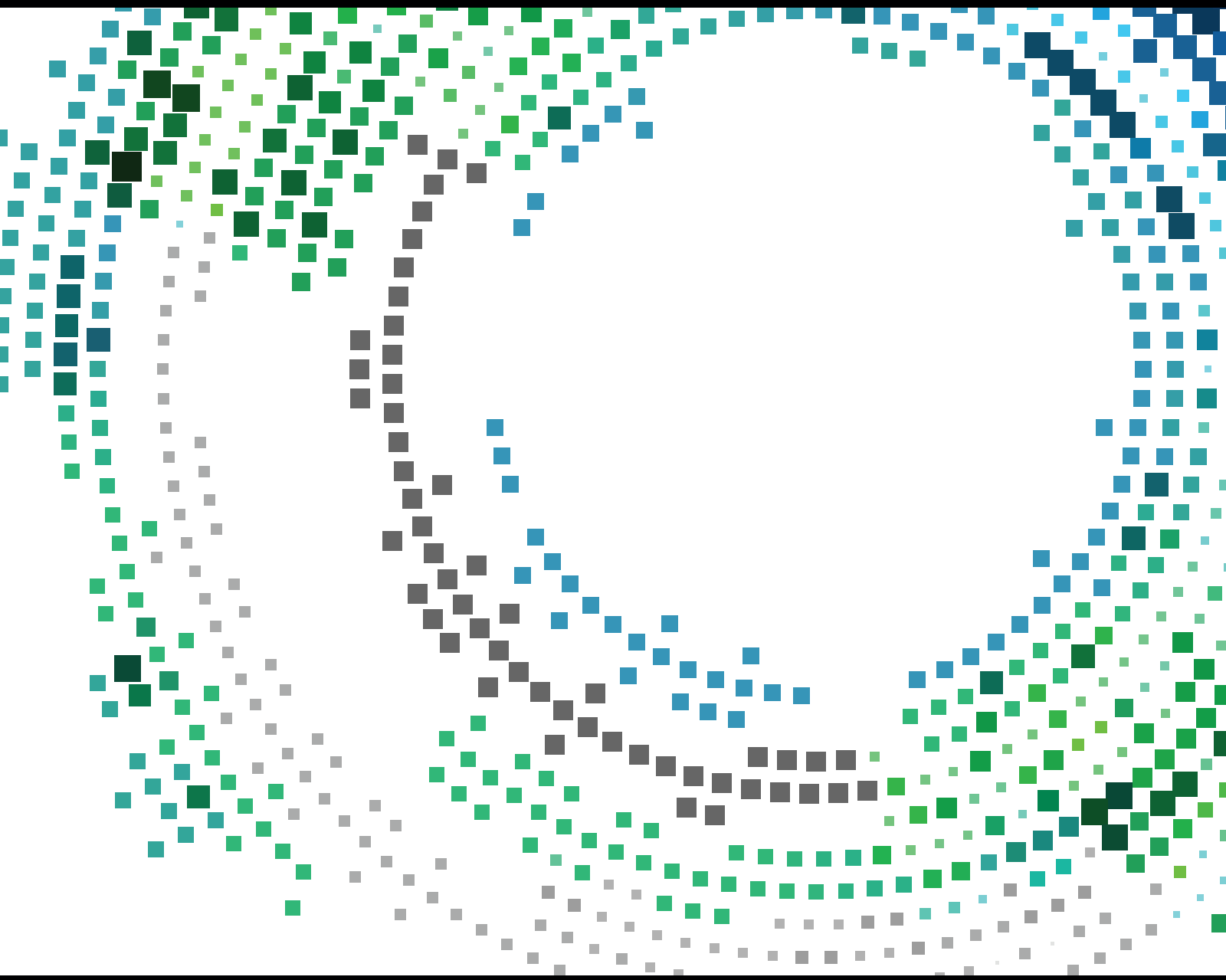
Mobile Information Systems

Challenges and Opportunities of Network Virtualization over Wireless Mobile Networks

Guest Editors: Xiaohong Jiang, Gabriel-Miro Muntean, George Ghinea, and Changqiao Xu



Challenges and Opportunities of Network Virtualization over Wireless Mobile Networks

Mobile Information Systems

Challenges and Opportunities of Network Virtualization over Wireless Mobile Networks

Guest Editors: Xiaohong Jiang, Gabriel-Miro Muntean, George Ghinea, and Changqiao Xu

Editorial Board

Contents

Hindawi
Mobile Information Systems
Volume 2017, Article ID 9053678, 2 pages
http://dx.doi.org/10.1155/2017/9053678

Editorial

Challenges and Opportunities of Network Virtualization over Wireless Mobile Networks

Xiaohong Jiang,[1] Gabriel-Miro Muntean,[2] George Ghinea,[3] and Changqiao Xu[4]

[1]*Future University Hakodate, Hakodate, Japan*
[2]*Dublin City University, Dublin, Ireland*
[3]*Brunel University London, Uxbridge, UK*
[4]*Beijing University of Posts and Telecommunications, Beijing, China*

Correspondence should be addressed to Xiaohong Jiang; jiang@fun.ac.jp

Received 6 February 2017; Accepted 6 February 2017; Published 26 March 2017

Enabled by the rising of mobile communication technologies, wireless mobile networks have become a crucial component of numerous Internet applications such as mobile sensing, video streaming, data dissemination, and social networks. With this growing trend, wireless mobile networks have the potential to reshape the way in which we lead our daily life. Large efforts have been made by the researchers recently, yet problems such as energy consumption, network resource allocation, and efficient data dissemination and transmission remain unsolved in wireless mobile networks.

Network virtualization as a concept enables abstraction and sharing of infrastructure and radio spectrum resource has become very popular in future communication technologies, such as 5G wireless network and cloud radio access networks (Cloud-RAN). With virtualization, the overall cost of equipment and management can be significantly reduced due to the increased network resource utilization. Additionally, the decoupling of network functionality from infrastructure makes it easier to migrate and test newer network architectures such as software-defined networks (SDN), content-centric networks (CCN), and emerging service and applications. Although network virtualization has been proved as a promising technology for future Internet, despite the opportunities brought by the network virtualization, several significant research challenges need to be addressed before the wide deployment mobile wireless network virtualization, including resource allocation and management, mobility, QoE guarantee of end users, and the future trend of wireless mobile network virtualization.

After thorough and meticulous reviews, we have selected 9 papers for this special issue.

Firstly, there are two articles dealing with the problem of dynamic resource caching for wireless mobile networks by making use of network virtualization technologies. In "A Novel Video Sharing Solution Based on Demand-Aware Resource Caching Optimization in Wireless Mobile Networks," S. Jia et al. propose a novel video sharing solution based on Demand-aware Resource Caching Optimization (VDRCO), in order to reduce network load and ensure user quality of experience (QoE). L. Zhong et al., in "A Novel Contribution-Aware Neighbor-Assist Video Delivery Solution over Mobile Content-Centric Networks," propose a novel contribution-aware neighbor-assist video delivery solution over mobile content-centric network (CNVD). CNVD designs the estimation methods of interest level and lookup capacity of nodes for video content, which enables the nodes to decide whether to build neighbor relationship with other nodes.

Leveraging network virtualization technologies, the paper by R. Zhang and S. Xiong entitled "A Novel Mobile Video Community Discovery Scheme Using Ontology-Based Semantical Interest Capture" proposes a novel mobile Video Community discovery scheme using ontology based semantical interest capture (VCOSI) to construct scalable and resilient node communities, which promotes video sharing capacity of video systems with the flexible and economic community maintenance.

The issue of dense wireless spaces for networking is addressed in the paper by M. Zhanikeev entitled "Improving Performance in Dense Wireless Spaces by Controlling Bulk Traffic." The author proposes a simple method called Bulk-n-Pick which minimizes the number of prolonged concurrent sessions by separating bulk from sync traffic. Moreover, the real-life experiments with various counts of wireless devices, bulk sizes, and levels of sync intensity are performed for realistic performance analysis.

The secure service composition for mobile clouds is addressed in the paper by N. Xi et al. entitled "Distributed Secure Service Composition with Declassification in Mobile Clouds." The authors propose a declassification mechanism for secure service composition based on cryptographic operations and information flow security requirements.

Accurate location service is also an important issue on wireless mobile networks. The paper by Q. Liu et al. entitled "A Wireless Location System in LTE Networks" proposes a new LTE location system in Centralized Radio Access Network (C-RAN), which makes channel and location measurement more available, allocation of baseband processing resources more flexible, and location service capability opening. In "GuideLoc: UAV-Assisted Multitarget Localization System for Disaster Rescue," A. Wang et al. propose GuideLoc, a highly efficient aerial wireless localization system, which uses the target guiding technology based on region division. By utilizing GuideLoc, the efficiency of search and rescue tasks can be improved.

Concerning mathematical modeling and network emulation, two papers focus on the dynamic topologies for virtualized sensor networks and emulation platform for delay tolerant networks, respectively. In "Diffusion Strategies for Distributed Kalman Filter with Dynamic Topologies in Virtualized Sensor Networks," S. Yang et al. propose a diffusion Kalman filter algorithm with dynamic topologies (DKFdt) to guarantee the accuracy of estimate values by sensors in physical layer of VSNs. The paper by H. Li et al. entitled "EmuStack: An OpenStack-Based DTN Network Emulation Platform (Extended Version)" proposes a large-scale real-time emulation platform named EmuStack, which aims at empowering network emulation to become as simple as network simulation.

Acknowledgments

We would like to thank all the reviewers for their hard work and for all their suggestions and comments, leading to improvements in the accepted papers. By compiling these papers, we hope that it will represent a useful starting point and stimulus for further research on network virtualization technologies in wireless mobile networks. We would like to take this opportunity to thank all authors for submitting their works to this special issue.

Xiaohong Jiang
Gabriel-Miro Muntean
George Ghinea
Changqiao Xu

Hindawi
Mobile Information Systems
Volume 2017, Article ID 1267608, 13 pages
http://dx.doi.org/10.1155/2017/1267608

Research Article

GuideLoc: UAV-Assisted Multitarget Localization System for Disaster Rescue

Anwen Wang,[1] Xiang Ji,[1] Dajun Wu,[2] Xuedong Bai,[1] Nana Ding,[1] Jing Pang,[1] Shaofeng Chen,[1] Xiaojiang Chen,[1] and Dingyi Fang[1]

[1]*School of Information Science and Technology, Northwest University, Xi'an, China*
[2]*Institute for Infocomm Research, Singapore*

Correspondence should be addressed to Dajun Wu; djwu@i2r.a-star.edu.sg and Dingyi Fang; dyf@nwu.edu.cn

Received 3 October 2016; Accepted 10 January 2017; Published 16 March 2017

Academic Editor: George Ghinea

Locating trapped targets using the signals of wireless devices such as mobile phones carried by targets is increasingly becoming the preferred scheme for disaster rescue. However, the accuracy of radio-localization technology currently adopted in the rescue is not high enough. To solve such problems, researchers proposed the approaches of Warwalking or Wardriving. However, both approaches are all limited by application scenarios. This paper proposes GuideLoc, a highly efficient aerial wireless localization system, which uses the target guiding technology based on region division. GuideLoc allows an unmanned aerial vehicle (UAV) to fly over a target device and provide position coordinates of the UAV as the target location information. During the process of positioning targets, based on the result of regional division, GuideLoc uses a number of fixed antennas to get the received signal strength indicator (RSSI) and angle of arrival (AOA) information of the target and estimates the target location using the information and correspondingly controls the UAV to fly towards the target. Averaging method is also applied for determining coordinates of the target. Experiment results show that GuideLoc can in short time achieve 2.7-meter positioning accuracy in average. Compared with other wireless localization methods using UAVs proposed in the literature, it shortened the flight distance by more than 50%. In addition, energy efficiency is also improved significantly.

1. Introduction

How many people died due to lack of timely rescue after an earthquake? Obtaining locations of trapped personnel quickly and accurately is the key to save more lives in mishaps and natural disasters such as earthquakes, fires, and straying in mountains. The industry is seeking to address this problem and improve the accuracy and efficiency of localization at scenes of disasters [1]. For example, UAVs were used in 2015 to search and rescue trapped people in the Yangtze River [2]. However, current UAV positioning approaches have not plan a flight path and they can only provide rough position estimation. This leads to time consumption and low accuracy significantly [3, 4]. Rescues would benefit tremendously from timely and accurate localization [5, 6]. For example, for an emergency rescue mission in the earthquake, many lives would be saved if the rescue is provided timely. Therefore, rescuers need to be aware of the positions of trapped persons quickly and accurately. Similarly, at the scene of fire, roughly knowing which floor trapped persons are located on is far from enough. It is better for rescuers to quickly and accurately know the exact locations of trapped persons, for example, whether or not the person is inside a specific room. In such case, we can rescue trapped persons timely and save many lives. In addition, fast and accurate positioning is also indispensable for some other emerging UAV applications. For instance, when someone is suddenly ill on a crowded street, if rescuers can find the exact position of the patient timely, the serious life threat of patient can be significantly reduced [7].

UAV positioning has recently been studied in the literature [1, 8–10]. Using the signal of wireless devices, such as mobile phones [11], carried by targets to locate targets is the common approach in this domain. One method is to use the aerial sensor network [12, 13], SensorFly [1, 9], which locates targets through the multiple anchor nodes deployed on the ground communicating with the node on the UAV.

The method needs to deploy anchor nodes in the field, but the deployment is difficult, expensive, and slow. Some other researchers propose to improve the localization accuracy by adopting multiple UAVs [10, 14–16]. However, this method requires the UAV to traverse the scanning area along a fixed flight path, which not only lacks flexibility, but also results in a longer searching path and a large amount of time and energy consumption. As another alternative, Warflying [17–19] directs the UAV to fly to any point of the open space to detect and accurately locate wireless target devices [8, 20]. Nevertheless, it needs to traverse the scanning target area according to the predetermined space filling curves no matter whether there exists a target or not [21], leading to low efficiency.

This paper introduces GuideLoc, a fast and accurate system for wireless localization of multiple targets based on the combination of applying UAVs and the method which uses received signal strength indicator (RSSI) and angle of arrival (AOA) to locate. Similar to current localization methods [22], GuideLoc uses RSSI [23] and AOA to estimate target positioning. Because the RSSIs sensed by the UAV from various directions are different and the direction in which the signal strength is the strongest corresponds to the direction of the target, the position of the target can be estimated. However, a key issue is that the localization is very rough and can generate serious positioning errors.

Different from these rough localization methods [24], GuideLoc controls the UAV to fly towards the target according to AOA information of the target. When GuideLoc arrives just over the target, it adopts its own GPS coordinates as the location data of the target, which can obtain the accurate location information. We propose a method which is mentioned in more detail subsequently to determine whether the UAV is over the target. At the same time, GuideLoc proposes the flight path planning algorithm for multitarget localization in an area. It needs to sequentially search each unit partition, calculate the flight path in each unit partition, and position each wireless target device in a unit partition along the flight path.

To illustrate GuideLoc [25], a simple example of aerial wireless localization is shown in Figure 1. Because the AOA and RSSI information of a target has certain deviation, it can cause inaccurate localization. Therefore, we need to adopt an advanced localization strategy to improve the accuracy. We allow the UAV to start from the starting point, fly along the planned path, and receive the wireless signal of each target device in real time during the flight process. When the UAV is over the target, the GPS coordinates of the UAV [26] are adopted as the location coordinates of the target device.

How can we determine whether the UAV is over the target device? The approach is to use the average of RSSIs. Specifically, if the average of RSSIs of the target is less than the threshold [24], the flight direction of the UAV needs to be adjusted for approaching to the target. Otherwise, it is deemed that the UAV has been over the target and the current GPS coordinates of the UAV are taken as the location coordinates of the target.

When the detection range of the UAV is smaller than the target area, how can the target detection [4, 27, 28] be accomplished in that area? To solve this problem, the whole area needs to be divided into a number of unit partitions [29]. Moreover, it should be noted that, in the real world, the targets are not evenly distributed in the target area. In some extreme cases, there is not any target device in some places.

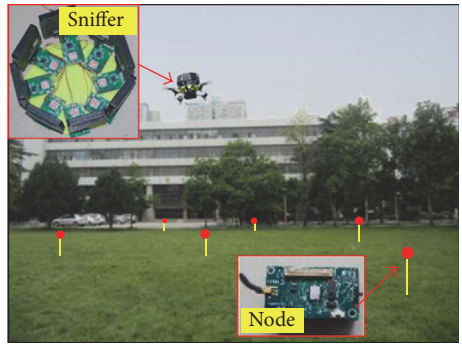


Figure 1: GuideLoc in localization.

In other cases, there could be multiple target devices in a unit partition. In order to complete the localization for all devices, the UAV will traverse the targets along a searching path. We present the design details in Section 5. Experiments verify the accuracy and rationality of the localization method in Section 7.

Summary of Experimental Results. We construct the GuideLoc system which consists of an ArduCopter with 8 wireless sniffers, a handheld controller, a software controller, and a target device locator. The performance of GuideLoc is evaluated using 12 wireless nodes deployed on our playground. In the experiments, we deploy 3, 6, 9, and 12 wireless nodes, respectively, as the targets and randomly distribute them on the football field. The transmission interval of the target is 1 second and the flight speed of the UAV is set to 5 m/s. In such condition, we compare GuideLoc with HAWK and obtain the following results:

(i) The positioning accuracy of GuideLoc can achieve 2.7 meters in average, which is slightly higher than that of HAWK;

(ii) The flight path length of HAWK is 2–5 times as long as that of GuideLoc under the different densities of target nodes.

Contribution

(i) It puts forward the first positioning system which combines target guiding with regional division. The positioning accuracy of the system is 2.7 meters in average in the area which is a 1 km × 1 km square, even in the case that targets are not evenly distributed in the area and wireless signal cannot cover the entire area.

(ii) The system can position targets in an area fast and accurately, which significantly reduces the power consumption of the UAV.

(iii) It proposes the averaging method for determining whether the UAV is over the target and verifies the

feasibility of the method in practice through the experiments.

(iv) It also provides a regional division method constrained by the one-hop communication coverage of a wireless device and verifies the effectiveness of the method using NS2 to simulate the positioning process.

2. Background

UAVs are unmanned aircrafts manipulated by radio remote control apparatus and self-provided program-control devices. Instead of a cockpit, an autopilot and program-control devices and so forth are installed inside a UAV [30]. The staff on the ground carry out tasks such as tracking [31] and positioning, remote controlling, and data transmission [32] by radar or other equipment.

The wireless localization system based on UAV consists of five components, namely, a multirotor helicopter, wireless sniffers, a handled controller, a software controller, and a target device locator [33]. The localization process is such that we configure a UAV equipped with a searching and positioning system and control the UAV to fly along a certain path. Because a target device periodically transmits the radio at some particular frequency [34], the directional antennas installed on the UAV can be used to collect RSSIs, estimate the AOA of the signal according to the RSSIs, and subsequently get the direction information of the target [22]. In addition, in the case of multiple targets, different address codes in the packets can be used to distinguish the signals from different targets.

The method mentioned above for searching and rescuing based on UAVs exhibits several features which are particularly meaningful for the localization problem.

(i) The biggest advantage of using a UAV is that it can fly at low height and obtain image data under simple conditions, even in the harsh natural environment [35]. In this way, we can reduce the search scope and rescue scale without costing a great deal of time, manpower, and material resources, which improves the effectiveness of the rescue, lowers search cost, and enhances the safety of rescuers.

(ii) With regard to RSSI technology [36–38], because the detection equipment is not sophisticated and its hardware is of low cost, we can obtain a more accurate RSSI value through averaging RSSI values from multiple measurements and reduce the impact of multipath and shadowing effect.

(iii) AOA data can be obtained by an antenna array or combining several receivers [22], which is easy to be realized in technology and could also bring about high positioning accuracy due to little noises in angle estimation.

(iv) Different address codes in the packets sent by the different devices can be used to distinguish the signals from different targets.

3. Overview

GuideLoc is an aerial wireless localization system which uses target guiding based on region division and is also energy-efficient. It can locate trapped targets quickly and accurately in the disaster rescue. Its flight distance is more than 50% shorter than that of other known wireless localization methods based on UAVs.

To realize the localization, GuideLoc collects wireless signals from the targets nearby. We have designed an experiment to simulate the disaster site in which 12 wireless nodes are randomly deployed as the targets to be searched on the football field. The transmission interval of the target signal is set to 1 second and the flight speed of the UAV is set to 5 m/s. GuideLoc presents the best searching path based on the genetic algorithm and uses wireless sniffers to collect RSSIs and AOA of a target. In addition, the GPS coordinates of the UAV are used to locate the target. As a result, GuideLoc achieves 2.7 meters of positioning accuracy on average.

To locate a target, GuideLoc goes through the following steps.

Step 1. If the target area is less than or equal to the one-top communication range of a wireless device, it will be regarded directly as a unit partition and the procedure goes to Step 2. Otherwise, the target area is divided into a number of unit partitions to ensure that each point in a unit partition can detect the signal from any other point and use the genetic algorithm to plan the optimal searching path for visiting each unit partition. (It will be explained in detail in Section 4.)

Step 2. Take the center of the nearest unit partition as the starting point of the UAV, fly to visit each unit partition, and detect the signals of wireless devices in real time to search targets in a unit partition.

Step 3. When the UAV arrives at a unit partition center, if GuideLoc can detect one or more new targets in its communication range, then go to Step 4; if not, it will determine whether it has finished the detection of all the unit partitions. If the detection has been finished, the localization process is over; if not, the UAV will fly to the center of the next unit partition along the searching path and repeat Step 3.

Step 4. GuideLoc obtains AOA and RSSI information of all targets, estimates the location of each target according to the information, uses the genetic algorithm to plan the optimal flight path for visiting each target with estimated locations, and then starts to traverse along the determined optimal path. (It will be illustrated in detail in Section 5.) When the UAV flies towards a target, the AOA and RSSIs of the target should be monitored in real time. The direction where signal strength is the strongest is considered as the direction of target. Then, the UAV is controlled to fly towards that direction. Afterwards, averaging method is used to judge whether the UAV is over the target; if yes, the GPS coordinates of the UAV are returned as position information of the target. (It will be explained in detail in Section 6.)

Step 5. After the UAV determines locations of all the targets in the current unit partition, it will fly to the next unit partition according to the planed path in Step 1. Return to Step 3.

We will discuss steps above and related technology in more detail in the following sections. In Section 4, we mainly discuss the region division; in Section 5, we describe the planning of best path for target localization in the target area; and then in Section 6, we discuss how to use the averaging method to locate a target.

4. Region Division

When the target area is larger than the one-hop communication range of a wireless device, dividing the target area must be done, because GuideLoc first needs to estimate the locations of all targets according to the received RSSI and AOA information sent by targets in a one-hop communication range, then plans the flight path according to the estimated locations, and last traverses each target. When the UAV flies right over a target, GuideLoc will take the GPS coordinates of the UAV as the target location information.

The purpose of region division is to ensure that any unit partition can be covered by the one-hop communication range of a wireless device, which can make GuideLoc work efficiently in each unit partition. Obviously, the UAV traverses each unit partition and locates the targets within each partition, which can solve the positioning problem when the target area is larger than a one-hop communication range. Of course, in order to shorten the flight distance, GuideLoc needs to traverse each unit partition along a shortest path.

4.1. Region Division. The core task of region division is to determine the size of a unit partition. There are two main limitations on the size: (1) Each unit partition can be covered by the one-hop communication range of a wireless device; (2) When the UAV flies at the center of unit partition, it can have received at least one data packet sent by each target in the unit partition. As a consequence, the factors influencing the division include the size of the target area, the radius r of a one-hop communication range, the UAV flight speed v, and the time interval t of packet transmission of a target.

In order to explain the method of regional division easily and clearly, we assume that the target area is a square with d on a side, and $N \times N$ unit partitions (the grid sections) are gotten after dividing the area, as shown in Figure 2. The essence of region division is to determine an optimal N. To ensure that GuideLoc can control the UAV to traverse all the targets in a unit partition on the basis of the real-time RSSI and AOA information, the UAV must receive a data packet sent by any other target anywhere in the unit partition, regardless of the position of the UAV in this partition. In other words, a unit partition is a subset of the intersection of all the one-hop communication ranges of targets in the partition. Because a unit partition is a square, the longest distance between any two points in the partition is the diagonal length. If the length is less than or equal to the radius r of a one-hop communication range, the UAV can receive the data packets

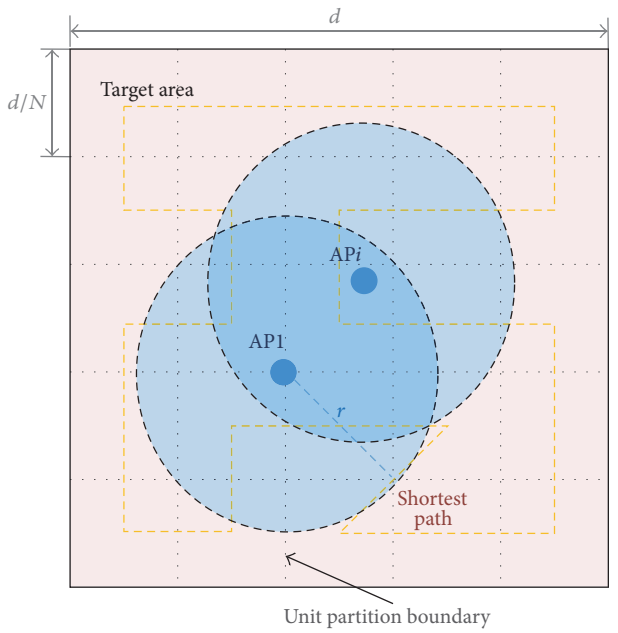


Figure 2: Illustration of regional division and shortest path for visiting every unit partition.

sent by any other targets in the unit partition, regardless of the UAV position in this partition. So the side length of a unit partition must satisfy

$$\frac{d}{N} \leq \frac{\sqrt{2}r}{2}. \tag{1}$$

If L represents the distance between the last traversed target in the previous unit partition and the center of the next partition, L satisfies formula (2). And the minimum value of L is obtained when the last traversed target in the previous partition is just at the center point of the adjacent edge of the previous and next partitions. It is $d/2N$. In order to ensure the UAV having traversed the last target in the previous partition can receive at least one packet from each target in the next unit partition before it reaches the center of the next partition, L must fulfill formula (3)

$$L \geq \frac{d}{2N}, \tag{2}$$

$$L > v \times t, \tag{3}$$

$$\frac{\sqrt{2}d}{r} \leq N < \frac{d}{vt}. \tag{4}$$

Combining formulas (1), (2), and (3), formula (4) can be derived. In the actual division, the UAV is expected to fly at the fastest speed for search and rescue. Therefore, the optimum value of N is $\sqrt{2}d/r$.

Discussion. A few points worth-noting are as follows:

(i) With regard to formula (3), since sending wireless signals is periodical [39], transmission interval t of a

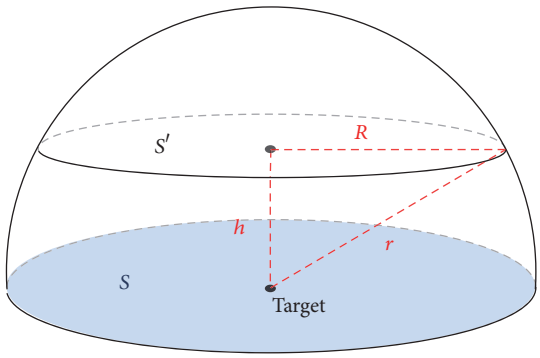


Figure 3: Effective range of UAV communicating with target.

target can be easily determined. The maximum flight speed of the UAV can also be set. Therefore, the minimum L can be inferred and it is also ensured that the sniffer can detect all the mobile devices in the target region during the process of localization.

(ii) GuideLoc can search and locate targets in 2D space by a UAV. Because the UAV flies in the air and trapped targets with wireless devices are on the ground, when the UAV is flying at a certain height, the range where the wireless device on the ground can communicate with the UAV should be a circular cross section S, which is formed by the UAV flight plane cutting through the hemispheric communication range of the wireless device. The relationship between the radius R of S, the UAV flight height h, and the radius r of the wireless communication range is as shown in Figure 3 and the following equation:

$$R = \sqrt{r^2 - h^2}. \tag{5}$$

4.2. Path Planning for Traversing All the Unit Partitions. After partitioning the area, GuideLoc needs to detect all the target devices in every unit partition so that all the targets in the whole area can be detected. Obviously, we should choose the shortest route for the UAV to traverse all the unit partitions and make sure that finally the UAV comes back to the start point after each unit partition is visited once and only once.

Planning the best searching path for all the unit partitions is the same as Traveling Salesman Problem (TSP). That is to say, the problem of flying path planning can be translated into TSP. In graph theory, this problem is to find out the Hamilton circle whose weight is the smallest in a given weighted connected undirected graph $G(V, E)$ and the vertexes of the graph G are the center points of the unit partitions.

We make use of genetic algorithm (GA) commonly used to solve TSP as a solution to obtain the shortest flight path.

5. Detection of Multitargets Inside a Unit Partition

While the UAV traverses each unit partition, GuideLoc needs to detect all the target devices in each unit partition one by one so that all the targets in the whole area can be detected.

In order to improve the accuracy of localization, we require the UAV to fly right over the target to be detected and use the GPS coordinates of the UAV as the position information of the target. Because the UAV can detect all the targets in a unit partition, we can obtain AOA and RSSI values of these targets, estimate the location of every target in the unit partition according to the information, and utilize the location information to plan the optimal flight path for visiting every target in this unit partition. Thus, we need a proper method to plan this optimal path in a unit partition.

Intuitively, in a unit partition, the UAV should choose a route whose total distance is the shortest and traverse every target only once. Therefore, when we plan the shortest path in a unit partition, we need to guarantee that the UAV will not only go through all the targets, but also pass the each target only once in this unit partition and finally arrive at the center of next unit partition.

5.1. Determination of the Optimal Searching Path. The problem of planning the shortest flight path is similar to a TSP. However, a key challenge is how to translate the path planning into a TSP. The translating process is shown as follows.

The goal of TSP is to search for an optimal cycle for traversing n points or to find a permutation $\pi(X) = \{V_1, V_2, \ldots, V_n\}$ of the natural subset $X = 1, 2, \ldots, n$ (an element of the set X represents a point number) to make formula (6) hold.

$$\min T_d = \min \left\{ \sum_{i=1}^{n-1} d\left(V_i, V_{i+1}\right) + d\left(V_n, V_1\right) \right\}. \tag{6}$$

In (6), $d(V_i, V_{i+1})$ represents the distance between point V_i and point V_{i+1}. V_1 is the starting point. But our problem is to find the shortest path to traverse all the points from the start to the end or search for a permutation $\psi(X) = \{M_1, M_2, \ldots, M_n\}$ of the natural subset $X = 1, 2, \ldots, n$ (an element of the set X represents a point number) and make formula (7) hold

$$\min T_d = \min \sum_{i=1}^{n-1} d\left(M_i, M_{i+1}\right). \tag{7}$$

In formula (7), $d(M_i, M_{i+1})$ represents the distance between point M_i and point M_{i+1}, M_1 is the starting point, and M_n is the destination. Since this problem is similar to a TSP, we will use GA that is usually applied to solve a TSP to settle this problem.

Algorithm. $d(M_n, M_1)$ is added to formula (7); that is to say, a virtual edge from the starting point to the destination is added and $d(M_n, M_1)$ is set as a minimum negative number; thus

$$\begin{aligned} \min T_d &= \min \sum_{i=1}^{n-1} d\left(M_i, M_{i+1}\right) + d\left(M_n, M_1\right) \\ &= \min \left\{ \sum_{i=1}^{n-1} d\left(M_i, M_{i+1}\right) + d\left(M_n, M_1\right) \right\}. \end{aligned} \tag{8}$$

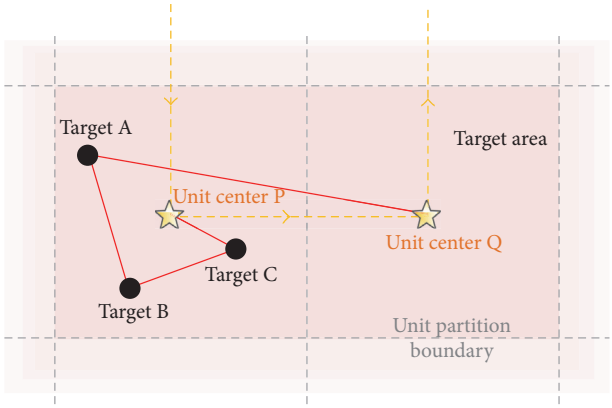


Figure 4: Shortest flight path within a unit partition.

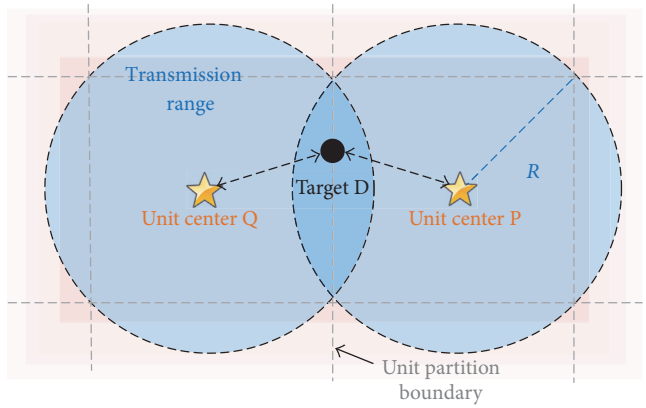


Figure 5: Illustration of target on the boundary between two adjacent unit partitions.

Comparing formula (8) with formula (6), formula (8) is just the expression of TSP. Our problem can be converted into a TSP and it also belongs to a kind of NP complete problem.

In our algorithm, we first figure out the estimated positions of each wireless device using their RSSI and AOA values. Thereafter, we take the center of the current unit partition as the starting point of the searching path within this unit partition and a vertex of the graph G', take the center of the next unit partition in the region searching path as the destination of the searching path within this unit partition and a vertex of the graph G', take all the targets in the current unit partition as the other vertices of the graph G', and take the distance between any two vertices in G' as an edge weight in G'. Finally we get a weighted connected undirected graph G'. In order to guarantee that the smallest Hamilton circle worked out must include the edge from the starting point to the destination, this edge is assumed to exist in the graph G' and its weight is supposed to be a very small negative.

Finally, we make use of the genetic algorithm to find out a Hamilton circle whose total distance is the shortest in the weighted connected undirected graph G'. Because the weight of the virtual edge between the starting point and the destination is assumed as a constant, after the virtual edge is deleted from the Hamilton circle having been found out, the rest of edges in the Hamilton circle just form the shortest path from the starting point to the destination through all the points.

That is to say, the path is the shortest flight path within a unit partition, for example, the path P-C-B-A-Q expressed using a solid red line as shown in Figure 4.

5.2. Determination of the Unit Partition Corresponding to the Target. When we apply the genetic algorithm to complete the localization of all the targets, a practical challenge arises: which unit partition does a target device belong to on earth?

After a region has been divided into some partitions, it is apparent that the positions of all the target devices located in this region are also determined. The distance between a target device and the center of the current unit partition is calculated using the RSSI values of the target device based on Log-Normal Shadowing Model (LNSM) [40]. If the distance is less than or equal to the threshold, the target device will belong to the current unit partition. Otherwise, it will not belong to this unit partition. However, since there exists a large error in the estimated distance according to the RSSI, a target device may be missed to be located due to the possible mistake made in determining which unit partition the target device belongs to.

The point D, for example, is on the boundary between two adjacent unit partitions in Figure 5. When the UAV arrives at the center of the right unit partition, if the estimated distance is longer than the actual distance, the UAV will judge that D does not belong to the unit partition.

Similarly, when the UAV arrives at the center of the left unit partition, if the estimated distance is slightly longer than the actual distance, the UAV also decides that it does not belong to the unit partition, which results in missing the detection of target device D.

In order to avoid missing any target, a threshold λ is set as the half of the diagonal length of a unit partition as formula (9) shown below:

$$\lambda = \frac{\sqrt{2}d}{2N}. \tag{9}$$

Then we can get a circular decision region. Its center is the center of a unit partition and its radius is the threshold. Apparently, missing a target device can be effectively avoided because the boundaries of any unit partition are all included inside a decision region.

Specifically, when the UAV approaches to the center of the right unit partition and makes use of RSSI values of the target device D to estimate the distance, even though the estimated distance is longer than the real distance, it would not give an erroneous judgment and can avoid missing the target D because D is still in the circular decision region.

Since the GuideLoc can mark the targets having been localized with their identification (i.e., their IP addresses), a target in an overlapping area between the two adjacent decision regions would not be localized more than once.

6. Single Target Localization

After determining the path of visiting each target in a unit partition, the GuideLoc will realize the fast and accurate positioning for a single target and then complete the accurate positioning for all the targets in the unit partition.

In many applications, reducing the complexity of data collection is required so as to position the target quickly and accurately. In this case, we need to estimate a target location by using a number of directional antennas to receive RSSI and AOA data of the target and utilizing a RSSI-based ranging method. However, the ranging method based on RSSI is a very rough positioning technology which could give rise to serious positioning errors.

For instance, when the geographical factors of the detection area remain constant, the main factors affecting RSSI values are the nongeographical factors, such as temperature, humidity, and air pressure. The effect of changes of such factors on the wireless signal transmission is ruleless. The fluctuation caused by them for ranging based on RSSI is not very significant. But they can cause greater impact on the distance estimation for a target whose propagation model has been fixed, result in RSSI-based ranging error becoming larger, and thus affect the positioning accuracy. Instead, GuideLoc uses the following method in order to finish the positioning accurately.

Method. GuideLoc controls the UAV to fly towards and gradually approach a target according to the angle information of the target. When the UAV reaches the point just right over the target, the GPS coordinates of the UAV are used as the position coordinates of the target.

Nevertheless, this method brings about a new challenge: how to determine whether the UAV has been just right over the target or not.

We designed two methods for this challenge. One is range method. The basic idea is as follows: when the UAV is exactly right over a target, ideally, the RSSIs from different directions almost all are equal, which means the range of the RSSI values closes to 0. But the experiment shows that it is not better than the averaging method. The idea of averaging method is as follows: when the UAV flies into the communication range of a target, it will receive wireless signals sent by the target. The RSSIs in different directions vary. In theory, the direction in which signal strength is the strongest is the direction of the corresponding target. Thus the UAV can arrive at the point which is just right over the target by flying towards the target direction. When the UAV flies towards a target and approaches it, the RSSI will gradually increase. The RSSI is the strongest when it is closest to the target. But it will gradually decrease when the UAV flies away from the target. Therefore, we can use the average of RSSIs to determine whether the UAV is just right over the target. The formula is

$$\overline{\mathrm{RSSI}} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{RSSI}_i, \tag{10}$$

where n is the number of antennas installed on the UAV (n is 8 in GuideLoc), i is an antenna label, and RSSI_i is the RSSI sensed by the antenna in i direction.

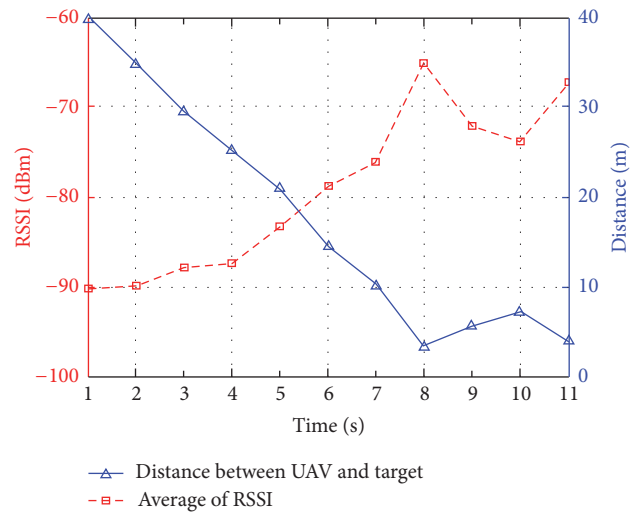


Figure 6: The result of the averaging method experiment.

In order to validate the feasibility of the averaging method in practice, an experiment is conducted with the related data used in the above experiments. The result is shown in Figure 6 where the blue line represents the change of the distance between the UAV and the target and the red line represents the change of the average of RSSIs. The figure shows that the positioning process can be divided into three stages. In the first stage (1 s–8 s), when the UAV flies towards and gradually approaches the target, the average of RSSIs gradually increases. In particular, the average of RSSIs reaches the maximum when the UAV is the closest to the target (8 s); in the second stage (8 s–10 s), the average of RSSIs gradually decreases as the UAV flies away from the target. In the last stage (10 s-11 s), as a result of only one target node deployed in the experiment, the UAV hovers around the target, which brings about a gentle fluctuation of the average of RSSIs. Therefore, the averaging method is feasible in practice to determine whether the UAV is just right over a target or not.

To reduce the energy consumption of the UAV, a threshold needs to be used as a key factor in the aerial wireless location algorithm based on target guiding. Specifically, GuideLoc takes a target as the next destination and controls the UAV to fly towards it. During the flight, the UAV constantly captures the wireless signals transmitted from the target and gets the new data. The average of RSSIs is subsequently calculated. If the average is less than the threshold, GuideLoc will update the direction information, adjust its flight direction, and continue to fly close to the target. Otherwise, the UAV is regarded as having been just right over the target, and the current position of the UAV is served as the position of the target. Then the current GPS coordinates of the UAV are transmitted to a Lenovo laptop X230 on the ground by an XBee wireless transceiver and the target positioning is completed.

The location algorithm based on target guiding is as shown in Algorithm 1.

```
(1)  Get measurement samples of the target;
(2)  Estimate AOA according to RSSI of different directions;
(3)  GuideLoc flies towards the target;
(4)  Capture new measurement samples of the target;
(5)  Use Formula (10) to calculate the average of RSSIs;
(6)  if (The average of RSSIs < −70)
(7)        Update AOA, adjust the direction of flight and
(8)        continues flying close to the target;
(9)  else
(10)       Arrive just right over the target, Locinfo = GPS;
(11) endif
(12) return Locinfo.
```

Algorithm 1: Algorithm of GuideLoc.

Discussion

(i) A threshold is used to determine whether the UAV is just over the target or not in the averaging method. The threshold setting is associated with the devices used. Different devices have their respective thresholds. The threshold is set to −70 dBm in our experiment.

7. Experiment and Analysis

After obtaining the division algorithm for determining the unit partition size and the target positioning algorithm, we need to verify their validity and feasibility and evaluate their performance. Firstly, we verify whether the unit partition size meets formula (4) through simulation and, thereafter, compare the performance of GuideLoc with that of HAWK using different densities of target nodes. Through the experiments, we obtain the positioning performance of the target wireless devices in the playground of our campus. However, since disaster rescue is often expected as fast as possible in real life, the effects of flight speed on the performance of the system also need to be studied.

7.1. Simulation Analysis. To verify whether the unit partition size is consistent with formula (4), we use NS2 to simulate the positioning process. Firstly, we configure the transmission range of a target wireless device as 300 m, the broadcast interval of a packet as 1 s, the flight speed of the UAV as 10 m/s, and the target area as a square area of 1000 m × 1000 m in which 30 wireless nodes are randomly deployed.

In the simulation, the UAV flies and searches along the unit partitions initially according to the optimal Hamilton circle. When a packet from a wireless node is received by the UAV, the distance between the node and the UAV is recorded. If the range of the RSSI values in different directions is not greater than 11 dBm, the current position of the UAV is taken as the position of the node and the distance between the node and the UAV is regarded as the positioning error. Subsequently, we use 0, 10, 20, 30, . . . , 190, 200 wireless nodes which are randomly and evenly distributed in the square area and run the simulation 30 times under each node density.

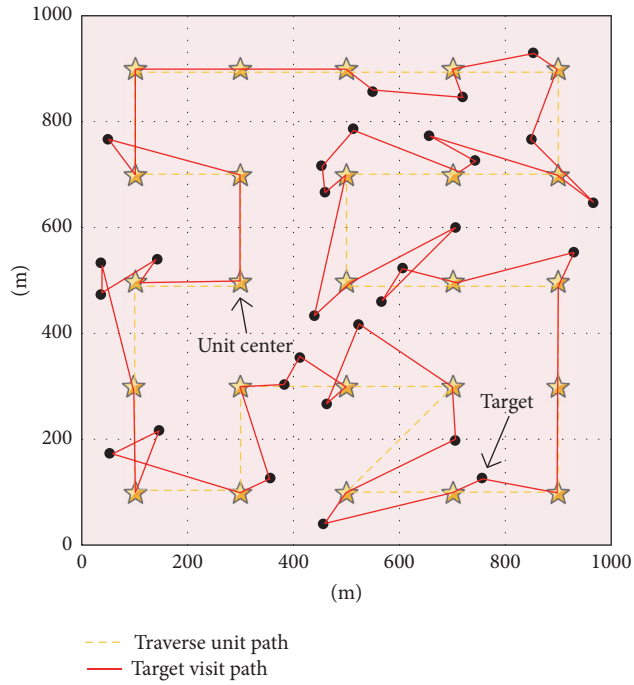


Figure 7: Flight route of GuideLoc: unit traversal path shows the traversal order of unit partitions and target visit path shows the path of visiting targets in each unit partition.

We apply the above simulation configuration to formula (4) to acquire the size of square unit partition. The target area is thus divided into 5 × 5 unit partitions as shown in Figure 7. The red solid line in the figure illustrates the flight path of GuideLoc in the positioning process. The UAV searches 25 unit partitions in order, so that it can locate the 30 mobile devices in the target area. At macrolevel, the UAV searches all the unit partitions along the smallest Hamilton circle (the yellow dashed line). At macrolevel, after the UAV flies through the center of the unit partition, it will fly towards the first target to be visited.

The simulation results are presented in Figure 8 where there are two CDF curves, respectively, representing the positioning errors of GuideLoc and HAWK under the 21 groups

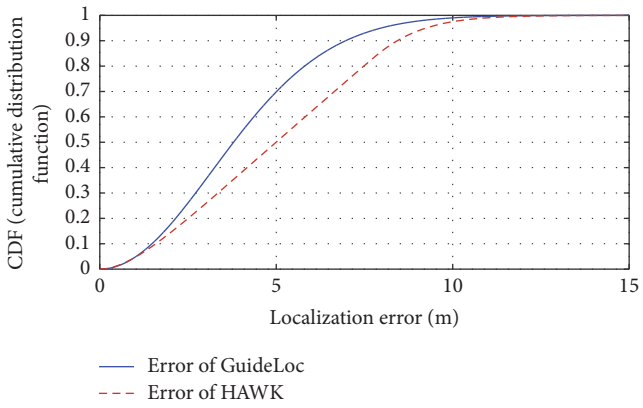


Figure 8: Localization errors of GuideLoc and HAWK.

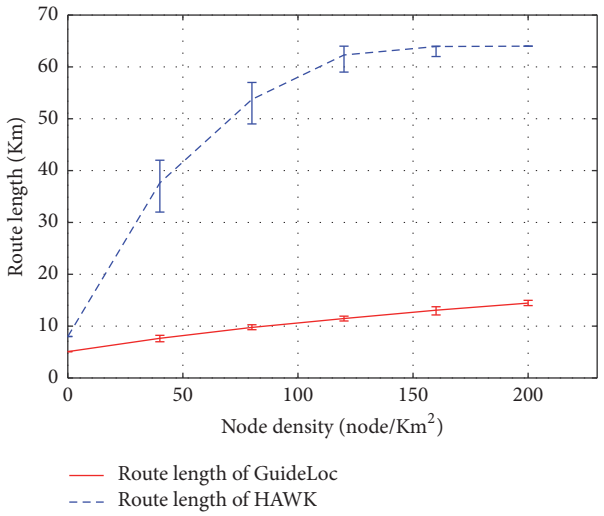


Figure 9: Flight path length of GuideLoc and HAWK under different node densities.

of node densities. The figure shows that the positioning accuracy of GuideLoc is slightly higher than that of HAWK. The median of the positioning error of GuideLoc is about 4 m, while that of HAWK is about 5 m. Moreover, the maximum error of GuideLoc is less than 12 m, while that of HAWK is less than 15 m.

In Figure 9, the length of the flight paths of GuideLoc and HAWK under the different node densities is shown. The length of the flight path of GuideLoc is obviously much shorter than that of HAWK. Under each node density, the length of the flight path of HAWK is 2–5 times as long as that of GuideLoc. When the node density is 0 (that is to say, there is not any target device in the target area), the flight path of HAWK is a level 3 Moore curve, the length of which is 8000 m, while the flight path of GuideLoc is the smallest Hamilton circle, the length of which is only 5082 m. The length of the flight paths in two methods is increasing with the increment of node deployment density, but the path length of HAWK increases quickly and the path length of GuideLoc increases slow.

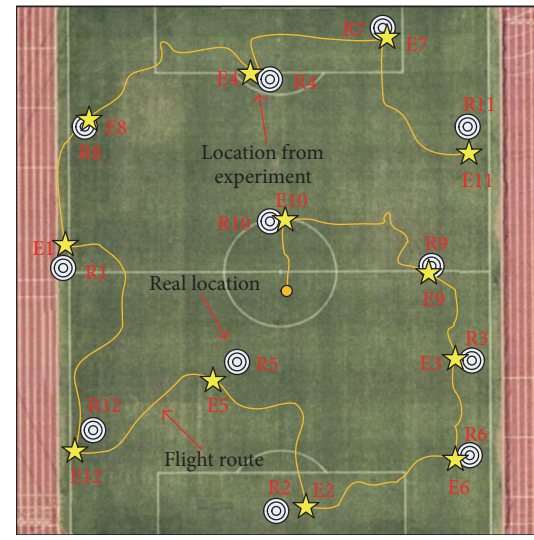


Figure 10: APs identified by GuideLoc.

7.2. Experiment Analysis. In this section, we obtain the results of wireless devices localization in the campus playground through real experiments and use them to evaluate the performance of GuideLoc. In the experiment, 3, 6, 9, and 12 wireless nodes are randomly distributed on the football field, respectively. We denote the beacon transmission interval of the target by t, which is 1 second. The flight speed of the UAV is set as 5 m/s. In order to ensure the positioning accuracy of HAWK and GuideLoc is on the same order of magnitude, a level 3 Moore curve is used as the flight path of HAWK.

However, because the UAV battery can only power its flight for about 15 minutes, a large scale experiment is not able to be conducted.

The experiment results are shown in Figure 10 which not only indicates the flight path of GuideLoc (the yellow solid line) in the localization process in the case of 12 targets deployed, but also marks the real and estimated locations of those targets (the white dot represents the real location; the yellow star represents the estimated location). Figure 11 shows the CDF curves of localization errors of GuideLoc and HAWK, respectively, in the case of different number of nodes. GuideLoc can achieve the accuracy of 2.7 meters in average. Figure 12, respectively, shows the total length of the flight paths of GuideLoc and HAWK in the case of different number of nodes. The proposed method makes use of the direction information of a target device so that the UAV can fly directly towards the target along the shortest flight path. As expected, when compared with HAWK, GuideLoc can significantly reduce the total flight distance and save the power consumption of the UAV.

We also need to evaluate the impacts of various flight speeds on the localization accuracy and time cost. For this purpose, in the experiment, only one wireless target node is deployed on the football field. The packet transmission interval of the node is set to 1 second. GuideLoc takes off from the edge of the football field to locate the target node and its

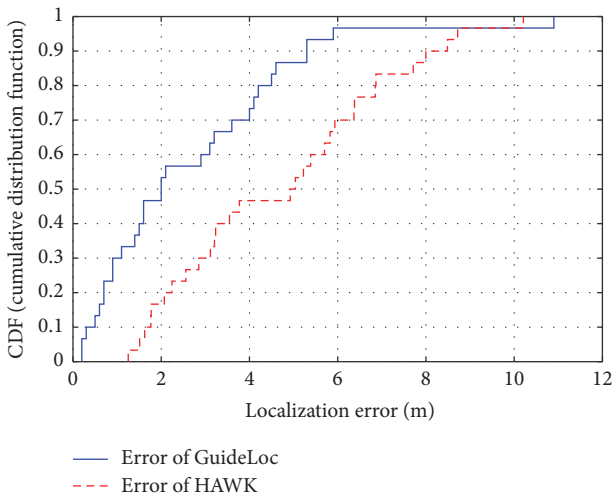


Figure 11: Localization errors of GuideLoc and HAWK.

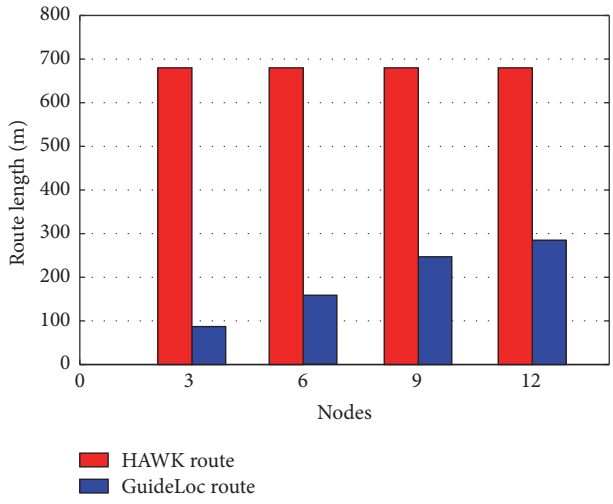


Figure 12: Route length of GuideLoc and HAWK.

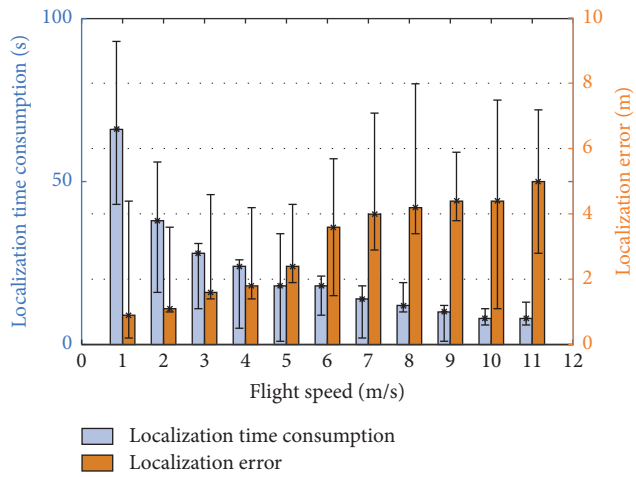


Figure 13: Localization error and time consumption of GuideLoc under the 11 different flight speeds.

flight speed is, respectively, set to 1, 2, 3 10, and 11 m/s. The experiment is repeated ten times under each flight speed.

Figure 13 shows the localization error and time consumption of GuideLoc under the 11 different flight speeds. As expected, the faster the flight speed, the less the number of measured samples and thus the greater the localization error. However, the faster the flight speed, the less the consumed the localization time and thus the higher the efficiency of search and rescue.

In short, the flight speed has an obvious impact on the performance of GuideLoc, and the localization error and time consumption are two contradictory parameters, which need to be carefully balanced.

Therefore, it is crucial to analyze and quantify the application requirements of users, so as to set the matching flight speed and related performance parameters.

7.3. Experiment Summary. We firstly verify the formula which restrains the unit partition size by the simulation and then compare the performance of GuideLoc and HAWK under the different densities of target nodes. Finally, we conduct the real-world experiments and acquire the corresponding results of wireless devices localization on the football field. The simulation results indicate that the localization accuracy of GuideLoc is slightly higher than that of HAWK and the length of the flight path of HAWK is 2–5 times as long as that of GuideLoc under the different densities of target nodes. The experiment results demonstrate that the localization accuracy of GuideLoc is 2.7 meters in average.

In summary, GuideLoc can directly fly towards a target device using the direction information of the target. Compared with HAWK utilizing space filling curves to traverse and scan, GuideLoc greatly reduces the total flight distance and saves the power consumption of the UAV.

8. Related Work

The early localization technologies for trapped targets in disaster rescue rely on Wardriving [8, 41] to collect RSSIs including direction information. Usually, the gradient information inferred from RSSIs is used to determine the location of an access point (AP) [20]. However, the technologies based on Wardriving are all limited by roads since it is difficult to reach some places, such as the top of high-rise buildings [29]. Thus the portability and flexibility of the system using this kind of technology are very limited.

Some researchers proposed heuristic rules (cluster [10]) based on the behavior of biological colony to establish an indoor cluster system and make a group of UAVs control their own flight. Each cluster member in the system is equipped with an airborne computer and a mini wireless camera [42], so multiple views can be collected to analyze through only one flight. Nevertheless, a common challenge for these technologies is that the localization effect is influenced by the cooperation effectiveness of these cluster members, that is, UAVs. Besides, it is very difficult to find out a hidden target in a practical application.

HAWK [19] uses a space filling curve as a searching path and determines the level of searching path according to the UAV flight speed, the packet transmission interval of target, and the wireless transmission range in order to ensure that all the wireless devices in a search area can be detected [43]. However, it lacks flexibility, since it must traverse and scan a target area along a fixed space filling curve, regardless of whether there are targets in the area, which results in a longer searching path and a significant amount of time and energy consumption.

The technology of searching and locating targets by using UAVs has some limitations. Firstly, one or more helicopters are used to form a fixed space filling curve instead of planning an optimal searching path in advance. Secondly, the location information of targets is obtained by cameras or RSSIs of sensors. Using cameras to get location information usually needs human involvement. In addition, it is of slow speed and easy to miss some targets. Furthermore, the localization accuracy cannot be guaranteed and targets that are only in sight can be monitored. RSSIs can only help us roughly estimate the target location; thus the error is significant.

In contrast, GuideLoc can divide a target area into a number of unit partitions in advance and use the genetic algorithm to plan an optimal searching path for traversing each unit partition. Then it uses the UAV to locate a target and takes the GPS coordinates of the UAV as the target location information. This method not only improves the localization accuracy, but also uses only one UAV to search and locate all the targets along an optimized searching path.

GuideLoc is related to a lot of technologies in locating wireless devices.

Firstly, the technology of collecting wireless signals is adopted for RSSI and AOA measurement. In some conditions, only one rotating directional antenna is used to measure AOA. GuideLoc also uses directional antennas, but it employs a number of fixed directional antennas. Each antenna is connected to a radio receiver, which enables the UAV to receive the wireless frames from all directions at the same time.

In addition, the common method for getting the location of wireless devices is to install a camera or a mini DVR on a searching device or use RSSIs to estimate [44]. Different from these methods, GuideLoc controls the UAV to fly just right over a target by estimating the target location based on RSSI and AOA and takes the current GPS coordinates of the UAV as the target location information, which can improve the localization accuracy.

Furthermore, in order to further improve the localization accuracy, the data in the digital compass on the UAV is used to enhance the calculation. Then the collected information composes a measurement sample ⟨target identification, GPS coordinates, UAV direction, antenna label, signal strength⟩, which is stored into flash memory for future calculation.

Finally, the current methods for searching and locating targets using UAVs are always dependent on space filling curves to determine the flight path of the UAV [45]. However, GuideLoc divides a target area into a number of unit partitions in advance and uses the genetic algorithm to plan an optimal searching path to traverse each unit partition, which shortens the flight distance and lowers the energy consumption.

9. Concluding Remarks

Nowadays, rescue in disaster such as earthquake rescue, expedition trapped, travelers lost, fire rescue all makes use of wireless devices to locate the trapped victims. However, most of the current search and rescue techniques are based on the auxiliary functioning of facilities or training, which causes low localization accuracy and large energy consumption. With regard to Wardriving, such kind of technology only applies to empty areas such as grasslands in which there are fewer and shorter obstructions. It does not deal with areas with more and higher coverings such as mountains and buildings. As for Warflying, it requests a UAV to traverse and scan a target area along a fixed flight path, even if there is no target in the area. Apparently, it lacks flexibility and leads to a longer flight path which results in large time and energy consumption, which could further lead to some bad consequences. For example, if the searching time is too long, the best rescue time may be missed and the lives of persons in distress will be endangered. Furthermore, because of the complex environment of disaster area and adverse communication, medical treatment, climate and topography, the high energy consumption and long searching time will affect the search and rescue greatly. Facing with the conditions of trapped persons' lives and the complex environment, if we rely on UAVs to search and locate them, the localization system that caters our needs must satisfy shorter time, lower energy consumption, and higher accuracy. However, the current localization systems cannot accomplish that task.

GuideLoc presents a localization method for search and rescue, which is not only fast, but also energy-saving and accurate. It can improve the efficiency of search and rescue tasks. The basic idea of our proposed method is to plan a traversing path after the target region division estimates the targets location based on RSSI and AOA and use the GPS coordinates of the UAV as that of the target when the UAV flies just right over the target.

We believe that GuideLoc is able to advance the search and rescue technology in natural disasters, field investigation, geological exploration, and other special field, provide technical support for saving lives timely and accurately, and guarantee the rescuers' safety to some extent. At the same time, it provides a new paradigm for searching and locating persons in distress by using UAVs.

Disclosure

Anwen Wang and Xiang Ji should be considered co-first authors.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contributions

Anwen Wang and Xiang Ji contributed equally to this work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (no. 61572402, no. 61272461, no. 61602382, no. 61272120, and no. 61602381) and the Science Foundation of Northwest University (no. 15NW32 and no. 15NW31).

References

[1] A. Purohit and P. Zhang, "Sensorfly: a controlled-mobile aerial sensor network," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 327–328, Berkeley, Calif, USA, November 2009.

[2] H. Zhou, "Research on visible light shooting method for transmission line inspected by unmanned aerial vehicle," *New Technology & New Products of China*, vol. 18, pp. 9–10, 2015.

[3] A. A. Heidari and R. A. Abbaspour, "Autonomous UAV path planning for search and rescue missions in post natural disaster assessment based on novel G-BFOA algorithm," *Journal of Geomatics Science and Technology*, vol. 3, no. 4, pp. 41–52, 2014.

[4] Z. Wang, H. Chen, Q. Cao, H. Qi, Z. Wang, and Q. Wang, "Achieving location error tolerant barrier coverage for wireless sensor networks," *Computer Networks*, vol. 112, pp. 314–328, 2017.

[5] J. Wang, D. Fang, Z. Yang et al., "E-HIPA: an energy-efficient framework for high-precision multi-target-adaptive device-free localization," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 716–729, 2016.

[6] J. Wang, X. Chen, D. Fang, C. Q. Wu, Z. Yang, and T. Xing, "Transferring compressive-sensing-based device-free localization across target diversity," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2397–2409, 2015.

[7] N. C. Sina, "Shanghai global carnival," 2003 http://news.sina.com.cn/c/2003-07-02/03581259997.shtml.

[8] A. P. Subramanian, P. Deshpande, J. Gao, and S. R. Das, "Drive-by localization of roadside WiFi networks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 718–725, Phoenix, Ariz, USA, April 2008.

[9] A. Purohit, Z. Sun, F. Mokaya, and P. Zhang, "SensorFly: controlled-mobile sensing platform for indoor emergency response applications," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Network (IPSN '11)*, pp. 223–234, ACM/IEEE, Chicago, Ill, USA, April 2011.

[10] O. Holland, J. Woods, R. De Nardi, and A. Clark, "Beyond swarm intelligence: the ultraswarm," in *Proceedings of the 2005 IEEE Swarm Intelligence Symposium, SIS 2005*, pp. 225–232, Pasadena, Calif, USA, June 2005.

[11] Y. Zou, G. Wang, K. Wu, and L. M. Ni, "SmartScanner: know more in walls with your smartphone!," *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2865–2877, 2016.

[12] G. Wang, S. Zhang, K. Wu, Q. Zhang, and L. M. Ni, "TiM: fine-grained rate adaptation in WLANs," *IEEE Transactions on Mobile Computing*, vol. 15, no. 3, pp. 748–761, 2016.

[13] H. Luo, K. Wu, Y. Gong, and L. M. Ni, "Localization for drifting restricted floating ocean sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9968–9981, 2016.

[14] P. B. Sujit and D. Ghose, "Search using multiple UAVS with flight time constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 491–509, 2004.

[15] S. V. Spires and S. Y. Goldsmith, *Exhaustive Geographic Search with Mobile Robots Along Space-Filling Curves*, Springer, Berlin, Germany, 1998.

[16] J. Scherer, S. Yahyanejad, S. Hayat et al., "An autonomous multi-UAV system for search and rescue," in *Proceedings of the 1st ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use (DroNet '15)*, pp. 33–38, ACM, Florence, Italy, May 2015.

[17] Wikipedia, "Warflying[eb/ol]," 2013 https://en.wikipedia.org/wiki/Warflying.

[18] R. Queen, "Wireless aerial surveillance platform [eb/ol]," 2014 https://rabbit-hole.org/.

[19] Z. Liu, Y. Chen, B. Liu, C. Cao, and X. Fu, "HAWK: an unmanned mini-helicopter-based aerial wireless kit for localization," *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 287–298, 2014.

[20] F. Comtech, "802.11 phocus array antenna system by fidelity comtech," 2014 http://www.fidelity-comtech.com/.

[21] V. De Araujo, A. P. G. S. Almeida, C. T. Miranda, and F. De Barros Vidal, "A parallel hierarchical finite state machine approach to UAV control for search and rescue tasks," in *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO '14)*, pp. 410–415, SCITEPRESS, Vienna, Austria, September 2014.

[22] Y. S. Lee, J. W. Park, and L. Barolli, "A localization algorithm based on AOA for ad-hoc sensor networks," *Mobile Information Systems*, vol. 8, no. 1, pp. 61–72, 2012.

[23] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. Ni, "We can hear you with wi-fi!," *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2907–2920, 2016.

[24] H.-H. Hsu and C.-C. Chen, "RFID-based human behavior modeling and anomaly detection for elderly care," *Mobile Information Systems*, vol. 6, no. 4, pp. 341–354, 2010.

[25] S. Chen, D. Fang, X. Chen, T. Xia, and M. Jin, "Aerial wireless localization using target-guided flight route," in *Proceedings of the ACM Conference on Special Interest Group on Data Communication (SIGCOMM '14)*, pp. 587–588, Chicago, Ill, USA, August 2014.

[26] Y. Kim, S. Lee, S. Lee, and H. Cha, "A GPS sensing strategy for accurate and energy-efficient outdoor-to-indoor handover in seamless localization systems," *Mobile Information Systems*, vol. 8, no. 4, pp. 315–332, 2012.

[27] Z. Wang, J. Liao, Q. Cao, and H. Qi, "Barrier coverage in hybrid directional sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 222–230, 2013.

[28] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, "Friendbook: a semantic-based friend recommendation system for social networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 3, pp. 538–551, 2015.

[29] P. Egglestone, D. Ansell, and C. Cook, "DEMO: uavs in crowd tagged mountain rescue," in *MindTrek*, p. 286, ACM, New York, NY, USA, 2013.

[30] T. Tomic, K. Schmid, P. Lutz et al., "Toward a fully autonomous UAV: research platform for indoor and outdoor urban search and rescue," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[31] X. Liu, M. Dong, K. Ota, L. T. Yang, and A. Liu, "Trace malicious source to guarantee cyber security for mass monitor critical infrastructure," *Journal of Computer & System Sciences*, 2016.

[32] A. Liu, Q. Zhang, Z. Li, Y. Choi, J. Li, and N. Komuro, "A green and reliable communication modeling for industrial internet of things," *Computers & Electrical Engineering*, 2016.

[33] A. Birk, B. Wiggerich, H. Bülow, M. Pfingsthorn, and S. Schwertfeger, "Safety, security, and rescue missions with an unmanned aerial vehicle (UAV): aerial mosaicking and autonomous flight at the 2009 european land robots trials (ELROB) and the 2010 response robot evaluation exercises (RREE)," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 64, no. 1, pp. 57–76, 2011.

[34] X. Fu, N. Zhang, A. Pingley, W. Yu, J. Wang, and W. Zhao, "The digital Marauder's map: a new threat to location privacy," in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops, (ICDCS '09)*, pp. 589–596, IEEE, Montreal, Canada, June 2009.

[35] M. A. Goodrich, L. Lin, and B. S. Morse, "Using camera-equipped mini-UAVS to support collaborative wilderness search and rescue teams," in *Proceedings of the International Conference on Collaboration Technologies and Systems (CTS '12)*, pp. 638–638, Denver, Colo, USA, May 2012.

[36] C. Liu, D. Fang, Z. Yang et al., "RSS distribution-based passive localization and its application in sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2883–2895, 2016.

[37] L. Shangguan, Z. Li, Z. Yang, and M. Li, "Otrack: towards order tracking for tags in mobile rfid systems," *IEEE Transactions on Parallel & Distributed Systems*, vol. 25, no. 8, pp. 2114–2125, 2013.

[38] L. Shangguan and K. Jamieson, "The design and implementation of a mobile RFID tag sorting robot," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '16)*, pp. 31–42, Singapore, June 2016.

[39] D. Gusenbauer, C. Isert, and J. Krosche, "Self-contained indoor positioning on off-the-shelf mobile devices," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN '10)*, September 2010.

[40] J. Xu, W. Liu, F. Lang, Y. Zhang, and C. Wang, "Distance measurement model based on RSSI in WSN," *Wireless Sensor Network*, vol. 2, no. 8, pp. 606–611, 2010.

[41] Wikipedia, "Wardriving and warwalking [eb/ol]," 2013 https://en.wikipedia.org/wiki/Wardriving.

[42] J. Scherer, S. Yahyanejad, S. Hayat et al., "An autonomous multi-UAV system for search and rescue," in *Proceedings of the 1st ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use (DroNet '15)*, pp. 33–38, Florence, Italy, May 2015.

[43] C. Han, K. Wu, Y. Wang, and L. M. Ni, "WiFall: device-free fall detection by wireless networks," in *Proceedings of the 33rd IEEE Conference on Computer Communications (IEEE INFOCOM '14)*, pp. 271–279, Toronto, Canada, May 2014.

[44] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu, "Relative localization of RFID tags using spatial-temporal phase profiling," in *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI '15)*, pp. 251–263, USENIX, Oakland, Calif, USA, May 2015.

[45] A. Karimoddini, H. Lin, B. M. Chen, and T. Heng Lee, "Hybrid formation control of the unmanned aerial vehicles," *Mechatronics*, vol. 21, no. 5, pp. 886–898, 2011.

Hindawi
Mobile Information Systems
Volume 2017, Article ID 6160489, 11 pages
http://dx.doi.org/10.1155/2017/6160489

Research Article

A Wireless Location System in LTE Networks

Qi Liu, Rongyi Hu, and Shan Liu

Network Technology Research Institute, China Unicom, Beijing 100084, China

Correspondence should be addressed to Shan Liu; liushancandy@163.com

Received 29 September 2016; Revised 23 December 2016; Accepted 15 January 2017; Published 16 February 2017

Academic Editor: Xiaohong Jiang

Personal location technologies are becoming important with the rapid development of Mobile Internet services. In traditional cellular networks, the key problems of user location technologies are high-precision synchronization among different base stations, inflexible processing resources, and low accuracy positioning, especially for indoor environment. In this paper, a new LTE location system in Centralized Radio Access Network (C-RAN) is proposed, which makes channel and location measurement more available, allocation of baseband processing resources more flexible, and location service capability opening. The location system contains more than two antenna clusters, and each of them gets time-difference-of-arrival (TDOA) of sounding reference signals (SRSs) from different antennas. Then, based on data provided by location measurement units (LMUs), the location information server calculates TDOAs and derives the users' position. Furthermore, a new location algorithm is raised which can achieve distributed antennas collaboration and centralized location computing. And an improved optimized algorithm with the best TDOA selection is proposed. Finally, simulations are given out to verify the efficiency of the proposed algorithm in this LTE location system.

1. Introduction

In recent years, wireless location technologies, especially technologies for indoor environment, have attracted attention of many powerful companies including network operators. Although the Global Positioning System (GPS) could provide personal location services, the 3rd Generation Partner Project (3GPP) has also proposed user location projects in LTE networks. Particularly, terminals maybe cannot receive GPS signals in indoor environment due to the weak received signal and multipath reflection. As LTE networks are being deployed around the world, location technologies with LTE networks need to be studied further, which could enhance user location services in all different environments [1–3].

There are mainly two challenges in traditional LTE location system [4, 5]. One is the limitation of indoor networks architecture, causing the difficulties of distinguishing positioning signals from DAS (distributed antenna system), and time synchronization among different base stations. And the other is the low positioning accuracy due to the location algorithm and the complex LOS/NLOS communication environment [6, 7].

According to the study of previous work [8–11], time-of-arrival (TOA), time-difference-of-arrival (TDOA), and hybrid location algorithms based on time-related measurements are the research hotspot at present. Most location systems used to take TOA/TDOA techniques. However, few location algorithms or schemes are proposed for LTE systems with 20 MHz bandwidth, especially for complex scenarios such as indoor positioning. So new studies or algorithms related to LTE systems are needed in order to improve positioning performances of cellular networks, especially location accuracy and implementation complexity [12, 13].

In this paper, a wireless location system in LTE networks based on C-RAN (Cloud Radio Access Networks) architecture is introduced [14–16]. To be specific, traditional cellular Radio Access Networks (RAN) usually consist of many standalone base stations. It is a proposed architecture for future cellular networks called C-RAN or Cloud-RAN, which allow distributed transmission of RF signals from remote radio heads (RRH), centralized processing of baseband signals in baseband units (BBU) pool, and real-time virtualization and cloud computing. For the cellular location services, these functions can better support dynamic computing resource

allocation and flexible selection of RF antennas, which would guarantee both communication quality and positioning accuracy. Meanwhile, high-precision synchronization among different base stations for TDOA measurement is easily solved by involving the LMU (location measurement unit), which can independently measure and calculate the TDOAs of different antennas. By the way, both stand-alone LMU and base station with LMU function are supported in our system.

In this paper, we also propose a new method that calculates and analyzes the TDOAs of sounding reference signals (SRSs) in uplink channel of C-RAN. According to the TDOAs of SRS and classical nonlinear least squares (NLLS) algorithm, a new location algorithm based on the LTE location system is proposed. It combines information from different antenna clusters provided by LMUs, which does not require synchronization among BSs. In addition, we propose an improved algorithm by optimizing the selection of TDOAs in order to enhance the accuracy of the users' location. Both the experiment test and simulation results show the performance gain of the proposed algorithms and verify the efficiency.

The remainder of the article is organized as follows. A wireless location system in LTE networks is given in Section 2, including the system model, the performance of SRS correlation for TDOA estimation, and the proposed location algorithms. In Section 3, the simulation and test results are described. Finally, Section 4 concludes the paper.

2. A Wireless Location System in LTE Networks

2.1. System Model. Generally, in order to ensure the location performance, location systems need to be low-latency and high-precision, especially for multiuser systems. So the location procedure shall be completed in a short time (e.g., 10 ms) in LTE networks, including channel measurement, large-scale computing, and information feedback. Meanwhile, with the development of positioning technology, a software update of the related equipment comes naturally. Traditional cellular network architecture obviously cannot support these features. However, in C-RAN architecture, operators can rapidly deploy or upgrade their networks and make full use of base station and antennas resources.

In this section, we introduce the user location mechanism in LTE system with the C-RAN architecture, where Uplink-Time Difference of Arrival (U-TDOA) technology is applicative. Here, location measurement of SRSs and channel measurement functions are achieved in LMUs, RF antennas or RRHs are flexibly selected and managed, and location algorithms and computing are centrally realized in BBU pool. Figure 1 gives the LTE location system model in C-RAN architecture.

Firstly, M ($M \in \mathbb{N}$) receiver units are used for collecting the LTE signals from the target User Equipment (UE) or Mobile Terminals (MT). In other words, there are M antenna clusters deployed and N_m ($N_m \in \mathbb{N}, N_m \geq 2$) antennas in cluster-$m$ ($m = 1, 2, \ldots, M$), and the antenna gap is usually set as 1 to 10 meters for more accurate location measurement. By the way, each location antenna cluster should contain at least two antennas to insure the available measurement of TOA/TDOA.

Secondly, there are M wireless location measurement units (LMUs) which connect the location servers in this system. In cluster-m, a LMU receives SRSs from each antenna in this cluster, extracts the time differences of SRSs from different antennas, and calculates TDOAs. It is worth mentioning that there is no need for synchronization between different LMUs or base stations.

Thirdly, LMUs report location measurement data to the location information server, including TDOAs and related antenna clusters' information. The server collects time differences from M clusters and gets user location information by the positioning algorithm. In this architecture, the time delay of location information transmitting will be less than traditional cellular architecture.

Figure 2 describes functions and signal processing in this location system. The RF front ends receive the LTE signals and transmit the signals to LMUs. LMUs process the baseband analog signals through Analog to Digital signals conversion (A/D) at first, then exact single user's SRS, and calculate TDOAs. Finally, location information server estimates the user's position through the location algorithm and stores the information in databases of BBU pool.

2.2. SRS Correlation Performances for TDOA Estimation. Two methods for TDOA estimation are widely used. One is calculating the cross-correlation between two received signals and getting TDOA [17]. But it does not work well in multipath environments especially for the indoor localization [18]. Another is estimating TOA at first by cross-correlation between the received signal and the transmitted pseudorandom sequence. And then the difference between the two TOA estimations is calculated, assuming that all receivers are synchronized. In this paper, TDOA is detected and estimated based on the second method, upon which the TDOA error model is built.

SRS is a reference signal in LTE networks. It is usually used to figure out the channel quality of uplink path. Mobile Terminals (MT) send SRS at the last symbol of a slot. A MT can transmit SRS signal every 2 subframes at the most and every 32 frames (320 subframes) at the least. A signaling parameter transmitted by eNodeB, named as SRS-ConfigIndex, tells UEs the periodicity of SRS transmission, and the period can be 2, 5, 10, 20, 40, 80, 160, and 320 milliseconds (ms) [5]. SRS is generated by Zadoff-Chu sequences, which are good candidates as their ideal correlation properties.

In this section, we evaluate SRS correlation performance in LTE networks by simulations. In the simulations, SRS is generated by a MT. Then SRS and useful information data are combined in a frame according to 3GPP LTE standards. The eNodeB receives the uplink signals and maps frequency signals to time domain, obtaining the Single-Carrier Frequency Division Multiple Access (SC-FDMA) signals. Then the received signals are correlated with the local SRS, which is the same with SRS generated by the MT. So the correlation peak is found. The peak time is what we want to get for our location system, which is helpful for obtaining the accurate

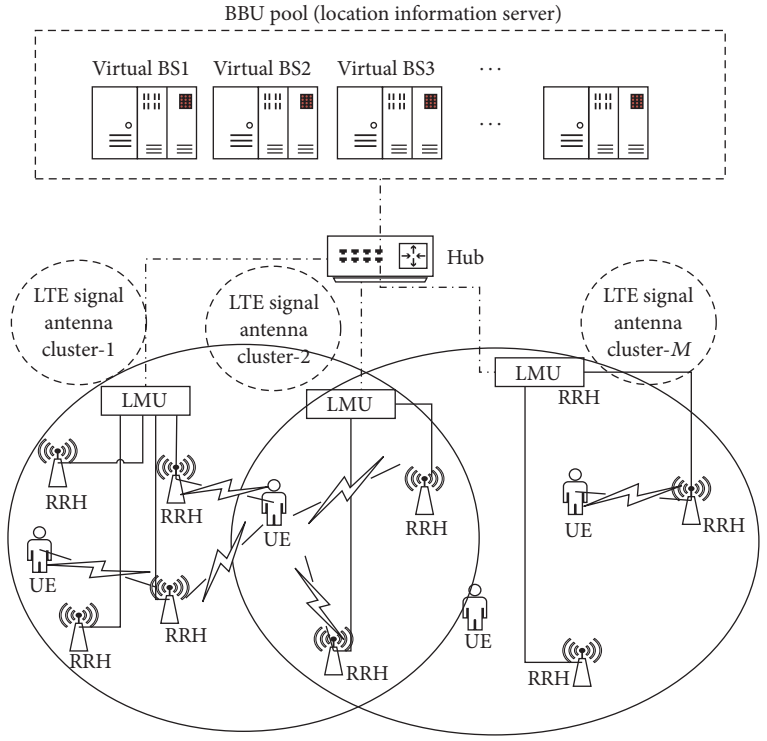


Figure 1: LTE location system in C-RAN architecture.

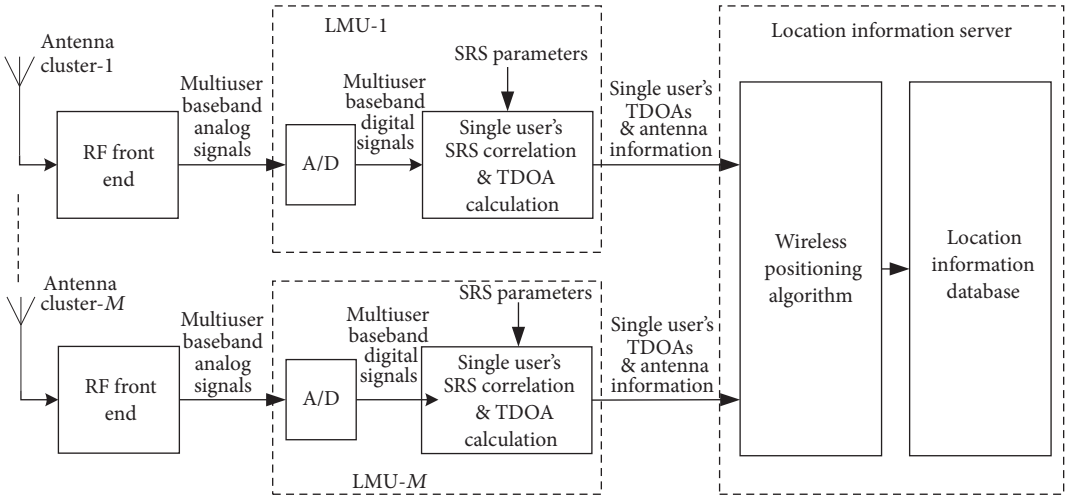


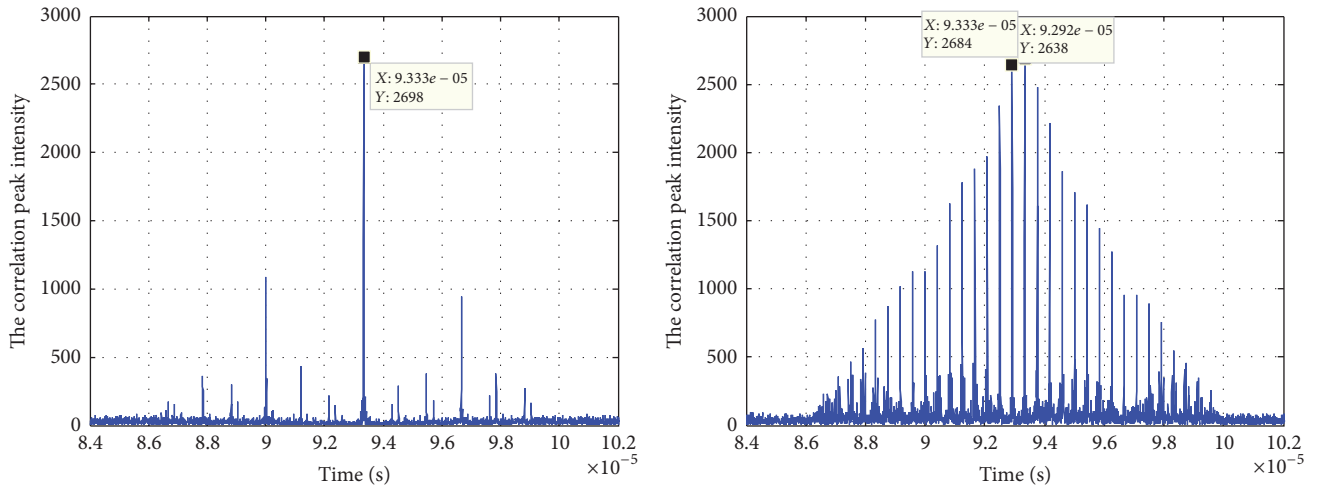
Figure 2: LTE location system function diagram.

TDOA. To be simplified, the Additive White Gaussian Noise (AWGN) channel model is applied in the simulations. And the default sampling rate is 30.72 MHz.

In LTE networks, SRSs from different users are transmitted by time-division mode or code-division mode. Their correlation results are shown in Figure 3. In time-division mode, as Figure 3(a) shows, SRS from a MT is easy to distinguish from others', which has a perfect performance of correlation peak. Then, we can easily get the TOAs and TDOAs of the different signals by detecting and analyzing their SRSs.

However, as Figure 3(b) shows, in the case of code-division mode, there are more side peaks in correlation results. The time interval between each side peak is approximately 410 ns. It is greater than the accuracy of time advance (TA), which is a parameter in LTE networks. So we can use TA to assist in finding the main peak time.

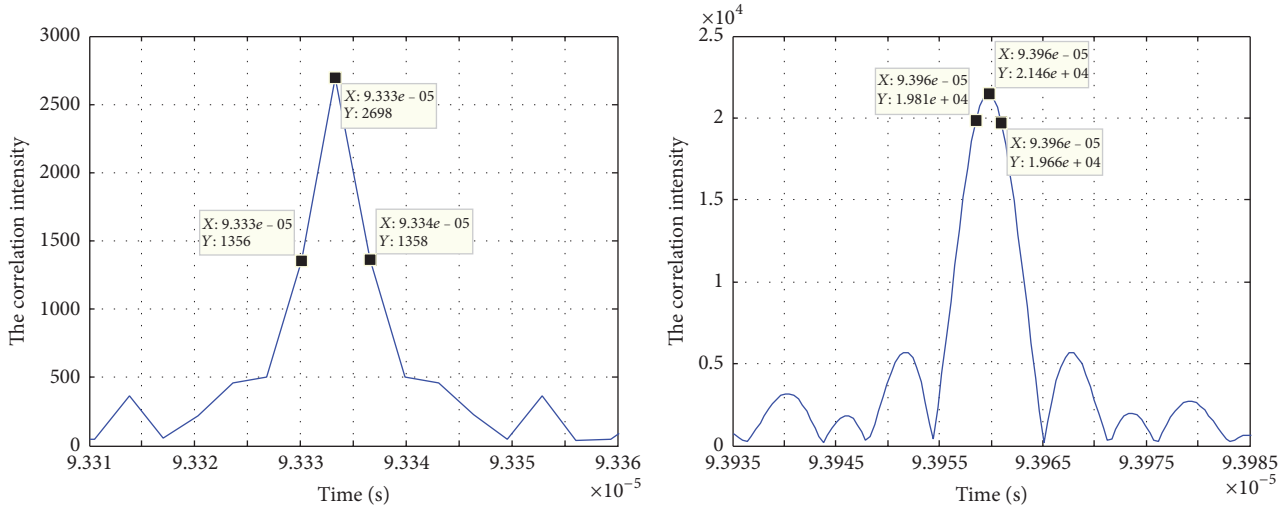
In addition, we can take advantage of oversampling before the SRS correlation, upon which the errors of time differences can be reduced. Figure 4 shows SRS correlation performance comparison between nonoversampling and oversampling. It is obvious that the correlation curve of oversampling is more



(a) Time-division mode

(b) Code-division mode

Figure 3: SRS correlation performance in time-division and code-division mode.



(a) Default sampling rate

(b) Oversampling

Figure 4: SRS correlation performance comparison between nonoversampling and oversampling.

smooth, and this could increase the accuracy of peak time with increased intensity of sample points. So oversampling can be used for TDOA detection.

2.3. Location Algorithm. First, we assume one cluster with N antennas in this location system, and $\mathbf{x}$ represents a MT's position to be estimated. $\|\mathbf{x} - \mathbf{x}_n\|$ is the true distance between antenna-n and the MT, where $\mathbf{x}_n$ represents antenna-n's position $n = 1, 2, \ldots, N$. Set antenna-1 as an anchor; then $h_n(\mathbf{x}) = \|\mathbf{x}-\mathbf{x}_n\|-\|\mathbf{x}-\mathbf{x}_1\|$ represents the difference of distance. It is defined that $\mathbf{h}(\mathbf{x}) = [h_2(\mathbf{x}), \ldots, h_n(\mathbf{x})]^{\mathrm{T}}$, where $[\cdot]^{\mathrm{T}}$ is the transpose operation. $\mathbf{r}$ represents measured difference of distance, shown as (1). Here, $\boldsymbol{\varepsilon}$ is measurement error and obeying normal distribution $(0, {\sigma_n}^2)$.

$$\mathbf{r} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\varepsilon}. \tag{1}$$

$\mathbf{h}(\mathbf{x})$ is linearized about an initial reference position $\mathbf{x}_0$ ignoring high-order terms by Taylor series expansion, shown in (2). Here $\mathbf{H}_0$ is the Jacobian matrix of $\mathbf{h}(\mathbf{x})$ at $\mathbf{x}_0$.

$$\mathbf{h}(\mathbf{x}) \approx \mathbf{h}(\mathbf{x}_0) + \mathbf{H}_0 * (\mathbf{x} - \mathbf{x}_0). \tag{2}$$

So (1) can be expressed as follows:

$$\begin{aligned} \mathbf{r} &= \mathbf{y} + \mathbf{h}(\mathbf{x}_0) - \mathbf{H}_0\mathbf{x}_0, \\ \mathbf{y} &= \mathbf{H}_0\mathbf{x} + \boldsymbol{\varepsilon}. \end{aligned} \tag{3}$$

Here, $\mathbf{y}$ is a linearized approximation of $\mathbf{r}$. Then $\mathbf{x}$ can be calculated by least squares (LS) estimation method, shown in (4). Here, $\mathbf{R}$ is the covariance matrix of $\boldsymbol{\varepsilon}$, and $(\cdot)^{-1}$ is the inverse matrix operation.

$$\hat{\mathbf{x}} = \left(\mathbf{H}_0^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_0\right)^{-1}\mathbf{H}_0^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}. \tag{4}$$

In addition, the algorithm also extends to the scenario of M clusters in the location system. Assuming that each cluster can have different number of antennas (at least 2), we can combine information getting from M clusters to calculate the MT's position by (4), where $\mathbf{R}'$ is the covariance matrix of $\boldsymbol{\varepsilon}'$:

$$\mathbf{y}' = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_M \end{bmatrix},$$

$$\mathbf{H}_0' = \begin{bmatrix} \mathbf{H}_{01} \\ \mathbf{H}_{02} \\ \vdots \\ \mathbf{H}_{0M} \end{bmatrix}, \tag{5}$$

$$\boldsymbol{\varepsilon}' = \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \\ \vdots \\ \boldsymbol{\varepsilon}_M \end{bmatrix}.$$

As the closed form of the estimation position cannot be derived, we apply an iterative method to minimize the error according to Taylor series expansion as follows. To be specific, the estimated position for the kth iteration should be derived as

$$\hat{\mathbf{x}}(k) = \left(\mathbf{H}_0'^{\mathrm{T}} \mathbf{R}'^{-1} \mathbf{H}_0'\right)^{-1} \mathbf{H}_0'^{\mathrm{T}} \mathbf{R}'^{-1} \mathbf{y}'. \tag{6}$$

Algorithm 1 (the proposed algorithm).

Step 1. Initialize the position $\mathbf{x}_0$, and calculate $\mathbf{y}'$, $\mathbf{H}_0'$, $\mathbf{R}'$ with (3) and (5).

Step 2 (iteration). (1) Estimate $\hat{\mathbf{x}}(k)$ with (5) and (6), where $k \in \mathbb{N}_+$ and $\hat{\mathbf{x}}(0) = \mathbf{x}_0$.

(2) If $\|\hat{\mathbf{x}}(k) - \hat{\mathbf{x}}(k-1)\| \le \sigma$, stop; else, continue;

(3) Set $\mathbf{x}_0 = \hat{\mathbf{x}}(k)$; go to Step 1.

Step 3. Return the final estimated position, until a preset number of iterations are reached or until convergence.

2.4. Improved Location Algorithm. In LTE systems, we cannot ensure the links between signal sources and receivers (or the receiving antenna clusters) are always in line of sight (LOS) due to the multipath. Though all TDOAs getting from M clusters can be gathered together for calculation, the bad estimated TDOAs caused by NLOS or penetration may make the performance of the proposed Algorithm 1 worse.

To deal with the bias caused by bad estimation, we proposed an improved algorithm of TDOA selection, which is mainly based on kNN (k-NearestNeighbor) algorithm, a nonparametric method used for classification and regression. First, by kNN classification, each estimated position calculated by randomly divided TDOAs is classified by a majority vote of its neighbors, with the estimation being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). Particularly if $k = 1$, it is simply assigned to the class of that single nearest neighbor. Then, by kNN regression (which is simplified as calculating the average of the values of k nearest neighbors), the best estimated positions and their reliable TDOAs with property value are selected, and the worst ones are got rid of.

The detailed procedure of the improved algorithm of TDOA selection is as follows. Firstly, a coarse selection is used. Set an upper limit of TDOA (T_{upper}) according to the maximum distance between receiving antennas in one cluster. Then, for all N_0 TDOAs, get rid of the obviously bad ones if

$$\mathrm{TDOA}(i) > T_{\mathrm{upper}}, \quad i = 1, 2, \ldots, N_0. \tag{7}$$

Secondly, the remaining TDOAs, whose number is assumed as N_1, will be selected in an accurate way. Here to simplify, we denote TDOAs as $T(i)$, $i = 1, 2, \ldots, N_1$. Choose M $(3 \le M \le N_1)$ of the TDOAs randomly,

$$A_1, A_2, \ldots, A_{C_{N_1}^M} \subset \{T(1), T(2), \ldots, T(N_1)\}, \tag{8}$$

where $A_i \triangleq \{T(x_{i1}), T(x_{i2}), \ldots, T(x_{iM})\}$ and $\{x_{i1}, x_{i2}, \ldots, x_{iM}\} \subset \{1, 2, \ldots, N_1\}$, $i = 1, 2, \ldots, C_{N_1}^M$.

And calculate the predicted positions with (5) and (6) in Algorithm 1 for all $C_{N_1}^M$ cases, expressed as $\mathrm{pos}(i)$ $(i = 1, 2, \ldots, C_{N_1}^M)$.

Then according to the distribution of positions, the range of all positions is divided into continuous J $(J \ge 4)$ equivalent segmented blocks. The cumulative probability of each segment j $(j = 1, 2, \ldots, J)$ is expressed as

$$P(j) = \frac{\sum_{i=1}^{C_{N_1}^M} p_j(i)}{C_{N_1}^M}, \quad j = 1, 2, \ldots, J, \tag{9}$$

where

$$p_j(i) = \begin{cases} 1, & \mathrm{dis}(i) \in \mathrm{segment}(j) \\ 0, & \mathrm{dis}(i) \notin \mathrm{segment}(j), \end{cases} \tag{10}$$

$$j = 1, 2, \ldots, J, \ i = 1, 2, \ldots, C_{N_1}^M.$$

Therefore, according to the cumulative probability distribution, we can get the densest segment, expressed as segment($j_{\max}$). Furthermore, the related positions in segment($j_{\max}$) are obtained, whose number is added up as L. Here, $C_{N_1}^M / J < L \le C_{N_1}^M$. Then the median position of L points can be easily obtained, called point A (pos_A). The distance between the positions and point A can be expressed as

$$\mathrm{dis}(i) = \left\|\mathrm{pos}(i) - \mathrm{pos}_A\right\|, \quad i = 1, \ldots, L. \tag{11}$$

By sorting all these distances, the former K $(K \le L)$ values can be achieved, which are considered as the nearest

neighbor points and most likely estimated positions of target location. Here, K depends on the size of sample data.

Finally, according to the K nearest neighbor points, the related TDOAs for $\text{pos}'(k)$ $(k = 1, 2, \ldots, K)$ are easy to achieve which are marked as $A_{jk} = \{T(x_{j1}), T(x_{j2}), \ldots, T(x_{jM})\}$, where $A_{jk} \in \{A_1, A_2, \ldots, A_{C_{N_1}^M}\}$, $j \in \{1, 2, \ldots, C_{N_1}^M\}$. Then we can calculate the times of selected TDOAs as the weight of TDOA(i):

$$W(i) = \sum_{k=1}^{K} \sum_{m=1}^{M} w_{k,m}(i), \tag{12}$$

where

$$w_{k,m}(i) = \begin{cases} 1, & \text{if } i = x_m(j),\ T(i) \in A(k) \\ 0, & \text{if } i \neq x_m(j) \text{ or } T(i) \notin A(k), \end{cases} \tag{13}$$

$$m = 1 \cdots M,\ k = 1 \cdots K,\ j = 1, 2, \ldots, C_{N_1}^M,\ i = 1 \cdots N_1.$$

By sorting the weights of these TDOAs, we can select the top N_2 $(M \leq N_2 \leq N_1)$ TDOAs, which are the optimized selection of TDOAs. Based on these TDOAs and Algorithm 1 above, we can recalculate the optimized estimated position.

The steps of the improved Algorithm 2 are as follows.

Algorithm 2 (the improved algorithm).

Step 1 (coarse selection). Set an upper limit of TDOA, and get rid of the bad TDOAs with (7).

Step 2 (accurate selection). (1) Calculate the positions with (5), (6), and (8) by random M TDOAs for all C_N^M cases.

(2) Calculate the cumulative probability of continuous J segmented blocks with (9), and get the densest segment $j_{\max}$ and the related L positions.

(3) Select the K nearest neighbor points with (11) after resorting the distances.

(4) Weight the related TDOAs with (12), and select top N_2 TDOAs.

Step 3. Calculate the final optimized position with the selected TDOAs using Algorithm 1.

3. Simulation and Test Results

3.1. Simulation Results. In this section, we mainly evaluate the performance of proposed location algorithms in the LTE location system through several simulations.

Some common simulation settings are as follows: SRS is configured in time-division mode among different users. To be simplified, the Additive White Gaussian Noise (AWGN) channel model is applied in the simulations. The default sampling rate is 30.72 MHz, and 10 times oversampling is used in this location system. We assume that the simulation scenario is a room whose size is 20 m × 20 m × 3 m and c $(c = 1, 2, \ldots, 5)$ antenna clusters are deployed in it. Each cluster contains m $(m = 2, 3, 4)$ antennas.

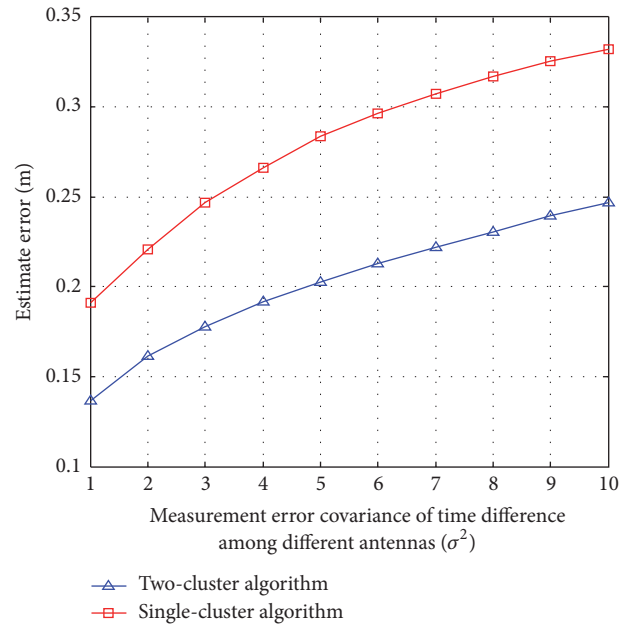


Figure 5: The performance of LTE location system between single cluster and 2 clusters.

In the first simulation group, users are randomly distributed. Firstly, performance comparisons between single cluster (with 4 antennas) and 2 clusters (one with 4 antennas and the other with 2 antennas) are given in Figure 5. The location accuracy of 2 clusters is better than that of single cluster. Its estimating error is improved about 25%.

Furthermore, Figure 6 shows the cumulative distribution function (CDF) of estimation error with different antenna clusters (clust_N), each of which has 4 antennas. Owing to the fact that simulation scenario (a room 20 m × 20 m × 3 m) is limited, the performance of 1 cluster is not quite good but seems to be nearly perfect when there are above 2 antenna clusters deployed. It is easy to find that the estimation error goes smaller with antenna clusters number increasing, and the estimation error is controlled within 1.1 meters when there are more than 2 clusters. Particularly, for 5 clusters, the estimation error can be even all controlled below 0.5 meters. In some degree, the results show that the more antenna clusters can bring better performance.

In the second simulation group, we assumed that 2 clusters are used and the user/MT moves in a certain area among the clusters. The location performances are compared when the user is in different positions. As shown in Figure 7, as the user moves from the edge of the area to the center in a line, the location accuracy is obviously improved. For example, when the user moves at the center areas among the antenna clusters like (3, 1) or (3.5, 1), the estimating error is lower than 0.1 meters, rather better than that of the edge areas especially at (0, 1) and (4.5, 1).

In the last simulation group, assuming the MT randomly distributed, we give out the performance by the cumulative distribution function (CDF) of estimation error for different clusters ($\text{clust}_N = 3, 4, 5$). And the comparison between the proposed algorithms, that is, Algorithm 1 based on the

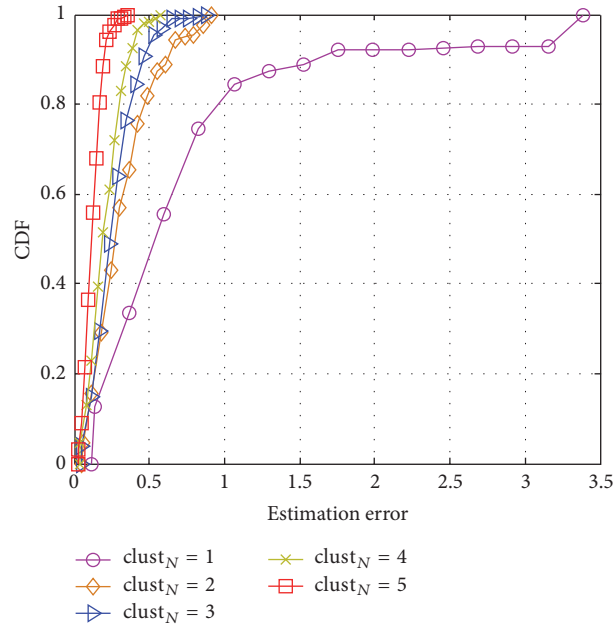


FIGURE 6: The performance of LTE location system with different antenna clusters, where clust$_N$ = 1, 2, . . . , 5.

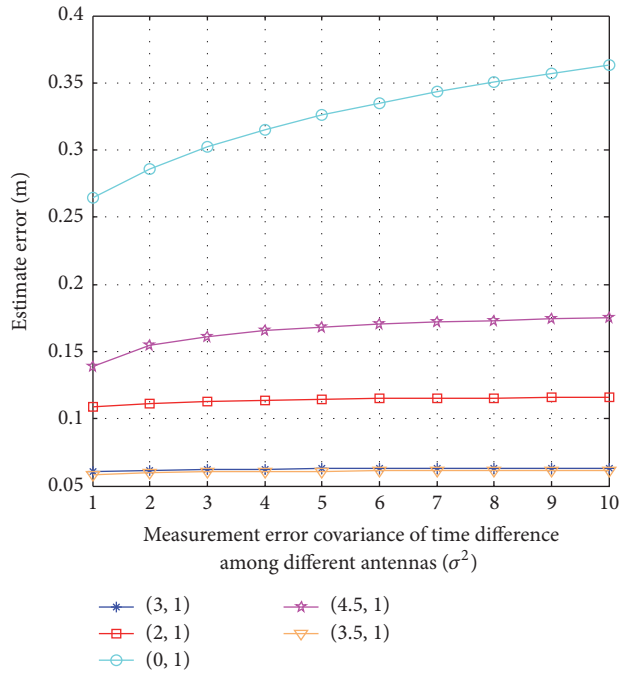


FIGURE 7: The performance for user in different positions in LTE location system.

weighted least square method and the improved Algorithm 2 based on TDOA selection, is also analyzed. Here, we assumed the good TDOAs (N_2) are mostly accounted for 75% of all (N_0).

As shown in Figure 8, it is easy to find that the estimation error goes smaller with clusters number increasing for both location algorithms. Meanwhile, for all three cases with

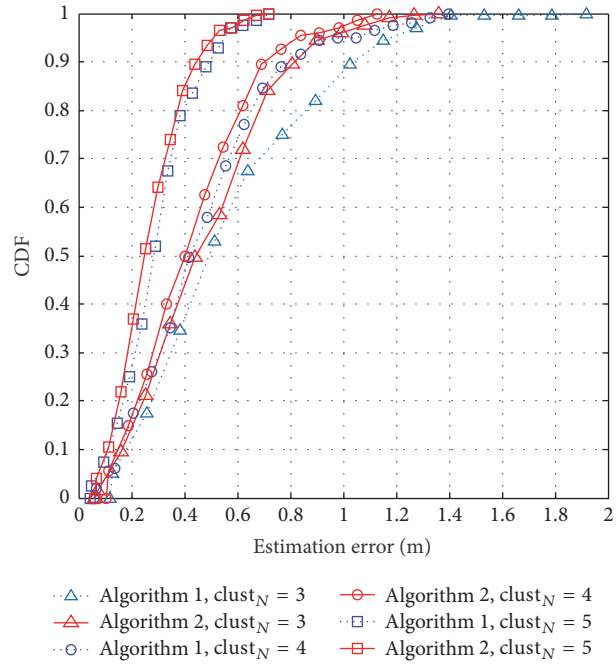


FIGURE 8: The performance comparison of the proposed two algorithms, where clust$_N$ = 3, 4, 5, and each has 4 antennas.

different clusters, the cumulative probability of estimation error with Algorithm 2 has a better performance, especially when there are 3 antenna clusters. At the expense of increased time complexity, Algorithm 2 can obviously improve the accuracy of location, which proves that the optimization of TDOA selection is effective and feasible.

3.2. Test Results. In order to testify the performance of the proposed system and algorithms, we have done some experiment in a conference room. The size of the room is 12.2 m ∗ 17 m ∗ 3.8 m and 2 antenna clusters are deployed in it; each cluster has 4 antennas. The experiment system is composed as we describe in Section 2.

LMUs are used to process the data received by antenna clusters, and the composition and features of LMU are presented in detail as Figure 9. Each unit has 4 receiving channels, and the signal in each channel is expected to be sampled approximately at the same time. Due to the fact that a small timing error may cause a large mistake during estimation for TDOA localization, it is imperative to precisely synchronize each channel. Through location measurement module, location data can be collected and sent to the location sever for analyzing, and the estimated position is finally calculated by the proposed algorithms. In a word, this design can release the demand of synchronization between base stations and save the cost of network side.

Firstly, we try to locate fixed targets in the conference room by using the deployed system. Figure 10 shows the positioning results of 6 fixed targets. Each picture presents the estimated position distribution of 50 test results, where the blue star is the target (as the center of a blue circle with radius

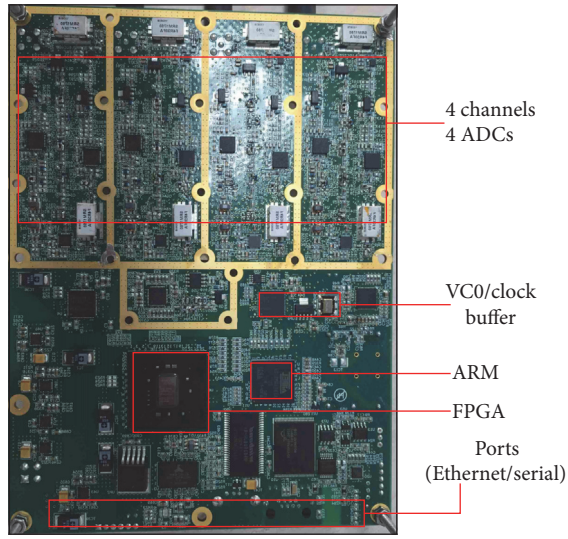


Figure 9: The composition of location measurement unit.

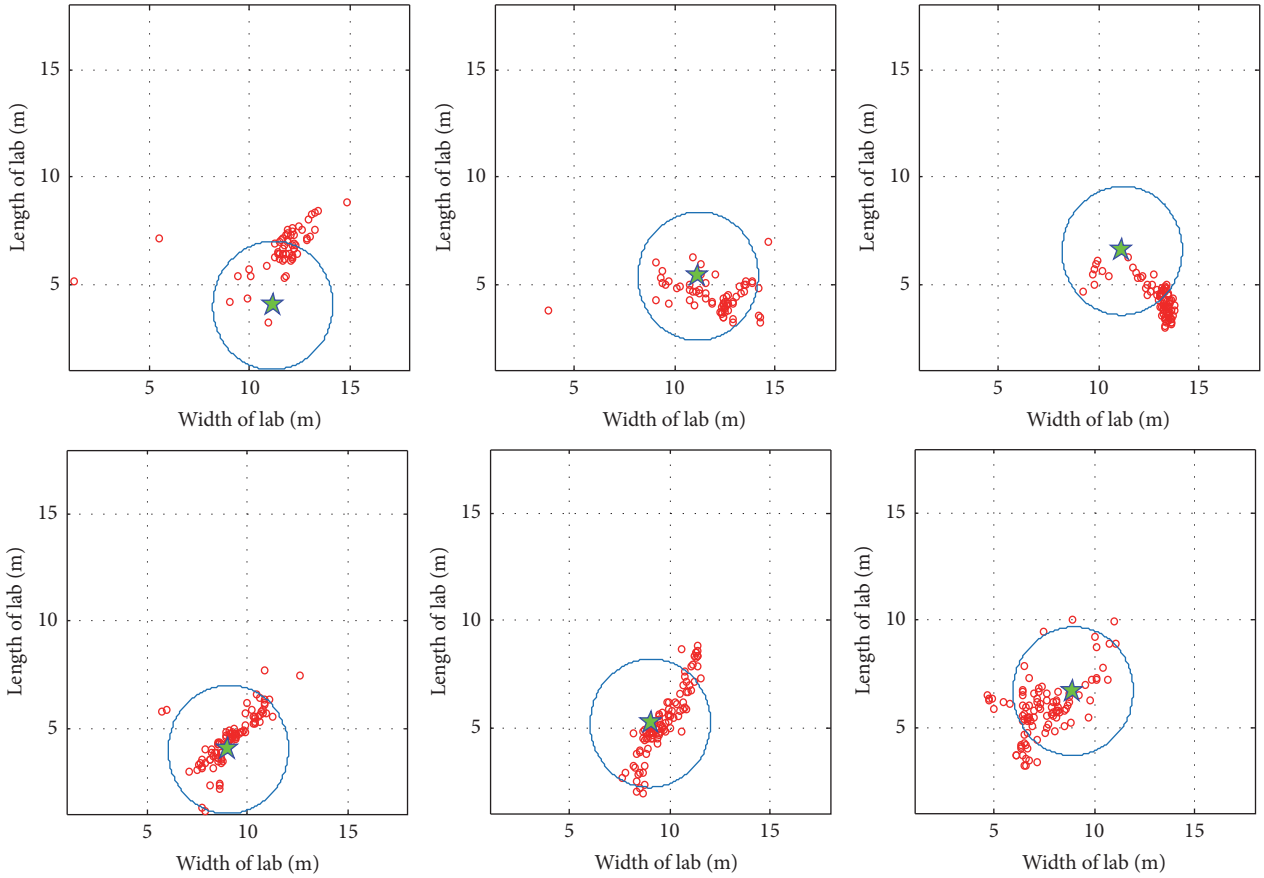


Figure 10: Multiple measurement results.

of 3 meters) and the red points are the 50 estimations. The final position results are shown in Figure 11 by aggregating the multiple measurement results.

From the statistic analysis about the positioning results in Table 1, as we can see, the possibility of measurement error less than 3 m is about 80%. And the maximum estimation error is 3 m, the minimum is 0.47 m, and the average is 1.3 m. Due to the complexity of real environment, the measurement error is larger than the simulation results but still can satisfy the indoor positioning precision.

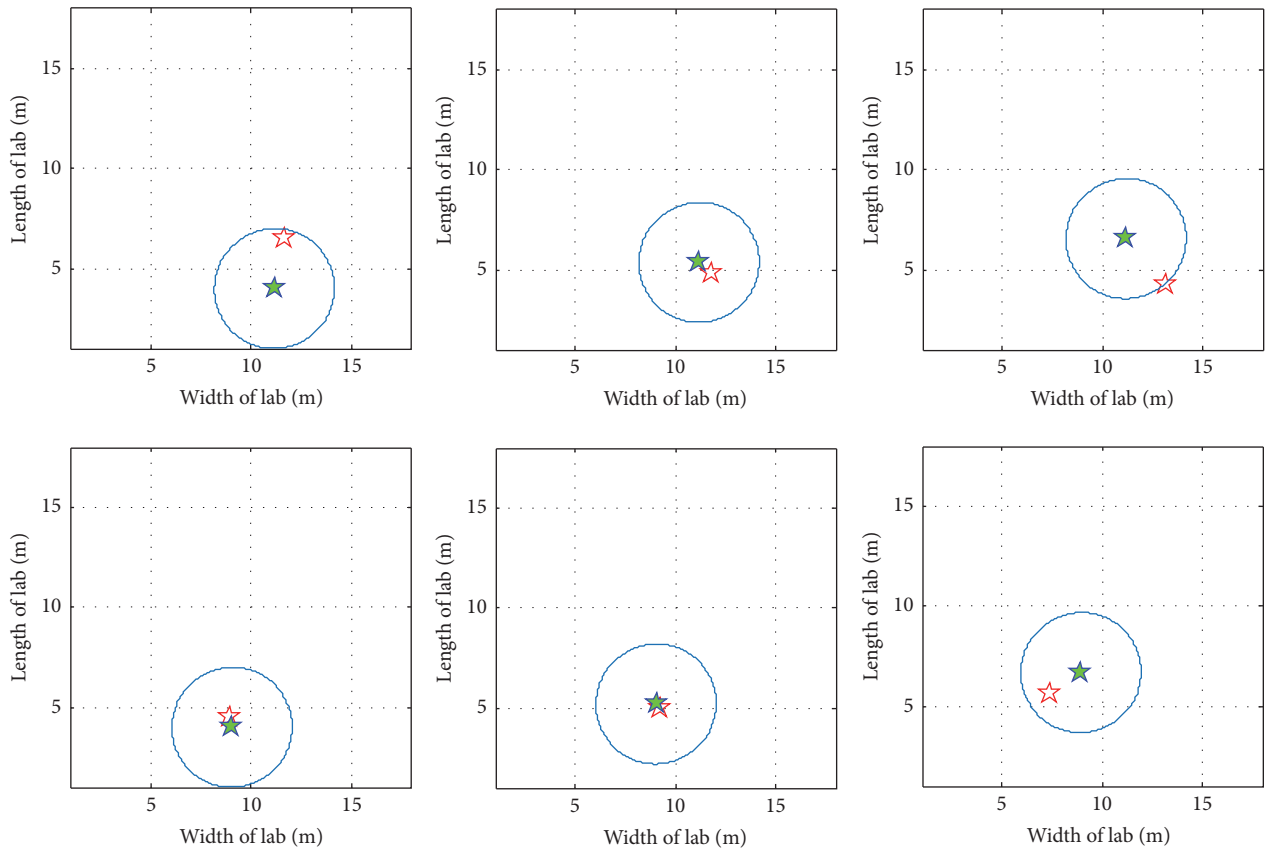


Figure 11: Final estimation of position.

Table 1: Statistic analysis of measurement results.

Number	<2 m (%)	<3 m (%)	<4 m (%)	Measurement error (m)
1	14.8	58.6	87.9	2.50
2	69.1	85.3	92.7	0.86
3	26.4	77.0	100	3.00
4	85.4	95.8	100	0.47
5	72.0	88.0	100	0.53
6	49.5	83.8	95.0	1.90
Average	53.5	79.3	92.0	1.30

Then we testify the UE's path tracing performance; Figure 12 presents the path tracking result of the UE moving at walking speed. The red line denotes the true path, and the blue line is the estimated trajectory. It can be seen that the estimated trajectory is very close to the true path and the measurement error is below 1 m. Comparing with the fixed measurement, the system of dynamic positioning is more stable and more precise. The possible explanation of the result is that UE's movement makes the error furtherly comply with Gaussian distribution, which can increase the precision by averaging multiple measurements.

In addition, both the simulation and test results show that the LTE location system and algorithm have good performance. In order to deal with the complexity of real scene, we improve our algorithm by selecting TDOA and get the final position result by aggregating multiple position points. This method can reduce the influence of multipath.

But our test environment is still relatively simple and hollowness. In a more complex environment, especially with serious multipath like indoor case, our algorithms may not work well. Other improved algorithms or multipath mitigation technologies should be taken into consideration. For example, in cellular communication systems, the hybrid TDOA and the Angle Of Arrival (AOA) location algorithm can reach higher accuracy than traditional TDOA method does.

And terminal-side hybrid locations with WiFi, Bluetooth, or other RF location systems are not precluded to improve the poor precision of wireless location in cellular network [19, 20]. Besides, the assistant positioning technologies of terminal-side equipment like inertial navigation and better filtering algorithms of tracking will be good for location estimation and prediction [21].

4. Conclusions

This paper introduces a new user location system in LTE networks with C-RAN architecture. In this system, uplink SRS

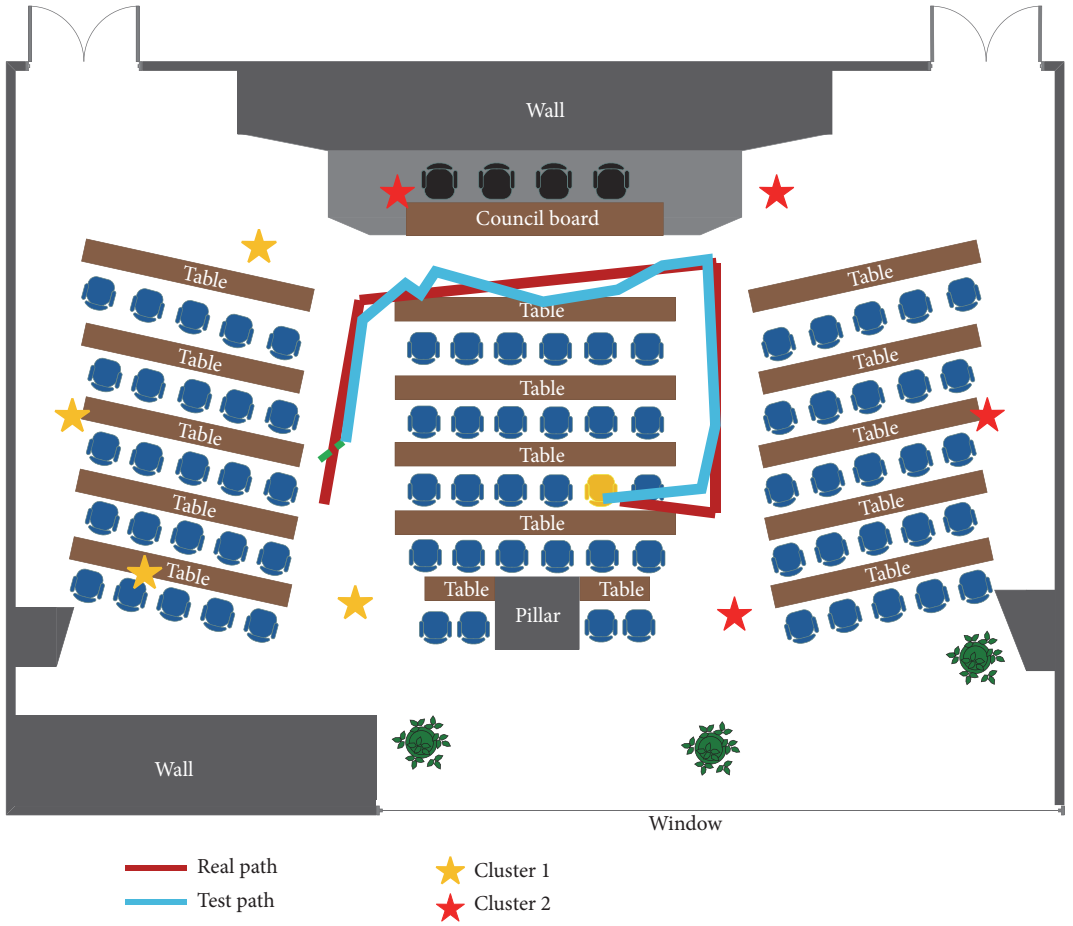


Figure 12: The test result of path tracking.

is used as monitoring signals. LMUs are responsible for signal detection and TDOA estimation. The location information server calculates the user's position through the proposed algorithms. Furthermore, based on the basic TDOA algorithm in most location systems, a new location algorithm is raised. It reduces synchronization demands among distributed antennas, which is verified to have a good performance on position estimation. In addition, to avoid the effects of NLOS and multipath overlap, we also propose an improved optimization algorithm by selecting the best TDOAs. Simulation results show that the proposed algorithm has performance gain and improves the efficiency and accuracy of the location system in C-RAN architecture.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by National Key Research and Development Plan Grants no. 2016YFB0502000. The authors would like to acknowledge the helpful pieces of advice from Project Partner Professor Deng Zhongliang in Beijing University of Posts and Telecommunications and Professor Liu Huaping, Wang Youxiang, Ph.D., Qiu Jiahui, Ph.D., Ma Yue, Zhang Wenhao, and Chen Yi.

References

[1] P. J. Voltz and D. Hernandez, "Maximum likelihood time of arrival estimation for real-time physical location tracking of 802.11a/g mobile stations in indoor environments," in *Proceedings of the Position Location and Navigation Symposium (PLANS '04)*, pp. 585–591, April 2004.

[2] C.-P. Yen and P. J. Voltz, "Indoor positioning based on statistical multipath channel modeling," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, article no. 189, 2011.

[3] Q. Liu and Y. Ma, "A wireless location system in LTE networks," in *Proceedings of the 2nd IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW '15)*, pp. 160–161, June 2015.

[4] S. Wei, L. Qi, L. Yiqun, B. Meng, C. Guoli, and Z. Hongke, "Small cell deployment and smart cooperation scheme in dual-layer wireless networks," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 929805, 7 pages, 2014.

[5] S. Sesia, I. Toufik, and M. Baker, *LTE—The UMTS Long Term Evolution*, John Wiley & Sons, 2011.

[6] H. Wu, Y. Chen, N. Zhang et al., "The NLOS error mitigation joint algorithm in hybrid positioning system combining DTMB and GPS," in *Proceedings of the China Satellite Navigation Conference (CSNC '12)*, pp. 287–296, Guangzhou, China, May 2012.

[7] W. Kim, J. G. Lee, and G.-I. Jee, "The interior-point method for an optimal treatment of bias in trilateration location," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1291–1301, 2006.

[8] R. Ye and H. Liu, "UWB TDOA localization system: receiver configuration analysis," in *Proceedings of the International Symposium on Signals, Systems and Electronics (ISSSE '10)*, pp. 205–208, September 2010.

[9] Y. Zhou, L. Jin, C. Jin, and A. Zhou, "FIMO: a novel WiFi localization method," in *Web Technologies and Applications*, vol. 7808 of *Lecture Notes in Computer Science*, pp. 437–448, Springer, Berlin, Germany, 2013.

[10] A. Catovic and Z. Sahinoglu, "The Cramer-Rao bounds of hybrid TOA/RSS and TDOA/RSS location estimation schemes," *IEEE Communications Letters*, vol. 8, no. 10, pp. 626–628, 2004.

[11] M. R. Gholami, S. Gezici, and E. G. Strom, "A concave-convex procedure for TDOA based positioning," *IEEE Communications Letters*, vol. 17, no. 4, pp. 765–768, 2013.

[12] R. Ye, S. Redfield, and H. Liu, "High-precision indoor UWB localization: technical challenges and method," in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB '10)*, pp. 1–4, IEEE, Nanjing, China, September 2010.

[13] B. Kempke, P. Pannuto, and P. Dutta, "Harmonia: wideband spreading for accurate indoor RF localization," in *Proceedings of the 1st ACM MobiCom Workshop on Hot Topics in Wireless (HotWireless '14)*, pp. 19–23, New York, NY, USA, September 2014.

[14] O. Simeone, A. Maeder, M. Peng, O. Sahin, and W. Yu, "Cloud radio access network: virtualizing wireless access for dense heterogeneous systems," *Journal of Communications and Networks*, vol. 18, no. 2, Article ID 7487951, pp. 135–149, 2016.

[15] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (C-RAN): a primer," *IEEE Network*, vol. 29, no. 1, pp. 35–41, 2015.

[16] R. Wang, H. Hu, and X. Yang, "Potentials and challenges of C-RAN supporting multi-RATs toward 5G mobile networks," *IEEE Access*, vol. 2, pp. 1200–1208, 2014.

[17] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[18] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man & Cybernetics Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.

[19] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: recent advances and comparisons," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[20] S. Bozkurt, S. Günal, U. Yayan, and V. Bayar, "Classifier selection for RF based indoor positioning," in *Proceedings of the 23rd Signal Processing and Communications Applications Conference (SIU '15)*, pp. 791–794, IEEE, Malatya, Turkey, May 2015.

[21] K. He, Y. Zhang, Y. Zhu, W. Xia, Z. Jia, and L. Shen, "A hybrid indoor positioning system based on UWB and inertial navigation," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP '15)*, pp. 1–5, Nanjing, China, October 2015.

Hindawi
Mobile Information Systems
Volume 2017, Article ID 7469342, 13 pages
http://dx.doi.org/10.1155/2017/7469342

Research Article

Distributed Secure Service Composition with Declassification in Mobile Clouds

Ning Xi, Di Lu, Cong Sun, Jianfeng Ma, and Yulong Shen

School of Computer Science and Technology and School of Cyber Engineering, Xidian University, Xian, China

Correspondence should be addressed to Di Lu; nijino2002@163.com

Received 7 October 2016; Accepted 21 December 2016; Published 14 February 2017

Academic Editor: Changqiao Xu

The regional and dynamic characteristics of mobile clouds pose a great challenge on information flow security during service composition. Although secure verification approaches based on standard noninterference provide a solid assurance on information flow security of composite service, too strict constraints on service components may cause the failure of composition procedure. In order to ensure the availability of composite service, we specify the declassification policies based on cryptographic operations to allow data to be legally declassified. And we propose the improved distributed secure service composition framework and approach, which can realize different cloud platforms in multiple domains, cooperate with each other to complete the declassification, and secure composition procedure. Through the experiment and evaluation, it is indicated that our approach provides a more reliable and efficient way for secure service composition in mobile clouds.

1. Introduction

Mobile devices (e.g., smartphone and tablet PC) are increasingly becoming more and more popular in human life as their portability, pervasive connectivity, and various applications (e.g., iPhone and Android Apps). Particularly, in recent years, more kinds of basic functions (e.g., computation, storage, and network) are offered by cloud computing as the software services for elastic management and rapid service delivery with low cost, such as SDS (Software Defined Storage) [1], SDN (Software Defined Network) [2], and cloud- based mobile Apps. With the explosion of mobile applications and the support of cloud computing, mobile computing based on clouds provides a new and promising paradigm for delivering IT services more effectively and conveniently [3]. Moreover, services provided by different clouds and mobile terminals can be composed together to form a more powerful applications [4, 5], for example, trip mode selection application composed by Positioning service, Walking Speed service, Bus Tracking service, and Arrival Estimate service [6].

However, because of the regional and heterogeneous characteristic of mobile networks, there are multiple clouds deployed in different network domains. Due to the multidomain feature of the mobile clouds, data located in different mobile terminals and domains may have different security levels, which poses a great challenge on the security of service composition across multiple mobile clouds. For instance, the personal medical records in e-health data center are with high security level, while the position of the ambulance is with lower security level. When these services are composed together for the patient's emergency, data with different security levels are transmitted among these services, respectively. If these services are composed in an insecure way, an operation in a service may transmit confidential data to a public object and cause the information leakage. Access control has been widely used for protecting sensitive information of individual service from being released to unauthorized attackers [7]. However, for a composite service in mobile networks, data may be processed by serval services from multiple clouds dynamically. Access control cannot detect the information leakage caused by the subsequent operations in other services. Therefore, information flow security is one of the major concerns about the service composition in mobile clouds.

In order to enforce the data security during the service composition, various security mechanisms have been proposed to validate the information flow in composite service based on type system, Petri nets, model checking, program

static analysis, and real-time monitoring. By using type system [8], Hutter and Volkamer [9] define a set of information flow security rules that check the service composition in a secure way during the compilation of the workflow code. Petri nets provide a formal way to model composite service and Accorsi and Wonnemann [10] can identify leaks by analyzing it. Model checking is an automatic verification way that can be used to detect information leaks [11]. Nakajima [12] embedded the lattice model into the Business Process Execution Language (BPEL), and verified the absence of invalid information flows based on model checking. Program analysis is used to construct the dependence among different inputs or outputs; then information flow control (IFC) policies can be designed according to the security requirements. There are two ways to analyze the software according to the different objects, that is, static analysis for source code and dynamic analysis for executable program. For static analysis, She et al. [13, 14] define the transformation factor to measure how likely the output would depend on the input data in different candidate services. In order to improve the accuracy of static analysis, PDG (Program Dependence Graph) is used to specify the dependence between the objects in composite service [15, 16]. Compared with static analysis, dynamic analysis is built on the real-time monitoring of executing program, which can provide more accurate way to check the illegal information flow during the running time [17]. But real-time monitoring increases the cost of service execution, which may decrease the QoS and interfere with users' experience, especially when dozens of services are composed together.

Based on the above approaches, many schemes for secure service composition among clouds are proposed to address the issues of the information leakage on cloud services. Bacon et al. [18] review a range of IFC models and implementations to identify opportunities for using IFC within a cloud computing context, including type system, static analysis, and runtime dynamic analysis. Chou [19] presents the CloudIFC (Cloud Information Flow Control) model to strictly control output information flows in cloud services. Based on the specific information flow control rules and the variables dependency obtained by static analysis, they propose a novel checking way by MapReduce to decrease the verification cost. Solanki et al. [20] develop a new access and information flow control paradigm for service based systems, namely, WS-AIFC, to secure the information flow among services. Based on the dependence list for each data object, WS-AIFC supports flexible cross-domain access and information flow validation. Considering multiple domain nature of clouds, we [21] propose a distributed information flow security verification framework and approach to provide a better load balance and reduce the verification cost effectively across multiple clouds.

Although the above approaches provide a solid assurance on information flow security of composite service, implementing these IFC policies in real applications is still a challenge. These policies aim at standard noninterference that characterizes the complete absence of any information flow or any causal flow from high-level entities to low level ones. However, this requirement is too strict that few services can satisfy it in real application. If all the candidate services fail in the verification, there is no available execution path, which causes the failure of the whole composite service. Meanwhile, in mobile clouds, services are bound together in a dynamic way during service composition, which means the security sensitivities of the input and output data may change when mobile terminal move into a new domain. Considering dozens of candidate services with similar service function, it will be a complex work on selecting appropriate components to compose users required application by type system, global model checking, or centralized static analysis. For type system, when user's initial inputs change, the service codes need to be rebuilt, which brings extra cost for the secure service composition. For global model checking and centralized static analysis, it is impractical to employ a centralized entity in multiple clouds to verify the information flow security. Moreover, the cost of verification can increase rapidly when the application involves more components and the number of the candidate services increases. First, the same service component has to be reverified in different composite services. Second, the state explosion problem arises if each service component is complicated.

Therefore, a distributed and efficient information flow control mechanism supporting declassifying or downgrading information is needed for the secure and reliable service composition in mobile clouds. Compared to the paper [22], we provide the following new extensions. Firstly, mobile cloud is a more complex scenario, which involves the cooperation of different cloud platforms in multiple domains during the composition, and we add more related works for a clear description. Secondly, we give more specific definitions on declassification operations and design an improved formal information flow security model supporting declassification. Thirdly, considering the limited energy and computing resource of mobile terminal, we improve the distributed secure service composition framework and algorithms for the involvement of cloud platforms, which can take over some load on service verification. Besides, more experiments and evaluations are executed for a deep analysis on our approach.

The rest of the paper is structured as follows. Section 2 gives a formal definition of the service chain model in mobile clouds. Section 3 presents the improved computation rules with declassifying information flow in service chain. In Section 4, we propose the secure service composition with declassification mechanism for service chain in mobile clouds. Section 5 evaluates the proposed approach. Section 6 concludes the paper.

2. Preliminaries

As shown in Figure 1, mobile cloud MC is a large-scale distributed environment which consists of multiple heterogeneous domains; that is, MC $= \{d_0, d_1, \ldots\}$. Domain d has various types of data resources R. And services provided by mobile terminals MT or cloud platforms can be CP composed into a more powerful application according to the different customer's requirement. For a clear description, each service provided by either terminals or clouds can be uniformly regarded as a service node in the domain; that is,

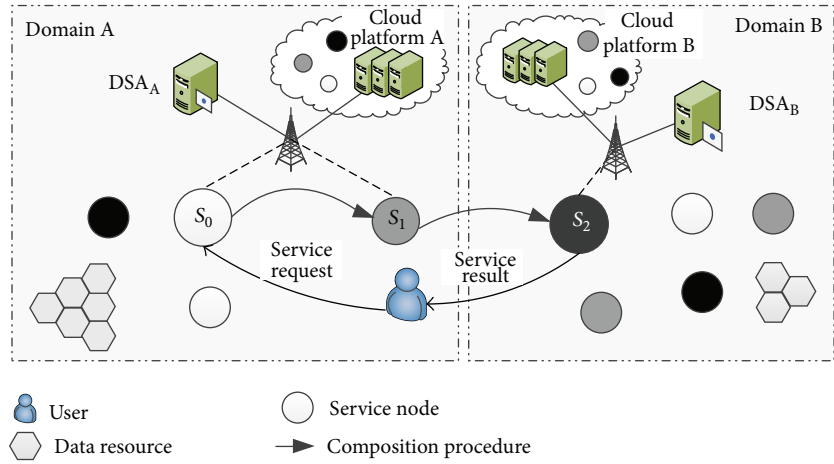


Figure 1: Service composition in mobile clouds.

SN $= \{sn_0, sn_1, \ldots\}$. There is also a security authority DSA in each domain for the management on security policies expressed by domain certificate DCe. Due to the limited energy and computation resources of mobile terminal, there is a cloud platform CP for processing more complex tasks. So domain d can be represented as $d = \langle SN, R, SA, DCe, CP\rangle$.

Referring to the definition in [21], each service s_i provided by service node SN can be represented as a tuple $s_i = \langle dom_i, In_i, Out_i, F_i, SCe_i\rangle$, where dom_i is the domain s_i belongs to; In_i is the input set of service; Out_i is the output set of service; F_i is the service function. SCe_i is the service certificate which specifies the security properties. For each service s_i, there is $In_i = \{In_i^M, In_i^D, In_i^L\}$, where In_i^M is the set of all inputs that s_i receives from its predecessor s_{i-1}; In_i^D is all the inputs from the domain resources $dom \cdot R$; In_i^L is all inputs from service node itself. In the same way, there is $Out_i = \{Out_i^M, Out_i^D, Out_i^L\}$, where Out_i^M is the set of all outputs that s_i sends to its successor s_{i+1}. Out_i^D is all outputs updated to the domain resources $dom \cdot R$. Out_i^L is all outputs written to its local storage.

Service chain SC is a simplified composite service with sequence structure, which can be represented as $SC = \langle CH, In_c, Out_c\rangle$. CH is the execution chain of services $\langle s_0, s_1, \ldots, s_{n+1}\rangle$. In CH, each service s_{i+1} only has one predecessor s_{i-1} and one successor s_{i+1}. For a clear description, s_0 and s_{n+1} are used to denote the initial user. In_{ch} and Out_{ch} are the inputs and outputs of SC including all the service components; that is, $In_{ch} = \cup\{In_i^M \cup In_i^D \cup In_i^L\}$, $0 \le i \le n+1$.

Due to the complex operations in service chain and dynamic network environment, the inner-service dependency $Dep_{inner}(o)$ and interservice dependency $Dep_{inter}(o)$ are defined to represent the flows between different inputs and outputs based on Program Dependence Graph (PDG) [16].

3. Secure Information Flow Model with Declassification in Service Chain

3.1. Multilevel Security Model. In order to represent different sensitivities of data resources in mobile clouds, multilevel security model is defined as $\langle SL, \le\rangle$, where SL is a finite set of security levels that is totally ordered by $\le$ [23].

For each input or output object o in s_i, we define Re: $In_i \cup Out_i \to SL_{ex}$ maps o to the required security level of data stored in it, while Pr: $In_i \cup Out_i \to SL_{ex}$ maps o to its clearance level which represents o can access the corresponding-level data. The required security levels will be computed according to the dependence of the input and output data, which is described as computation rules in the following sections. The clearance levels are provided by the objects who want to access the data, which can be specified in service certificates.

3.2. Secure Information Flow with Standard Noninterference. For data with different security requirements, the computation rules (CRs) on required security level are defined in [16] as follows:

CR1. $\forall u \in In_i^D \cup In_i^L, Re(u) = Pr(u)$.

CR2. $\forall u \in In_i^M, Re(u) = Re(u)$ where $v \in Out_{i-1}^M \wedge v \in Dep_{inter}(u)$.

CR3. $\forall u \in Out_i, Re(u) = \sqcup_{max} Re(v) v \in In_i \cup Out_j$, $j \le i$, and $v \in Dep(u) \cup Dep_{inter}(u)$.

Based on the standard noninterference, we propose a strong security definition on information flow for composite service in [16].

Definition 1. The information flow in service chain SC is considered secure if $\forall u \in In_{ch} \cup Out_{ch}$, u satisfies $Pr(u) \ge Re(v)$, where $v \in In_{ch} \cup Out_{ch}$ and $v \in Dep_{inner}(u) \cup Dep_{inner}(u)$.

In this definition, it is considered secure when there is no flow from a high-level object to another low level one across all service components. However, the strong security constraints enforce the fact that the flow of information must comply with the security level ordering and do not tolerate any exceptions. To deal with real application, with the execution of the composite service, the required security levels of inputs or outputs become higher and higher according to the above CRs, which is so strict that fewer candidate service components can satisfy. In this case, it would lead to

a high failure rate on service composition. Therefore, more general flow policy allowing data declassification needs to be proposed to improve the availability of composite service.

3.3. Secure Service Composition Model with Cryptographic Operations. Due to the strong security condition, declassification operations are needed for the secure service composition. Cryptographic operations are promising ways of maintaining data confidentiality and integrity, for example, encryption and digital signature. Through the cryptographic operations, processed secret data can be transmitted into a public object, which realizes the declassification of data. Therefore, extra cryptographic operations $\mathrm{En}(o, \mathrm{key})$ and $\mathrm{De}(o, \mathrm{key})$ can be add to the service function F_i for each service.

$$
\begin{aligned}
f &::= a; f \\
a &::= \mathrm{skin} \mid \mathrm{input}(\mathrm{var}, e) \mid \mathrm{var}_p := e \mid a; a \\
&\quad \mid \text{if } (e) \text{ then } a \text{ else } a \\
&\quad \mid \text{while } (e)\ a \mid \mathrm{output}(\mathrm{var}, e) \\
&\quad \mid \mathrm{En}(\mathrm{var}, \mathrm{key}) \mid \mathrm{De}(\mathrm{var}, \mathrm{key}) \\
e &::= \mathrm{var}_p \mid e\,\mathrm{Re} \\
R &::= + \mid - \mid = \mid <
\end{aligned}
$$

For $\forall \mathrm{var} \in \mathrm{In}_i \cup \mathrm{Out}_i$, var_p and var_c represent the plaintext and ciphertext of var, and the encryption and decryption operation on var are defined as $\mathrm{En}(\mathrm{var}, \mathrm{key})$ and $\mathrm{De}(\mathrm{var}, \mathrm{key})$. Because of the low efficiency on homomorphic encryption [24], the traditional cryptographic operations are considered in this paper. As shown in F_i's definition, the classified data var_c cannot be directly processed by regular operations which may cause the plaintext of var to not be recovered. But the basic input, output, encryption, and decryption are still supported by F_i for classified data var_c.

When the data in var is encrypted, it provides more secure way to transmit var, and the attacker needs to work harder to crack the ciphertext which depends on the security of encryption algorithm E and the key *key*. Thus we use $\mathrm{Re}(\langle E, \mathrm{key}\rangle)$ to represent the security level of classified data var_c. Encryption with more complex algorithm and key means $\mathrm{Re}(\langle E, \mathrm{key}\rangle)$ is lower. And the security level of var with reencryption depends on the strongest algorithm and key. When var_c is decrypted, the data of var is no longer protected by encryption, and the security level of var returns to its original value. According to the analysis above, we can extend the basic computation rules as follows:

CR4. $\forall u \in \mathrm{In}_i \cup \mathrm{Out}_i$, if u is encrypted by $\langle E, \mathrm{key}\rangle$, there is $\mathrm{Re}(u) = \mathrm{Re}(u_p) \wedge \mathrm{Re}(\langle E, \mathrm{key}\rangle)$.

CR5. For the ciphertext u_c, if u is decrypted, there is $\mathrm{Re}(u) = \mathrm{Re}(\mathrm{De}(u_c, \mathrm{key})) = \mathrm{Re}(u_p)$.

In traditional definition on standard noninterference, high security level data are not allowed to transfer to an object with lower level. The encryption operation may violate the requirements on standard noninterference. But the attacker still cannot obtain the sensitive data if he cannot crack the ciphertext, which is still considered secure although the sensitive data is transferred to an object with lower clearance. In order to specify the special downgrading flow in composite service, an extended definition on inner dependence is proposed as follows.

Definition 2. For $\forall u \in \mathrm{Out}_i, \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ represents the set of inputs that u depends on, where Enc is the pair of encryption algorithm and key that u adopts; Dec is the pair of decryption algorithm and key that dependent inputs adopt. Then $\forall v \in \mathrm{In}_i$ and $\wedge v \in \mathrm{Dep}_{\mathrm{inner}}(u)$, there are four cases to consider:

(1) v is plaintext and u outputs as the plaintext; there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\phi,\phi}(u)$.

(2) v is plaintext but u outputs as the ciphertext encrypted by $\langle E_u, \mathrm{key}_u\rangle$; there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\langle E_u, \mathrm{key}_u\rangle,\phi}$.

(3) v is ciphertext but u outputs as the plaintext; it means v is decrypted with $\langle E_v, \mathrm{key}_v\rangle$ during the execution of service. Then there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\phi,\langle E_v, \mathrm{key}_v\rangle}$.

(4) v is ciphertext and u also outputs as the ciphertext; there are three different cases:

  (1) If v is decrypted with $\langle E_v, \mathrm{key}_v\rangle$ during the execution of service, it means v is operated as plaintext and u is encrypted by another encryption algorithm and key. Then there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\langle E_u, \mathrm{key}_u\rangle,\langle E_v, \mathrm{key}_v\rangle}$.

  (2) If v is not decrypted but u is reencrypted by $\langle E_u, \mathrm{key}_u\rangle$, we can obtain $v \in \mathrm{Dep}^{\mathrm{inner}}_{\langle E_u, \mathrm{key}_u\rangle,\phi}$.

  (3) If v is not decrypted and u is not reencrypted, there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\phi,\phi}$.

Based on the extend inner dependence, interdependence can be defined recursively as follows.

Definition 3. $\forall u \in \mathrm{In}^M_i \cup \mathrm{Out}_i, \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ represents the set of inputs or outputs in different services that u depends on. EnS is the set of pairs of the encryption algorithm and key that is used during the execution path, while DeS represents the set of all decryption operations. For each $v \in \mathrm{In}_j \cup \mathrm{Out}^M_j, v \in \mathrm{Dep}_{\mathrm{inter}}(u), 0 \le j < i \le N$, there are three cases to consider:

(1) $i = j + 1$: $\forall u \in \mathrm{In}^M_i$ and $v \in \mathrm{Out}^M_j$, if $v \in \mathrm{Dep}_{\mathrm{inter}}(u)$, there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(u)$.

(2) $i = j + 1$: $\forall u \in \mathrm{In}^M_i$ and $v \in \mathrm{In}^M_j$, if $\exists w \in \mathrm{Out}^M_j$, $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_v,\mathrm{Dec}_v}(w)$, and $w \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS}_w,\mathrm{DeS}_w}(u)$, there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$, where $\mathrm{EnS} = \mathrm{Enc}_v \cup \mathrm{EnS}_w$ and $\mathrm{DeS} = \mathrm{Dec}_v \cup \mathrm{DeS}_w$.

(3) $i = j + 1$: $\forall u \in \mathrm{Out}_i, v \in \mathrm{Out}^M_j$, if $\exists w \in \mathrm{In}^M_i, v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS}_v,\mathrm{DeS}_v}(w)$ and $w \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{Enc}_w,\mathrm{Dec}_w}(u)$, there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$, where $\mathrm{EnS} = \mathrm{EnS}_v \cup \mathrm{Enc}_w$ and $\mathrm{DeS} = \mathrm{DeS}_v \cup \mathrm{Dec}_w$.

(4) $i = j + 1$: $\forall u \in \mathrm{Out}_i$ and $v \in \mathrm{In}_j$, if $\exists w_1 \in \mathrm{Out}^M_j, w_2 \in \mathrm{In}^M_i$, $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_v,\mathrm{Dec}_v}(w_1)$, $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS}_{w1},\mathrm{DeS}_{w1}}(w_2)$

and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_{w2},\mathrm{Dec}_{w2}}(u)$, there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$, where $\mathrm{EnS} = \mathrm{Enc}_v \cup \mathrm{EnS}_{w1} \cup \mathrm{Enc}_{w2}$ and $\mathrm{DeS} = \mathrm{Dec}_v \cup \mathrm{Des}_{w1} \cup \mathrm{Dec}_{w2}$.

(5) $i > j + 1$: $\forall u \in \mathrm{In}^M_i$ and $v \in \mathrm{In}_j \cup \mathrm{Out}^M_j$, if $\exists w \in \mathrm{In}_k \cup \mathrm{Out}_k$, $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{Ens}_v,\mathrm{Des}_v}(w)$, and $w \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{Ens}_w,\mathrm{Des}_w}(w)$, $j < k < i$, there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$, where $\mathrm{EnS} = \mathrm{EnS}_v \cup \mathrm{EnS}_w$ and $\mathrm{DeS} = \mathrm{DeS}_v \cup \mathrm{DeS}_w$.

Based on the extend inner and interdependence, the improved security definition on information flow for composite service can be presented as follows.

Definition 4. The information flow in service chain SC is considered secure if $\forall u, v \in \mathrm{In}_c \cup \mathrm{Out}_c$, $v \in \mathrm{Dep}_{\mathrm{inner}}(u) \cup \mathrm{Dep}_{\mathrm{inter}}(u)$ satisfies the following conditions:

(1) $\forall u, v \in \mathrm{In}_i \cup \mathrm{Out}_i$, and $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc},\mathrm{Dec}}(u)$,

(i) if $\mathrm{Enc} - \mathrm{Dec} = \phi$, there is $\Pr(u) \geq \mathrm{Re}(v)$;
(ii) if $\mathrm{Enc} - \mathrm{Dec} \neq \phi$, there is $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u \rangle)$, where $\langle E_u, \mathrm{key}_u \rangle \in \mathrm{Enc} - \mathrm{Dec}$.

(2) $\forall u \in \mathrm{In}_i \cup \mathrm{Out}_i$, $v \in \mathrm{In}_j \cup \mathrm{Out}_j$, $0 \leq j < i \leq N$, and $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{Ens},\mathrm{Des}}(u)$,

(i) if $\mathrm{EnS} - \mathrm{DeS} = \phi$, there is $\Pr(u) \geq \mathrm{Re}(v)$;
(ii) if $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, there is $\Pr(u) \geq \prod^{1 \leq x \leq N_e}_{\min} \mathrm{Re}(\langle E_x, \mathrm{key}_x \rangle)$, where $\langle E_x, \mathrm{key}_x \rangle \in \mathrm{Ens} - \mathrm{Des}$.

According to Definition 4, two different types of flow are considered separately, that is, unclassified and classified flow. For the unclassified flow, it must satisfy the traditional information noninterference constraints, that is, the clearance on each input or output in s_i must be no less than the required security level, which depends on all related inputs and outputs in s_i and its predecessor. For the classified flow, data security depends on the encryption operation, so it can be considered secure that the clearance of the input or output is equal or greater than the required security level of the strongest encryption operation.

Based on improved information flow security definition, we can deduce the security constraints on each service as the following theorem.

Theorem 1. *The information flow in service chain SC with N steps is considered secure if each s_i in SC satisfies the following conditions:*

(1) $\forall u \in Out_i, v \in In_i$, *and* $v \in Dep_{\mathrm{inner}}(u)$,

(a) *if u is not encrypted, there is* $\Pr(u) \geq \mathrm{Re}(v)$;
(b) *if u is encrypted by* $\langle E_u, key_u \rangle$, *there is* $\Pr(u) \geq \mathrm{Re}(\langle E_u, key_u \rangle)$.

(2) $\forall u \in In^M_i, v \in Out^M_{i-1}$, *and* $v \in Dep_{\mathrm{inter}}(u)$,

(a) *if u is not encrypted, there is* $\Pr(u) \geq \mathrm{Re}(v)$;
(b) *if u is encrypted by* $\langle E, key \rangle$, *there is* $\Pr(u) \geq \mathrm{Re}(\langle E, key \rangle)$.

Proof. First, let $N = 1$; then there are two service components involved in the service chain, that is, s_0 and s_1. □

Case 1. Inner information flow in each service component is considered first; that is, $\forall u \in \mathrm{Out}_0$, $v \in \mathrm{In}_0$, and $v \in \mathrm{Dep}_{\mathrm{inner}}(u)$.

(1) Condition (1)(a) provides that for each $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc},\mathrm{Dec}}(u)$ where $\mathrm{Enc} - \mathrm{Dec} = \phi$, there is $\Pr(u) \geq \mathrm{Re}(v)$.

(2) Condition (1)(b) provides that for each $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc},\mathrm{Dec}}(u)$ where $\mathrm{Enc} - \mathrm{Dec} \neq \phi$, there is $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u \rangle)$.

In the same way, we can get the information flow is also secure in s_1.

Case 2. Information flow between s_0 and s_1 is considered; that is, $\forall u \in \mathrm{In}_1 \cup \mathrm{Out}_1$, $v \in \mathrm{In}_0 \cup \mathrm{Out}_0$, and $v \in \mathrm{Dep}_{\mathrm{inter}}(u)$.

(1) $\forall u \in \mathrm{In}_1$, $v \in \mathrm{Out}_0$, and $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$, according to Definition 3(1), there is $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ where $\mathrm{EnS} = \phi$ and $\mathrm{DeS} = \phi$, and condition (2) provides $\Pr(u) \geq \mathrm{Re}(v)$.

(2) $\forall u \in \mathrm{In}_1$, $v \in \mathrm{In}_0$, and $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$, according to Definition 3(2), there is $\exists w \in \mathrm{Out}^M_j$, $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_v,\mathrm{Dec}_v}(w)$, and $w \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(u)$.

(i) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} = \phi$, w is not encrypted. Condition (1)(a) provides $\Pr(w) \geq \mathrm{Re}(v)$, and condition (2)(a) provides $\Pr(u) \geq \mathrm{Re}(w)$. Therefore, $\Pr(u) \geq \mathrm{Re}(v)$.
(ii) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, w is encrypted by $\langle E_w, \mathrm{key}_w \rangle$. There is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_w, \mathrm{key}_w \rangle\}$. Condition (1)(b) provides $\Pr(w) \geq \mathrm{Re}(\langle E_w, \mathrm{key}_w \rangle)$. Condition (2)(a) and CR 4 provide $\Pr(u) \geq \mathrm{Re}(w) = \mathrm{Re}(\langle E_w, \mathrm{key}_w \rangle)$.

(3) $\forall u \in \mathrm{Out}_1$ and $v \in \mathrm{Out}_0$, according to Definition 3(3), there is $\exists w \in \mathrm{In}^M_1$ and $v \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(u), w \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_w,\mathrm{Dec}_w}(u)$.

(i) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} = \phi$, u is not encrypted. Condition (1)(a) provides $\Pr(u) \geq \mathrm{Re}(w)$, and condition (2)(a) a provides $\Pr(w) \geq \mathrm{Re}(v)$. Therefore, $\Pr(u) \geq \mathrm{Re}(v)$.
(ii) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, u is encrypted by $\langle E_u, \mathrm{key}_u \rangle$. There is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_u, \mathrm{key}_u \rangle\}$. Condition (1)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u \rangle)$.

(4) $\forall u \in \mathrm{Out}_1$, and $v \in \mathrm{In}_0$, according to Definition 3(4), there is $\exists w_1 \in \mathrm{Out}^M_0$, $w_2 \in \mathrm{In}^M_1$, $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_v,\mathrm{Dec}_v}(w_1)$, $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\varphi,\varphi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc}_{w2},\mathrm{Dec}_{w2}}(u)$, and there is $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ where $\mathrm{EnS} = \mathrm{Enc}_v \cup \mathrm{Enc}_{w2}$ and $\mathrm{DeS} = \mathrm{Dec}_v \cup \mathrm{Dec}_{w2}$.

(i) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS},\mathrm{DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} = \phi$, there are two different cases:

(a) For $\mathrm{Enc}_v \cup \mathrm{Enc}_{w2} = \phi$ and $\mathrm{Dec}_v \cup \mathrm{Dec}_{w2} = \phi$, CR 3 provides $\mathrm{Re}(w_1) \geq \mathrm{Re}(v)$. Condition (1)(a) provides $\Pr(u) \geq \mathrm{Re}(w_2)$ and condition (2)(a) provides $\Pr(w_2) \geq \mathrm{Re}(w_2) = \mathrm{Re}(w_1)$. Therefore, $\Pr(w) \geq \mathrm{Re}(v)$.

(b) For $\mathrm{Enc}_v = \{\langle E_v, \mathrm{key}_v\rangle\}$, $\mathrm{Dec}_v = \phi$, $\mathrm{Enc}_{w2} = \phi$, and $\mathrm{Dec}_{w2} = \{\langle E_v, \mathrm{key}_v\rangle\}$, CR 5 provides $\mathrm{Re}(w_{2_p}) \geq \mathrm{Re}(v)$ and condition (1)(a) provides $\Pr(u) \geq \mathrm{Re}(w_{2_p})$, so $\Pr(u) \geq \mathrm{Re}(v)$.

(ii) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, there are four different cases:

(a) For $\mathrm{Enc}_v = \{\langle E_v, \mathrm{key}_v\rangle\}$, $\mathrm{Dec}_v = \phi$, $\mathrm{Enc}_{w2} = \phi$, and $\mathrm{Dec}_{w2} = \phi$ where $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_v, \mathrm{key}_v\rangle\}$, condition (2)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_v, \mathrm{key}_v\rangle)$.

(b) For $\mathrm{Enc}_v = \phi$, $\mathrm{Dec}_v = \phi$, $\mathrm{Enc}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, and $\mathrm{Dec}_{w2} = \phi$ where $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, condition (2)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_v, \mathrm{key}_v\rangle)$.

(c) For $\mathrm{Enc}_v = \{\langle E_v, \mathrm{key}_v\rangle\}$, $\mathrm{Dec}_v = \phi$, $\mathrm{Enc}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, and $\mathrm{Dec}_{w2} = \{\langle E_v, \mathrm{key}_v\rangle\}$ where $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, condition (2)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_v, \mathrm{key}_v\rangle)$.

(d) For $\mathrm{Enc}_v = \{\langle E_v, \mathrm{key}_v\rangle\}$, $\mathrm{Dec}_v = \phi$, $\mathrm{Enc}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, and $\mathrm{Dec}_{w2} = \phi$ where $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_v, \mathrm{key}_v\rangle, \langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, condition (2)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_{w2}, \mathrm{key}_{w2}\rangle) \geq \min\{\mathrm{Re}(\langle E_v, \mathrm{key}_v\rangle), \mathrm{Re}(\langle E_{w2}, \mathrm{key}_{w2}\rangle)\}$.

Based on the above analysis and Definition 4, information flow between s_0 and s_1 is secure.

Therefore, Theorem 1 is true when $N = 1$.

Then we assume Theorem 1 is true when $N = n$, and the proof on $N = n + 1$ is presented as follows.

Case 1. Inner information flow in service component s_{n+1} is considered; that is, $\forall u \in \mathrm{Out}_{n+1}, v \in \mathrm{In}_{n+1}$, and $v \in \mathrm{Dep}_{\mathrm{inner}}(u)$.

(1) Condition (1)(a) provides that for each $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{Enc} - \mathrm{Dec} = \phi$, there is $\Pr(u) \geq \mathrm{Re}(v)$.

(2) Condition (1)(b) provides that for each $v \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{Enc} - \mathrm{Dec} \neq \phi$, there is $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u\rangle)$.

And above assumption provides that information flow in $s_0, s_1, \ldots, s_n$ is secure.

Case 2. The assumption provides that information flow among first n service step is secure. Then the interinformation flows between s_{n+1} and former services are considered; that is, $\forall u \in \mathrm{In}_{n+1} \cup \mathrm{Out}_{n+1}$, $v \in \mathrm{In}_j \cup \mathrm{Out}_j$, and $v \in \mathrm{Dep}_{\mathrm{inter}}(u)$, $0 \leq j \leq n + 1$.

According to Definition 3(5) and Lemma 1 in [16], there is $\exists w_1 \in \mathrm{Out}^M_n$, $w_2 \in \mathrm{In}^M_{n+1}$, $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$, $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\varphi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}(\mathrm{Enc}, \mathrm{Dec})(u)$.

(1) For $u \in \mathrm{In}_{n+1}$ there is $u = w_2$.

(i) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} = \phi$, there is for $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$, and $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \phi$. Condition (2)(a) provides $\Pr(u) \geq \mathrm{Re}(w_1)$ and the assumption provides $\mathrm{Re}(w_1) \geq \mathrm{Re}(v)$. So there is $\Pr(u) \geq \mathrm{Re}(v)$.

(ii) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, there is for $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$ and $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(u)$ where $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$. Condition (2)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_n, \mathrm{key}_n\rangle)$. So there is $\Pr(u) \geq \bigcap^{1\leq i\leq n}_{\min} \mathrm{Re}(\langle E_{vi}, \mathrm{key}_{vi}\rangle)$.

(2) For $u \in \mathrm{Out}_{n+1}$, the following cases are considered:

(i) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} = \phi$, there are two cases:

(a) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$, and $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2), w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \phi$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \phi$, condition (1)(a) provides $\Pr(u) \geq \mathrm{Re}(w_2)$. CR 2 provides $\mathrm{Re}(w_1) = \mathrm{Re}(w_2)$. The assumption provides $\mathrm{Re}(w_1) \geq \mathrm{Re}(v)$. So there is $\Pr(u) \geq \mathrm{Re}(v)$.

(b) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1), w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \{\langle E_v i, \mathrm{key}_v i\rangle, 1 \leq i \leq n\}$ and $\mathrm{Dec}_{w2} = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$, CR 5 provides $\mathrm{Re}(w_{2_p}) \geq \mathrm{Re}(v)$ and condition (1)(a) provides $\Pr(u) \geq \mathrm{Re}(w_{2_p})$, so there is $\Pr(u) \geq \mathrm{Re}(v)$.

(ii) If $\forall v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(u)$ satisfies $\mathrm{EnS} - \mathrm{DeS} \neq \phi$, there are five cases:

(a) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1), w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \{\langle \mathrm{E}_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \phi$, condition (1)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_n, \mathrm{key}_n\rangle)$. So there is $\Pr(u) \geq \bigcap^{1\leq i\leq n}_{\min} \mathrm{Re}(\langle E_{vi}, \mathrm{key}_{vi}\rangle)$.

(b) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1), w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, there is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\} \cup \langle E_{w2}, \mathrm{key}_{w2}\rangle$. Condition (1)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_{w2}, \mathrm{key}_{w2}\rangle)$. So there is $\Pr(u) \geq \bigcap^{1\leq i\leq n}_{\min} \mathrm{Re}(\langle E_{vi}, \mathrm{key}_{vi}\rangle)$.

(c) For $\in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1), w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \phi$ but $\mathrm{Dec}_{w2} = \{\langle E_{vn}, \mathrm{key}_{vn}\rangle\}$, there is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle, 1 \leq i \leq n\}$. Condition (1)(b)

provides $\Pr(u) \geq \mathrm{Re}(\langle E_{n-1}, \mathrm{key}_{n-1}\rangle)$. So there is $\Pr(u) \geq \bigcap_{\min}^{1\leq i\leq n} \mathrm{Re}(\langle E_{vi}, \mathrm{key}_{vi}\rangle)$.

(d) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$, $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle,\ 1 \leq i \leq n\}$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$ but $\mathrm{Dec}_{w2} = \{\langle E_{vn}, \mathrm{key}_{vn}\rangle\}$, there is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{vi}, \mathrm{key}_{vi}\rangle,\ 1 \leq i \leq n\} \cup \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$. Condition (1)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_{w2}, \mathrm{key}_{w2}\rangle)$. So there is $\Pr(u) \geq \bigcap_{\min}^{1\leq i\leq n} \mathrm{Re}(\langle E_{vi}, \mathrm{key}_{vi}\rangle)$.

(e) For $v \in \mathrm{Dep}^{\mathrm{inter}}_{\mathrm{EnS,DeS}}(w_1)$, $w_1 \in \mathrm{Dep}^{\mathrm{inter}}_{\phi,\phi}(w_2)$, and $w_2 \in \mathrm{Dep}^{\mathrm{inner}}_{\mathrm{Enc,Dec}}(u)$ where $\mathrm{EnS}_v - \mathrm{DeS}_v = \phi$ and $\mathrm{Enc}_{w2} - \mathrm{Dec}_{w2} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$, there is $\mathrm{EnS} - \mathrm{DeS} = \{\langle E_{w2}, \mathrm{key}_{w2}\rangle\}$. Condition (1)(b) provides $\Pr(u) \geq \mathrm{Re}(\langle E_{w2}, \mathrm{key}_{w2}\rangle)$.

Based on the above analysis and Definition 4, information flows between s_{n+1} and former services s_i where $i < n + 1$ are secure.

Therefore, Theorem 1 is also true when $N = n + 1$.

In conclusion, Theorem 1 is true.

Based on the above Theorem 1, we can propose an improved service composition mechanism supporting declassification operations. The specific declassification policies (DPs) are presented as follows.

DP 1. For $\forall v \in \mathrm{In}_i, u \in \mathrm{Out}_i, v \in \mathrm{Dep}_{\mathrm{inner}}(u)$, if $\Pr(u) < \mathrm{Re}(v)$, then u needs to be encrypted by $\langle E_u, \mathrm{key}_u\rangle$ which satisfies $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u\rangle)$.

DP 2. For $\forall v \in \mathrm{Out}_i^M$, $u \in \mathrm{In}_{i+1}^M$, $v \in \mathrm{Dep}_{\mathrm{inter}}(u)$, if $\Pr(u) < \mathrm{Re}(v)$, then u needs to be encrypted by $\langle E_u, \mathrm{key}_u\rangle$ which satisfies $\Pr(u) \geq \mathrm{Re}(\langle E_u, \mathrm{key}_u\rangle)$.

According to the declassification policies, when the provided security level of u cannot satisfy the strict conditions, cryptographic operations are adopted to assist in declassifying the required security level which can also hold the information flow security.

4. Secure Service Composition with Declassification in Mobile Network

4.1. Secure Service Composition with Declassification Framework in Mobile Network. In the mobile cloud system with multiple domains, there are serval candidate services with same functions but different providers, which can be denoted by $s_{i,j} \in S_i$, $0 \leq i \leq N$, $0 \leq j \leq |S_i|$. Traditional secure service composition approaches are based on standard information flow verification technique where insecure candidate service is filtered. However, it may be so strict that few candidate services can satisfy in real application, which leads to the failure of service composition. Based on the declassification policies, we can propose an improved secure service composition framework supporting declassification operations, which is shown in Figure 2.

This framework is constructed as a distributed secure service composition framework involving three main kinds of entities, that is, Cloud Platform (CP), Candidate Services (CS), and Domain Security Authorities (DSA). Considering the limited energy and computation resources of mobile terminals, the verification procedure is executed by CPs. DSAs are responsible for the management on the security certificate SCe for each service node. SCe includes the provided security levels of input and outputs, the dependencies between the input and output and its public key. If the service node is fixed one, that is, services are provided by cloud platform, the certificate is generated when the service is first deployed in cloud platform. If the service node is mobile one, that is, services are provided by mobile terminal, the certificate is generated when the terminal first moves into this domain.

During the verification, all candidate services send their dynamic input data and certificates to the cloud platform to finish the verification procedure. There are two different scenarios, that is, inner-domain and interdomain verification. For inner-domain verification, candidate services CP and DSA in the same domain are involved in the verification. For interdomain verification, the participant entities include not only candidate services but also two CPs and SAs in the corresponding domains.

Comparing to the traditional verification procedure in [16], declassification based on cryptographic operations is executed automatically to recover the insecure information flows against the declassification policies. If the information flow security verification returns failure, each insecure component needs to negotiate a session key with its adjacent nodes for the encryption and decryption during the service execution. For clear description, we mainly focus on the declassification procedure in this paper.

4.2. Cryptographic Operations for Declassification in Service Composition

4.2.1. Cryptographic Operation Agent. Based on the Theorem 1, basic cryptographic operations must be supported by each service node to realize the declassification of information flow during the service composition. There are many relevant security specifications which have been proposed to protect data confidentiality and integrity during service execution, such as XML Encryption and Signature, WS-Security, SAML (Security Assertion Markup Language), XACML (XML Access Control Markup Language), and XKM- S (XML Key Management Specification) [25]. By developing the basic security functions supported by these specifications, a cryptographic operation agent (COA) can be designed and deployed in each service node, mobile terminal, or cloud platform, to execute the declassification operations, which is shown in Figure 3.

The cryptographic operation agent is composed of three function modules, that is, key negotiator, encryptor and decryptor. Key negotiator is responsible for the key management including key generation, negotiation with other services, key storage, and update. Encryptor and decryptor are responsible for data encryption and decryption during the service. There are two phases for agent to complete the declassification procedure, that is, key negotiation and data encryption and decryption.

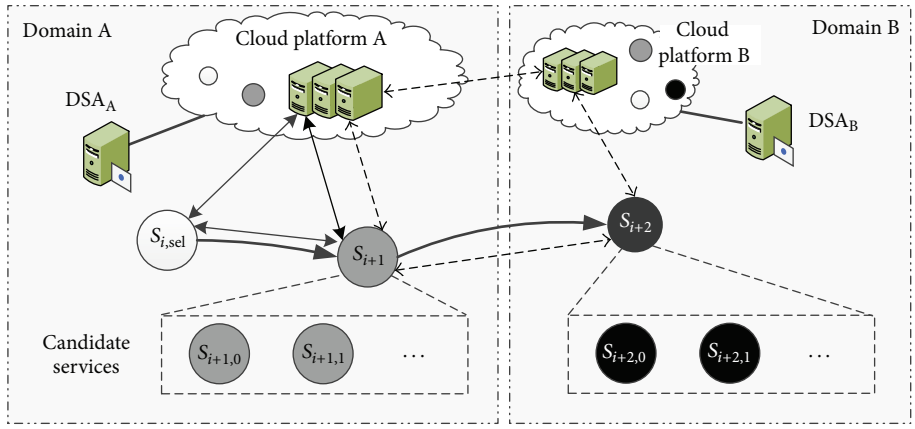


Figure 2: Distributed secure service composition with declassification framework.

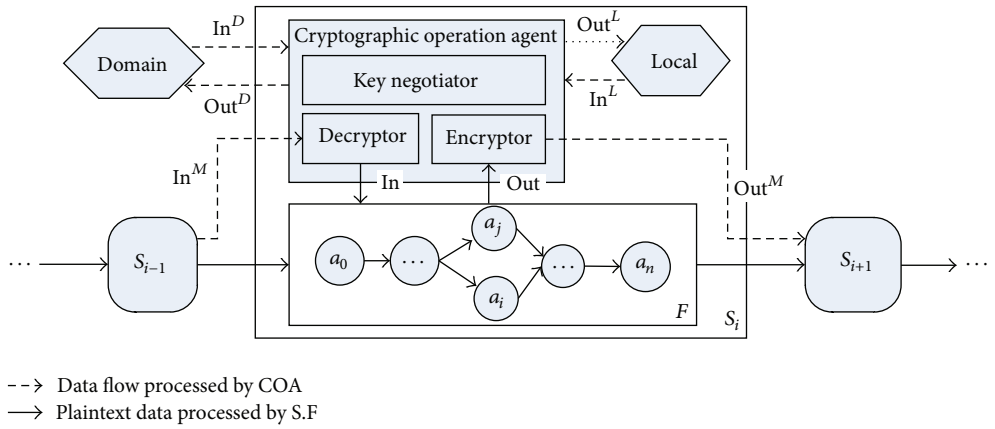


Figure 3: Cryptographic operation agent in each service node.

4.2.2. Key Negotiation Phase. Key negotiation phase is the preparation phase for the data declassification, which is also the most critical step. In this phase, for each insecure input or output $u \in \text{In}_i \cup \text{Out}_i$, related two services negotiate for generating the appropriate encryption algorithm and key $\langle E, \text{key} \rangle$ to ensure the information flow security according to DPs. There are two kinds of negotiation process due to multiple domains, that is, inner-domain negotiation and interdomain negotiation, which is shown in Figures 4 and 5. The procedure of key negotiation follows the specification of XKMS (XML Key Management Specification).

When the key negotiation begins between two adjacent service nodes, both certificates containing their own public keys are delivered to the opponents. Then the random number protected by public key is transferred to each other at the fourth and seventh step. And finally the session key *key* is computed based on these random numbers with a standard key generation algorithm. Meanwhile the encryption algorithm E can also be negotiated during this procedure. In order to ensure the information flow security in the following composition, the length of the key, the complexity of the random number, the key generation algorithm, and the encryption algorithm must satisfy the requirements on security level. The pseudocode of key negotiation is presented as Algorithm 1.

4.2.3. Data Declassification Phase. The data declassification phase is activated after the procedure of secure service composition. During the service execution, the COA encrypts the insecure inputs and outputs to realize the declassification on high-level data by using the session key. Meanwhile, it also realizes decryption on the cipher data for normal processing of service function.

4.3. Distributed Secure Service Composition with Declassification Algorithm across Multiple Mobile Clouds. During the secure service composition, cloud platform CP verifies the service chain by service step based on Theorem 1. For each candidate service s_i, CP first verifies whether the input objects satisfy the security condition, then compute the required

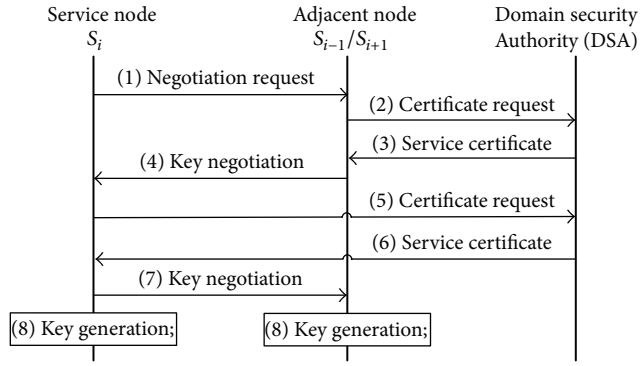


Figure 4: Key negotiation in the domain.

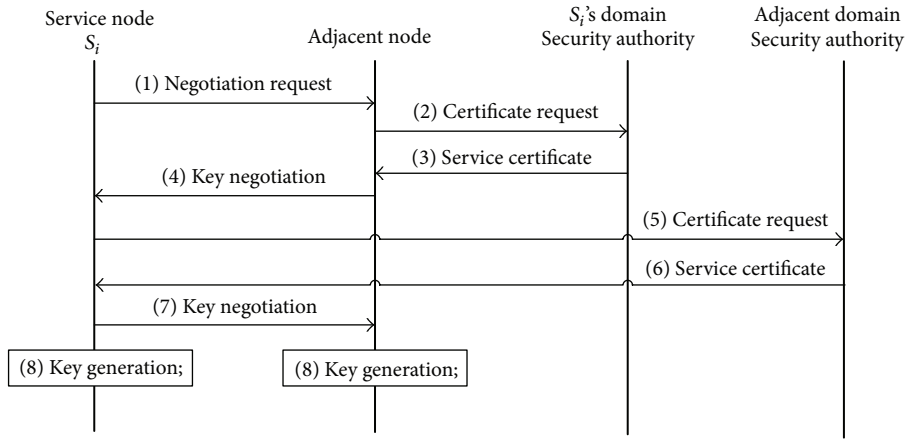


Figure 5: Key negotiation across the domain.

security level for each output objects, and finally verify whether the output objects satisfy the security condition. Meanwhile, if there is an input or output object of s_i which fails to satisfy the strict security constraints, the key negotiation is executed automatically between the related services. In this case, the procedure also returns true unless key negotiation is failed. The pseudocode of verification and declassification for adjacent services is presented as Algorithm 2.

Based on the verification and declassification procedure, we propose a distributed secure service composition with declassification algorithm for mobile clouds. The composition procedure is executed in a distributed way, that is, different cloud platforms in multiple domains need cooperation with each other to finish the whole procedure. There are three types of messages defined for the control on the execution of the procedure, that is, start_message, failure_message, and success_message. First, each cloud platform CP waits for the start message to start the composition procedure. Then CP receives the intermediate result of composition from the start message, including the required security level of predecessor's output and all executable path. After that, CP generates all possible execution paths based on intermediate result and the candidate services located in its domain and verifies them. For each path p that passed the verification, CP pushes it into passed path set PP and records its required security levels of outputs, which can be grouped as an intermediate result for the next step composition. If there is no legal path, CP would send the failure message to user to announce the failure of composition. If the final service step located in this domain, CP would send the success message with all passed path to user. If there are other steps in different domain, CP would send the start message with the intermediate result to the next cloud platform to continue the verification procedure. The distributed secure service composition with declassification algorithm is shown as Algorithm 3.

5. Experiments and Evaluations

The information flow security can be ensured by Theorem 1. And the basic comparison of different verification approaches is presented in Table 1.

According to Table 1, compared to [9, 13–16, 19–21], only [12] and our approach in this paper support the declassification during service composition which can ensure the availability of the composite service. Besides, approaches in [9, 12–15] all work in a centralized way while [16, 19–21] and

Input: s_i, s_{i-1} or s_{i+1}
Output: True or **False** $\langle \text{key}_{i,0}^{M}, E_{i,0}^{M} \rangle, \ldots, \langle \text{key}_{i,n}^{M}, E_{i,n}^{M} \rangle$ **and** $\langle \text{key}_{i,0}^{\text{Out}}, E_{i,0}^{\text{Out}} \rangle, \ldots, \langle \text{key}_{i,n}^{\text{Out}}, E_{i,n}^{\text{Out}} \rangle$.
(1) //$\text{In}_{i,\text{insec}}^{M}$ and $\text{Out}_{i,\text{insec}}$ represents insecure input and outputs in s_i
(2) **if** input is s_{i-1} **then**
(3) **for** each input $\text{in}_{i,x}^{M} \in \text{In}_{i,\text{insec}}^{M}$ **do**
(4) Negotiate_Requests(s_{i-1}, $\text{dom}_{i-1} \cdot \text{SA}$)
(5) **if** Key_Computation($\text{key}_{i,x}^{M}$, $E_{i,x}^{M}$, $\Pr(\text{in}_{i,x}^{M})$) == **False then**
(6) //**False** means two service components can't generate appropriate $\langle \text{key}_{i,x}^{M}, E_{i,x}^{M} \rangle$ which satisfies $\Pr(\text{in}_{i,x}^{M}) \geq \text{Re}(\text{key}_{i,x}^{M})$, else it return **True**.
(7) **return False**
(8) **end if**
(9) **end for**
(10) **else**
(11) **for** each output $\text{out}_{i,y} \in \text{Out}_{i,\text{insec}}$ **do**
(12) Negotiate_Requests(s_{i+1}, $\text{dom}_{i+1} \cdot \text{SA}$)
(13) **if** Key_Computation($\text{key}_{i,y}^{\text{Out}}$, $E_{i,y}^{\text{Out}}$, $\Pr(\text{out}_{i,y})$) == **False then**
(14) **return False**
(15) **end if**
(16) **end for**
(17) **end if**
(18) **return True.** $\langle \text{key}_{i,0}^{M}, E_{i,0}^{M} \rangle, \ldots, \langle \text{key}_{i,n}^{M}, E_{i,n}^{M} \rangle$ and $\langle \text{key}_{i,0}^{\text{Out}}, E_{i,0}^{\text{Out}} \rangle, \ldots, \langle \text{key}_{i,n}^{\text{Out}}, E_{i,n}^{\text{Out}} \rangle$

Algorithm 1: Key Negotiation(·).

Input: s_i, s_{i-1}, s_{i+1}
Output: True or **False**
(1) **for** each $\text{in}_{i,x}^{M} \in \text{In}_{i}^{M}$ **do**
(2) **if** Verification($\text{in}_{i,x}^{M}$) == **False then**
(3) $\text{In}_{i,\text{insec}}^{M} \leftarrow \{\text{in}_{i,x}^{M}\} \cup \text{In}_{i,\text{insec}}^{M}$
(4) **end if**
(5) **end for**
(6) **if** Key_ Negotiation(s_i, s_{i-1}) == **False then**
(7) **return False**
(8) **end if**
(9) Compute Out Required (s_i, s_{i-1})
(10) **for** each $\text{out}_{i,y} \in \text{Out}_{i}$ **do**
(11) **if** Verification($\text{out}_{i,y}$) == **False then**
(12) $\text{Out}_{i,\text{insec}} \leftarrow \{\text{out}_{i,y}\} \cup \text{Out}_{i,\text{insec}}$
(13) **end if**
(14) **end for**
(15) **if** Key_ Negotiation(s_i, s_{i+1}) == **False then**
(16) **return False**
(17) **end if**
(18) **return True**

Algorithm 2: Adjacent Verify & Declass(·).

Table 1: Basic comparison.

	Approach	Framework	Information declassifying
Hutter and Volkamer [9]	Type system	Centralized	×
Nakajima [12]	Model checking	Centralized	√
She et al. [13–15]	Program analysis	Centralized	×
Chou [19]	Program analysis	Distributed	×
Solanki et al. [20]	Program analysis	Distributed	×
Xi et al. [21]	Model checking	Distributed	×
Xi et al. [16]	Program analysis	Distributed	×
This paper	Program analysis	Distributed	√

Table 2: Simulation Configuration.

Network settings	
Simulator	NS-3
Field (m^2)	1000×1000
Cloud domain	3
Cloud platform	3
Mobile nodes	100
Radio type	802.11g
Mobility model	RandomWalk
Backbone network	1 Gbps
Security settings	
Security level	U, C, S, T

our approach is distributed, which is more appropriate for the verification across multiple domains.

In addition, we evaluate the performance of typical approaches, that is, our approach, global model checking [12], centralized program analysis [15], and distributed verification [20, 21] in multiple scenarios by using NS-3 [26]. Basic encryption functions are provided by OPENSSL library [27]. The basic settings of multiple clouds in mobile network are shown as Table 2.

Our simulated mobile network covers about $1000 \times 1000\,\text{m}^2$, which involves three cloud domains, three cloud platforms, and about 100 mobile nodes. For each domain in mobile network, there is one cloud platform and random number of mobile nodes. The communication adopts advanced 802.11g technology, and the mobility model for each node uses the standard *RandomWalk* model. For the backbone network, it connects different cloud domains with wired Gigabit Ethernet. Based on the settings of network, we develop multiple services and deploy them to the cloud platform and specific mobile nodes for dynamic service composition. Meanwhile, we define four different security levels for the information used in service composition, that is, unclassified (U), confidential (C), secret (S), and top secret (T), according to the standard multilevel security model which has been widely applied in government and military systems [23].

Based on the designed mobile network, we simulate the service composition process in multiple mobile clouds. During the simulation, we investigate the success rate and

Input: $s_i, s_{i+1}, \ldots, s_{i+n} \in \mathrm{dom}_{\mathrm{CP}}, \mathrm{CP}_{s_{i-1}}, \mathrm{CP}_{s_{i+n+1}}$
Output: **True** or **Flase**
(1) wait for start_message
(2) $\mathrm{Res}_{s_{i-1}} = \mathrm{ReceInterRes}(\mathrm{CP}_{s_{i-1}})$
(3) $\mathrm{AP} = \mathrm{GenAllPath}\ (\mathrm{Res}_{s_{i-1}} \cdot \mathrm{PP}, s_{i+1}, \ldots, s_{i+n})$
(4) **for** each path $p \in \mathrm{AP}$ **do**
(5) **for** each step m from $i-1$ to $i+n$ **do**
(6) **if** Adjacent Verify & Declass $(s_i, s_{i-1}, s_{i+1}) ==$ **True then**
(7) RCount++
(8) **end if**
(9) **end for**
(10) **if** RCount $== n$ **then**
(11) Push p into $\mathrm{Res}_{s_{i+n}} \cdot \mathrm{PP}$
(12) **for** each $\mathrm{out}^M_{i+n} \in s_{i+n}$ **do**
(13) $\mathrm{Res}_{s_{i+n}} \cdot \mathrm{Re}(\mathrm{out}^M_{i+n}) = \mathrm{Re}(\mathrm{out}^M_{i+n})$
(14) **end for**
(15) **end if**
(16) **end for**
(17) **if** $|\mathrm{PP}| == 0$ **then**
(18) send failure_message to user S_0
(19) **else**
(20) **if** $i + n == N$ **then**
(21) send success message to user s_{N+1}
(22) **else**
(23) send start message to next cloud platform $\mathrm{CP}_{s_{i+n+1}}$
(24) **end if**
(25) **end if**

Algorithm 3: Distributed Compos & Declass(·).

Table 3: Experiment scenario.

Success rate	
Candidate Number	0–20
Service step	1–15
Approaches	Our approach, Solanki et al. [20], She et al. [15], Xi et al. [21]
Composition Time	
Candidate Number	1–15
Service step	1–8
Approaches	Our approach, Xi et al. [21], She et al. [15], Nakajima [12]

time cost on the composition with the different number of service steps and candidate services. The variations of the simulation are shown in the Table 3.

Figures 6 and 7 show the success rate of service composition with different number of candidate services and service steps. With the increase of candidate service number, the composition procedures are more likely to be succeed. Because of the looser security constraints, the success rate of our approach increases much faster than the other approaches. Besides, for [15, 20, 21], the rate decreases vastly when there are too many service steps involved in composite service. With the execution of the verification, the requirements on the inputs and outputs become more strict and fewer candidate services can satisfy it. For our approach, most candidate services can still satisfy the information flow security constraints because of the declassification on data.

Figures 8 and 9 show the cost on composition time with dynamic candidate service number and service step. When the number of candidate service or service step is small, the difference on time cost is not too much. However, for global model checking way [12] and centralized program analysis [15], they are all centralized verification approaches in which all possible composite services must be verified one by one. Therefore, with the increase of candidate services, the complexity of modeling the composite service increases vastly, and it is a time-consuming work to check the complicated model. For our approach and [21], they avoid the repetitive verification on some candidate services, so it provides an more efficient way for secure service composition as the increase of candidate services. Besides, because the key negotiation procedure is executed by related candidate services, the cloud platform can continue verifying other candidate services. Both procedures can br executed in a parallel way. Therefore, the extra effort is not evidence compared with that in [21].

6. Conclusion

In this paper, we propose a declassification mechanism for secure service composition based on cryptographic operations and information flow security requirements. Considering the multidomain characters of mobile clouds, a distributed secure service composition with declassification framework and approach is proposed to overcome the high-rate

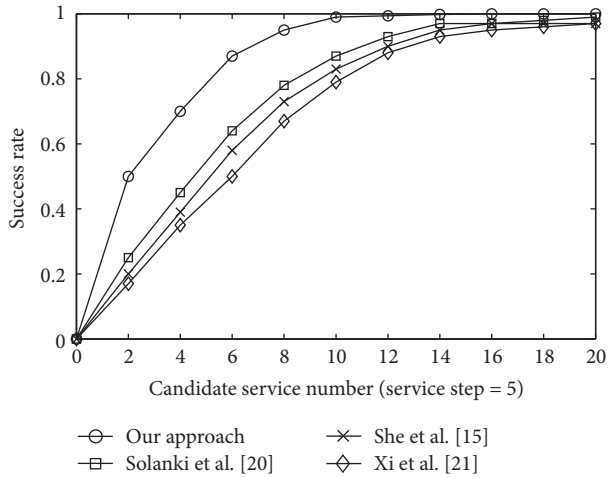


Figure 6: Success rate of composition with candidate service number.

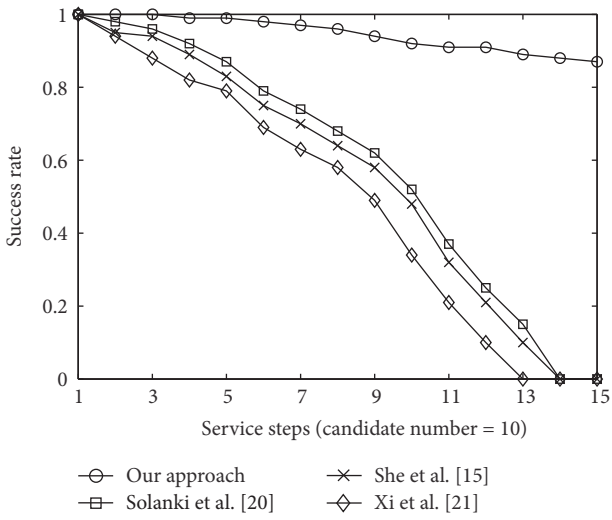


Figure 7: Success rate of composition with service step.

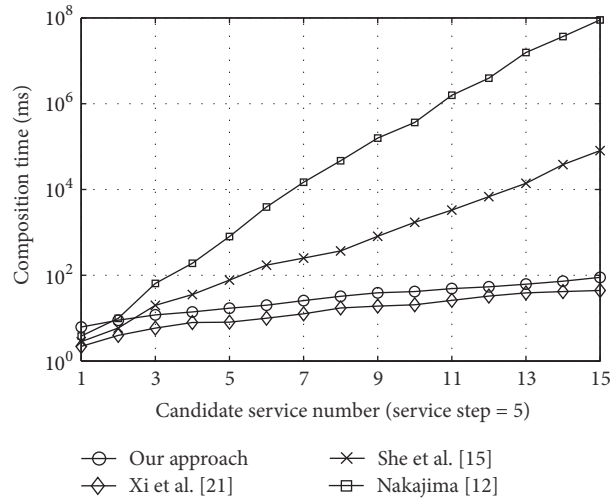


Figure 8: Composition time with candidate service number.

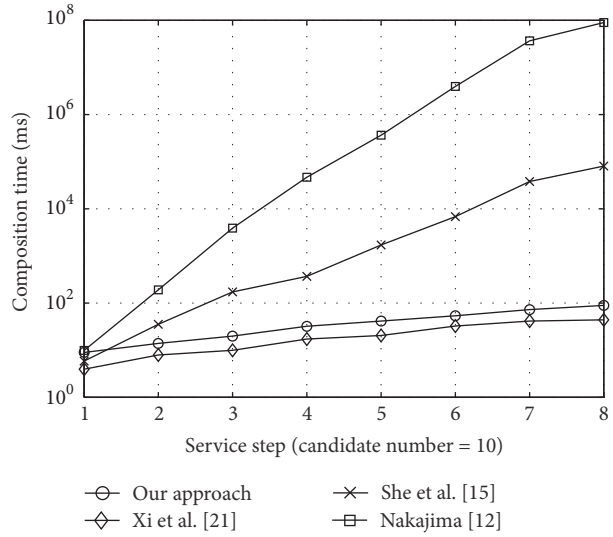


Figure 9: Composition time with service step.

failure of composition, which is caused by too strict security constraints in the traditional composition methods. Through the evaluation on NS-3, the results show our approach can improve the success rate of service composition effectively while the additional cost can be affordable. More dynamic declassification policies for service composition with complex structure will be considered in the future.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (61502368, 61303033, and U1405255), the National High Technology Research and Development Program (863 Program) of China (no. 2015AA017203 and no. 2015AA016007), the Fundamental Research Funds for the Central Universities (XJS14072 and JB150308), Natural Science Basis Research Plan in Shaanxi Province of China (Grant no. 2016JM6034), Xi'an Technology Research Project (CXY1402), and the Aviation Science Foundation of China (no. 20141931001).

References

[1] P.-J. Maenhaut, H. Moens, B. Volckaert, V. Ongenae, and F. De Turck, "Design of a hierarchical software-defined storage system for data-intensive multi-tenant cloud applications," in *Proceedings of the 11th International Conference on Network and Service Management (CNSM '15)*, pp. 22–28, IEEE, Barcelona, Spain, November 2015.

[2] S. Sezer, S. Scott-Hayward, P. Chouhan et al., "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, 2013.

[3] N. Chalaemwongwan and W. Kurutach, "Mobile cloud computing: a survey and propose solution framework," in *Proceedings of the 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON '16)*, pp. 1–4, Chiang Mai, Thailand, June 2016.

[4] N. Chen, N. Cardozo, and S. Clarke, "Goal-driven service composition in mobile and pervasive computing," *IEEE Transactions on Services Computing*, 2016.

[5] F. Paganelli, M. Ulema, and B. Martini, "Context-aware service composition and delivery in NGSONs over SDN," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 97–105, 2014.

[6] C. Groba and S. Clarke, "Opportunistic composition of sequentially-connected services in mobile computing environments," in *Proceedings of the IEEE 9th International Conference on Web Services (ICWS '11)*, pp. 17–24, IEEE, Washington, DC, USA, July 2011.

[7] C. Danwei, H. Xiuli, and R. Xunyi, "Access control of cloud service based on UCON," in *Proceedings of the IEEE International Conference on Cloud Computing*, pp. 559–564, Springer, Beijing, China, December 2009.

[8] D. Volpano, C. Irvine, and G. Smith, "Sound type system for secure flow analysis," *Journal of Computer Security*, vol. 4, no. 2-3, pp. 167–187, 1996.

[9] D. Hutter and M. Volkamer, "Information flow control to secure dynamic web service composition," in *Security in Pervasive Computing*, pp. 196–210, Springer, Berlin, Germany, 2006.

[10] R. Accorsi and C. Wonnemann, "Static information flow analysis of workflow models," in *Proceedings of the 2nd International Symposium on Services Science, ISSS 2010 and 3rd International Conference on Business Process and Services Computing (BPSC '10— INFORMATIK '10)*, pp. 194–205, October 2010.

[11] R. Dimitrova, B. Finkbeiner, M. Kovács, M. N. Rabe, and H. Seidl, "Model checking information flow in reactive systems," in *Verification, Model Checking, and Abstract Interpretation*, pp. 169–185, Springer, Berlin, Germany, 2012.

[12] S. Nakajima, "Model-checking of safety and security aspects in web service flows," in *Web Engineering*, pp. 488–501, Springer, Berlin, Germany, 2004.

[13] W. She, I.-L. Yen, B. Thuraisingham, and E. Bertino, "The SCIFC model for information flow control in web service composition," in *Proceedings of the IEEE International Conference on Web Services (ICWS '09)*, pp. 1–8, IEEE, Los Angeles, Calif, USA, July 2009.

[14] W. She, I.-L. Yen, B. Thuraisingham, and E. Bertino, "Policy-driven service composition with information flow control," in *Proceedings of the IEEE 8th International Conference on Web Services (ICWS '10)*, pp. 50–57, Miami, Fla, USA, July 2010.

[15] W. She, I.-L. Yen, B. Thuraisingham, and E. Bertino, "Security-aware service composition with fine-grained information flow control," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 330–343, 2013.

[16] N. Xi, J. Ma, C. Sun, and T. Zhang, "Decentralized information flow verification framework for the service chain composition in mobile computing environments," in *Proceedings of the IEEE 20th International Conference on Web Services (ICWS '13)*, pp. 563–570, Santa Clara, Calif, USA, July 2013.

[17] E. J. Schwartz, T. Avgerinos, and D. Brumley, "All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask)," in *Proceedings of the 31st IEEE Symposium on Security and Privacy (SP '10)*, pp. 317–331, IEEE, Oakland, Calif, USA, May 2010.

[18] J. Bacon, D. Eyers, T. F. J.-M. Pasquier, J. Singh, I. Papagiannis, and P. Pietzuch, "Information flow control for secure cloud computing," *IEEE Transactions on Network and Service Management*, vol. 11, no. 1, pp. 76–89, 2014.

[19] S.-C. Chou, "Controlling information flows in SaaS cloud services," in *Proceedings of the 7th International Conference on Computing and Convergence Technology (ICCIT, ICEI and ICACT) (ICCCT '12)*, pp. 651–656, Seoul, Korea, December 2012.

[20] N. Solanki, T. Hoffman, I. L. Yen, F. Bastani, and S. S. Yau, "An access and information flow control paradigm for secure information sharing in service-based systems," in *Proceedings of the IEEE 39th Annual Computer Software and Applications Conference (COMPSAC '15)*, vol. 1, pp. 60–67, July 2015.

[21] N. Xi, C. Sun, J. Ma, and Y. Shen, "Secure service composition with information flow control in service clouds," *Future Generation Computer Systems*, vol. 49, pp. 142–148, 2015.

[22] N. Xi, C. Sun, J. Ma, Y. Shen, and D. Lu, "Distributed secure service composition with declassification in mobile network," in *Proceedings of the International Conference on Networking and Network Applications (NaNA '16)*, pp. 254–259, IEEE, Hakodate, Japan, July 2016.

[23] D. E. Denning, "A lattice model of secure information flow," *Communications of the Association for Computing Machinery*, vol. 19, no. 5, pp. 236–243, 1976.

[24] C. Gentry and S. Halevi, "Implementing Gentry's fully—homomorphic encryption scheme," in *Advances in cryptology—EUROCRYPT 2011*, vol. 6632 of *Lecture Notes in Computer Science*, pp. 129–148, Springer, Heidelberg, Germany, 2011.

[25] S. Luo, J. Hu, and Z. Chen, "Implementing attribute-based encryption in web services," in *Proceedings of the IEEE 8th International Conference on Web Services (ICWS '10)*, pp. 658–659, IEEE, Miami, Fla, USA, July 2010.

[26] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," in *Proceedings of the SIGCOMM Demonstration*, vol. 14, 2008.

[27] P. Chandra, M. Messier, and J. Viega, *Network Security with Openssl*, O'Reily, 2002.

Hindawi
Mobile Information Systems
Volume 2017, Article ID 3725898, 14 pages
http://dx.doi.org/10.1155/2017/3725898

Research Article

A Novel Video Sharing Solution Based on Demand-Aware Resource Caching Optimization in Wireless Mobile Networks

Shijie Jia,[1,2] Ruiling Zhang,[2] Shengli Jiang,[2] and Mingchuan Zhang[3]

[1]*Central Plains Economic Zone Wisdom Tourism Cooperative Innovation Center, Luoyang Normal University, Luoyang, Henan 471934, China*
[2]*Academy of Information Technology, Luoyang Normal University, Luoyang, Henan 471934, China*
[3]*Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China*

Correspondence should be addressed to Ruiling Zhang; ruilingzhang@163.com

Received 29 October 2016; Accepted 22 December 2016; Published 19 January 2017

Academic Editor: George Ghinea

The video services in wireless mobile networks attract massive mobile video users, which triggers the demand for huge network traffic. In order to reduce network load and ensure user quality of experience (QoE), the MP2P-based video systems need to implement high-efficiency management and scheduling of video resources to optimize distribution and promote sharing with offloading traffic in underlaying networks. The traditional solutions do not efficiently address the problems caused by demand variation and node mobility for distribution optimization and resource sharing, so that promotion of video sharing performance in wireless mobile networks for user QoE and reduction of network load still is a significant challenge. This paper proposes a novel video sharing solution based on Demand-aware Resource Caching Optimization in wireless mobile networks (VDRCO). VDRCO models the process of video dissemination in overlay networks in terms of the epidemic model and discusses the main influence factors for video dissemination. By estimation for user demand and video distribution variation, VDRCO makes use of clustering nodes with similar video demand and movement behaviors to achieve distribution optimization and efficient video sharing. Extensive tests show how VDRCO achieves much better performance results in comparison with a state-of-the-art solution.

1. Introduction

The deployment and popularity of video streaming services in wireless mobile networks are benefited from fast increase of wireless bandwidth and continuous improvement [1]. The promotion of wireless bandwidth capacity such as WiMAX, 4G, and 5G can accommodate massive network traffic brought by high-definition video application [2], which improves visual experience of users; on the other hand, the various technologies of network access such as MANETs, VANETs, and cellular network enable users to conveniently watch video content via smart terminal [3]. As Figure 1 shows, by leveraging the network virtualization methods, the video systems employ the mobile peer-to-peer (MP2P) technologies to achieve large-scale deployment of video applications and promote video sharing in wireless mobile networks [4–9]. The MP2P-based video systems make use of management and scheduling for video resources in overlay networks to promote video sharing performance. The efficiency of video sharing is a key factor for system quality of service (QoS) and user quality of experience (QoE). The increase in the scale of video users brings huge traffic load for the core network. Offloading video traffic in underlaying networks can relieve load of the core network and reduce delay of video lookup and transmission [10, 11]. However, this requires video requesters always can find and connect with geographic adjacent providers to fetch desired videos (called "near-end video fetching") instead of geographic far providers (called "far-end video fetching"). The near-end video fetching in wireless mobile networks relies on the two perspectives: (1) the efficient resource management achieves optimal video distribution to ensure local balance between supply and demand of resources in geographic area; (2) the efficient resource scheduling supports fast discovery of

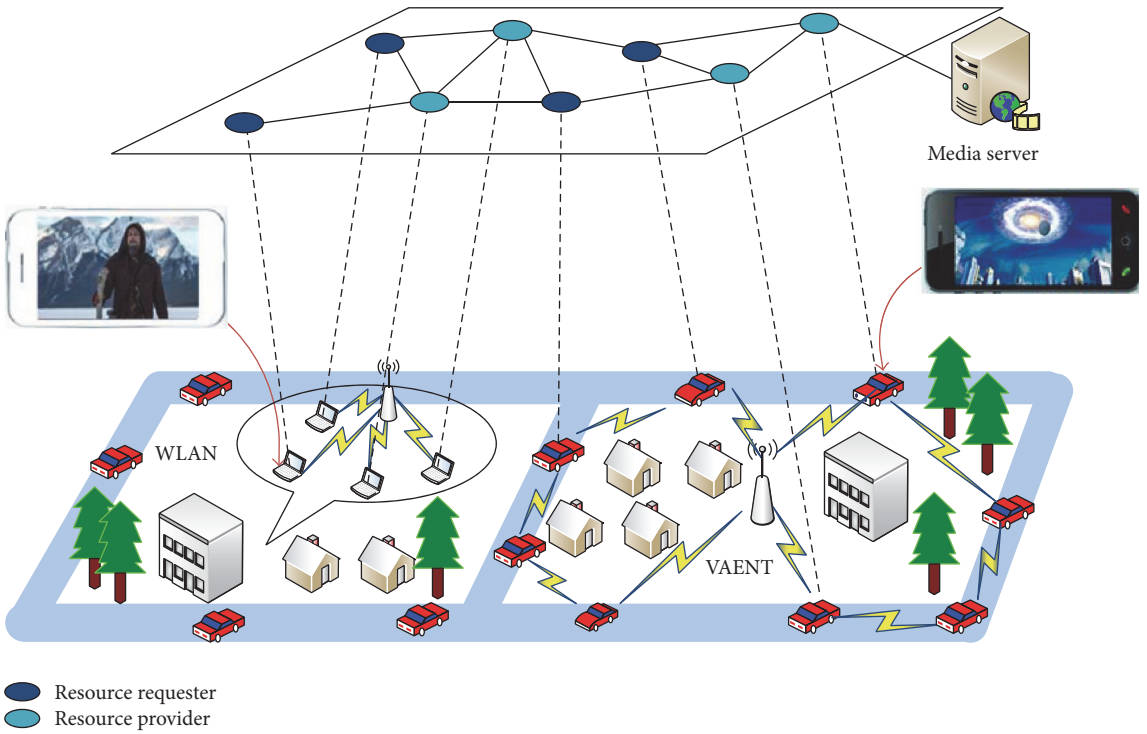


Figure 1: MP2P-based video streaming services in wireless mobile networks.

geographical adjacent video providers on the basis of the optimized distribution.

Recently, massive studies focus on making use of video caching and scheduling to promote sharing performance. For instance, the cooperative caching methods can make use of unoccupied storage space of nodes to cache popular videos in the future [12, 13]. The prefetching methods enable the nodes prestore the watched videos with high probability in the future, which reduces startup delay of request nodes and ensures smooth playback [14, 15]. The compelling caching assignment methods make use of the available storage resources to implement caching optimization in terms of variation of video distribution [16–21]. The social-based resource scheduling methods further promote resource sharing performance by cache integration in terms of social relationship [22–25]. However, the above methods do not efficiently address the negative influence caused by demand variation and node mobility for balance stability of video distribution, which results in low-efficiency scheduling and high-cost management [26]. In fact, user interest for video content is the determining factor for requesting or removing a video. The interest variation drives the nodes to request the desired videos regardless of whether near-end or far-end fetching. In other words, if the requested videos cannot be met by geographical adjacent video providers, the requesters also search geographical far providers, which increase delay of video lookup and transmission. Moreover, when the nodes lose interest for videos in local buffer, they remove the videos in order to cache interested videos. Obviously, variation of node interest is an important influence factor for balance stability of video distribution. On the other hand, the mobile nodes carry the own video resources and move from a geographical area to anther geographical area, which also causes variation of video distribution. A key issue for promotion of video sharing performance in wireless mobile networks is how to address the problems caused by interest variation and node mobility.

In this paper, we propose a novel video sharing solution based on Demand-aware Resource Caching Optimization in wireless mobile networks (VDRCO). By making use of the epidemic model, VDRCO models video dissemination process in overlay networks and analyzes the four influence factors for video dissemination and distribution: number and influence of initial providers and attraction and number of disseminated videos. The influence of initial providers and the attraction of disseminated videos bring severe influence for user demand. The number of initial providers and disseminated videos influences resource distribution in the process of video dissemination. Based on the above analysis, VDRCO designs the two measurement methods for user demand and video distribution variation. By clustering the nodes in terms of interaction frequency and similarity of interest and mobility, VDRCO further designs a video sharing strategy to promote efficiency of near-end video sharing based on the measurement results of user demand and video distribution variation. Simulation results show that VDRCO can obtain much better performance of video delivery in comparison with a state-of-the-art solution.

2. Related Work

Recently, many studies employ caching and scheduling methods to improve resource distribution and delivery. For instance, the asymmetric cooperative resource caching approach proposed in [13] enables the nodes to build cache layer, so that request and response of data are transmitted to the cache layer, which efficiently reduces overhead of data copying and decreases end-to-end transmission delay. In [14], Kozat et al. designed the three cache mechanisms in P2P-based systems: (1) cache prefetching: the peers prefetch all content even if the content is not desired; (2) opportunistic cache update: if a cached content is last in local buffer, the peers can replace the content; (3) hybrid scheme composed of the above two strategies, which can efficiently reduce the server load. In [16], Tan and Massoulié designed the caching strategies in P2P system based on the two scenarios: (1) pure P2P network and (2) distributed server networks. In pure P2P network, each peer assigns one cache slot for the most popular content and the remaining cache slots are assigned for the more popular content. In distributed server networks, the peers randomly cache content according to the overall popularity distribution and switch out content from cache space with uniformly probability. In [18], a commercial caching system is designed, which includes a video retailer (VR) and multiple network service providers (NSPs). By caching popular videos in the rented SBSs, the VR leases small-cell base stations (SBS) of NSPs to provide better local video services to the mobile users. In [19], Mokhtarian and Jacobsen designed the four algorithms: (1) a flexible ingress-efficient algorithm; (2) an offline caching algorithm based on awareness of future requests; (3) an optimal offline caching algorithm; (4) an adaptive ingress control algorithm, handling the demand for content caching on the servers of CDNs. In [20], a cache replacement method, Trend-Caching, is proposed, which optimizes caching performance according to the popularity trends of video content. Trend-Caching makes the decision of caching and removing videos in terms of popularity trend of video content, which achieves fast convergence and improves cache hit rate.

In order to further promote resource sharing performance, massive researchers have leveraged other technologies, such as the social network, to improve efficiency of resource caching and scheduling. For instance, in [22], SAVE considers channel switching as the interactions between channels and clusters channels with high-frequency interactions. SAVE builds contacts between channels with high-frequency and low-frequency interactions, which enables the nodes to stay in the same channel clusters or makes use of interchannel contacts to join a new channel instead of accessing server. The nodes also make use of the maintained friend lists to record common-interest channels and watching time shared by nodes, which achieves fast channel switching and reduces server load. Zhang et al. designed a resource prefetching strategy for video services based on online social network [24]. Because the popular videos are watched with high probability, the nodes can prefetch prefix chunks of popular videos, which reduce startup delay and achieves smooth playback. In [23], social-P2P groups nodes with common interests and close social relationship into a cluster by investigation of social interests and relationships in social network. Furthermore, the relatively stable nodes in each cluster also are clustered in terms of the Distributed Hash Table (DHT). When any node requests video content, it sends request message to content providers by making use of the DHT structure, which improves lookup success rate and reduces lookup delay.

Moreover, some resource caching methods in new network architecture such as Content-Centric Networks (CCNs) and Information-Centric Networks (ICNs) are proposed. In [27], CodingCache makes use of network coding and random forwarding to support multipath routing in CCNs and improve diversity of cached content along different paths, which increases cache hit rate with reasonable cache privacy. Si et al. investigated the use of harvested bands for caching video contents closed to the interested nodes [28], in order to improve the performance of video distribution. Based on transforming the allocation of harvested bands (a Markov decision process) into a partially observable Markov decision process and a multiarmed bandit formulation, a spectrum management mechanism is designed, which improves efficiency of proactive video caching and spectrum utilization.

3. Model of Video Dissemination

The management for video resources cached by nodes in overlay networks focuses on achieving optimal video distribution to support high-efficiency video sharing. The local balance between supply and demand of videos is an important characteristic of optimal video distribution. For instance, when a node n_i always can find the video providers which cache the requested video from one-hop neighbor nodes of n_i, the one-hop neighbor relationship can minimize the number of forwarding request message and video data, which ensures low delay of video lookup and transmission and reduces risk of video data loss. Obviously, the local balance between supply and demand of videos can efficiently support near-end video fetching (e.g., the above one-hop video fetching) to promote video sharing performance.

The maintenance for stability of local balance of video resources in wireless mobile nodes is a significant challenge. This is as interest variation and node mobility bring severely negative influence for local balance of video resources. For instance, a node n_i has watched all videos cached in one-hop neighbor nodes of n_i, so that n_i loses interests for videos with high probability. In the other words, n_i does not request videos which have been watched and may fetch new videos from other nodes instead of one-hop neighbor nodes. Obviously, the interest variation results in imbalance of video distribution. On the other hand, because the mobile nodes are the carriers of video resources, the node mobility makes the video resources move in geographical area, which also results in variation of video distribution in geographical area. The maintenance for stability of local balance needs continuous adjustment (caching new videos and removing old videos) in terms of interest variation and node mobility. The variation of user interests can change the number of available resources in video systems, which determines the

performance of video lookup. The node mobility only leads to the variation of geographical distribution of video resources instead of changing total amount of available video resources, which determines the performance of video delivery.

3.1. Construction of Video Dissemination Model. For convenience, several notations used in sections are defined in Notations (notations used by models). The video distribution is continuous variation in the whole process of video dissemination. This is as the video dissemination process has obvious periodicity. When a new video v_i appears in overlay networks, the nodes have potential interest for v_i. Some nodes cache v_i and push v_i to other contacted nodes, which increases the range and speed of dissemination of v_i. At the moment, number of cached v_i in overlay networks fast increases. When the most of nodes have watched v_i, they lose the interests for v_i and remove v_i in local buffer in terms of their storage capacity. The number of cached v_i in overlay networks fast decreases. Therefore, the video dissemination process can be influenced by variation of user interests instead of the node mobility. The initial providers of v_i act as the infected nodes. The nodes interested in v_i are considered as the susceptible nodes. The nodes which have watched v_i are considered as the immune nodes. In the process of video dissemination, the susceptible nodes actively send request messages to search v_i (pull mode) and the initial providers send the advertisement messages to push v_i for other nodes (push mode). The two modes of pull and push enable v_i to be spread in overlay networks. Therefore, the dissemination process of v_i follows the epidemic model [29]. N is the total number of nodes in overlay networks. $N = N(S)+N(I)+N(R)$, where $N(S)$ is the number of nodes interested in v_i; $N(I)$ is the number of nodes which have cached v_i; $N(R)$ is the number of nodes which have been uninterested in v_i and have removed v_i in local buffer. $S = N(S)/N$, $I = N(I)/N$, and $R = N(R)/N$ denote the ratio between $N(S)$, $N(I)$, $N(R)$, and N, respectively. At the time t, S, I, and R meet the equation $S(t) + R(t) + I(t) = 1$ and the dissemination model of v_i can be defined as

$$\begin{aligned} \frac{dI}{dt} &= \lambda SI - \mu I, \\ \frac{dS}{dt} &= -\lambda SI, \\ \frac{dR}{dt} &= \mu I, \end{aligned} \tag{1}$$

where λ and μ are the increasing rate of nodes which request and remove v_i, respectively. If the content of v_i is wonderful, the probability of requesting v_i for nodes in overlay networks is high. If the nodes which have cached v_i have strong influence level for other nodes, the pushed v_i is accepted with high probability. Therefore, the wonderful content and strong influence level of push nodes make the value of λ keep high levels, so that v_i can be fast disseminated to the whole overlay networks (the value of I also keeps fast increase). If the nodes cached in v_i have watched v_i, they lose the interests for v_i and remove v_i in local buffer, so that the values of μ also keep high levels (the value of R fast increases).

3.2. Analysis for Video Distribution Variation. We further describe variation process of λ and μ to discuss the variation of distribution of v_i.

(1) Initially, because the number of initial providers is less and there are some nodes which are uninterested in v_i in overlay networks, the dissemination rate of v_i is low and the value of λ slowly increases. At the moment, because the nodes which cache v_i do not finish the playback for v_i, the value of μ is 0.

(2) With the increase in the dissemination time, if the content of v_i is wonderful, the large amount of nodes make use of pull mode to request and cache v_i; if the nodes which have cached v_i have strong influence level for other nodes, the pushed v_i easily is accepted. The value of λ fast increases and reaches the peak. At the moment, because some nodes have watched v_i and lose the interests for v_i, they remove v_i in local buffer. The value of μ keeps the increase trend.

(3) When the increment of number of nodes interested in v_i decreases, the value of λ decreases. At the moment, because the large amount of nodes have watched v_i and remove v_i, the value of μ fast increases and reaches the peak.

(4) When the number of nodes interested in v_i keeps relatively stable at the later stage of dissemination process, the value of λ keeps low level. Because the number of nodes which remove v_i decreases, the value of μ fast decreases and also keeps low level.

(5) Finally, when all nodes finish the playback and remove v_i, the values of I and S are 0, respectively (the values of λ and μ are 0). In fact, the number of nodes in overlay networks always keeps dynamic variation in the real video services. The new mobile nodes join into the video systems, so the values of I and S always keep slight fluctuation in the reasonable range.

If the nodes finish the playback and immediately remove v_i, the values of λ and μ have the same variation process. If the nodes finish the playback and do not immediately remove v_i in order to serve other request nodes, the values of μ have longer and more moderate variation process than those of λ. In fact, we only consider the latter case in terms of the characteristic of resource sharing with each other in MP2P-based video systems.

On the other hand, the values of λ and μ reflect the variation levels of distribution of v_i. The attraction of video content and the influence of providers are the two impact factors for the values of λ and μ and variation of user interests. For instance, the two nodes are interested in the NBA and frequently share related videos with each other. If a node pushes videos of the CBA to another node, the latter may accept the pushed videos with high probability. However, if a node pushes videos of the World Cup to another node, the latter may reject the pushed videos with high probability. This is as the pushed videos go beyond the demand (interest) ranges of the latter (the latter is immune for the pushed videos). The wonderful video content can

attract large amount of nodes and the high-level influence of providers increases the probability of push success, which speeds up the video dissemination. As we know, the fast dissemination can increase the provision of available resources to reduce startup delay of request nodes caused by shortage of resources. In order to efficiently improve rate and range of video dissemination, a key issue is how to be aware of user demand and select appropriate push nodes.

Except for attraction of video content and influence of providers, number of initial providers and number of disseminated videos are the two influence factors for the dissemination process. The large amount of initial providers not only ensures enough provision of available resources but also increases the value of λ due to extension of push range. Moreover, the dissemination of multiple videos in overlay networks not only leads to the fast change of user interests but also causes the competition for bandwidth and storage resources of nodes. For instance, the small number of new videos lengthens the process of video dissemination. This is as the nodes do not fast lose the interests for the existing videos in overlay networks. The content providers do not eagerly remove v_i to fetch new videos due to limited storage space. Otherwise, if the massive new videos appear in overlay networks, the nodes continually request more interested videos, which reduce the stay time of v_i in local buffer of content providers. This speeds up the convergence of process of video dissemination, which results in the fast variation of distribution of v_i. Therefore, in order to improve video sharing performance, the management and scheduling of video resources need to consider awareness of user demand, selection of push nodes in terms of estimation of common interests between nodes, and monitoring of variation level of video distribution.

4. VDRCO Detailed Design

As we know, the optimal video distribution can improve resources sharing performance of video systems and reduce delay of video lookup and delivery. The accurate awareness for user demand and distribution variation level is significant for optimization of video distribution. The awareness for user demand enables the video systems to be aware of caching what resources. The awareness for distribution variation level enables the video systems to be aware of period time of scheduling resources.

4.1. Awareness of User Demand. When the users are interested in videos, they send request messages to fetch and cache the requested videos. Obviously, the requested videos can embody the interests of users and the interest range of users is limited. For instance, a user has the highest interest level for the NBA-related videos among all videos. The user pays more time to search and watch the NBA-related videos. On the other hand, because the users may push some interested videos to other users, the pushed videos also reflect the interests levels of users. Therefore, the videos which are watched for a long time or are pushed by users can represent the interest of users. The interest range can be considered as demand range of users.

In order to accurately measure the interest range, we cluster the videos watched by users and define the interest range in terms of the interest levels of users for videos. The set of clustered videos is considered as the interest range of users. Let $S_v(n_i)$ denote the set of watched videos of n_i. The similarity S_{jk} between any two videos $v_j \in S_v(n_i)$ and $v_k \in S_v(n_i)$ can be calculated according to our previous work [30]. If S_{jk} is the largest among the similarity values between v_k and other items in $S_v(n_i)$, v_j and v_k form a new video subset $S_{\text{sub}_j} = (v_j, v_k)$, as Figure 2(a) shows. Similarly, if S_{hk} denotes the similarity between $v_h \in S_v(n_i)$ and $v_k \in S_v(n_i)$ and S_{hk} is larger than the similarity values between v_h and other items in $S_v(n_i)$, v_h joins into S_{sub_j}, as Figure 2(b) shows. After the iteration of video clustering, the items in $S_v(n_i)$ are grouped into several video subsets; namely, $S_v(n_i) = (S_{\text{sub}_a}, S_{\text{sub}_b}, \ldots, S_{\text{sub}_j})$, as Figure 2(c) shows. Because any video in each subset S_{sub_j} only has a similarity value with another video in S_{sub_j}, all items in S_{sub_j} can be ranked in terms of the ascending similarity; namely, $S_{\text{sub}_j} = (v_a, v_b, \ldots, v_j)$. In fact, the video subsets may include many noise items. For instance, $v_s \in S_{\text{sub}_j}$ and the similarity value S_{sk} between v_s and v_k is larger than the values between v_s and other items in $S_v(n_i)$. If the value of S_{sk} is very small and the similarity values between v_s and other items in S_{sub_j} are also very small, v_s is unsuitable to become the member in S_{sub_j}; namely, v_s is a noise item in S_{sub_j}. This is as the above video clustering only considers the similarity between v_s and v_k and neglects the similarity between v_s and other items in S_{sub_j}. In order to remove the noise items in video subsets, we firstly calculate the average value of similarity among all items in S_{sub_j} according to the following equation:

$$\overline{S}\left(S_{\text{sub}_j}\right) = \frac{\sum_{c=1}^{|S_{\text{sub}_j}|} \sum_{e=1}^{|S_{\text{sub}_j}|-1} S_{ce}}{\left|S_{\text{sub}_j}\right| \left(\left|S_{\text{sub}_j}\right| - 1\right)}, \quad \overline{S}\left(S_{\text{sub}_j}\right) \in [0, 1], \tag{2}$$

where $|S_{\text{sub}_j}|$ returns the number of items in S_{sub_j}. The initial average value of similarity among items in S_{sub_j} is defined as $\overline{S}(S_{\text{sub}_j})^{(1)}$. Because all items in S_{sub_j} have been ranked in terms of the ascending similarity and v_a is first item with minimum similarity, v_a is removed from S_{sub_j} and other items form a new set $S^{(2)}_{\text{sub}_j}$. The average value of similarity among all items in $S^{(2)}_{\text{sub}_j}$ is recalculated according to (2); namely $\overline{S}(S_{\text{sub}_j})^{(2)}$. If $\overline{S}(S_{\text{sub}_j})^{(2)} < \overline{S}(S_{\text{sub}_j})^{(1)}$, v_a is not a noise item and is readded to $S^{(2)}_{\text{sub}_j}$. Otherwise, if $\overline{S}(S_{\text{sub}_j})^{(2)} > \overline{S}(S_{\text{sub}_j})^{(1)}$, v_a is considered as a noise item. Although the similarity value between v_a and another item in S_{sub_j} is high, v_a may have the low similarity with other items in S_{sub_j}. Therefore, when v_a is removed, the average value of similarity in $S^{(2)}_{\text{sub}_j}$ increases. After making the decision of retaining or removing v_a, $S^{(2)}_{\text{sub}_j}$

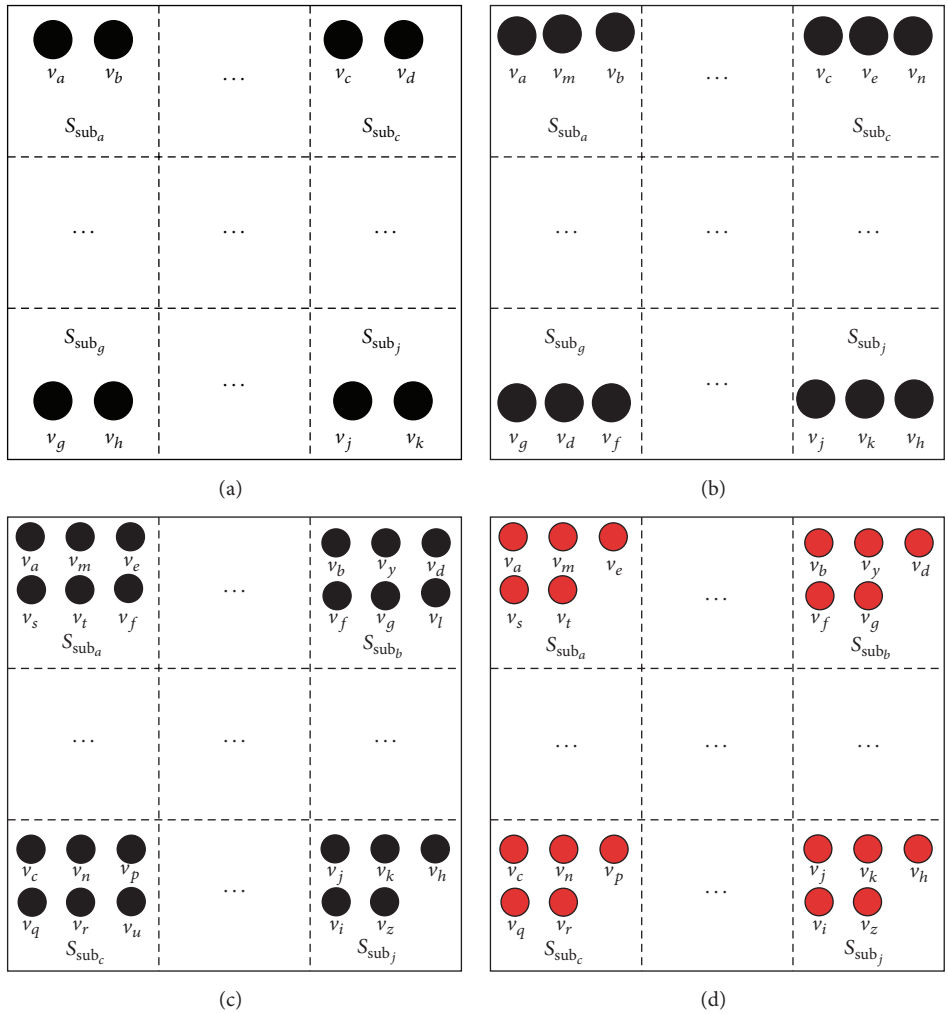


Figure 2: Estimation of interest range.

continues to remove v_b. Other items in $S^{(2)}_{\text{sub}_j}$ form a new subset $S^{(3)}_{\text{sub}_j}$. If $\overline{S}(S_{\text{sub}_j})^{(3)} > \overline{S}(S_{\text{sub}_j})^{(2)}$, v_b is a noise item. Otherwise, if $\overline{S}(S_{\text{sub}_j})^{(3)} < \overline{S}(S_{\text{sub}_j})^{(2)}$, v_b is not a noise item and is readded to $S^{(2)}_{\text{sub}_j}$. By iteration of the above process, the noise videos in S_{sub_j} are removed, as Figure 2(d) shows. Similarly, the noise items in all video subsets can be removed and form a new video subset. We continue to cluster and refine items in subset composed of noise videos according to the above process. The condition of convergence is that each noise video forms a subset.

In order to extract the demand regions of n_i from the video subsets, we firstly make use of the watched time and pushed frequency to estimate the interest level for any video v_j according to the following equation:

$$I_i\left(v_j\right) = \alpha \frac{l_i\left(v_j\right)}{L_i} + \beta \frac{f_i\left(v_j\right)}{F_i}, \quad \alpha, \beta \in [0,1], \tag{3}$$

where $l_i(v_j)$ is the watched time of a node n_i for v_j; L_i is the total playback time of n_i for all watched videos; $f_i(v_j)$ is the pushed frequency of n_i for v_j; F_i is the total push frequency of n_i for all watched videos. α and β are the weight values of watched time and push frequency, respectively, and $\alpha + \beta = 1$. Therefore, the value of $I_i(v_j)$ is in the range $[0, 1]$. If a video $v_j \in S_{\text{sub}_j}$ has the highest interest level among all items in S_{sub_j}, v_j acts as the deputy video of S_{sub_j}. Each video subset S_{sub_j} has an average value of interest level according to the following equation:

$$\overline{I}\left(S_{\text{sub}_j}\right) = \frac{\sum_{c=1}^{|S_{\text{sub}_j}|} I_i\left(v_c\right)}{\left|S_{\text{sub}_j}\right|}. \tag{4}$$

Further, the average value of interest level of all items in $S_v(n_i)$ can be defined as

$$\overline{I}\left(S_v\left(n_i\right)\right) = \frac{\sum_{c=1}^{|S_v(n_i)|} \overline{I}\left(S_{\text{sub}_j}\right)}{\left|S_v\left(n_i\right)\right|}, \tag{5}$$

where $|S_v(n_i)|$ returns the number of video subsets included in $S_v(n_i)$. If the value of $\overline{I}(S_{\text{sub}_j})$ is greater than $\overline{I}(S_v(n_i))$, S_{sub_j}

is considered as a demand region of n_i. Therefore, the set of demand regions can be defined as $D_i = (d_1, d_2, \ldots, d_k)$. When n_i requests and watches a new video v_d, the similarity between v_d and deputy videos in all demand regions is calculated. If the similarity between v_d and a deputy video v_j in S_{sub_j} is the largest similarity among all deputy videos, v_d is added to S_{sub_j}. By the purification for the subset, if v_d is a noise video, the similarity values between v_d and all noise videos are calculated. v_d is added to the video subset corresponding to the noise video which has the largest similarity; otherwise, if v_d is not a noise video, it stays in S_{sub_j}.

4.2. Awareness of Variation Level of Video Distribution. When the nodes want to watch new videos and the local buffer is full, they remove the videos which have been watched. The replacement of cached videos in local buffer of nodes directly results in the variation of video distribution in overlay networks. The stay time of cached videos in local buffer is an important factor for the measurement of replacement. For instance, when a request node n_j wants to watch a video v_k, it sends the request message in overlay networks and waits for the returned video data. If a video provider n_i stores v_k and receives the request message of n_j, it needs to transmit video data to n_j and cannot remove v_k in local buffer. In other words, the more the number of received request messages is, the longer the stay time of cached videos is. Obviously, the period time of video dissemination is the impact factor for stay time. The time of downloading v_k from n_i can be defined as

$$\text{TR}_i(v_k) = \frac{\text{size}_k}{\overline{B}_i\left(1 - \overline{P}_i\right)}, \tag{6}$$

where size_k is the size of v_k; $\overline{B}_i$ and $\overline{P}_i$ are the average values of bandwidth and packet loss rate of n_i in the process of transmission. We assume that the video providers remove v_k in local buffer when they do not receive the request messages during a short period time Δt. If the number of request messages received by n_i is NR_i during a period time $t_a < \text{NR}_i \times \text{TR}_i(v_k)$, the stay time of v_k in local buffer of n_i is $\text{NR}_i \times \text{TR}_i(v_k) + \Delta t$. If $\overline{B}_i$ is equal to the playback rate of requesters, the transmission time of v_k is equal to the playback time of requesters. If the number of requesters and providers is NR_k and NP_k for v_k during the period time $t_b - t_a$, respectively, and $\text{NR}_k < \text{NP}_k$, the NR_k video requesters cache and watch v_k and also become video providers of v_k. The number of cached v_k in networks is $2 \times \text{NR}_k$ and the number of v_k removed by providers is $\text{NP}_k - \text{NR}_k$. This is as the $\text{NP}_k - \text{NR}_k$ providers do not receive the request messages for v_k. The variation level of v_k distribution during $t_b - t_a$ is defined as $\text{VL}_k = \text{NR}_k/(\text{NP}_k - \text{NR}_k)$.

On the other hand, the push between nodes also drives the video dissemination except for the video request. For instance, a provider n_i stores a video v_k and pushes v_k to another node n_j. If n_j accepts the pushed v_k, it returns an acknowledgement message to n_i and receives the video data from n_i. Otherwise, if n_j is uninterested in v_k, it neglects the pushed v_k. The push success rate also is an important factor for period time of video dissemination. If the number of successful push for v_k during the period time $t_b - t_a$ of video dissemination is NU_k, the variation level of v_k distribution during $t_b - t_a$ is defined as $\text{VL}_k = (\text{NR}_k + \text{NU}_k)/(\text{NP}_k - \text{NR}_k - \text{NU}_k)$, where $\text{NP}_k > \text{NR}_k + \text{NU}_k$. Obviously, $(\text{NR}_k + \text{NU}_k)/(t_b - t_a)$ and $(\text{NP}_k - \text{NR}_k - \text{NU}_k)/(t_b - t_a)$ denote the increase and decrease rate of v_k, respectively ($\lambda = (\text{NR}_k + \text{NU}_k)/(t_b - t_a)$ and $\mu = (\text{NP}_k - \text{NR}_k - \text{NU}_k)/(t_b - t_a)$). The variation level of video distribution can be defined as $\text{VL}_k = \lambda/\mu$.

4.3. Video Sharing Strategy. Based on the awareness results for user demand and distribution variation level, the video systems make use of video caching and scheduling to optimize distribution and achieve high-efficiency sharing in the process of video dissemination. For instance, if the two nodes n_i and n_j frequently share cached videos with each other, n_i/n_j pushes videos which may be interested by n_j/n_i in terms of results of demand awareness, which increases probability of push success. When their video demand is out of original range and has high variation level, they can adjust cached videos in local buffer to maintain the stability of balance of video supply and demand between them. Moreover, if the nodes have similar demand and are clustered into a node set, the similar demand ensures the videos cached by them can meet the requirement with each other. The nodes in the same set share cached videos and adjust local cached videos according to variation of demand and cached resources in the set, which achieves balance between supply and demand in the whole overlay networks.

On the other hand, the construction of logical links between items in node sets relies on similarity level of demand and does not consider the factor of their geographical location. The adjacent geographical location between video providers and requesters can reduce number of relay nodes in the path of data transmission, which decreases delay of data transmission and probability of data loss and relieves traffic load of networks. The mobility results in dynamic geographical location of nodes, which brings severely negative influence for video delivery. For instance, the one-hop data transmission between requesters and providers may become multihop transmission with the continuous movement. The estimation of mobility similarity between nodes can measure the stability of relative distance between nodes. The construction of node sets (clustering nodes in overlay networks) needs to investigate similarity of demand and mobility of nodes, which achieves local balance between video supply and demand in geographical area.

We investigate the interaction (push and pull) between nodes to estimate similarity of demand. The high-frequency and successful interaction denotes that the nodes can meet the demand for the videos with each other, which increases lookup success rate and reduces lookup delay. The total number of interaction between any two nodes n_i and n_j is defined as

$$F_{ij} = f_i^{\text{push}} + f_j^{\text{push}} + f_i^{\text{pull}} + f_j^{\text{pull}}, \tag{7}$$

where f_i^{push} and f_i^{pull} are the number of videos pushed and requested by n_i for n_j; f_j^{push} and f_j^{pull} are the number of

videos pushed and requested by n_j for n_i. The accepted video push and the successful response for the video request are considered the successful interaction. $\mathrm{IR}_{ij} = F_{ij}^{(s)}/F_{ij}$ denotes the interaction success rate where $F_{ij}^{(s)}$ is the number of successful interaction. The higher the value of IR_{ij} is, the more efficient the video dissemination is. Moreover, the high-frequency interaction and high interaction success rate also reflect the closeness and stability of relationship between nodes. The close and stable relationship between nodes avoids the frequent message exchange to fetch state information with each other such as cached videos, which reduces the energy consumption of mobile nodes caused by huge message overhead.

Because the mobile nodes have low capacity in bandwidth, computation, storage, and energy, they difficultly bear the load for the estimation of mobility similarity with each other in large geographic areas. We investigate the stability of one-hop neighbor relationship between nodes to estimate their mobility similarity. The estimation of mobility similarity in small geographic areas can reduce the load of mobile nodes. If the two nodes have one-hop neighbor relationship at the time t_a, the two nodes are considered as encounter at t_a. The more the number of encounter is, the longer the period time of stable one-hop neighbor relationship is. The stability st of one-hop relationship between nodes can be estimated according to our previous work [31]. Therefore, the relationship level between any two nodes n_i and n_j can be defined as

$$\mathrm{GR}_{ij} = \mathrm{IR}_{ij} \times \left(1 - \frac{\operatorname{arccot}\left(\mathrm{st}_{ij}\right)}{\pi}\right). \tag{8}$$

Because the values of st_{ij} are in the range $(-\infty, +\infty)$, the values of $1 - \operatorname{arccot}(\mathrm{st}_{ij})/\pi$ are in the range $[0, 1]$ and $\mathrm{GR}_{ij} \in [0, 1]$. Any node n_i records the information of nodes which encountered n_i and interacted with n_i and builds a node set GN_i. n_i estimates the relationship level with all items in GN_i according to (8). We cluster the nodes in overlay networks into multiple node subsets by employing the above demand region construction method in Section 4.1. If n_j has the highest relationship level with n_i among GN_i, n_i and n_j form a node set GS_j. If another node n_k also has the highest relationship level with n_j among items in GN_k, n_k joins into GS_j. After the iteration of the above clustering process, the built node subsets continue to remove the noise nodes according to the variation of average values of relationship between nodes in subsets. Finally, the nodes in overlay networks are grouped into multiple node subsets; namely, $\mathrm{NS} = (\mathrm{GS}_1, \mathrm{GS}_2, \ldots, \mathrm{GS}_n)$. The nodes in the same subsets are aware of the deputy videos corresponding to demand regions with each other by periodical exchange of messages.

The nodes preferentially send request messages for the videos to members in the same subsets. Before n_i requests a video v_k, it firstly calculates the similarity between v_k and deputy videos of other nodes in current subset; namely, $\mathrm{LS}_i = (S_{ak}, S_{bk}, \ldots, S_{nk})$, where S_{ak} denotes the similarity value between v_a and v_k. If the similarity between the deputy video v_h and v_k is the largest among all deputy videos and a node n_j in current subset stores v_h, n_i sends the request message to n_j. If n_j also stores v_k in local buffer, n_j directly returns an acknowledge message and transmits data of v_k for n_i. Otherwise, if n_j does not store v_k, it returns a lookup failure message to n_i. n_i uses a threshold Thr_i to filter some items in LS_i. If the similarity values between v_k and items in LS_i is greater than Thr_i, these deputy videos are marked as the selected videos. n_i sends the request messages to the nodes which cache the selected videos except for n_j. If these nodes store v_k, they return the acknowledge messages to n_i and n_i selects a node as the provider. Otherwise, if these nodes do not store v_k, n_i broadcasts the request message in the whole network after receiving the lookup failure messages.

On the other hand, because the nodes can be aware of number of request for v_k and number of nodes which cache v_k in subsets by message exchange, they estimate the variation level VL_k of distribution for v_k during a period time. If VL_k is the minimum value among the variation levels of all videos in current subset, v_k can be preferentially removed in local buffer of nodes after a period time Δt. Otherwise, if VL_k is not the minimum value among the variation levels of all videos in current subset and the nodes carrying v_k do not receive the request message during a period time $(1 + \mathrm{VL}_k)\Delta t$, these carriers can remove v_k from their local buffer.

5. Testing and Test Results Analysis

5.1. Testing Topology and Scenarios. The performance of the proposed solution VDRCO compares with a state-of-the-art community-based video sharing solution AMCV [32]. We use in NS-2 to model and implement the two solutions VDRCO and AMCV. The 400 mobile nodes are deployed in the $1000 \times 1000\,\mathrm{m}^2$ area and have 200 m signal range. The mobile speed of nodes is in the range $[1, 20]$ m/s. The simulation time is set to 400 s. The data transmission protocol is UDP and the wireless routing protocol is DSR. In order to reduce the influence caused by network congestion for the simulation results, the server and the mobile nodes have 20 Mb/s and 10 Mb/s bandwidth, respectively. The transmission rate of video data (the playback rate of mobile nodes) is set to 128 kb/s.

The 200 mobile nodes request videos and join the video system, which are considered as the system members. The number of video files is 30 and the length of each file is 100 s. The 200 playback logs are generated for the system members, which are used to define the requested resources and playback time of system members. After simulation startup, the system members finish the first request for the videos in terms of the playback logs following the Poisson distribution from $t = 0$ s to $t = 100$ s. The requested resources and playback time of each system member in playback logs are randomly assigned. The number of videos requested by nodes in playback logs is 5 and the system members quit the system when they have watched all videos. The number of resources cached in each system member is 3. In VDRCO, the 200 historical playback logs are generated, which are used to estimate demand regions of the 200 system members. The historical logs of encounter and interaction of system

members are generated. The system members are clustered in terms of their historical logs of encounter and interaction before the simulation. α and β are set to 0.5, respectively. The threshold Thr of all system members is set to 0.2. Δt is set to 20 s.

On the other hand, the 400 movement behavior logs are generated for the 400 mobile nodes. In the movement behavior logs, the location, speed, and moving target of mobile nodes are randomly assigned. The mobile nodes move from the initial location. When the mobile nodes arrive at the assigned target, they do not stay and continue to move in terms of the new speed and moving target defined in movement behavior logs. In VDRCO, the members in node subsets are reestimated the relationship between them in terms of the new movement traces and make the selection: leaving current subset and joining new subset or staying current subset.

AMCV is suitable for the performance comparison with VDRCO. AMCV has the flexible deployment strategy to implement video file sharing in wireless mobile networks. In AMCV, the nodes which request the same video chunk are grouped into a community. In the simulation, AMCV clusters the nodes which request the same video file into a community. The number of communities in AMCV is in the range (0, 30]. The maintenance strategy in AMCV also is suitable for the video file sharing. (2) The sharing strategy of video chunks in AMCV is suitable for the video file sharing. We assign the ID for the 30 video files, so the communities in AMCV can build the static connections with each other. We also generated the 5000 historical playback logs which include ID of videos requested by nodes. AMCV can construct the dynamic connections between communities according to the request behaviors of nodes. α and β are also set to 0.5, respectively. PT_b and PT_r are set to 0.2 and 0.12, respectively. The value of UT is defined as 60 s. (3) AMCV and VDRCO make use of community-based MP2P network architecture and estimation for the video request behaviors of nodes to support high-efficiency video sharing. AMCV relies on the association relationship between videos requested by nodes to build the dynamic connections between communities and support the fast video lookup. VDRCO investigates the content similarity level between videos requested by nodes and node mobility to build the node subsets and support the video sharing between nodes in subsets. Before simulation startup, the system members also are clustered in terms of the community construction method.

5.2. Performance Evaluation. The performance comparison between VDRCO and AMCV includes the three aspects average startup delay (ASD), caching utilization rate, and packet loss rate (PLR).

(1) ASD. Let $SD = t_r - t_v$ denote the definition of startup delay. t_r is the time of sending the packet containing request message at the application layer; t_v is the time of receiving first packet containing video data at the application layer.

Figure 3 shows the variation of ASD of the two solutions VDRCO and AMCV with increasing simulation time. $ASD = \sum_{i=1}^{m} SD_i/m$ denotes the average startup delay where m is the number of nodes which have finished startup during a time interval $T = 20$ s. The two curves corresponding to VDRCO and AMCV keep the fall trend from $t = 220$ s to $t = 400$ s after the fast rise from $t = 0$ s to $t = 200$ s. The VDRCO results fast increase from $t = 0$ s to $t = 60$ s, also keep slow rise from $t = 80$ s to $t = 120$ s, and reach the peak value at $t = 220$ s. The AMCV results keep relatively stable rise from $t = 0$ s to $t = 120$ s and reach the peak values at $t = 220$ s. Although the VDRCO results are higher than those of AMCV from $t = 0$ s to $t = 60$ s, the red curve corresponding to VDRCO is lower than that of AMCV from $t = 80$ s to $t = 400$ s and the peak value of VDRCO is less than that of AMCV.

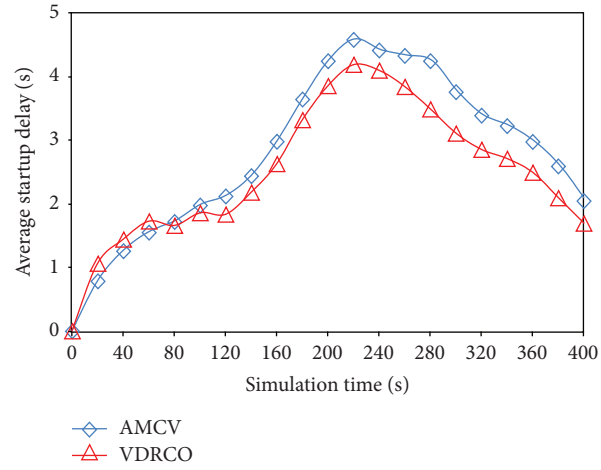


Figure 3: Average startup delay against simulation time.

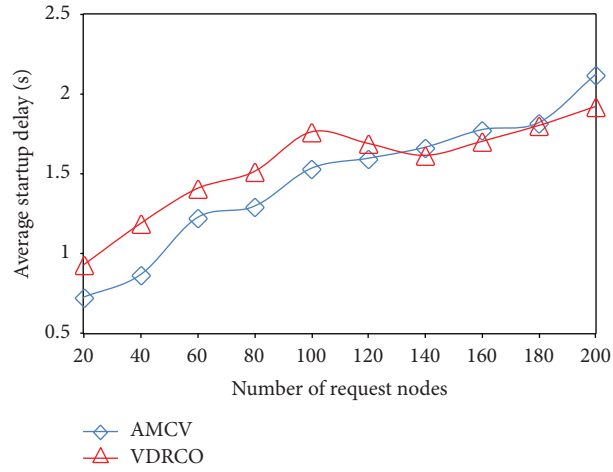


Figure 4: Average startup delay against number of request nodes.

Figure 4 shows the variation of ASD of VDRCO and AMCV with increasing number of request nodes. $ASD = \sum_{i=1}^{m} SD_i/m$ also denotes the average startup delay where m is the number of nodes which have finished startup during a time interval of every 20 nodes requesting videos. The red curve corresponding to VDRCO keeps fast rise with increasing number of request nodes from 0 to 100 and also falls with increasing number of request nodes from 100 to 140. Finally, the red curve continues to keep the rise trend

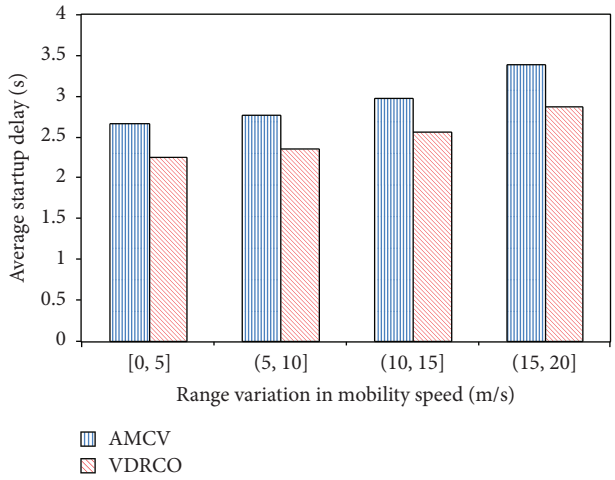


Figure 5: Average startup delay against range variation of mobile speed.

with increasing number of request nodes from 140 to 200. The blue curve corresponding to AMCV always fast increases in the process of the 200 request nodes joining. Although the red curve of VDRCO is higher than that of AMCV with increasing number of request nodes from 0 to 120, the VDRCO results are better than those of AMCV in the increase process of request nodes from 140 to 200.

Figure 5 shows the results of ASD of VDRCO and AMCV in the different range of mobile speed where the mean values of startup delay of all nodes during the whole simulation corresponding to the four range of mobile speed denote ASD. The bars corresponding to VDRCO and AMCV keep the rise trend with the increase in the mobile speed. The red bars of VDRCO are lower than those of AMCV in all range of mobile speed.

In AMCV, the system nodes which request the same video are grouped into a node community and these communities rely on the static and dynamic connections to support the node movement between communities (namely, video lookup). If there are the communities corresponding to all videos in overlay networks, the request nodes can fast find the video providers by making use of the connections between communities to forward the request messages. Therefore, the ASD results of AMCV in the initial simulation process are better than those of VDRCO. However, the system members continuously move between communities with the variation of requested videos, which leads to the dynamic and fragile community structure. In the other words, the interest variation for the video content results in the generation and disappearance of communities, which brings severely negative influence for the lookup success rate. If the request messages cannot be forwarded to the corresponding communities, they only are forwarded to the server. Therefore, the ASD values of AMCV fast increase. On the other hand, AMCV does not consider the mobility of mobile nodes in the process of video delivery, so that the far-end access for the video resources is not avoided. The fast increase in the number of requested videos brings large amount of data traffic, which leads to the network congestion. The ASD values of AMCV keep high level in the process of network congestion from $t = 140$ s to $t = 340$ s, which means that the ASD values of AMCV are subjected to the severe influence of network congestion. When some system members have watched the 5 videos and quit the system, the decrease of traffic relieves the congestion level, so that the ASD values of AMCV can quickly decrease. Moreover, the mobility of mobile nodes may result in the increase of hop number between requesters and providers, so the increase of mobile speed also causes the rise of ASD results of AMCV. The ASD values of AMCV with low mobile speed are less than those with high mobile speed. In VDRCO, the system members are clustered in terms of the frequency of interaction and encounter between them and preferentially request and push the video content in current subsets. Because the system members in subsets continuously estimate the relationship between them according to the current frequency of interaction and encounter, the variation of subset structure brings severely negative influence for the sharing performance. Initially, the new movement traces lead to the change of subset structure; namely, the system members leave current subsets and join other subsets, which increase the ASD values of VDRCO. Once the subset structure keep relatively stable state, the intrasubset video sharing relies on the close geographical distance to obtain low transmission delay of request messages and video data. Further, the system members in the same subsets are aware of the demand regions with each other in the process of video sharing and dynamically regulate the local cached resources according to the demand variation and video dissemination. The adjustment of cached resources based on the awareness of demand variation ensures local balance between supply and demand to promote the intrasubset sharing and reduce lookup delay. Therefore, the ASD values of VDRCO can keep the level of decrease and stability from $t = 60$ s to $t = 120$ s. With the fast increase in the number of requested videos, VDRCO also is subjected to the network congestion from $t = 140$ s to $t = 340$ s, but the ASD values of VDRCO still are less than those of AMCV. The peak value of VDRCO is less than that of AMCV and the ASD decrease rate of VDRCO also is higher than that of AMCV. On the other hand, because the system members in the same subsets keep frequent encounter and have the stable one-hop neighbor relationship, the near-end video delivery reduces the influence caused by network congestion. The ASD values of VDRCO keep the rise trend with the increase in the mobile speed, but the increment of ASD values of VDRCO is less than that of AMCV.

(2) Caching Utilization Rate. Making use of video resources cached in local buffer of system members to serve the requesters denotes the caching utilization. Let N_{CU} and N_{RM} denote the number of caching utilization and all request messages, respectively. The caching utilization rate can be defined as $R_{CU} = N_{CU}/N_{RM}$.

Figure 6 shows the variation of caching utilization rate of VDRCO and AMCV every 40 s. The curves of VDRCO and AMCV have the rise trend during the whole simulation process. The blue curve corresponding to AMCV fast increases from $t = 40$ s to $t = 320$ s and keeps stable rise from $t = 340$ s

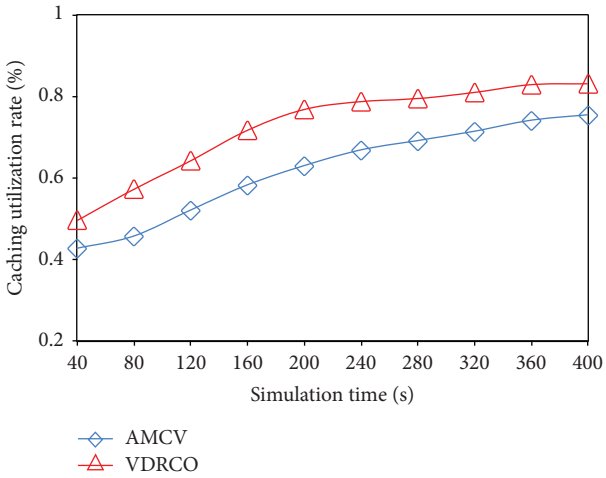


Figure 6: Caching utilization rate against simulation time.

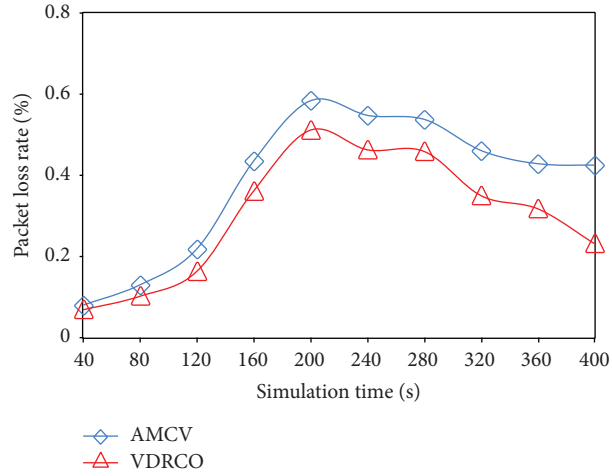


Figure 7: Average packet loss rate against simulation time.

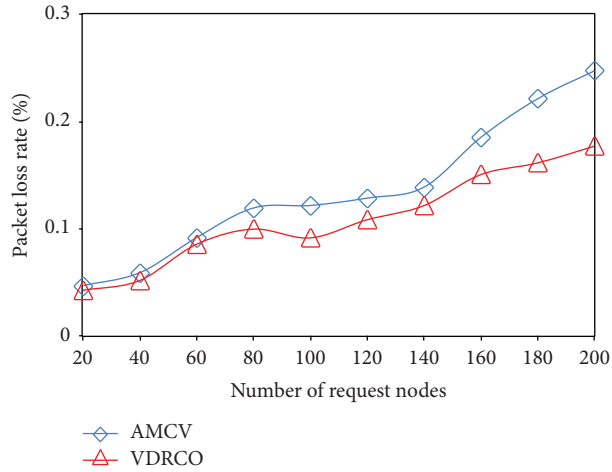


Figure 8: Average packet loss rate against number of request nodes.

to $t = 400$ s. The red curve corresponding to VDRCO has the stable increase from $t = 200$ s to $t = 400$ s after a fast rise from $t = 40$ s to $t = 180$ s. The results of caching utilization rate of VDRCO are higher than those of AMCV.

In AMCV, the system members firstly send the request messages to the broker members in current communities. The broker members make use of the maintained connections with other communities to forward the request messages. If the requested videos are not played by the system members, the communities corresponding to the requested videos also are not built and the request nodes need to fetch the requested video resources from the server. Although the members in communities which have watched the requested videos and do not immediately remove the videos, they continue to request new videos, quit current communities, and join the new communities. The original videos cached by them are not used to serve other requesters. This is as these members have quit the original communities, which results in the link disconnection with the broker members of original communities. On the other hand, the interests of users have random and extensive characters; it is difficult to all videos being watched during the same period time. Therefore, the caching utilization rate of AMCV always keeps the lower level than those of VDRCO. In VDRCO, the system members in the same subsets have high-frequency interaction and similar interests, so they make use of local cached videos to serve with each other. Even if the system members do not fetch the desired video resources from the nodes in current subsets, they require other nodes in subsets forward the request messages or broadcast the request messages to obtain the requested videos from the nodes in overlay networks. Therefore, the results of caching utilization rate of VDRCO are higher than those of AMCV in the whole process of simulation. Moreover, the high caching utilization rate also means that VDRCO reduces the load of the server to obtain higher system scalability than AMCV.

(3) Packet Loss Rate. The ratio between number of lost data at the application layer in the transmission process and number of total number of data sent by video providers and the server denotes the packet loss rate.

Figure 7 shows the PLR variation of VDRCO and AMCV with increasing simulation time where the PLR values during a time interval $T = 40$ s denote the PLR. The two curves corresponding to VDRCO and AMCV have similar variation process of PLR. The results of VDRCO and AMCV decrease from $t = 200$ s to $t = 400$ s after slow rise from t $= 40$ s to $t = 120$ s and fast increase from $t = 120$ s to $t = 200$ s. The blue curve of AMCV is higher than the red curve of VDRCO where AMCV's results have higher increment and peak value than those of VDRCO from $t = 40$ s to $t = 200$ s. The decrement of AMCV's results is lower than that of VDRCO from $t = 240$ s to $t = 400$ s.

Figure 8 shows the PLR variation of VDRCO and AMCV with increasing number of request nodes. Because the 200 system members request the videos following the Poisson distribution from $t = 0$ s to $t = 100$ s, the PLR results of VDRCO and AMCV with the increase in number of request nodes from $t = 0$ s to $t = 100$ s are shown in Figure 8. The two curves of VDRCO and AMCV have the similar variation

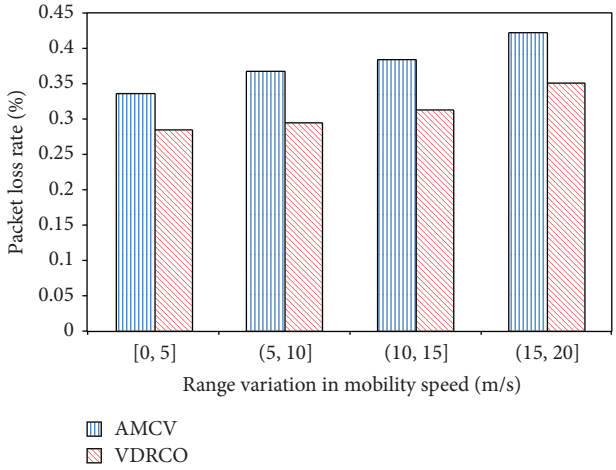


Figure 9: Average packet loss rate against range variation of mobile speed.

trend with variation of number of request nodes from 20 to 200. The red curve corresponding to VDRCO firstly keeps the fast rise with increasing number of request nodes from 20 to 80, experiences a transient fall from 80 to 100, and also continues to quickly increase from 120 to 200. The blue curve corresponding to AMCV firstly has a fast rise with increasing number of request nodes from 20 to 80, experiences a stable process from 100 to 140, and continues to quickly increase from 160 to 200. The blue curve of AMCV is higher than that of VDRCO during increase process in number of request nodes.

Figure 9 shows the PLR variation of VDRCO and AMCV in different range of mobile speed. The bars of VDRCO and AMCV keep the rise trend with increasing mobile speed. The red bars of VDRCO keep the low levels in the low range of mobile speed $[0, 5]$ and $(5, 10]$ and have the fast rise in the high range of mobile speed $(10, 15]$ and $(15, 20]$. The blue bars of AMCV also have low levels in the range of mobile speed $[0, 5]$, keep the slow rise with the variation of mobile speed from $(5, 10]$ and $(10, 15]$, and have the distinct increase in $(15, 20]$. The PLR results of VDRCO are less than those of AMCV in the four range of mobile speed.

AMCV does not consider the negative influence caused by the node mobility for the video delivery; namely, AMCV groups the system members into communities according to the videos requested by them. The forwarding process of request messages only reflects the logical distance between requesters and providers. The neglect for the geographical distance between requesters and providers leads to the increase in the probability of far-end video fetching, which brings the two types of negative influence for the video delivery. (1) The video delivery with geographical long distance increases the load core networks, which aggravates the level of network congestion. When the network congestion occurs, the PLR values of AMCV keep high levels during a long period time from $t = 200$ s to $t = 400$ s, even if some system members continuously quit the video system. (2) There is high risk of packet loss in the process of data transmission with geographical long distance. For instance, if the receiving buffer of a relay node is full, the relay node drops the subsequent data; if there are many relay nodes in the data transmission path, the fast movement of some relay nodes leads to the frequent change of data transmission path which increases the probability of packet loss. Therefore, AMCV has the higher PLR values than VDRCO. VDRCO groups the system members into multiple subsets according to the frequency of interaction and encounter. The system members in the same subsets have stable one-hop neighbor relationship, so they achieve near-end video fetching with high probability. The near-end video fetching not only relieves the load of core network to reduce the level of network congestion but also avoids the risk of packet loss caused by node mobility in the process of data transmission. The PLR values of VDRCO keep relatively low level from $t = 200$ s to $t = 280$ s and fast decrease from $t = 280$ s to $t = 400$ s, which means that the negative influence caused by the network congestion for VDRCO is low relative to AMCV. Moreover, because VDRCO considers the stability of mobility of system members in subsets (namely, stability of one-hop neighbor relationship), the PLR values of VDRCO keep low increase in the range of mobile speed $[0, 5]$, $(5, 10]$ and $(10, 15]$. Although the PLR values of VDRCO are subjected to the distinct negative influence in the high mobile speed $(15, 20]$, VDRCO's PLR results are better than those of AMCV.

6. Conclusion

In this paper, we propose a novel video sharing solution based on Demand-aware Resource Caching Optimization in wireless mobile networks (VDRCO). By construction of video dissemination process model in terms of the epidemic model, VDRCO discusses the main influence factors for the dissemination and distribution of videos. By estimation for user demand range and video distribution variation level, VDRCO groups the nodes into multiple subsets in terms of interaction frequency and similarity of interest and mobility, which optimizes video distribution. VDRCO further designs video sharing strategies to balance local supply and demand and make use of near-end video fetching to improve sharing performance. The simulation results also show how VDRCO has lower startup delay, higher caching utilization rate, and lower packet loss rate than AMCV.

Notations

n_i: A mobile node i
v_i: A video resource i
N: Total number of nodes in overlay networks
$N(S)$: Number of nodes interested in v_i
$N(I)$: Number of nodes which have cached v_i
$N(R)$: Number of nodes which lose interest for v_i and have removed v_i
λ: Increasing rate of nodes which request v_i
μ: Increasing rate of nodes which remove v_i
$S_v(n_i)$: Set of videos watched by n_i
S_{sub_j}: A video subset in $S_v(n_i)$

S_{sk}: Content similarity between v_s and v_k
$\overline{S}(S_{\text{sub}_j})$: Average value of similarity among all items in S_{sub_j}
$l_i(v_j)$: Playback time of n_i for v_j
$f_i(v_j)$: Frequency of n_i pushing v_j
$\overline{I}(S_{\text{sub}_j})$: Average value of interest level of n_i for all items in S_{sub_j}
$\overline{I}(S_v(n_i))$: Average value of interest level of n_i for all items in $S_v(n_i)$
D_i: Set of demand regions of n_i
$\text{TR}_i(v_k)$: Time of downloading v_k from n_i
VL_k: Variation level of distribution of v_k
F_{ij}: Total number of interaction between n_i and n_j
IR_{ij}: Interaction success rate between n_i and n_j
GR_{ij}: Relationship level between n_i and n_j
GN_i: Set of nodes which encountered and interacted with n_i.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant nos. 61501216, 61272015, 61050004, and U1404611, the Basic and Frontier Technology Research Project of Henan Province (142300410303), the Science and Technology Plan Projects (Openness & Cooperation) of Henan Province (152106000048), the Program for Science & Technology Innovation Talents in the University of Henan Province under Grant no. 16HASTIT035, and the Inter-Governmental Science and Technology Cooperation Project under Grant no. 2016YFE0104600.

References

[1] J. Yoon, H. Zhang, S. Banerjee, and S. Rangarajan, "Video multicast with joint resource allocation and adaptive modulation and coding in 4G networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1531–1544, 2014.

[2] M. Hashem Eiza, Q. Ni, and Q. Shi, "Secure and privacy-aware cloud-assisted video reporting service in 5G-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7868–7881, 2016.

[3] C. A. Chan, W. Li, S. Bian et al., "Assessing network energy consumption of mobile applications," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 182–191, 2015.

[4] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang, and G.-M. Muntean, "Performance-aware mobile community-based VoD streaming over vehicular Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1201–1217, 2015.

[5] L. Yang and W. Lou, "A contract-ruled economic model for QoS guarantee in mobile peer-to-peer streaming services," *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1047–1061, 2016.

[6] Y. Sun, Y. Guo, Z. Li et al., "The case for P2P mobile video system over wireless broadband networks: a practical study of challenges for a mobile video provider," *IEEE Network*, vol. 27, no. 2, pp. 22–27, 2013.

[7] J.-L. Kuo, C.-H. Shih, C.-Y. Ho, and Y.-C. Chen, "A cross-layer approach for real-time multimedia streaming on wireless peer-to-peer ad hoc network," *Ad Hoc Networks*, vol. 11, no. 1, pp. 339–354, 2013.

[8] S. Jia, C. Xu, A. V. Vasilakos, J. Guan, H. Zhang, and G.-M. Muntean, "Reliability-oriented ant colony optimization-based mobile peer-to-peer VoD solution in MANETs," *Wireless Networks*, vol. 20, no. 5, pp. 1185–1202, 2014.

[9] A. Fiandrotti, V. Bioglio, M. Grangetto, R. Gaeta, and E. Magli, "Band codes for energy-efficient network coding with application to P2P mobile streaming," *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 521–532, 2014.

[10] C. Xu, J. Zhao, and G. Muntean, "Congestion control design for multipath transport protocols: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2948–2969, 2016.

[11] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean, "CMT-QA: quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2193–2205, 2013.

[12] O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Cooperative download in vehicular environments," *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 663–678, 2012.

[13] J. Zhao, P. Zhang, G. Cao, and C. R. Das, "Cooperative caching in wireless P2P networks: design, implementation, and evaluation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 229–241, 2010.

[14] U. C. Kozat, Ö. Harmanci, S. Kanumuri, M. U. Demircin, and M. R. Civanlar, "Peer assisted video streaming with supply-demand-based cache optimization," *IEEE Transactions on Multimedia*, vol. 11, no. 3, pp. 494–508, 2009.

[15] C. Xu, F. Zhao, J. Guan, H. Zhang, and G.-M. Muntean, "QoE-driven user-centric vod services in urban multihomed P2P-based vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2273–2289, 2013.

[16] B. Tan and L. Massoulié, "Optimal content placement for peer-to-peer video-on-demand systems," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 566–579, 2013.

[17] L. Zhou, Z. Yang, Y. Wen, and J. J. P. C. Rodrigues, "Distributed wireless video scheduling with delayed control information," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 889–901, 2014.

[18] J. Li, J. Sun, Y. Qian, F. Shu, M. Xiao, and W. Xiang, "A commercial video-caching system for small-cell cellular networks using game theory," *IEEE Access*, vol. 4, pp. 7519–7531, 2016.

[19] K. Mokhtarian and H. Jacobsen, "Flexible caching algorithms for video content distribution networks," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–14, 2016.

[20] S. Li, J. Xu, M. van der Schaar, and W. Li, "Trend-aware video caching through online learning," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2503–2516, 2016.

[21] L. Zhou, Z. Yang, J. J. P. C. Rodrigues, and M. Guizani, "Exploring blind online scheduling for mobile cloud multimedia services," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 54–61, 2013.

[22] H. Shen, Y. Lin, and J. Li, "A social-network-aided efficient peer-to-peer live streaming system," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 987–1000, 2015.

[23] H. Shen, Z. Li, and K. Chen, "Social-P2P: an online social network based P2P file sharing system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 10, pp. 2874–2889, 2015.

[24] S. Zhang, Z. Shao, M. Chen, and L. Jiang, "Optimal distributed P2P streaming under node degree bounds," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 717–730, 2014.

[25] H. Shen, Z. Li, Y. Lin, and J. Li, "Socialtube: P2P-assisted video sharing in online social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2428–2440, 2014.

[26] C. Xu, S. Jia, L. Zhong, and G.-M. Muntean, "Socially aware mobile peer-to-peer communications for community multimedia streaming services," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 150–156, 2015.

[27] Q. Wu, Z. Li, G. Tyson, S. Uhlig, M. A. Kaafar, and G. Xie, "Privacy-aware multipath video caching for content-centric networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2219–2230, 2016.

[28] P. Si, H. Yue, Y. Zhang, and Y. Fang, "Spectrum management for proactive video caching in information-centric cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2247–2259, 2016.

[29] A. Dadlani, M. S. Kumar, S. Murugan, and K. Kim, "System dynamics of a refined epidemic model for infection propagation over complex networks," *IEEE Systems Journal*, vol. 10, no. 4, pp. 1316–1325, 2016.

[30] R. Zhang and S. Xiong, "A novel mobile video community discovery scheme using ontology-based semantical interest capture," *Mobile Information Systems*, vol. 2016, Article ID 5279425, 12 pages, 2016.

[31] S. Jia, C. Xu, J. Guan, H. Zhang, and G.-M. Muntean, "A novel cooperative content fetching-based strategy to increase the quality of video delivery to mobile users in wireless networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 370–384, 2014.

[32] C. Xu, S. Jia, L. Zhong, H. Zhang, and G.-M. Muntean, "Ant-inspired mini-community-based solution for video-on-demand services in wireless mobile networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 322–335, 2014.

Hindawi
Mobile Information Systems
Volume 2017, Article ID 8013971, 11 pages
http://dx.doi.org/10.1155/2017/8013971

Research Article

Improving Performance in Dense Wireless Spaces by Controlling Bulk Traffic

Marat Zhanikeev

School of Management, Tokyo University of Science, Fujimi 1-11-2, Chiyoda-ku, Tokyo 102-0071, Japan

Correspondence should be addressed to Marat Zhanikeev; maratishe@gmail.com

Received 7 October 2016; Accepted 13 December 2016; Published 9 January 2017

Academic Editor: Changqiao Xu

The growing number of wireless devices nowadays often results in congestion of wireless channels. In research, this topic is referred to as networking in dense wireless spaces. The literature on the topic shows that the biggest problem is the high number of concurrent sessions to a wireless access point. The obvious solution is to reduce the number of concurrent sessions. This paper proposes a simple method called Bulk-n-Pick which minimizes the number of prolonged concurrent sessions by separating bulk from sync traffic. Aiming at educational applications, under the proposed design, web applications would distribute the main bulk of content once at the beginning of a class and then rely on small messages for real time sync traffic during the class. For realistic performance analysis, this paper first performs real-life experiments with various counts of wireless devices, bulk sizes, and levels of sync intensity. Based on the experiments, this paper shows that the proposed Bulk-n-Pick method outperforms the traditional design even when only two concurrent bulk sessions are allowed. The experiment shows that up to 10 concurrent bulk sessions are feasible in practice. Based on these results, a method for online performance optimization is proposed and validated in a trace-based emulation.

1. Introduction

Wireless standards have recently developed from 802.11g [1] to 802.11n [2] which offers higher throughput and can use multiple antennas (MIMO), making it possible to achieve rates up to 300 Mbps. The development continues with 802.11ac standard with further improvements gradually entering practice. More details on the universe of wireless standards can be found in an excellent survey in [3]. Development of future 802.11 protocols, specifically the 802.11ax standard, can be found in [4] and is close in spirit to this paper in that the new protocol places the problem of wireless interference in the center.

Dense wireless space is a separate topic in recent literature [5]. It is part of the overall guidelines for WLAN deployment [6] in terms of both high density of client devices and dense packing of multiple Access Points (APs). For example, the study in [7] measured interferences in spaces with up to 50 APs and shows how interference depends specifically on *bulk size*, defined in this paper as simply the volume of transmitted information but also hinting at its relative size, among several other parameters (distance from AP, etc.). Several recent research papers come with measurement results that are similar to the ones presented in this paper, specifically in that they support the notion of *abrupt deterioration* in performance when interference exceeds a given threshold [4, 8].

Wireless interference is also taken into account by research on *wireless beacons* [9]. The main advantage of beacons is that they can be used without association between clients and APs, which adds more flexibility in multi-AP spaces. However, when frame rate of beacons is increased and a driver/application hack is used to allow to stream data continuously over multiple beacon frames, interference also becomes a major problem.

This paper focuses on the case of *high density in client devices*. Specifically, an educational class with one AP and up to 20 client devices is used as the main application scenario in this paper. In Japan, there are guidelines that cover LAN deployment in schools but are silent about classes based on WLANs [10]. The problem in WLAN-based classes [11] is interference that causes *reliability problems* in web applications. A study in [12] discusses this problem based on real-life measurements and shows that web applications start having

reliability problems at 10+ and become difficult to manage at 20+ devices, under a level of traffic intensity which is common/standard in educational classes. Poor reliability here refers to HTTP requests which would return empty results due to failed or interrupted wireless sessions. The advice in [12] is to resolve the issue by *designing* web applications accordingly, that is, resending requests on failure and so forth. However, such a *resilience* feature has a side effect, where multiple resends translate into longer completion times for downloads, on average. Moreover, congestion in such cases can be further degraded when multiple clients are trying to recover interrupted connections at the same time. By comparison, the methods proposed in this paper resolve this congestion problem completely.

The core proposal in this paper is the *Bulk-n-Pick* method for multiparty (focus in this paper is on the one-to-many case) data exchange in dense wireless spaces. The name of the method can be read as *bulk-and-pick* and *bulk-then-pick*, both correctly identifying the important capabilities offered by the method. The core idea is to minimize wireless congestion by *scheduling bulk transfers* (or, in a broader sense, controlling them), which is covered by the *bulk* part of the method. The *pick* part refers to the *real time sync traffic* which references portions of the main (predownloaded) bulk. The above reliability problem is mostly removed because the *pick* traffic carries only small sync messages, thus reducing the probability of a high number of overlapping wireless sessions. Provided that the bulk is downloaded in the background (not making users wait), such a design supports a high level of interactivity in web applications running over the local wireless network.

The proposed design makes good practical sense in educational classes. Students can take turns downloading the bulks at the beginning of the class (or separate sessions/portions of class) and then enjoy reliable syncs during the class. Moreover, the optimization problem formulated in this paper generalizes this design to situations in which bulk can be downloaded freely at any time, while server side automatically reacts to and resolves wireless congestion as it occurs.

The specific contributions in this paper are as follows. A new experiment was conducted to fill the gaps found in the earlier dataset in [12]; with the new dataset, it is now possible to model a dense wireless space for both the traditional and the proposed methods. The practical metric in analysis is *completion time* which directly indicates how much time it requires for the entire class (assuming each student has its own wireless device) to complete the download of the main bulk. Results show that the class under the Bulk-n-Pick method can reduce the completion time of 100 Mbyte downloads (for all clients) down to 15 minutes versus over an hour for conventional classes.

Further analysis in this paper introduces the *density* metric which describes the frequency of successful downloads at a given time. Together with the *rate,* a metric describing the true throughput at application level, the optimization problem proposed in this paper can resolve congestion in wireless spaces in real time and with a high degree of flexibility.

This paper has the following structure. Section 2 discusses the research literature which, in addition to the above overview of the *dense wireless* topic itself, covers several adjacent areas of research. Section 3 introduces the core proposal referred to as the *Bulk-n-Pick* method, and Section 4 describes the dataset collected from real measurement experiments conducted in a mock wireless class; the dataset is used as the basis for trace-based analysis further in the paper. Section 5 presents simple analysis derived directly from the dataset. Section 6 formulates the optimization problem of the *decision rules* type, which makes it possible to run a wireless class in the online version of the Bulk-n-Pick mode, that is, when bulk traffic can be exchanged freely, without using a predefined schedule. Section 7 describes the setup for the emulation analysis of the proposed optimization method, results of which are discussed in Section 8. The paper is concluded in Section 9.

2. Related Work

The literature directly discussing the topic of *dense wireless spaces* was reviewed in Section 1. This section discusses the literature which does not deal with this topic directly but is either closely related to it or can be used as part of the method proposed in this paper. Earlier stage of this proposal can be found at [13]; that study conducted early tests of client-server coordination in wireless classrooms which provided further motivation and finally led to this proposal. Note that [13] only conducted preliminary and rather simple practical tests, while this paper proposes a generic optimization framework and analyzes performance over a wide variety of conditions.

Arguably, the closest related topic to dense wireless spaces is that of *wireless beacons*. Beacons refer to a technology that relies on the use of the beacon frame defined for all 802.11* standards. Beacon frames can normally hold only 256 bytes of information, but there are methods and even drivers implemented in hardware/software that can stream larger bulks via multiple consecutive beacon frames; this is referred to as beacon stuffing [9]. Beacons offer the obvious advantage: they work in the *broadcast* mode, while traditional dense wireless spaces are considered as unicast environments. The *broadcast* feature is also often referred to as *nonassociative* use of WiFi Access Points (AP)s; this is because the client device can listen to one or multiple beacons (APs) without having previously established a connection with each beacon. This is also an obvious convenience, since the default use of client devices today does not allow for multiple parallel connections to multiple APs. This paper will revisit the subject of beacons several times in this paper but, otherwise, places the subject out of scope. While this technology can offer a drastic decrease in wireless congestion, it would be difficult to implement it in practice at the current level of availability of custom-made devices. Future publications will repeatedly revisit the subject, following advances in hardware and software.

Unexpectedly, roadside infra, described by the 802.11p standard, has a strong relation to dense wireless spaces. There is current research that has already incorporated this relatively recent standard as part of the future vehicle-to-roadside

as well as vehicle-to-vehicle communications [14, 15]. To deal with the congestion at the roadside, the new methods adopt the same general approach that is found in *cognitive* and *opportunistic* wireless networks [16]. Since this is the part that is related to the topic of dense wireless spaces, we can ignore the vehicle-specific parts of the technology and focus on its cognitive/opportunistic parts. They are revisited further in this section.

4G+ wireless technologies are also closely related to the topic of dense wireless spaces. A good survey of 4G+ technologies can be found in [17]. 4G+ is mostly about the LTE-A suite of technologies. It is truly a suite of technologies that includes Coordinated Multipoint (CoMP), Multiple-Input-Multiple-Output (MIMO), Self-Organized Networks (SON), and others, each defined for a separate application in dense local wireless networks. Note that while 4G+ technology itself is still considered the wide-area access technology, via the concept of the *microcell* (eNodeB, femtocell, etc.), the *local-access* part of the technology also becomes important. The problem of (local) interference is considered to be a major issue in such networks [18], opening new venues for cognitive and opportunistic networking as part of 4G+ [16].

Here, it is important to understand the underlying nature of local connectivity in 4G+ networks. Device-to-Device (D2D) and Machine-to-Machine (M2M) technologies represent the peer-to-peer part of 4G+. However, while the true P2P assumes that the two devices are in direct communication with each other, the D2D/M2M technologies in 4G+ assume that the devices communicate via the microcells which are part of the larger 4G+ infrastructure. While it is not found in current literature, support for the true P2P would fully enable mobile clouds based on 4G+ suite of technologies. Given the CoMP and SON research, some steps are already being made in this direction.

Cognitive and opportunistic components are a major part of 4G+ and specifically the 5G technologies discussed in literature. Since microcells and traditional base stations play independent roles in establishing and maintaining wireless end-to-end paths, energy efficiency and, for terminals and spectrum, efficiency for the entire system are important factors. Discussions in [19] provide good background on energy/spectrum efficiency in the larger context of cognitive wireless networking. The term *spectral efficiency* in 4G+ research is the direct counterpart of the term *wireless congestion control* in this paper.

There is an obvious relation between the proposal in this paper and the cognitive/opportunistic approach. The latter achieves higher efficiency of wireless use by sensing and opportunistically putting to use openings in both time and spectrum. In this paper, we know in advance that dense wireless spaces (like educational classes as the main example in this paper) are already congested, so the proposal in this paper can be presented and is a means that allows for lowering congestion to a manageable level by separating traffic into heavy (bulk) and light (pick, sync) portions and carefully controlling the former. Arguably, the two approaches are two sides of the same coin.

The following literature specifically targets *dense wireless networks* and can therefore be compared to this paper. In [4], the author discusses the 802.11ax protocol, the release of which is planned for 2019. 802.11ax places the problem of *dense wireless spaces*, which it is defined as high density of APs (itself coming from high density of users), in the center of its new features, which include new *beamforming* techniques but, more importantly, a much higher degree of coordination between network and user sides. The proposal in this paper can be viewed as a form of such coordination where, in the absence of widespread adoption of 802.11ax, sources and destinations of wireless traffic have to coordinate in an ad hoc manner.

The work in [8] has a similar proposal but focuses on 4G+ networks, specifically the LTE-A standard. Again, [8] argues that client-network coordination is the only feasible solution to the problem of congestion in dense wireless spaces. Given the centralized nature of the LTE-A protocol, [8] advocates that, under high interference, some small cells can be turned off in order to reduce the level of interference. Note that this solution is very similar to the *Bulk-n-Pick* method in this paper, where interference is controlled by coordinating sessions of bulk transfer. Even with these similarities, the contexts between this paper and [8] are, naturally, drastically different.

At least one recent paper on the subject of *dense wireless spaces* contrasts with this paper as well as the [4, 8] above in that it views the solution in terms of a *multidimensional convex optimization* problem. Numeric simulations in [20] show that performance deteriorates on a fairly slow curve and compares with the curve that represents the case of *perfect beamforming* (i.e., 100% distinction/separation of each wireless client) in terms of the shape. This paper argues that the measurement studies explained within this paper as well as those presented in [4–8] and others offer the overwhelming amount of real-life evidence that supports the discussion in this paper.

With the above description in mind, it is necessary to distinguish this paper from the proposals in [4, 8]. Just like this paper, both [4, 8] propose a form of coordination between clients and network in purely WiFi [4] and 4G+ [8] networks. Both depend on technologies that are yet to be adopted for widespread use. This paper, on the other hand, depends on the conventional WiFi stacks and is therefore feasible under the current level of wireless technology. The unique distinction of this paper is that it measures and models throughput at *application level* which is different from that at lower levels; the traditional analysis target for current literature [4, 8]. This paper shows that this difference is substantial. Specifically, measurement shows that some HTTP requests can fail midway under high interference and have to be repeated. At application level, this translates into abnormally long *completion times* which, at lower protocol levels, may take several failed transmissions until the final successful one.

3. Proposal: The Bulk-n-Pick Method

This section provides more details on the proposed Bulk-n-Pick method and on how it compares with the traditional method.

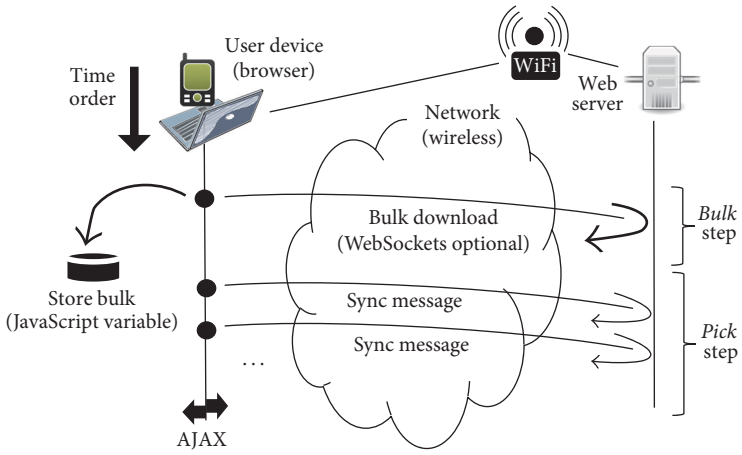


Figure 1: Roles and procedures within one Bulk-n-Pick cycle. Thickness of curves represents relative difference in traffic size.

In the traditional method, data exchange in a dense wireless space happens in real time, disregarding bulk size. Large bulks are often split into smaller pieces (this paper uses 100 kbyte pieces) in order to be able to trace the status of the download, but otherwise the download process happens at the time when it is required by the application logic. This logic also describes downloads based on web sockets [21], which can be disrupted without losing all the previously received data and restarted from the current position. Note that statistically WebSockets and piece-wide downloads should result in similar performance, as far as congestion in dense wireless spaces is concerned.

The *Bulk-n-Pick method* proposes a different logic for web applications. Figure 1 shows one Bulk-n-Pick cycle. The total bulk of content is assumed to be packed into a single binary file and distributed to all the clients prior to using it; normally, the download will happen at the beginning of a Bulk-n-Pick cycle. For simplicity, the analysis further in this paper will assume that the total bulk is 100 Mbytes in size. Once the bulk is distributed to all the clients, the Bulk-n-Pick method assumes that the exchange between the server and multiple clients is reduced to *sync messages* which specify actions. For actions that need to use a portion of the previously downloaded bulk, sync messages would include the respective reference (offset and length in the binary bulk). For example, a sync message may say *show image* and specify the location of the binary data for the image in the bulk. Since most modern browsers support HTML5, the bulk can be handled in the raw binary form (Blob, ByteArray, etc.) without having to assign a meaningful structure to the entire bulk; the latter can simply be stored as a binary string.

On the other hand, *localStorage,* another major function in HTML5, is not necessary as the web applications (written in JavaScript) can keep the bulk in its runtime memory in form of a normal JavaScript variable (array, hash, etc.). Experiments show that runtime memory can easily hold several hundreds of Mbytes of data while *localStorage* on most modern browsers is restricted to 5 Mbytes in the default setting. Moreover, the nature of the bulk in the proposed web application is such that it only needs to exist for a given short-term browser session, after which it can be safely discarded.

Note that Bulk-n-Pick method is multipurpose in nature and can go beyond the educational use described in this paper as well as earlier works in [11, 12]. For example, it can be applied to various over-the-network activities like indexing [22], where the application logic would be changed into short-term sessions, each with its own bulk and sync messages. The proposed method is applicable as long as it is implemented in a dense wireless space where devices need to exchange relatively large bulks at random times. Merging and scheduling bulk transfers at a preparation stage helps alleviate congestion at a later time.

The justification for the *Bulk-n-Pick* method is offered in Section 4 when presenting results of a measurement study within a mock wireless class. Measurement results show that separating bulk traffic from small-size sync messages is a way to control the level of interference in the dense wireless space. Note that this method requires *coordination* between client and server sides, which is a common ground between this paper and recent proposals in [4, 8].

4. The Wireless Class Dataset

The predecessor dataset in [12] used 3 APs and randomized *bulk size* (size of server's replies to HTTP requests from clients). Both parameters make it difficult to model Bulk-n-Pick situations which focus on *separation of bulk from picks* (small syncs) and expect simple wireless classes with only one AP. The new experiment and the resulting dataset explained in this section resolve these problems.

The following setup was used for the new experiment. The population of client devices consisted of 8 tablets and 12 notebooks, with no specific reason for this split except for the physical limits of hardware available at the time. All the devices were tightly packed within a small space (the area of 2 desks) to avoid the effect of distance to the only AP. Note that the dataset in [12] emulated a real class layout in which distance to AP is a nonnegligible variable. Both [12] and experiments in other literature [5]

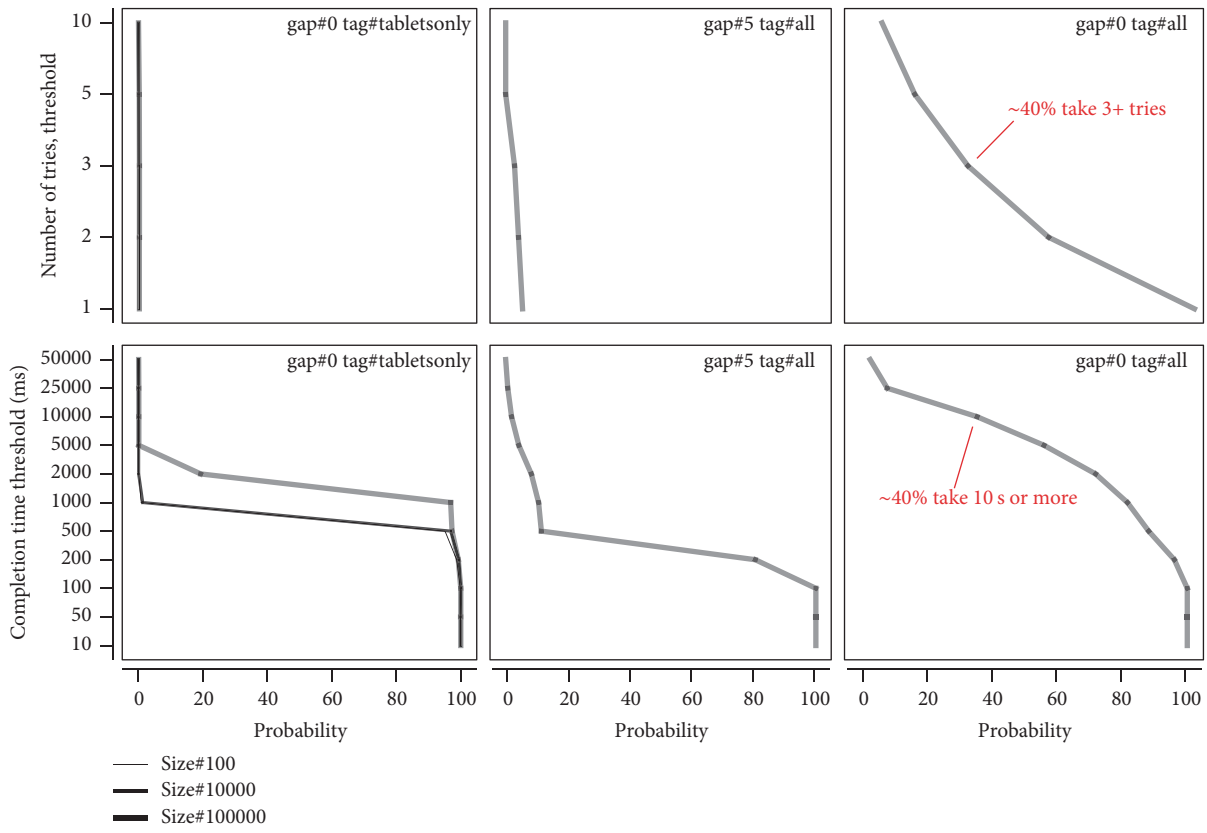


FIGURE 2: Visualization of the dataset used as the basis for analysis in this paper. Request *size* is in *bytes* and request *gap* in *seconds*. Left-to-right columns are for *tablets only with no gap*, *all devices with* 5 s *gap*, and *all devices with no gap*. Rows are for *retry count* and *completion time threshold*, respectively.

points to strong dependence of data rate to distance from AP.

Each run in the new experiment used a fixed *bulk size* for each session, selected randomly from the list of 100 bytes, 10 kbytes, and 100 kbytes. The *gap* parameter was used to define the time interval between requests by each client; *5s* and *0s* gaps were used, representing the intensity of traffic exchange. It was assumed that the 100 Mbyte bulk, the fixed size of the total bulk of content in this paper, was to be downloaded gradually in pieces using multiple 100 kbyte downloads, while 100 byte requests were used for later syncs. The 100*k* pieces are necessary from the practical point of view; they allow the web application to shown to human users the current status of the download. Status can also be shown when using web sockets [21] but that method requires more programming effort without offering any functional improvement. In dense wireless spaces, continuous sockets for each client would increase congestion and would be frequently disrupted. More discussion on pieces versus sockets is offered further in this paper.

To avoid statistical bias, 10 sessions were conducted for each unique combination of the above parameters, and the average was selected as the representative value. Following the same general goals for statistical analysis as were explained in [12], experimental results are processed in such a way as to show probability of a given number of resent web requests and a given total completion time, for each unique configuration. The term *probability* here is a purely statistical concept, easily derived from occurrence frequency of a given event.

Figure 2 visualizes the new dataset. In order to present larger plots, axes are shown only for edge plots but are the same for all plots in the visualization (however, each row has a different vertical axis).

The top row shows how many tries (repeated requests) it took to download a piece (100 bytes, 10 kbytes, and 100 kbytes) while the bottom row shows how much time in total it took to complete each request *(completion time)*, including the multiple tries. The visualization logic is such that each horizontal coordinate (virtual cut) represents *the probability* for a given request to encounter the corresponding value on the vertical scale. The probability is calculated as *occurrence ratio* based on the statistics collected in experiments. The value is cumulative, which means that 0% of requests experience the value above the largest (applies to both tries and completion time) and 100% of requests experience the value above (threshold value included) the smallest in the dataset. All the values in between are produced by lowering the threshold and counting the ratio of requests that produced values above it. Annotations in the rightmost column explain a randomly selected bullet, as an illustration of the visualization logic. Note that the probability curves in Figure 2 are used as models

in analysis further in this paper. The models simply pick a given value from the dataset based on its probability and use it to emulate a given wireless classroom. Note that running such experiments in real classes is too costly and is left to future practitioners of the proposed technology.

The distinct experimental conditions that form columns in Figure 2 are as follows. The 1st column is from the experiment which used only the 8 tablets *(tabletsonly)* and zero gap between requests. This column is also the only one that shows all the three curves for individual sizes: the 10 k and 100 byte curves were found to be the same and therefore are redundant in other columns; note that this point can be considered as an experimental proof of the advantages offered by the Bulk-n-Pick method as 100-byte traffic can only be used for real time sync.

The second and 3rd columns show only the 100 k curves for the entire population of devices (8 tablets plus 12 notebooks are tagged as *all*) and 5 s and 0 s gaps, respectively.

Note that 2nd and 3rd columns are drastically different. Since only the *gap* parameter is different, it can be considered as the main factor in the congestion experienced by all the devices in the 3rd column. In the upper plot, we see that most requests are retried with about 40% of requests taking three of more retries to complete. This effect is also shown in the bottom plot in the greatly deteriorated performance in completion time. This experiment supports the general findings in existing literature [5] which argues that duration of sessions has a major effect in dense wireless spaces.

The above experiment supports the following modeling used as basis for the analysis further in this paper. The 1st column of plots and specifically its 100 k curves in Figure 2 are used directly to model the Bulk-n-Pick model because only a subset of devices participate in bulk transfers; this assumption is valid for up to 8 devices tested in the experiment. Note that 1st column shows that there are no (or very few) requests which required 2+ tries to complete, while distribution of completion time for 100 k bulk is only slightly above that for 10 k and 100-byte bulks.

The 100 k curves in the 3rd column are used to model the traditional method as all the devices share the same wireless channel and use it only for bulk traffic (no separation of bulk from sync).

More details on modeling are provided further in this paper. This dataset is used for two distinct analyses. First a simple analysis is performed directly based on this dataset. Then, the dataset is transformed into the *density-rate* space which is used for additional emulation and analysis based on the optimization problem formulated further in this paper.

5. Trace-Based Analysis

The following setup is used for analysis. First, the dataset and basic modeling method were explained in Section 4 and are used as the basis. Secondly, the analysis objective is simplified by only emulating the initial bulk download and keeping the sync part of web application sessions out of scope. This is a valid simplification as the experiment described in the previous section showed that 100 byte and even 10 kbyte message did not cause congestion even with

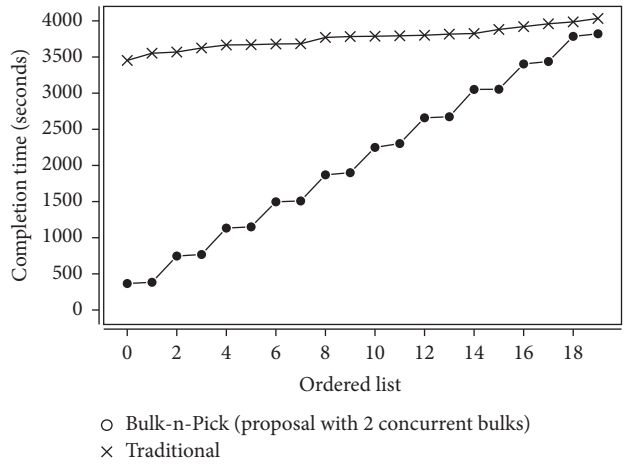


Figure 3: Example emulation run with 2 parallel sessions in the proposed *Bulk-n-Pick* model. Each result is plotted as distribution in increasing order of values, which also happens to be the time order.

20 devices in class. In other words, by separating bulk from pick messages and scheduling bulk downloads, the proposed method resolves the congestion problem and can assume that sync traffic is handled in normal (uncongested) conditions. For the sake of the argument, mild congestion can sometimes occur even for sync traffic, in which case sync messages might fail. However, two failures in sequence are rare, which means that resilience can be easily achieved by repeating the sync message. Note that this is different from the heavy congestion discovered in [12] where large-volume messages would have to be repeated 3–5 times and would take much longer to complete (lower rates due to congestion).

The *completion time* metric of performance refers to the completion of all bulk downloads for all the devices in class, which means that it is the sum of all individual bulk downloads. As was explained above, all the bulks are downloaded in 100 k pieces. The number of parallel sessions for the Bulk-n-Pick model is a variable in emulation but is kept within the range found in the above experiments so that analysis results remain valid.

Figure 3 shows an example plot for 2 concurrent sessions in the Bulk-n-Pick model compared to the traditional model in which all devices compete for the channel. Note that *two concurrent sessions* does not refer to the total number of downloads but is simply the specific scheduling quota which allows for at most two devices to download the bulk at the same time. The schedule is asynchronous and the next download starts soon after the previous one completes, allowing for a small gap required for control messages.

The plot shows time progress of individual completion times for each device while the last (rightmost) value on each curve is the final *completion time* as per the definition above. As can be expected, the Bulk-n-Pick downloads are fast for each pair, hence the stepwise shape of the curve, but it still takes time for the entire class to complete. Under the traditional model, all the devices complete at roughly the same time. The overall outcome is that the proposed Bulk-n-Pick method is better than the traditional method even

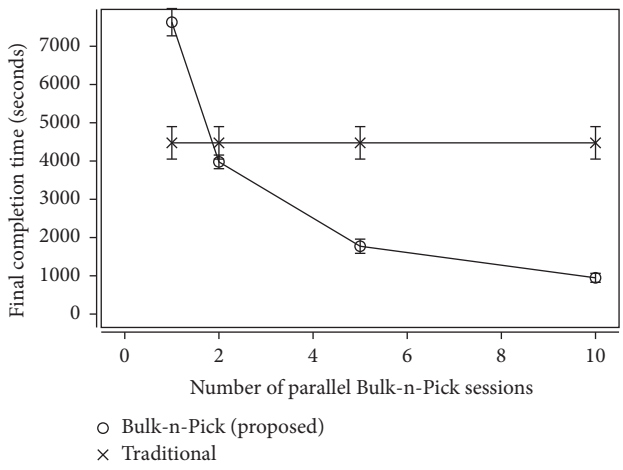


Figure 4: Overall performance plotted for an increasing array of parallel sessions. Bullets for the traditional model are copied in each bullet because its performance does not depend on the number of parallel sessions.

when only two devices are allowed to download the bulk concurrently.

Figure 4 shows the overall performance ranging from 1 to 10 concurrent devices (under Bulk-n-Pick model). The traditional model shows the same performance for each horizontal position as it does not depend on the concurrency parameter. Each bullet has the 3σ bounds marked to visualize variance of data aggregated in a given bullet. The results show that the Bulk-n-Pick method performs poorer than the traditional method only when *concurrency* is set to 1 but starts outperforming it from 2 and more concurrent bulk sessions. With 5 concurrent sessions, it takes about 30 m for the entire class to complete the download. With 10 devices, the total completion time is down to about 15 minutes, a reasonable preparation time for an educational class [11]. The value for 10 devices is extrapolated from the experiments for 8 tablet devices as explained in Section 4 and is therefore very close to experimental conditions.

6. Online Density Optimization

This section opens the second part of the proposed method, specifically, the method that governs the online optimization of performance under unknown (in advance) load. Note that this is a major distinction from the analysis in Section 5 where the core assumption was that bulk would be scheduled in advance. The optimization in this section removes this assumption and instead monitors congestion in real time. The control side is still present; that is, when a high level of congestion is detected, the server side can control it by decreasing the number of devices allowed to run concurrently. This level of control can be achieved with ease with existing software. For example, web server can adjust *sync interval* for all clients, thus forcing more sparse traffic conditions. Direct scheduling (e.g., "come back in 1 minute") command can be scheduled for individual clients when bulk transfer is in question. However, note that this *scheduling* does not mean that the entire schedule is being optimized. Rather, this *scheduling* refers to relative delays issued to individual clients in such a way that the noncongested level of traffic in the wireless space can be maintained. In this respect, although the proposed optimization has an indirect effect on the schedule of requests, the problem below is formulated as *decision rules* rather than a *scheduling problem*.

Let us assume that we have a history of n records of recent performance in the local wireless space. While the records are collected, the window of k most recent records is kept separately for real time monitoring. Let us also denote transmission rate as R, bulk volume as S, and density as D.

The *density* metric D requires special attention. It represents *intensity* of HTTP requests sent by all clients, calculated as the number of requests per unit of time. However, since the core topic of this paper is referred to in academic literature as *dense wireless space*, it was considered that the same name would be more appropriate for the metric as well. Although the *density* metric represents *throughput* (or *goodput* meaning the throughput of successful requests), it does not directly represent the physical level of congestion. Rather, it is measured at server side, which can only observe successful transfers. As was mentioned earlier in this paper, when the wireless space is congested beyond a certain level, connections carrying HTTP requests can fail midway and even result in empty replies returned to the client. The robust application logic deals with it by resending requests multiple times. Since the *density* metric is measured at the server side, it cannot capture all the failed requests and only sees those that have completed successfully. However, using *density* D, *rate* R, and *volume* S together, it is possible to clearly identify congested conditions, as done in the next section. Note that this framework is standard and shared with other literature on the subject, majority of which analyzes *throughput* as a function of *interference*.

The number of concurrent clients is denoted as m, and the constant M_{bulk} stands for the maximum possible number of concurrent clients that can transfer bulk in parallel without causing congestion; this paper will use 8, as was identified by experiments and discussed earlier in this paper. In practice, this number can vary depending on the settings. For example, in multi-AP classes or when using high-end APs, a larger number of concurrent clients can be supported.

For a practically feasible optimization logic, it is important to find the threshold beyond which conditions are considered as *congested*. Generically, let us define α as a *margin* relative to the highest values observed for R, S, and D throughout all the available history (not the recent window k but the entire history of n samples). Then, we can define the margins of *perfect performance* as

$$\begin{aligned} R_{\max} &= \max\left(R_i\right) \quad \forall D_i > \alpha D_{\max},\ i \in n, \\ S_{\max} &= \max\left(S_i\right) \quad \forall D_i > \alpha D_{\max},\ i \in n. \end{aligned} \tag{1}$$

These rules basically define boundaries uncongested operation based on threshold levels of *volume* S and *rate* R. Note that *density* D is used to identify both thresholds. This is a major condition for the success of the proposed method. It is assumed that requests come at the maximum

possible rate, referred to as *zero gap* earlier in this paper. This allows for a simple practical assumption that *density* (as defined above) decreases only in response to changes in traffic or congestion conditions, the two directly related to each other. When presenting the dataset earlier in this paper, it was pointed that performance deterioration is best represented as a step function, that is, drastically worsening shortly after a given threshold.

In the above equations for thresholds, this artifact is translated into the assumption that uncongested performance has to stay within the α margin from the best possible performance, with the latter measured in terms of request density D. The figure in the next section will revisit this discussion with a visual proof to this assumption.

Given these conditions, the online performance optimization method at the current time slot i is then formulated as the decision on whether or not to impose a limitation on the number of concurrent client devices. Assuming that M_{bulk} is the constant representing the maximum number of allowed concurrent devices (8 comes from experiments and is used from this point on), the decision rule is written as

$$m_i = \begin{cases} M_{\text{bulk}}, & \text{when } \text{avg}_{j \in k} R_j < \text{R}_{\max}, \\ \infty & \text{otherwise}, \end{cases} \quad (2)$$

where ∞ stands for *unlimited*, that is, free traffic exchange by all clients at *zero gap*.

Note that the above formulation takes the form of *decision rules* based solely on *performance statistics collected in real time*. The rules apply as long as the distribution of *density* D is similar to the one described above. Here, experimental data collected for this paper is supported by several other academic papers [5–7]. Specifically, the notion of *abrupt deterioration* of performance (a kind of saturation) under high interference is shared with several recent works [4, 8, 20], all containing measurement data similar to the one presented in this paper. The thresholds in the proposed rules can be adjusted for a given practical situation by detecting the step-function point in the distribution curve, as explained above. This supports the claim that the proposed optimization problem is generic in nature. See the next section for details on how the thresholds are identified in practice.

7. Emulation Setup

Figure 5 is the same dataset as was introduced earlier in this paper but this time represented in the space of *density* and *rate*. In this form, the dataset is intended to support the optimization method in the previous section and provide the visual means of its interpretation.

The map in Figure 5 was generated in the following way. First, all measurements for *gap* above zero were discarded; this is because the proposed optimization method depends on the continuous exchange of traffic in the *zero-gap* mode. Then, the dataset was replayed in time using the window k of 3 s worth of measurement samples, with 1 s step. For each window, the selected single measurement result (the center of current window) provides the size S (0.1 k, 10 k, and

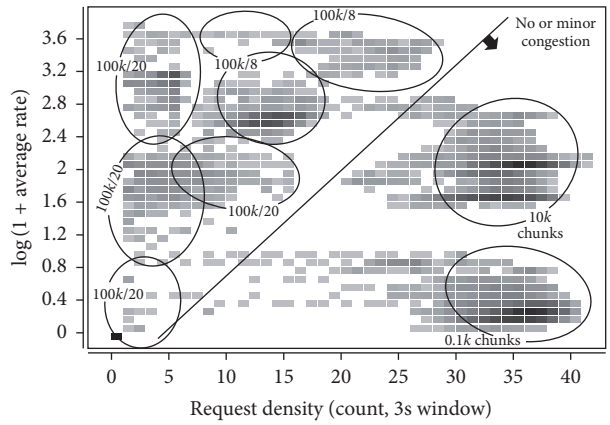


Figure 5: Visualization of the subset of the original dataset used for the emulation analysis in this and the next sections.

100 k as explained above), while the entire window provides the number of requests (i.e., density D) and the average rate R. The average rate is calculated as the average value across all the samples within the window, where each sample would contain its own individual rate (calculated based on the completion time of an individual request).

After manual confirmation using the raw data, the areas in Figure 5 were marked with volumes 0.1*k*, 10*k*, and 100*k* and, on the right side, number of devices (8, 20). There is a diagonal line which separates the noncongested from congested performance, as marked in the figure.

Let us consider how congestion increases, based on the visual proof in Figure 5. Vertical column around $D = 35$ is the area of *perfect performance*, which was found for all measurements for 0.1*k* and 10*k* traffic volumes; that is, the network would not get congested when all 20 client devices would continuously receive *10 k* chunks of data.

The left side of the diagonal line is fully dictated by traffic which uses *100 k* chunks. Here, the areas are split into two groups. Several areas at the upper range of *rate* (note that the vertical scale is in log) come from experiments with 8 concurrent devices. The rest of the areas, including those at the bottom of the range, come from experiments with 20 concurrent devices.

The map is shown as the grid pattern, where each cell is filled with a color that represents the relative occurrence frequency. Most of the areas have darker areas in the middle, which is a natural dispersion pattern which would have a central mode and some scattered samples nearby. Note that the darkest cell in the map is at near absolute zero; this is part of a 100*k/20* area which was observed in extremely congested conditions, when some clients had to send several repeated requests before succeeding. This is reflected in both lower density of requests (near zero) and very low rate.

The spatial distribution of areas for 8 versus 20 devices is interesting. There is a clear visual pattern that states that, in congested conditions, by decreasing the number of concurrent devices from 20 to 8, one can not only increase the density of requests but also drastically improve (0.5–1.5

magnitudes on the log scale) transmission rates (by extension, number of retries) for individual requests.

This, in a nutshell, is the visual way to explain the proposed optimization method. First, congestion is detected when the diagonal line is crossed (α margin is a variable) and *rate* decreases below the maximum achievable rate on the right side of the diagonal. Then, by limiting the number of allowed concurrent devices, the overall performance is improved by increasing both density and rate.

Based on this general approach, emulations for analysis further in this paper were conducted as follows. Measurement samples behind the visual in Figure 5 were selected for replay based on their relative occurrence frequency. Initially, the default state assumes that 20 client devices perform traffic exchange with the server concurrently. This means that samples from measurements with 8 devices would not be selected. Otherwise, the samples are selected randomly, forming a continuous stream of requests which are monitored using the window of $k = 100$ samples.

Random selection frequently leads to conditions, when current window of samples experiences congestion. Then, based on the thresholds and decision rules explained in Section 6, the server may decide to switch to the 8-device mode. In this case, random selection of samples is limited only to the measurement data coming from experiments with 8 devices. When congestion is no longer detected, emulation switches back to the default conditions and resumes random selection of input data from the 20-device subset of the dataset.

Many simulation runs were conducted to cover a range of values for a (0.9, 0.8, 0.7, 0.6, 0.5) and remove the randomness effect. Each simulation run lasted for 5000 randomly selected samples.

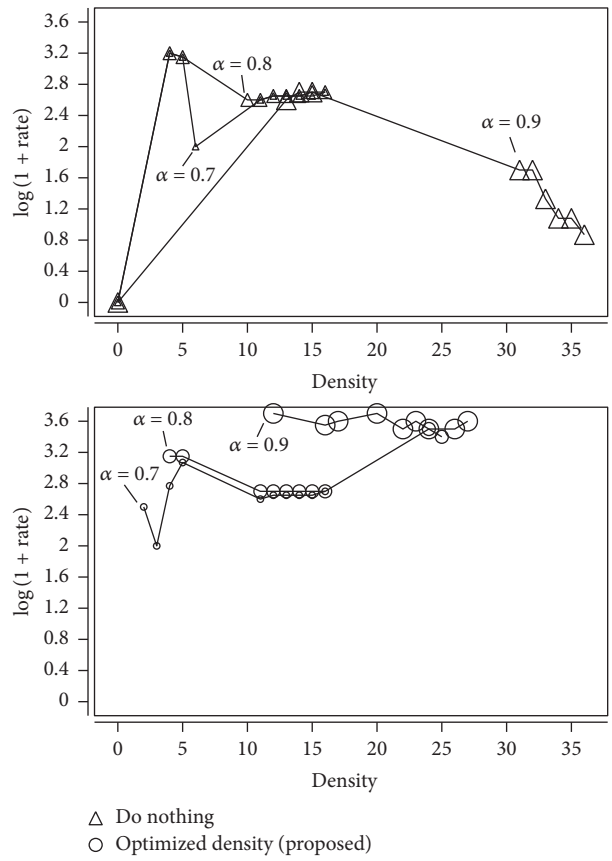


Figure 6: Performance trajectories for the two methods of coping with dense wireless spaces. Each plot has three curves in the decreasing order of bullet size, each for the margin values $\alpha = 0.9$, $\alpha = 0.8$, and $\alpha = 0.7$, respectively.

8. Analysis of Emulation Results

Since this part of the analysis assumes that some actions should be taken in real time, it makes sense to give proper names to the competing methods. The *Do Nothing* method will represent the traditional approach, which does not, in fact, do anything about congestion. The *optimized density* method implements the proposed optimization problem and the visual heuristic explained in Section 7. Unfortunately, as the review of literature above showed, there are no other rivals at the time to be included into the comparison.

Figure 6 shows trajectories for performance under the *Do Nothing* (above) and the proposed (below) methods. The term *trajectory* refers to the curve that represents dynamics in *rate* with variable *density*. Data for the trajectories was filtered in such a way that the samples *beyond* the threshold (i.e., within the margins of the perfect performance) were not included in the plots. In other words, only the data, which represents the emulated *congested* conditions, is shown. Larger bullets represent larger margins α. There curves are plotted for each method, each for its own margin.

Let us deal with the easiest part of the plots first. The near-zero area in the *Do Nothing* method is the legacy from the dataset map in Figure 5. Since it is not within the margins of *perfect performance* and since the traditional method does not optimize the performance in real time, this area is retained. In other words, the traditional technology is expected to keep suffering from congested wireless spaces. Otherwise, performance trajectories for the *Do Nothing* method gather around the same coordinates, with the exception of the $\alpha = 0.9$ case, when it sharply moves into the area of *density* of 30–35. This is purely due to the high (close) margin, which in this case classifies some of the samples very close to perfect performance as those under congestion.

For the proposed method, the two smaller trajectories are also very similar to each other. Both experience relatively high and stable *rate* across a wide range of values for *density*. However, when $\alpha = 0.9$, the trajectory is drastically different in that it shrinks to a relatively focused area with high values in both *rate* and *density*. This is an interesting effect that hints that it is better to react to changes in *density* early on, even with deviations at 10% from the level of *perfect performance*.

By comparison, the *Do Nothing* method is a convex function, which offers poor performance at both ends of the curve and relatively good performance only close to its peak. The proposed method, on the other hand, offers stable rates across a wide range of density conditions. The side feature of the proposed method is its relative stability to the α parameter. Based on this dataset, the value of $\alpha = 0.9$ is

recommended as it represents a monitoring routine which reacts quickly to deviations from perfect performance and thus achieves very high rates and a smaller range of density.

9. Conclusion

This paper proposed the new Bulk-n-Pick method for data transfer in dense wireless spaces. The specific case analyzed in this paper is the *one-to-many* transfers as found in educational classed (lecturer to students) but the method itself is applicable to the more general *many-to-many* case.

The core idea of the proposed method is to separate *bulk* from *sync* traffic and strive to download the bulk via a controlled schedule of concurrent sessions. Experiments in this paper show that 8 concurrent devices cause very little congestion, and the further analysis shows that with 10 concurrent devices a class of 20 clients can complete all the individual downloads in 15 minutes, while the same class would take over an hour in the traditional mode when all the devices would compete for the same channel under extreme congestion.

The analysis in this paper was divided into two parts. The first part performed simple analysis based on the raw experimental results. The second part focused on a general method for optimizing performance in real time. There, an optimization problem was formulated as a combination of *thresholds* and *decision rules* that were triggered when performance exceeded a given *margin* of deviation from the near-perfect performance. The main distinction between the two analyses is that the first one basically proposed a scheduling method, while the second one represents a generic solution to the same problem.

The optimization-centric analysis showed that the dataset could be represented as a *density-to-rate* map, which could serve as a visual guide to the steps that should be undertaken towards optimized performance. The visual also served as a proof for the decision rules formulated earlier. The data behind the visual was used for the trace-based emulation analysis, which showed that only the proposed optimization can maintain a steady average rate within the wireless space regardless of the current density (intensity) of requests. It was also found that the proposed method is advised to react swiftly even to relatively minor deviations from the perfect level of performance as such strategy would result in yet higher rates and a shorter range of observed densities.

This paper was set solely in the one-to-many setting, a good example for which is an educational class. Future publications on the subject will be generalized to many-to-many wireless environments. This will introduce new elements into the problem, some of which were briefly mentioned in this paper.

Competing Interests

The author declares that they have no competing interests.

References

[1] IEEE Standard, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: further higher-speed physical layer extension in the 2.4 GHz band," IEEE Standard 802.11g, supplement to part 11, 2003.

[2] "Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," IEEE Standard 802.11n, 2009.

[3] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, "The IEEE 802.11 universe," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 62–70, 2010.

[4] B. Bellalta, "IEEE 802.11ax: high-efficiency WLANS," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.

[5] Aerohive Networks, "Aerohive design and configuration guide: high-density WiFi networks," Aerohive Whitepaper, 2013.

[6] B. Bing, *Emerging Technologies in Wireless LANs: Theory, Design, and Deployment*, Cambridge Press, Cambridge, UK, 2008.

[7] Y. Xiao and J. Rosdahl, "Throughput and delay limits of IEEE 802.11," *IEEE Communications Letters*, vol. 6, no. 8, pp. 355–357, 2002.

[8] B. Soret, K. Pedersen, N. T. K. Jørgensen, and V. Fernández-López, "Interference coordination for dense wireless networks," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 102–109, 2015.

[9] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman, "Beacon-stuffing: Wi-Fi without associations," in *Proceedings of the 8th IEEE Workshop on Mobile Computing Systems and Applications (HOTMOBILE '07)*, pp. 53–57, IEEE, Tucson, Ariz, USA, February 2007.

[10] Japan Ministry of Internal Affairs and Communications, *School LAN: Layouts and Recommendations for Building Local Area Networks in Schools*, Japan Ministry of Internal Affairs and Communications, 2007.

[11] M. Zhanikeev, T. Tomoto, and H. Watanabe, "Time to play to know (T2P2K): first report on a digital classroom project in grades 4 through 6 of elementary schools," *IEICE Technical Report on Educational Technology (ET)*, vol. 112, no. 374, pp. 37–42, 2013.

[12] M. Zhanikeev, "Experiments on practical WLAN designs for digital classrooms," *IEICE Communications Express*, vol. 2, no. 8, pp. 352–358, 2013.

[13] M. Zhanikeev, "Bulk-n-pick method for one-to-many data transfer in dense wireless spaces," in *Proceedings of the IEEE International Conference on Networking and Network Applications (NaNA '16)*, pp. 90–94, IEEE, Hakodate, Japan, July 2016.

[14] R. K. Schmidt, T. Leinmüller, E. Schoch, F. Kargl, and G. Schäfer, "Exploration of adaptive beaconing for efficient intervehicle safety communication," *IEEE Network*, vol. 24, no. 1, pp. 14–19, 2010.

[15] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular Ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.

[16] S. Glisic, *Advanced Wireless Networks: Cognitive, Cooperative and Opportunistic 4G Technology*, John Wiley & Sons, 2009.

[17] E. Dahlman, S. Parkvall, and J. Skold, *4G LTE LTE-Advanced for Mobile Broadband*, Academic Press, 2014.

[18] M. Rumney, *LTE and the Evolution to 4G Wireless, Design and Measurement Challenges*, John Wiley & Sons, 2013.

[19] C. X. Mavromoustakis, A. Bourdena, G. Mastorakis, E. Pallis, and G. Kormentzas, "An energy-aware scheme for efficient spectrum utilization in a 5G mobile cognitive radio network architecture," *Telecommunication Systems*, vol. 59, no. 1, pp. 63–75, 2015.

[20] Y. Shi, J. Zhang, B. O'Donoghue, and K. B. Letaief, "Large-scale convex optimization for dense wireless cooperative networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 18, pp. 4729–4743, 2015.

[21] M. Zhanikeev, "Experiments with application throughput in a browser with full HTML5 support," *IEICE Communications Express*, vol. 2, no. 5, pp. 167–172, 2013.

[22] M. Zhanikeev, "A new practical design for browsable over-the-network indexing," in *Proceedings of the International Conference on Information Science, Electronics and Electrical Engineering (ISEEE '14)*, pp. 1686–1690, IEEE, Sapporo, Japan, April 2014.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 5352165, 12 pages
http://dx.doi.org/10.1155/2016/5352165

Research Article

A Novel Contribution-Aware Neighbor-Assist Video Delivery Solution over Mobile Content-Centric Networks

Lujie Zhong,[1] Shijie Jia,[2,3] and Mu Wang[4]

[1]*Information Engineering College, Capital Normal University, Beijing 100048, China*
[2]*Central Plains Economic Zone Wisdom Tourism Cooperative Innovation Center, Luoyang Normal University, Luoyang, Henan 471934, China*
[3]*Academy of Information Technology, Luoyang Normal University, Luoyang, Henan 471934, China*
[4]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100086, China*

Correspondence should be addressed to Shijie Jia; shjjia@gmail.com

Received 13 September 2016; Accepted 10 November 2016

Academic Editor: George Ghinea

In this paper, we propose a novel contribution-aware neighbor-assist video delivery solution over mobile content-centric network (CNVD). CNVD allows the nodes to build and maintain neighbor relationship with other nodes to share cached video resources and achieve unicast-based video lookup. CNVD constructs an estimation model of contribution of neighbor nodes by investigation of lookup delay, number of cached videos, lookup success rate, and geographical distance. CNVD designs the estimation methods of interest level and lookup capacity of nodes for video content, which enables the nodes to decide whether to build neighbor relationship with other nodes. A maintenance method of neighbor relationship between nodes is proposed, which enables the nodes to update valid time of neighbor relationship in terms of contribution of neighbor nodes and decide whether to remove neighbor relationship in terms of the current valid time of neighbor relationship. Further, CNVD designs a contribution-based video lookup algorithm, which reduces lookup delay and improves lookup success rate. Extensive tests show how CNVD achieves much better performance results in comparison with other state-of-the-art solutions.

1. Introduction

The prominent advancement of bandwidth and networking technologies in wireless mobile networks greatly promotes development of mobile Internet applications, such as social, e-commerce, and multimedia [1]. The video streaming services rely on provision of rich visual content and convenience of access using mobile devices to become the most popular applications [2–5] (e.g., the mobile video users in China have been four hundred and forty million in 2016). The video services focus on user quality of experience (QoE) [6]. Rich visual content and convenient access for video services in wireless mobile networks attract large amount of users. The large-scale video access consumes massive network bandwidth and results in strong competition between users for the network bandwidth, which brings severely negative influence for startup delay of users. The dynamic and complex condition of video delivery in wireless heterogeneous networks increases the probability of video data loss, so that the distorted frames reduce user QoE. Obviously, the increase in upload bandwidth supply and video delivery performance is very important for improving user QoE. By the investigation of node mobility to address the problems of dynamic path of data transmission, the P2P/MP2P-based video systems make use of the resources of online video users to increase the efficiency and capacity of bandwidth supply in wireless mobile networks [7–10]. However, the limited resources of bandwidth, computation, storage, and energy of mobile devices result in the limited available resources in overlay networks, which difficultly meets the demand for increasing huge video traffic.

Content-centric networking (CCN), a brand-new framework, focuses content based on newly designed protocol stack instead of host and employs the all-to-cache method

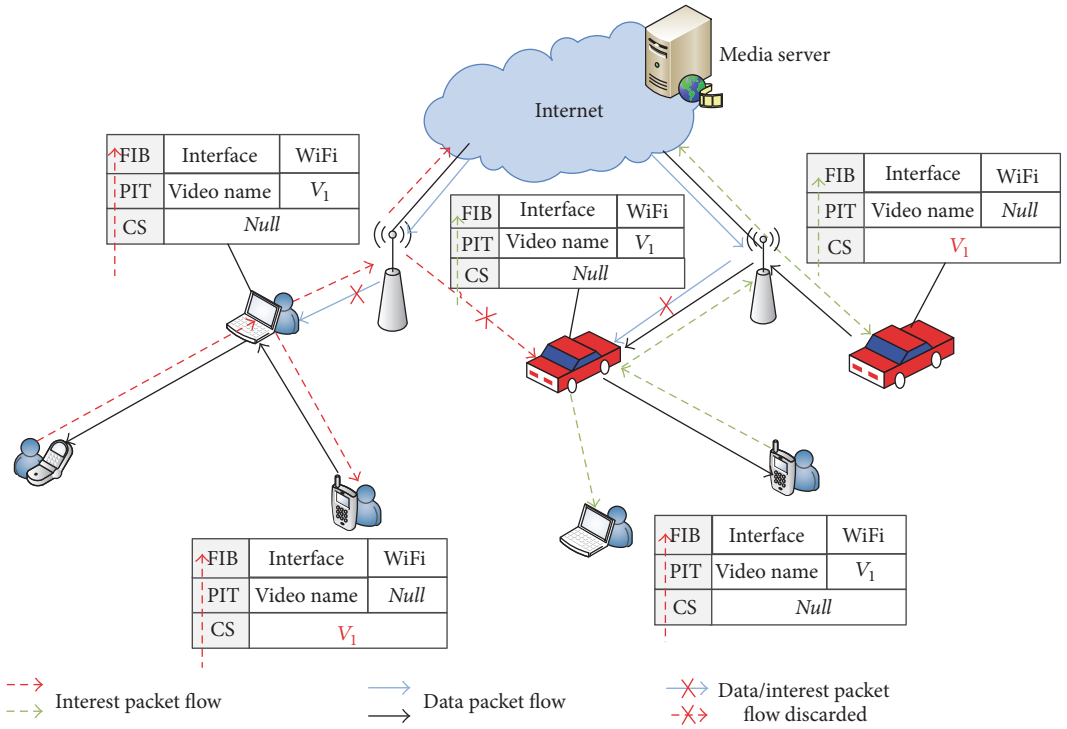


Figure 1: Multimedia streaming services in mobile content-centric networks.

to achieve nearby content fetching [11], which reduces delay of content lookup and delivery. Figure 1 illustrates content delivery process in mobile content-centric networks (MCCNs). Each mobile node in MCCNs maintains the data structures: content store (CS), pending interest table (PIT), and forwarding information base (FIB) [12]. The mobile nodes send interest packets to their neighbor nodes in order to obtain the desired video data. If the neighbor nodes have cached the requested video data in CS, the data is directly returned to the request users to the request nodes. Otherwise, if the neighbor nodes do not cache the requested video data locally, the mobile nodes record the incoming interface of interest packets in PIT and broadcast to all neighbors; if the mobile nodes have recorded the information of interest packets, they discard the received interest packets. If the neighbor nodes firstly receive the interest packets, they forward the interest packets to the next-hop nodes; if the mobile nodes receive the same interest packets, they discard the received packets. When the interest packets are forwarded to the content providers which cache the requested video data, the providers return the video data along the reverse searching path. At the moment, the intermediate nodes (relay nodes) in the data reverse path can cache the returned data in order to have nearby access to the data in the future.

The all-to-cache, the traditional caching method in MCCNs, consumes large number of storage resources of nodes to support nearby content fetching in order to offload traffic in underlying network, which leads to huge waste of storage resources. The low capacities of storage, computation, and energy of mobile devices difficultly support the massive consumption. On the other hand, because the nodes are the content carriers, the content movement in geographic area with movement of carriers brings huge negative influence for content caching effectiveness. Moreover, the traditional broadcasting-based content lookup approach in MCCNs also wastes huge network bandwidth. The network congestion caused by large-scale request results in long startup delay. In order to address the problems that existed in traditional methods, some unicast-based content delivery solutions in MCCNs make use of the collected and maintained content information to achieve precise and economic content lookup. However, the nodes need to maintain large number of content carriers in order to fast search content providers. The larger the scale of maintained information is, the higher the probability of fast and successful lookup is. The maintenance of large-scale content information not only brings redundancy link between nodes, but also leads to the overload of mobile nodes due to low capacities. Otherwise, the small-scale maintenance reduces lookup hit rate, which also results in long startup delay. In order to ensure high QoE of users and reduce maintenance cost of nodes, a key issue is how to design the economic and efficient content delivery solutions.

In this paper, we propose *a novel contribution-aware neighbor-assist video delivery solution over mobile content-centric networks (CNVD)*. CNVD allows the mobile nodes to build and maintain neighbor relationship with other mobile nodes to share cached videos and support unicast-based video lookup. In order to achieve economic maintenance for the video content in MCCNs to support efficient lookup and transmission of videos, CNVD designs an estimation method of contribution level of neighbor nodes by investigation for lookup success rate, number of cached videos,

forwarding delay of interest packets and video data, and geographical distance of selected neighbor nodes in lookup and transmission paths. A maintenance strategy of neighbor nodes composed of construction and removal of neighbor relationship in terms of contribution levels of neighbor nodes is proposed, which reduces maintenance cost of neighbor nodes. Further, CNVD designs a contribution-based video lookup algorithm, which decreases the delay of lookup and transmission of videos by selection of appropriate next-hop nodes in terms of contribution levels of neighbor nodes. Simulation results show how CNVD achieves much better performance results in comparison with other state-of-the-art solutions.

2. Related Work

The content lookup in MCCNs [12, 13] mainly employs the broadcasting-based method. When the content requesters want to search content, they broadcast interest packets in the whole network. The mobile nodes which receive the interest packets of requesters become the relay nodes in the lookup path of content. If the relay nodes cannot store the requested content, they continue to broadcast the received interest packets to their neighbor nodes. The broadcasting-based method consumes large amount of network bandwidth, which increases risk of network congestion and wastes battery energy of mobile nodes. In order to reduce the cost of content lookup, Rehman et al. proposed a timer-based interest forwarding (REMIF) method [14]. After the relay nodes in the lookup path of content receive the interest packets, they monitor the channel state for a specific time. If the duplicated interest packets are received within the time window, the relay nodes discard the duplicated interest packets in terms of the recorded information in PIT. Otherwise, the relay nodes forward interest packets to other neighbor nodes. However, the time window results in the extra delivery latency, which is unsuitable for video streaming services. Yu et al. proposed a neighborhood-aware interest forwarding method (NAIF) [15], which forwards interest packets according to statistics information of interest forwarding. The relay nodes in the lookup path of content decide to broadcast or drop the received interest packets according to the forwarding rate. By the adaptive adjustment of forwarding rate, NAIF promotes efficiency of interest forwarding and reduces consumption of network bandwidth. However, NAIF does not consider selection of appropriate next-hop neighbor nodes. If the interest packets always are forwarded to the nodes which cache the requested content with high probability in the process of content routing, the interest packets quickly are satisfied, which reduces the content lookup delay. Otherwise, the interest packets still need to experience long-term forwarding, which increases the lookup delay and reduces user QoE. The performance of interest forwarding easily is influenced by the selection of neighbor nodes.

In order to further promote efficiency of interest forwarding, several content lookup methods employ unicast-based interest forwarding. Bian et al. proposed a geo-based forwarding strategy for NDN in VANETs environment [16]. The location position information of data source is added into data name in the process of naming data. By periodical exchange of location information among one-hop neighbor nodes, if the nodes receive interest packets, they select the forwarders with close geographical distance, which reduces delay of interest forwarding. Qian et al. proposed a probability-based adaptive forwarding scheme (PAF) based on the ant colony optimization (ACO) [17]. PAF makes use of the ACO to calculate the probability of next-hop relay nodes according to the performance measurement such as delay. The better the results of performance measurement are, the higher the selection probability of next-hop nodes is. The probability-based selection of next-hop nodes improves the delivery quality and balances the load of interest forwarding among nodes. The centrality-based data dissemination method is proposed in [18]. By investigation for social contact patterns and interests of mobile users to measure the social centrality of nodes, the nodes which receive interest packets makes use of unicast or multicast to forward the interest packets to next-hop nodes in terms of centrality values. The selected next-hop nodes have the high centrality value and have high probability of encountering nodes, which also achieves high performance of lookup and dissemination of data. In fact, the centrality-based method relies on a priori knowledge of centrality values of mobile nodes. The mobile nodes need to continuously calculate and exchange centrality values of other encountered nodes, which consumes large amount of resources of mobile nodes. Ahmed et al. proposed a unicast-based forwarder selection (RUFS) in order to address the problems caused by interest broadcast storm [19]. RUFS requires each vehicle node to share statistic information of satisfied interests with neighbor nodes. All neighbor nodes maintain local neighbors satisfied list (NSL) to store cached content information, which efficiently promotes content lookup performance by selection of optimal interest forwarder. However, the mobile nodes have the low capacities in the resources of energy, computation, storage, and bandwidth; they do not bear high-cost maintenance for large-scale items in NSL. Otherwise, the small-scale maintenance of cached content information increases the risk of content lookup failure. Obviously, the unicast-based methods promote content lookup performance and reduce the consumption of bandwidth in the process of interest forwarding; the key issue is how to balance content-aware cost and lookup performance for the content delivery in MCCNs.

3. Model of Contribution Capacity

3.1. Estimation of Contribution Capacity of Neighbor Nodes. For convenience, some notations which are used in current and following sections are defined in Notations. The maintenance of neighbor nodes supports forwarding messages to destination nodes, so the selection of relay nodes is very important for the forwarding performance. The efficient selection of neighbor nodes in MCCNs not only speeds up convergence process of video lookup, but also promotes lookup hit rate. CNVD requires each node to maintain serval logical neighbor nodes in order to support video lookup

(these neighbor nodes are not always one-hop neighbors in geographic area). For instance, n_i makes use of the mapping relationship between cached content and nodes to build and maintain a neighbor node list $\mathrm{NL}_i = (n_a, n_b, \ldots, n_k)$. When n_i needs to watch video content, it sends an interest packet to an item n_h in NL_i. n_h shoulders the responsibility of lookup of requested content. n_h checks local cached videos and searches video information carried by neighbor nodes of n_h. If n_h and n_h's neighbor nodes do not have the requested video, n_h selects a neighbor node as next-hop node to forward the interest packet. The neighbor node selection is very important for the video lookup performance. If the nodes cache large amount of videos in local buffer, the request for videos may be responded by them with high probability. If the nodes maintain large amount of neighbor nodes, they are aware of the information of videos cached in neighbor nodes, which increases the probability of successful response for the requested videos. In order to obtain high success rate and performance of content lookup, CNVD constructs an estimation model of contribution capacity of neighbor nodes to estimate the performance of content lookup of neighbor nodes.

The delay-sensitive video services have high requirement for the content delivery performance (lookup and transmission of content). The startup delay of video requesters is an important evaluation parameter for the quality of service (QoS) of video systems. The startup delay includes lookup and transmission delay of video data, which is the time span from sending interest packets to receiving video data of supporting video playback. The requesters always hope that the startup delay meets the requirement of the own QoE. Let d_{ui} denote the upper bound of startup delay of n_i in terms of the requirement of n_i's QoE (d_{ui} is the maximum value of sustainable startup delay of n_i). n_h is a neighbor node of n_i. When n_h receives video request of n_i and helps n_i search video provider, d_{rh} is defined as the generated startup delay in the process of lookup and transmission of video data. d_{rh} is the real startup delay of n_i. If n_h enables d_{rh} to be less than d_{ui}, n_i considers that n_h successfully helps n_i search the requested video. n_i continues to maintain the logical link between n_i and n_h. Otherwise, if n_h receives the interest packet of n_i and cannot make d_{rh} in $[0, d_{ui}]$, n_i considers that the lookup task assigned by n_i for n_h is failure. n_i needs to consider whether to remove the logical link between n_i and n_h. n_i considers the logical link between n_i and n_h is valueless for the lookup failure, so that the maintenance of logical link between them wastes valuable resources of bandwidth, computation, and energy of n_i. Therefore, we make use of d_{ui} and d_{rh} to calculate contribution value of a lookup task assigned by n_i for n_h according to the following equation:

$$C_h\left(\mathrm{VC}_j\right) = \begin{cases} w_h r_h \left(1 - \dfrac{d_{rh}}{d_{ui}}\right), & d_{rh} < d_{ui}, \\ 0, & d_{rh} \geq d_{ui}, \end{cases} \quad (1)$$

where VC_j is the category of requested video v_a and $v_a \in \mathrm{VC}_j$, which points out the limited range for video lookup. $C_h(\mathrm{VC}_j)$ is the contribution value of n_h for the assigned lookup task corresponding to VC_j. $d_{rh} < d_{ui}$ denotes that current video lookup is successful for n_h; $1 - d_{rh}/d_{ui}$ is the distance ratio between d_{rh} and d_{ui} where $1 - d_{rh}/d_{ui} \in (0, 1)$. The less the value of d_{rh} is, the higher the contribution value of n_h is. Otherwise, $d_{rh} \geq d_{ui}$ denotes that current video lookup is failure for n_h. The contribution value of n_h is 0. For instance, n_i sends an interest packet to the neighbor node n_h for a video content v_a at the time t_s. When n_i receives the requested data at the time t_r, it calculates the real startup delay by $d_{rh} = t_r - t_s$ and estimates contribution value of n_h according to (1). In MCCNs, there are the two main influencing factors for the startup delay of video requesters: number of relay nodes in lookup path and delivery capacity of interest packets and video data of these relay nodes. The number of relay nodes determines the forwarding frequency of interest packets and video data. The more the number of relay nodes is, the longer the forwarding delay of interest packets and video data is, which results in long startup delay. The selection of relay nodes (neighbor nodes) is a key factor for reducing startup delay. If the neighbor nodes cache large amount of videos related to the requested videos, the demand of requesters is satisfied with high probability. Otherwise, if the neighbor nodes cache small amount of videos and these cached videos are not related to the requested videos, the interest packets still continue to be forwarded, which increases the lookup delay. On the other hand, because the relay nodes in the paths of lookup and transmission of video data are the logical neighbor nodes with each other, the geographical distance and communication quality between relay nodes are neglected. The long geographical distance between relay nodes may increase the delay of lookup and transmission of video data; the low communication quality between relay nodes (e.g., network congestion) not only results in the increase in delay of lookup and transmission of video data, but also causes the loss of interest packets and video data. In order to investigate the influence of supply and delivery capacity of neighbor nodes for the performance of lookup and delivery of video data, we add the two impact factors $w_h \in [0, 1]$ and $r_h \in [0, 1]$ for $1 - d_{hr}/d_{hl}$. w_h and r_h denote the weight values generated by capacity of video supply and delivery of neighbor nodes, respectively.

3.2. Estimation of Video Supply Capacity of Neighbor Nodes. The video supply capacity of neighbor nodes is very important for reducing the length of paths of video lookup and transmission in MCCNs. We firstly investigate lookup success rate (LSR) and number of cached videos (NCVs) to calculate the values of w_h of neighbor nodes. The LSR reflects the resource supply capacity of neighbor nodes for the video requesters. If a neighbor node n_h always meets the demand of requesters for the requested videos, n_h has high video supply capacity to help n_i fast search the requested videos, which reduces the lookup delay. There is a close relation between LSR and NCV. The number of cached videos maintained by n_h includes local videos of n_h and videos cached in neighbor nodes of n_h. The more the number of maintained videos is, the higher the probability of request satisfied in the maintained videos is. The investigation of LSR reflects the video supply capacity of selected neighbor nodes. On the other hand, the NCV of n_h and n_h's neighbor nodes is the dynamic variation

with the interest for the video content. The low capacity of storage, computation, bandwidth, and energy makes the mobile nodes only cache small amount of videos. In order to watch new videos, the mobile nodes need to remove the old videos and cache the new videos in local buffer with limited size. The investigation of NCV reflects the influence level of variation of video distribution maintained by n_h for the video supply capacity of selected neighbor nodes. Therefore, we make use of LSR and NCV of all relay nodes in lookup paths to estimate supply capacity of selected neighbor nodes. The LSR of any neighbor node n_h of n_i for a video category VC_j can be obtained according to the following equation:

$$R_{ih}\left(\mathrm{VC}_j\right) = \frac{RN_s}{RN_t}, \tag{2}$$

where $R_{ih}(\mathrm{VC}_j)$ is n_h's LSR for n_i's request of videos in VC_j; RN_s and RN_t denote successful and total lookup number, respectively. CNVD employs a provider-feedback-based lookup performance estimation method. Because the providers are the terminal point of lookup paths, they can collect the information of relay nodes in paths, such as LSR and number of cached videos. After the neighbor node n_h of requester n_i receives the interest packet, n_h adds number of cached videos corresponding to VC_j and LSR of next-hop node n_j selected by n_h into the interest packet. n_j also adds the above information into the interest packet. After the iteration, the provider n_p adds the collected information of relay nodes and the number of cached videos into the returned data. After n_i receives the data, it is aware of the supply capacity of relay nodes and provider. n_i obtains two datasets $RS_i = (R_{ih}, R_{ij}, \ldots, R_{ip})$ and $NS_i = (NR_{ih}, NR_{ij}, \ldots, NR_{ip})$. R_{ij} and NR_{ij} denote LSR and NCV of relay nodes and provider in the paths of lookup and transmission of video data, respectively. Because the grey relational coefficient (GRC) can measure the relation level of variation process of two curves [20], we make use of the GRC to estimate the relational level between RS_i and NS_i. The items in RS_i and NS_i are normalized according to the following equation:

$$x^*(\mathrm{att}) = \frac{x(\mathrm{att}) - \mathrm{lower}_{\mathrm{att}}}{\mathrm{upper}_{\mathrm{att}} - \mathrm{lower}_{\mathrm{att}}}, \quad x^*(\mathrm{att}) \in [0,1], \tag{3}$$

where att denotes the attribution of estimation parameter such as LSR and NCV; $x(\mathrm{att})$ is the value of estimation parameter; $\mathrm{upper}_{\mathrm{att}}$ and $\mathrm{lower}_{\mathrm{att}}$ are the upper and lower bound of values of estimation parameter (minimum and maximum values). The relational level between RS_i and NS_i can be calculated according to the following equation:

$$\mathrm{GRC}(R, NR) = \frac{1}{\sum \theta_{\mathrm{att}} \left| x^*(\mathrm{att}) - 1 \right| + 1}, \quad \mathrm{GRC} \in [0,1], \tag{4}$$

where θ_{att} is the weight value of $x^*(\mathrm{att})$; $\mathrm{GRC}(R, NR)$ is the relational value between RS_i and NS_i, which denotes the relational level of two curves composed of items in RS_i and NS_i. The higher the $\mathrm{GRC}(R, NR)$ is, the more similar the interests of relay nodes for cached videos in the lookup path are. For instance, the LSR and NCV of relay nodes keep the rise/fall trend, which means that the variation process of LSR and NCV meets the condition of rise/fall of LSR with increase/decrease of NCV. The relay nodes also have similar interests for the content related to v_a. If the LSR of relay nodes keeps fall/rise trend with increase/decrease of NCV, the relay nodes do not have the common interests for the content in VC_j, which may increase the risk of lookup failure. We use $\mathrm{GRC}(R, NR)$ to assign the value of w_h; namely, $w_h = \mathrm{GRC}(R, NR)$.

3.3. Estimation of Video Delivery Capacity of Neighbor Nodes. Except for the supply capacity of neighbor nodes, we also investigate the video delivery capacity of relay nodes in lookup and transmission paths to estimate relational level between geographical distance and forwarding delay between them. The forwarding delay reflects the delivery performance of interest packets and video data in lookup and transmission paths. The geographical distance and communication quality between relay nodes are the main influencing factors for the forwarding delay. The neighbor nodes form a logical lookup path, so the long geographical distance between logical relay nodes enables the interest packets or video data to experience many geographical relay nodes, which increases forwarding delay. Generally, the larger the geographical distance between logical relay nodes is, the longer the forwarding delay of interest packets and video data is [21–23]. Moreover, the low communication quality between relay nodes also increases the forwarding delay of interest packets and video data or causes loss of interest packets and video data, even if the two relay nodes have close geographical distance. The investigation of forwarding delay between logical relay nodes denotes the video delivery capacity of neighbor nodes. On the other hand, the geographical distance reflects the stability of mobility of relay nodes. The mobility makes the geographical distance continuously change, which influences the delay of lookup and transmission of video data. The investigation of geographical distance between logical relay nodes denotes the influence level of node mobility for the video delivery capacity of neighbor nodes. Similarly, all nodes (the requester n_i, logical relay nodes, and the provider n_p) in the lookup and transmission paths add their timestamp of forwarding interest packets and geographical location into the interest packets. n_p collects the information of timestamp and geographical location of all nodes. Let $GS_i = ((x_i, y_i), (x_h, y_h), \ldots, (x_p, y_p))$ and $TS_i = (t_i, t_h, \ldots, t_p)$ denote the set of timestamp and geographical location, respectively, where x_i and y_i denote the abscissa and vertical coordinates of n_i and t_i is the timestamp of forwarding interest packet of n_i. n_p also adds GS_i and TS_i into the returned data. When n_i receives the requested data from n_p, it calculates the forwarding delay and geographical distance between nodes according to the following equation:

$$d_{ih} = t_h - t_i, \quad \mathrm{gd}_{ih} = \sqrt{\left(x_i - x_h\right)^2 + \left(y_i - y_h\right)^2}, \tag{5}$$

where d_{ih} denotes the forwarding delay of interest packets between n_i and n_h; gd_{ih} denotes the geographical distance between n_i and n_h. GS_i and TS_i are converted to GD_i and TD_i;

namely, $GD_i = (gd_{ih}, gd_{hk}, \ldots, gd_{hk})$ and $TD_i = (d_{ih}, d_{hk}, \ldots, d_{lp})$. n_i makes use of (3) to normalize items in GD_i and TD_i and further makes use of (4) to calculate relational level between forwarding delay and geographical distance between GD_i and TD_i. We use $GRC(d, gd)$ to assign the value of r_h; namely, $r_h = GRC(d, gd)$. n_i calculates and records the contribution of n_h for the lookup of v_a. The forwarding delay and geographical distance between relay nodes keep the same rise/fall trend, which means that the variation process of forwarding delay and geographical distance meets the condition of rise/fall of forwarding delay with increase/decrease of geographical distance. There is the good communication quality between relay nodes. If the forwarding delay between relay nodes keeps fall/rise trend with increase/decrease of geographical distance, there is the bad communication quality between relay nodes such as network congestion, which brings high risk of packet loss and long delay. The requesters should avoid the reusage of current lookup path for the subsequent video lookup.

4. CNVD Detailed Design

4.1. Construction of Neighbor Relationship. The nodes maintain the logical connections with their neighbor nodes in order to fast fetch desired video content. For instance, if the neighbor nodes of a node n_h store large number of video resources, the videos requested by n_h may be cached by the neighbor nodes. Obviously, the nodes tend to preferentially construct the neighbor relationship with the nodes cached large-scale resources. However, if a node n_i caches large amount of video resources and n_h is uninterested in the cached videos of n_i, n_i also does not meet the demand of n_h. Otherwise, if n_h is interested in large amount of videos cached by n_i, n_h preferentially constructs the neighbor relationship with n_i. The large amount of cached videos and the similar interests enable the request of n_h to be met by n_i with high probability. n_h sends an invitation message to n_i where the message includes the information of videos cached in n_h and n_h's neighbor nodes; namely, $VI_h = (VL_a, VL_b, \ldots, VL_k)$. VL_a is a video list and includes the ID of videos corresponding to the video category VC_a. Because n_h is aware of the information of videos stored in neighbor nodes by message exchange, the videos stored in neighbor nodes also are considered as the available resources of n_h. If n_i has the high interested degree (interest level) for videos cached in n_h, the video request of n_i is met by n_h with high probability. n_i estimates the interest level for the videos cached in n_h according to the following equation:

$$I_{ih} = \frac{\sum_{c=a}^{s} |VL_c - UV_c|}{\sum_{c=a}^{s} |VL_c|}, \quad I_{ih} \in [0, 1], \tag{6}$$

where s is the number of video categories interested by n_i; UV_c is the set of videos uninterested by n_i corresponding to the video category VC_c. $|VL_c - UV_c|$ returns the number of items in the difference set between VL_c and UV_c. $\sum_{c=a}^{s} |VL_c - UV_c|$ returns the number of interested videos of n_i. $I_{ih} \geq TH_i$ denotes that n_i is interested in the videos cached by n_h where TH_i is the threshold of interest level of n_i. n_i accepts the invitation of n_h and constructs the neighbor relationship with n_h. Otherwise, if $I_{ih} < TH_i$ denotes that n_i is uninterested in the videos cached by n_h, n_i rejects the invitation of n_h.

The scale of video resources cached in n_h and n_h's neighbor nodes is limited. If n_i's interest for the video content is out of range of resources cached in n_h and n_h's neighbor nodes (the requested videos are not located in the set of videos cached in n_h and n_h's neighbor nodes), n_i still needs to search the requested videos with the help of neighbor nodes. Except for the supply capacity of local resources, the resource lookup capacity of nodes also is an important factor for the construction of neighbor relationship. For instance, n_h and n_h's neighbor nodes only cache small amount of videos, but n_h's neighbor nodes have strong lookup capacity (high contribution values for lookup tasks assigned by n_h). n_i also may consider the construction of neighbor relationship with n_h. Even if n_h and n_h's neighbor nodes do not provide one-hop and two-hop video access for n_i, they may successfully search the videos requested by n_i and enable the startup delay meet the demand of n_i's QoE. n_i also accepts the invitation of n_h and constructs the neighbor relationship with n_h. n_h sends an invitation message to n_i where the message includes the contribution values of all neighbor nodes of n_h; namely, $S_{con} = (CL_a, CL_b, \ldots, CL_k)$. CL_a is the list of contribution values of neighbor nodes corresponding to the video category VC_a. n_i estimates the lookup capacity of n_h for the video category VC_a according to the following equation:

$$LP_{ah} = \frac{\sum_{e=1}^{g} \overline{C_e}}{g}, \qquad \overline{C_x} = \frac{\sum_{b=1}^{v} C_b}{v}, \tag{7}$$

where $\overline{C_x}$ is the average contribution value of a neighbor node n_x of n_h corresponding to VC_a; v is the number of video lookup tasks assigned by n_h for the requested videos in VC_a; g is the number of nodes which accept the lookup tasks assigned by n_h for the videos in VC_a. If n_i is interested in multiple video categories, it estimates the video lookup capacity of n_h for multiple video categories according to the following equation:

$$\overline{LP}_h = \frac{\sum_{e=1}^{k} \omega_e \times \overline{LP_{eh}}}{k}, \quad \omega_e \in (0, 1), \tag{8}$$

where ω_e is the weight value of the video category VC_e; namely, there are different weigh values between video categories; k is the number of video categories interested by n_i. Let $\overline{LP}_i$ be the video lookup capacity of n_i by making use of (8), where $\overline{LP}_i$ and $\overline{LP}_h$ are corresponding to the same video categories. If $\overline{LP}_h > \overline{LP}_i$, n_i constructs the neighbor relationship with n_h. CNVD allows the nodes construct the neighbor relationship with other nodes according to the interest level for the cached videos and video lookup capacity with each other.

4.2. Removal of Neighbor Relationship. The nodes which build logical links not only need to consume bandwidth to maintain

the state with each other, but also are responsible for the video lookup. If a neighbor node n_h always requests n_i to help search desired videos and does not provide satisfied lookup performance for n_i, the logical link between n_i and n_h is insignificant and redundant for n_i. In order to save the resources of bandwidth, computation, and energy, CNVD allows n_i to remove the redundant logical link with n_h (not all links always are maintained). In order to construct or keep the neighbor relationship with the nodes with strong lookup capacity and large amount of interested videos, the nodes need to store more videos and maintain the links with more nodes. The link removal is the punishment of node selfishness, but the video sharing performance in the whole network is not reduced by link removal.

The contribution of neighbor nodes is the important metric for the maintenance of logical links. The cached videos and maintained links of n_i and neighbor nodes provide video supply service with each other. The contribution values reflect the video supply capacities of neighbor nodes. We define a time T_{ih} to denote valid period of link between n_i and n_h. n_i or n_h removes the link between them when the time span of link is greater than T_{ih}. For instance, if the contribution value of n_h for n_i is 0 (n_i does not require n_h search videos or all video lookup of n_h is fail), n_i removes the link between n_h and n_i when the valid time of their link is greater than T_{ih}. If n_h successfully searches the video requested by n_i and has a corresponding contribution $C_h(\mathrm{VC}_j)$, the valid time of link between them is defined as

$$\mathrm{VT}_{ih} = \mathrm{VT}^*_{ih} + T_{ih} \times C_h\left(\mathrm{VC}_j\right), \quad C_h\left(VC_j\right) \in [0,1], \quad (9)$$

where VT^*_{ih} is the remaining time of link; VT_{ih} is current valid time of link (neighbor relationship). Obviously, the higher the contribution of neighbor nodes is, the longer the valid time of link is. The maintenance method of neighbor relationship based on the valid time of link reduces the consumption of resources of bandwidth, computation, and energy and promotes the video sharing.

4.3. Neighbor Nodes Discovery and Video Lookup. Initially, all nodes do not construct the neighbor relationship with other nodes. The nodes send invitation messages to their one-hop nodes. Because the nodes do not have neighbor nodes, the video lookup capacity of inviters is 0. The one-hop nodes decide whether to accept the invitation by (6). Once the two nodes construct neighbor relationship, they maintain the logical link by periodical exchange of messages containing information of cached videos. If the link is overtime at a node side, the node removes current link.

On the other hand, if a node n_i wants to watch a video $v_a \in \mathrm{VC}_j$, n_i checks the local cached videos because it is aware of videos cached in all neighbor nodes by message exchange. If the requested video is cached in a neighbor node n_h, n_i makes use of the valid link to send the interest packet to n_h. After n_h checks local cached videos, n_h directly returns the video data to n_i. If the requested video is not cached in all neighbor nodes of n_i, n_i requests the help of neighbor nodes to search the requested video data. n_i selects a neighbor node n_a as the next-hop node where n_a has the highest contribution value corresponding to VC_j among all neighbor nodes. After n_a receives the interest packet, it check its videos and the videos cached in neighbor nodes. If n_a and n_a's neighbor nodes do not have the requested video, n_a also selects a neighbor node as the next-hop node according to the contribution value of all neighbor nodes corresponding to VC_j. In order to avoid the loop circuit in the lookup process, the requesters and relay nodes add the information of their neighbor nodes into the interest packet. After iteration of the above process, if the content provider is found, the provider returns the video data to n_i and n_i updates the contribution of n_a. Otherwise, if the interest packet is overtime, n_i broadcasts the interest packet. The pseudocode of the process of video lookup is detailed in Algorithm 1. The number of relay nodes in lookup path and the number of their neighbor nodes determine the complexity of Algorithm 1. Therefore, the complexity of Algorithm 1 is $O(n)$.

After the nodes successfully fetch the requested videos with the help of neighbor nodes (the lookup delay meets the requirement of QoE of requesters), they record the information of providers. If the nodes have sufficient resources of computation, bandwidth, and energy to maintain a new link, they send the invitation messages to the providers along the lookup path. The providers decide whether to accept the invitation according to the capacities of video supply and lookup of inviters. If the providers reject the invitation, the inviters remove the information of providers. The rejection of providers drives the inviters continuously to find more appropriate neighbor nodes.

5. Testing and Test Results Analysis

5.1. Testing Topology and Scenarios. We compare the performance of CNVD with RUFS which is a state-of-the-art unicast-based CCN forwarding strategy [19]. CNVD and RUFS were modeled and implemented in Network Simulator 3 (NS-3). The more the number of mobile nodes is, the more the scale of requested video data is, which brings severe network congestion. In order to reduce the influence of network congestion for the experiment effect, 200 mobile nodes are considered as vehicular nodes and are distributed in a $2000 \times 2000\ \mathrm{m}^2$ square area which has five horizontal and five vertical streets with two lanes. The bandwidth of mobile nodes is 10 Mb/s. The mobility results in the variation of geographical distance between mobile nodes to influence the delay of lookup and transmission of video data. The random movement behaviors cannot reflect the real movement environment, so the movement behaviors of mobile nodes follow the Manhattan mobility model [24]. In order to simulate the real urban environment, the movement speed varies from 15 m/s to 20 m/s. 33 road side units (RSUs) are evenly deployed in the square area and provide initial video data for mobile nodes. The mobile nodes and RSU equip IEEE 802.11p WAVE network interface to support data transmission. The maximum transmission unit (MTU) of network is set to 1500 B. The size of content store (CS) in each node is 10000 MTU, which is almost equal to 5% of the total size of video content. The signal range of mobile nodes is set to 250 m. The simulation time is 1000 s.

```
(1)  flag = 0; j = 0;
(2)  /* NL is neighbor set of node; v_a ∈ VC_j is video content requested by requester
     n_i; RS is set of relay nodes in lookup path.*/
(3)  for (k = 0; k < |NL_i|; k++)
(4)    if NL_i[k] includes v_a
(5)      n_i sends interest packet to NL_i[k]; flag = 1; break;
(6)      flag = 1;
(7)      break;
(8)    end if
(9)  end for
(10) if (flag = 0)
(11)   n_i sends interest to neighbor RS[j] with the most contribution for VC_j;
(12)   while (flag = 1 or j > TTL)
(13)     if RS[j]'s neighbor NL[h] includes v_a
(14)       RS[j] forwards interest to NL[h];
(15)       flag = 1;
(16)     else RS[j] forwards interest to RS[j + 1] with the most contribution for
         VC_j; j++;
(17)     end if
(18)   end while
(19) end if
(20) if (flag = 0)
(21)   N_i broadcasts interest;
(22) end if
```

Algorithm 1: Process of video lookup.

We group 100 video files into 20 video categories where the length of each file is 100 s. Before the simulation, we created 200 playback logs to define playback behaviors of 200 mobile nodes. The 200 mobile nodes watch diverse video content according to the created 200 playback logs where the watched time is random. When the nodes have watched a video, they request new videos according to the playback logs. 200 mobile nodes join the video system following the Poisson distribution. In CNVD, the valid time T of link between nodes is set to 20 s. The threshold of interest level of all nodes is set to 0.5 and the number of neighbor nodes maintained by each node is in the range $[1, 10]$. θ_{att} is set to 0.5. The values of ω corresponding to all video categories are 0.5. The upper bound of startup delay of all request nodes is set to 5 s.

5.2. Performance Evaluation. The performance of CNVD is compared with RUFS in terms of lookup latency, cache hit ratio, playback freeze frequency, and maintain overhead, respectively.

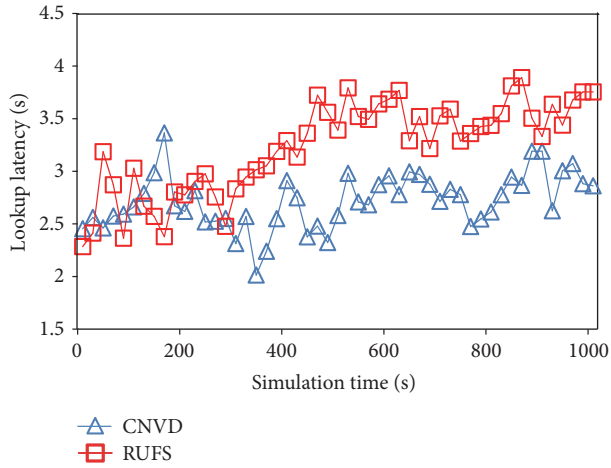


Figure 2: Lookup latency versus simulation time.

5.2.1. Lookup Latency. The lookup latency is defined as the time span between the time when the requester sends the interest packet and the time when the provider receives the interest packet.

We use mean values of all lookup latency during every 20 s as the average lookup latency. Figures 2 and 3 show the performance of lookup latency of CNVD and RUFS in terms of the variation in simulation time and number of mobile nodes. As Figure 2 shows, the blue curve corresponding to CNVD first experiences fast increase before t = 200 s and decreases to the lowest point (2 s) at t = 400 s. The lookup latency of CNVD fluctuates around 3 s and keeps relatively stable after t = 600 s. The red curve corresponding to RUFS has a severe fluctuation before t = 200 s, slightly increases from t = 200 s to t = 600 s, and maintains slight fluctuation after t = 600 s. Obviously, the lookup latency of CNVD is lower than that of RUFS.

We use mean values of all lookup latency during every 20 nodes as the average lookup latency. Figure 3 shows the variation of lookup latency of two solutions CNVD and RUFS when the number of nodes increases from 20 to 200. The black bars corresponding to CNVD have slight rise from 20

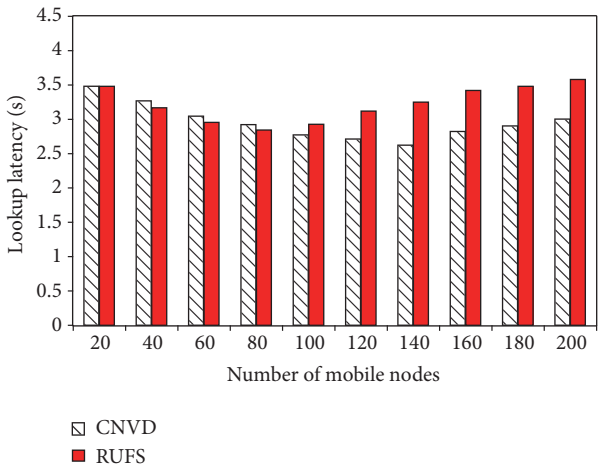


Figure 3: Lookup latency versus number of mobile nodes.

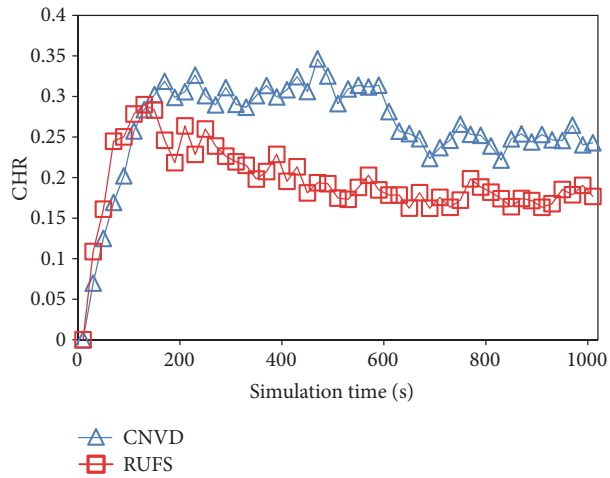


Figure 4: CHR versus simulation times.

to 140 after fast fall from 140 to 200 where the lookup latency of CNVD is between 2.5 s and 3 s. The red bars corresponding to the RUFS also have increase trend after fast decrease. The range of lookup latency of RUFS is [2.8, 3.5]. The lookup latency of RUFS is higher than those of CNVD.

Initially, the small amount of mobile nodes requests and caches video content. The RSUs provide the initial video resources for the request nodes, so that the lookup latency keeps low levels. With increasing number of request nodes, the increase in the scale of requested videos brings huge video traffic, which leads to the network congestion and causes the rise of lookup latency. When the nodes have watched all videos, they quit the system. The decreasing traffic relieves the congestion level and enables the lookup latency fall. In CNVD, the nodes build the neighbor relationship based on the capacity of video supply and lookup and maintain the neighbor relationship in terms of forwarding delay, geographical distance, lookup success rate, and number of cached videos. The neighbor nodes with strong capacity of video supply and delivery reduce the amount of interest forwarding and decrease lookup latency. The link removal mechanism drives the nodes to continuously find more appropriate neighbor nodes with similar interests and strong capacity of video supply and delivery, which promotes the lookup performance of neighbor nodes. Therefore, CNVD can enable the lookup latency to keep low level with slight jitter. In RUFS, the selection of relay nodes relies on the forwarding capacity of neighbor nodes. However, the forwarding capacity of neighbor nodes only investigates the opportunistic encounter with other nodes, which cannot guarantee the validation of content location information. Additionally, RUFS neglects mobility of mobile nodes, which also brings negative influence for video delivery performance; namely, the dynamic network topology caused by node mobility may result in frequent change of paths of packet forwarding, which increases the lookup latency. Although CNVD does not consider the mobility of mobile nodes, the nodes investigate the variation trend of geographical distance and transmission latency in the process of maintenance of neighbor nodes. Therefore, the lookup latency of CNVD is lower than that of RUFS.

5.2.2. Cache Hit Ratio (CHR). The cache hit ratio is defined as

$$\mathrm{CHR} = \frac{HN}{RN}, \quad \mathrm{CHR} \in [0, 1], \tag{10}$$

where RN is the total number of interest packets received by nodes; HN is the number of requests satisfied by videos cached in nodes. The high CHR denotes the video providers can efficiently supply the cached videos to reduce the amount of unsuccessful lookup. Figures 4 and 5 illustrate the results of CHR of CNVD and RUFS in terms of the variation in simulation times and number of mobile nodes, respectively.

We use mean values of all CHR during every 20 s as the average CHR. As Figure 4 shows, the two curves experience similar rise trend from $t = 0$ s to $t = 200$ s. The blue curve corresponding to CNVD keeps rise from $t = 200$ s to $t = 400$ s, reaches the peak 0.36 at $t = 460$ s, and has a fall with slight fluctuation from $t = 400$ s to $t = 1000$ s. The red curve corresponding to RUFS has the decrease trend with slight fluctuation from $t = 100$ s to $t = 1000$ s. The blue curve of CNVD is higher than that of RUFS.

We use mean values of all CHR during every 20 nodes as the average CHR. As Figure 5 shows, the black bars corresponding to CNVD have a fast rise trend with increasing number of mobile nodes where the increase rate is gradual from 100 to 200. The red bars corresponding to RUFS also show the rise trend, but the CHR results of RUFS always keep continuous jitter. The CHR of CNVD is almost 20% higher than that of RUFS.

Initially, the small number of mobile nodes joins the system and requests video content. The sent interest packets also are less. The RSUs provide the initial video data, so that the CHR values keep fast rise. The video content is fast disseminated to the whole network with the help of content caching. With the increase in the number of request members, the number of requested videos also fast increases.

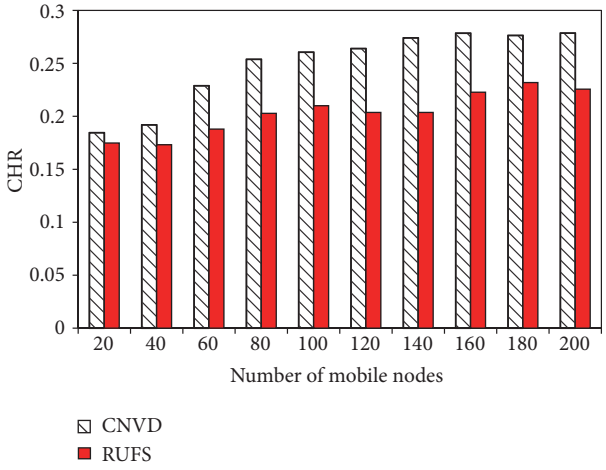


Figure 5: CHR versus number of mobile nodes.

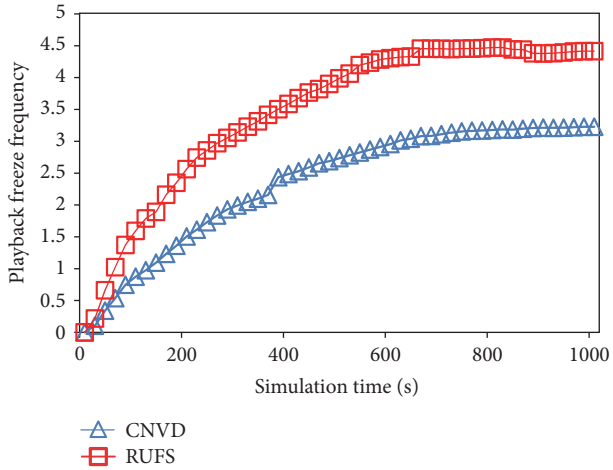


Figure 6: Playback freeze frequency versus simulation time.

When the request nodes need to fetch videos from the network instead of RSUs, the CHR values decrease due to the unbalanced distribution of content. In CNVD, the nodes build the neighbor relationship based on the capacity of video supply and lookup. The similar interests between neighbor nodes ensure the neighbor nodes cache and request the similar videos. The neighbor nodes with large amount of cached videos can support the video request falling in the content cached in neighbor nodes with high probability. The investigation of lookup success rate and number of cached videos in the process of maintenance of neighbor nodes drives the nodes continuously to find more appropriate neighbor nodes with higher lookup success rate and larger number of cached videos, which further promotes the sharing performance between neighbor nodes and increases the CHR values. Therefore, CNVD can obtain high CHR. RUFS only collects the content location information by opportunistic encounter with other nodes and does not ensure the mobile nodes always obtain the location information of requested content. Additionally, the node mobility also leads to fast invalidation of collected information. Therefore, the CHR of RUFS fast decreases and keeps low levels after $t = 100$ s.

5.2.3. Playback Freeze Frequency. The times of occurrence of playback freeze per second are used to denote the playback freeze frequency during the whole simulation time.

We use mean values of all playback freeze frequency during every 20 s as the average playback freeze frequency. Figure 6 shows the variation of playback freeze frequency of the two solutions with increasing simulation time. The curves of CNVD and RUFS experience fast rise with slight jitter from $t = 0$ s to $t = 1000$ s. The increment of RUFS's results is higher than that of CNVD.

In RUFS, the nodes collect the location information of video content by making use of opportunistic message exchange. The node mobility speeds up the invalidation of collected information. Moreover, the nodes only collect information of cached content of adjacent nodes, so that the small-scale collection of content information increases the risk of lookup fail. Therefore, the playback freeze frequency of RUFS keeps fast rise during the whole simulation; namely, the QoE of nodes is relatively low. In CNVD, the nodes which build the neighbor relationship have strong capacity of video supply and lookup. Moreover, the nodes remove the logical link with neighbor nodes in terms of the contribution by the investigation for capacity of video supply and delivery of neighbor nodes, which promotes the sharing performance such as high lookup success rate and low lookup delay. Therefore, the playback freeze frequency of CNVD keeps low increment; namely, the nodes in CNVD can obtain high QoE.

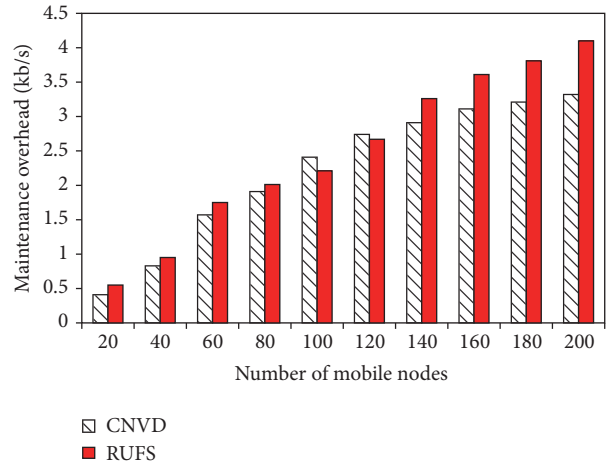


Figure 7: Maintenance overhead versus number of mobile nodes.

5.2.4. Maintenance Overhead. The average bandwidth used to maintain node state and exchange content information every second is defined as the maintenance overhead.

Figure 7 shows the variation of maintenance overhead of the two solutions with the increase in the number of mobile nodes. The black and red bars of CNVD and RUFS keep fast rise trend with the growth of system scale. The black bars corresponding to CNVD keep lower increment than those of

RUFS, even if the maintenance overhead of CNVD is higher than that of RUFS from 100 to 120.

In RUFS, the nodes maintain the information of recently satisfied interests of one-hop neighbor nodes. In order to ensure validity of exchanged information, there is the high-frequency exchange of messages between nodes. Therefore, RUFS has high maintenance overhead. In CNVD, the upper bound of number of neighbor nodes is 10. The small number of neighbor nodes does not bring high maintenance cost. Moreover, in order to further reduce the maintenance cost, the nodes remove the link with neighbor nodes in terms of the contribution. Because the nodes which keep long-term neighbor relationship have high lookup performance, the limited number of neighbor nodes does not bring more negative influence for the content lookup. Therefore, the maintenance overhead of CNVD is lower than that of RUFS.

6. Conclusion

In this paper, we propose a novel contribution-aware neighbor-assist video delivery solution over mobile content-centric network (CNVD). CNVD constructs the estimation model of contribution of neighbor nodes by investigation of lookup success rate, number of cached videos, forwarding delay, and geographical distance. In order to ensure the neighbor nodes have high capacity of video supply and lookup, before construction of neighbor relationship, the nodes estimate interest levels for cached content, measure the lookup performance, and further decide whether to build the neighbor relationship. In order to stimulate nodes to improve utilization rate of cached content and find more appropriate neighbor nodes with strong capacity of video supply and lookup, CNVD designs a removal method of neighbor relationship. The nodes which keep long-term neighbor relationship have strong capacity of video supply and lookup, which promotes video sharing performance and ensures QoE of the request nodes. CNVD designs a contribution-based video lookup algorithm, which achieves fast video lookup. The simulation results show how CNVD has lower lookup latency, high cache hit ratio, lower playback freeze frequency, and lower maintenance overhead than RUFS.

Notations

n_i: Mobile node i
VC_j: Video category j
$C_h(\mathrm{VC}_j)$: Contribution of n_h for video lookup of n_i in VC_j
d_{hr}: The real startup delay of n_i
d_{hl}: The lower bound of startup delay of n_i's QoE requirement
w_h: Weight value of video supply capacity of n_h
r_h: Weight value of video delivery capacity of n_h
R_{ij}: n_j's lookup success rate for video request of n_i
NR_{ij}: Number of videos cached by neighbor node n_j of n_i
d_{ih}: Forwarding delay of interest packets or video data between n_i and n_h
gd_{ih}: Geographical distance between n_i and n_h
VL_a: Video list including ID of videos in VC_a
UV_c: A set of uninterested videos of n_i for videos in VC_c
TH_i: Threshold of interested degree of n_i for video content
CL_a: Set of contribution values of neighbor nodes of n_i for videos in VC_a
ω_e: Weight value of VC_e
LP_{ah}: Estimation value of lookup capacity of n_h for request of videos in VC_a
VT_{ih}: Valid time of neighbor relationship between n_i and n_h.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant nos. 61402303 and 61501216 and the Project of Beijing Municipal Commission of Education under Grant KM201510028016 and the Science and Technology Plan Projects (Openness and Cooperation) of Henan Province (152106000048).

References

[1] S. Chen and J. Zhao, "The requirements, challenges, and technologies for 5G of terrestrial mobile telecommunication," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 36–43, 2014.

[2] L. Zhou, Z. Yang, Y. Wen, and J. J. P. C. Rodrigues, "Distributed wireless video scheduling with delayed control information," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 889–901, 2014.

[3] D. Bethanabhotla, G. Caire, and M. J. Neely, "Adaptive video streaming for wireless networks with multiple users and helpers," *IEEE Transactions on Communications*, vol. 63, no. 1, pp. 268–285, 2015.

[4] S. Jia, C. Xu, J. Guan, H. Zhang, and G.-M. Muntean, "A novel cooperative content fetching-based strategy to increase the quality of video delivery to mobile users in wireless networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 370–384, 2014.

[5] C. Xu, S. Jia, L. Zhong, and G.-M. Muntean, "Socially aware mobile peer-to-peer communications for community multimedia streaming services," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 150–156, 2015.

[6] C. Xu, F. Zhao, J. Guan, H. Zhang, and G.-M. Muntean, "QoE-driven user-centric vod services in urban multihomed P2P-based vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2273–2289, 2013.

[7] N. Kumar and J.-H. Lee, "Peer-to-peer cooperative caching for data dissemination in urban vehicular communications," *IEEE Systems Journal*, vol. 8, no. 4, pp. 1136–1144, 2014.

[8] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang, and G.-M. Muntean, "Performance-aware mobile community-based VoD streaming over vehicular Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1201–1217, 2015.

[9] Y. Zhao, Y. Liu, C. Chen, and J. Zhang, "Enabling P2P one-view multiparty video conferencing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 73–82, 2014.

[10] C. Xu, S. Jia, L. Zhong, H. Zhang, and G.-M. Muntean, "Ant-inspired mini-community-based solution for video-on-demand services in wireless mobile networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 322–335, 2014.

[11] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '09)*, pp. 1–12, ACM, Rome, Italy, December 2009.

[12] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, "VANET via named data networking," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS '14)*, pp. 410–415, Toronto, Canada, April 2014.

[13] L. Zhang, A. Afanasyev, J. Burke et al., "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[14] R. A. Rehman, T. D. Hieu, H.-M. Bae, S.-H. Mah, and B. S. Kim, "Robust and efficient multipath Interest forwarding for NDN-based MANETs," in *Proceedings of the 9th IFIP Wireless and Mobile Networking Conference (WMNC '16)*, pp. 187–192, Colmar, France, July 2016.

[15] Y.-T. Yu, R. B. Dilmaghani, S. Calo, M. Y. Sanadidi, and M. Gerla, "Interest propagation in named data manets," in *Proceedings of the International Conference on Computing, Networking and Communications (ICNC '13)*, pp. 1118–1122, San Diego, Calif, USA, January 2013.

[16] C. Bian, T. Zhao, X. Li, and W. Yan, "Boosting named data networking for efficient packet forwarding in urban VANET scenarios," in *Proceedings of the 21st IEEE International Workshop on Local and Metropolitan Area Networks (LANMAN '15)*, pp. 1–6, Beijing, China, April 2015.

[17] H. Qian, R. Ravindran, G. Q. Wang, and D. Medhi, "Probability-based adaptive forwarding strategy in named data networking," in *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM '13)*, pp. 1094–1101, Ghent, Belgium, May 2013.

[18] W. Gao and G. Cao, "User-centric data dissemination in disruption tolerant networks," in *Proceedings of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM '11)*, pp. 3119–3127, Shanghai, China, April 2011.

[19] S. H. Ahmed, S. H. Bouk, and D. Kim, "RUFS: RobUst forwarder selection in vehicular content-centric networks," *IEEE Communications Letters*, vol. 19, no. 9, pp. 1616–1619, 2015.

[20] A. Razzaq and A. Mehaoua, "Layered video transmission using wireless path diversity based on grey relational analysis," in *Proceedings of the IEEE International Conference on Communications (ICC '11)*, pp. 1–6, Kyoto, Japan, June 2011.

[21] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean, "CMT-QA: quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2193–2205, 2013.

[22] C. Xu, Z. Li, L. Zhong, H. Zhang, and G.-M. Muntean, "CMT-NC: improving the concurrent multipath transfer performance using network coding in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1735–1751, 2016.

[23] C. Xu, Z. Li, J. Li, H. Zhang, and G.-M. Muntean, "Cross-layer fairness-driven concurrent multipath video delivery over heterogeneous wireless networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 7, pp. 1175–1189, 2015.

[24] F. Bai, N. Sadagopan, and A. Helmy, "The important framework for analyzing the impact of mobility on performance of routing protocols for Ad Hoc networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 383–403, 2003.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 6540207, 15 pages
http://dx.doi.org/10.1155/2016/6540207

Research Article

EmuStack: An OpenStack-Based DTN Network Emulation Platform (Extended Version)

Haifeng Li, Huachun Zhou, Hongke Zhang, Bohao Feng, and Wenfeng Shi

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Haifeng Li; 13111026@bjtu.edu.cn

Received 21 August 2016; Accepted 25 October 2016

Academic Editor: Xiaohong Jiang

With the advancement of computing and network virtualization technology, the networking research community shows great interest in network emulation. Compared with network simulation, network emulation can provide more relevant and comprehensive details. In this paper, EmuStack, a large-scale real-time emulation platform for Delay Tolerant Network (DTN), is proposed. EmuStack aims at empowering network emulation to become as simple as network simulation. Based on OpenStack, distributed synchronous emulation modules are developed to enable EmuStack to implement synchronous and dynamic, precise, and real-time network emulation. Meanwhile, the lightweight approach of using Docker container technology and network namespaces allows EmuStack to support a (up to hundreds of nodes) large-scale topology with only several physical nodes. In addition, EmuStack integrates the Linux Traffic Control (TC) tools with OpenStack for managing and emulating the virtual link characteristics which include variable bandwidth, delay, loss, jitter, reordering, and duplication. Finally, experiences with our initial implementation suggest the ability to run and debug experimental network protocol in real time. EmuStack environment would bring qualitative change in network research works.

1. Introduction

The current Internet is based on a number of key assumptions on communication system, including a long-term and stable end-to-end path, small packet loss probability, and short round-trip time. However, many challenging networks (such as sensor/actuator networks and ad hoc networks) cannot satisfy one or more of those assumptions. Excited enough, there have been increasing efforts to support these challenging networks on some special delay and interrupt scenes [1, 2]. In particular, in order to adapt Internet to these challenging environments, Fall proposes Delay Tolerant Networks (DTN) [3]. The key idea of DTN is custody transfer [4] which adopts the hop-by-hop reliable delivery to guarantee the end-to-end reliability. DTN was initially invented for the deep space communication, while currently it has been gradually applied in wireless sensor networks, ad hoc networks, and even satellite networks.

In DTN areas, related research works such as routing and congestion control strategies have obtained many achievements [5, 6] along with a number of DTN implementations such as DTN2, ION, and IBRDTN [7–9]. However, many problems [10, 11] such as security and contact plan design have not been resolved yet.

In order to further study DTN architecture, many experimental platforms have been designed. Koutsogiannis implements a testbed to evaluate space-suitable DTN architectures and protocols with many deep space communication scenarios [12]. The DTN testbed can support about ten nodes experimental topology. Based on the generic-purpose wireless network bench, Beuran designs a testbed named QOMB [13]. QOMB has a good support for emulating a large-scale mobile networks, but it wastes lots of hardware resources since none of virtual computing technology is employed. Thus, QOMB lacks a monitoring system; the experimental fidelity cannot be guaranteed especially in the large-scale scene. Komnios introduces the SPICE testbed [14] for researching space and satellite communication. SPICE is equipped with special hardware and it can accurately emulate the link characteristics between the space and ground stations. However, due to the introduction of professional hardware, SPICE is hard to be imitated by other researchers.

Meanwhile, without using network virtualization technology, the emulation topology of SPICE is fixed and will be changed difficultly.

With the advancement of network and compute virtualization technology, it becomes much easier to design and implement a scalable and flexible emulation platform than before. In this work, EmuStack, a network emulation platform for DTN, is introduced. Our design objective is enabling EmuStack to support a large-scale, real-time, and distributed network emulation and provide synchronous and dynamical precise management for topology and link characteristics. For example, Docker container technology [15] is utilized as the compute virtualization technique into efficiently virtualize several physical emulation nodes into hundreds of virtual emulation nodes. By integrating Linux Traffic Control (TC) utility [16] with OpenStack [17], EmuStack can achieve more fine-grained control of the virtual topology and link characteristics. Meanwhile, OpenStack is composed of various independent modules; thus it possesses a good support for the development of the other functionalities in EmuStack. To improve the performance of EmuStack, many OpenStack subprojects are adopted. An example is Ceilometer [18] which is developed lightly and integrated into EmuStack for ensuring experimental fidelity and monitoring, alarming, and collecting relevant data.

As we have a deeper insight into our initial work [19], in this paper, we further present details of controlling link characteristics and analyze the reason for link rate-limiting difference between the Ethernet device of virtual emulation node and the TAP device of physical emulation node. Moreover, we further introduce EmuStack scalability and performance and discuss their main factors. Additionally, we provide one more DTN experiment to better evaluate and demonstrate the performance of EmuStack.

The remainder of this paper is organized as follows. In Section 2 we introduce the related work. In Sections 3 and 4, we present architectural design, implementation of EmuStack and thoroughly discuss performance of EmuStack. Then we reproduce two published classic DTN experiments and compare and analyze the key experimental results in Section 5. Finally, in Section 6, we conclude this paper along with future works.

2. Related Work

Recently, with the advancement of container virtualization technology, network researchers show their interest in employing container to construct their experimental platforms to support their large-scale topology experiments. Emulab [20] is one of the well-known testbeds using the container virtualization in Linux. Due to the efficiency of container, Emulab possesses a good support for scalability. Although these technologies introduced in Emulab are not the latest now, the design philosophies are still helpful for current researchers to design large-scale test bed. Additionally, Lantz et al. [21] designed Mininet based on container virtualization technique including processes and network namespaces technique. Mininet can support SDN and run on a single computer. Handigol et al. [22] further improved Mininet performance with enhancements to resource provisioning, isolation, and monitoring system. Besides, Handigol replicated a number of previously published experimental results and proved that Linux Container (LXC) [23] technology is not only lightweight but also possesses a good fidelity and performance. In order to perform an in-depth performance evaluation of LXC, Xavier et al. [24] conducted a number of experiments to evaluate various compute virtualization technologies and finally proved that LXC virtualization has a near-native performance on CPU, memory, disk, and network. Therefore, in EmuStack, we employ Docker container (based on LXC) as compute virtualization technology.

OpenStack is an open-source reference framework mainly for developing private and public cloud, which consists of loosely-coupled components that can control hardware pools of compute, network, and storage resources. OpenStack is composed of many different independent modules, and anyone can add additional components into OpenStack to meet their requirements. Therefore, OpenStack is definitely a good choice for developing emulation platform.

3. Architectural Design

This section describes the overall architecture design of EmuStack from the perspective of hardware and software.

3.1. Hardware. Figure 1 shows EmuStack hardware structure (where gray rectangles stand for primary services installed). EmuStack hardware can be composed of only several physical nodes (general-purpose computer). There are two types of physical nodes: network emulator and physical emulation nodes. Network emulator is the core hardware which is a physical node equipped with multiple NICs in EmuStack and it plays multiple roles. It is not only an OpenStack controller node which manages compute and network resources and an OpenStack network node which manages virtual emulation networks, but also an emulation orchestrator which is responsible for creating emulation parameters and orchestrating the whole resources of CPU, memory, and network. In addition, physical emulation node is a compute node of OpenStack, which hosts all virtual emulation nodes and executes the emulation control commands from network emulator.

In EmuStack, there are two types of physical networks, namely, the management network and emulation network. Management network carries management traffic which consists of lightweight control information and usually does not become the determinant of performance. Emulation network transfers emulation data which consumes much bandwidth and would vary greatly with different DTN protocol experiments. Therefore, the physical emulation network possibly becomes the main limitation of EmuStack. For several physical nodes system of EmuStack, adopting the star structure can solve the emulation data traffic bottleneck problem, as shown in the bottom right of Figure 1. In this structure, all emulation NICs of physical emulation nodes are directly connected to those of network emulator. NICs of network emulator are attached to an Open vSwitch bridge, where the "internal" device named after itself is assigned an IP address belonging to the emulation network. In practice, this physical emulation

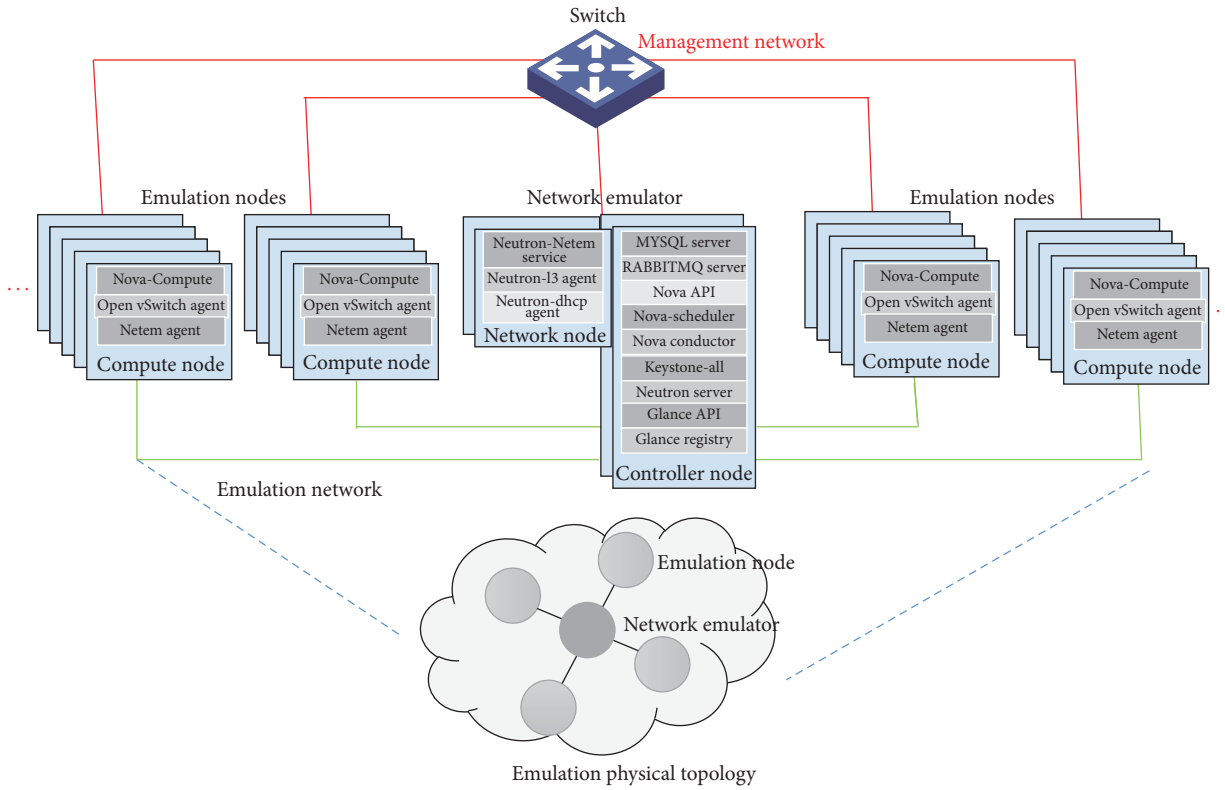


Figure 1: EmuStack hardware structure.

network structure can meet most requirements of our DTN research works; however if researchers want to construct the EmuStack system that consists of dozens of physical nodes, this structure would become infeasible since network emulator would not have enough NICs to directly connect to all the emulation NICs of physical emulation nodes. For system with dozens of physical nodes, physical emulation network can employ several physical switches to carry the emulation data as management network does. In this scheme, as the first step, we need to determine which one of physical NICs on network emulator (and physical emulation nodes) will carry management traffic. Then we connect all the remaining NICs of network emulator (and physical emulation nodes) to those physical switches. Those physical switches ports will need to be specially configured to allow trunked or general traffic. Finally, for EmuStack system with hundreds of physical nodes, as a part of the future work, we will extend network emulator to support distributed processing and enable multiple network emulators to exist in EmuStack.

3.2. Software. Figure 2 describes EmuStack software synopsis involving network emulator, physical emulation node, and virtual emulation node. As the key component of EmuStack, network emulator carries many open-source services and customized service extensions. Nova service and the core plugin in Neutron service is attended to initialize virtual emulation nodes and virtual emulation network, respectively. Additionally, these services also have the ability to create, modify, and delete virtual emulation nodes and virtual emulation network. Neutron-Netem service is responsible for generating experimental parameters and data to dynamically control experimental program, topologies, and link characteristics. Meanwhile, in order to provide sufficient fidelity and reduce experimental complexity at the same time, we adopt Telemetry Management (Ceilometer) [18] service to monitor and collect hardware resources and experimental data. In addition, Keystone [25], Horizon [26], and Glance [27] are utilized to provide the support for managing authentication, authorization, service catalog, web interface, and image services. Besides, as a part of the future work, on the basis of OpenStack Heat service, we will develop the orchestrator to more efficiently and flexibly orchestrate the distributed hardware resource management [28]. Most of those services are open-source projects and available in OpenStack; hence we only need to integrate them to meet most EmuStack design requirements. In order to implement synchronous, dynamic, precise, and real-time emulation control service, we design and implement the Neutron-Netem service and Neutron-Netem agent, which will be further discussed in Section 4.

As shown in the bottom left of Figure 2, physical emulation node is regarded as a compute node in OpenStack where virtual emulation nodes are hosted. Physical emulation node runs Nova-Compute service driven by the Docker hypervisor to manage virtual emulation nodes and Open vSwitch agent to execute the managing emulation network commands (including create, modify, and delete function) from network emulator. Open vSwitch agent employs two Open vSwitch (OVS), "OVS for emulation" and "OVS for

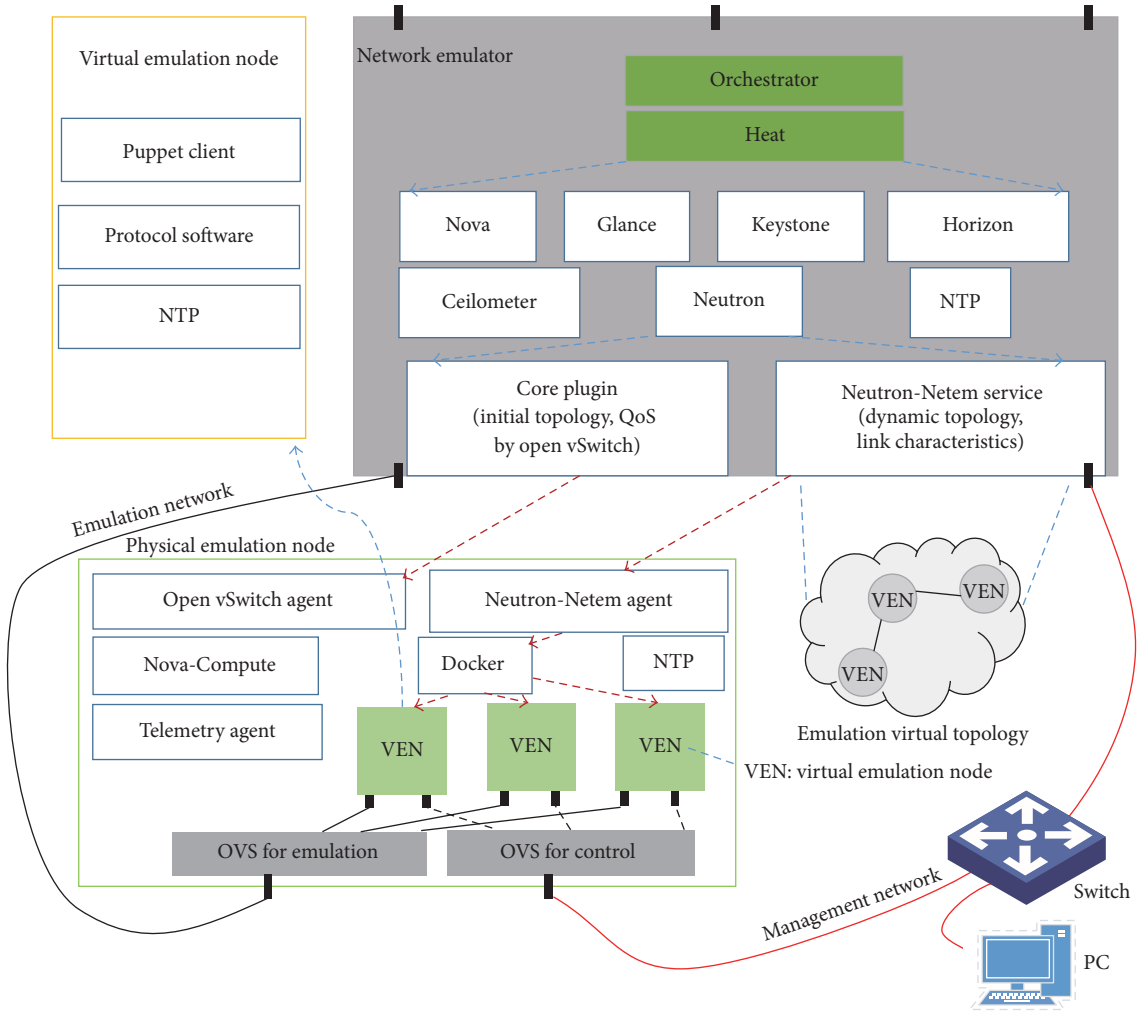


Figure 2: Synopsis of EmuStack software.

control" to manage virtual emulation networks and virtual management networks, respectively. Open vSwitch agents manage virtual networks by configuring flow rules on the above two OVS. Moreover, as the agent of Ceilometer service in network emulator, Telemetry Agent is responsible for publishing collected data to network emulator through the management network and creating alarms once collected data breaks the monitoring rules. Finally, Neutron-Netem agent is designed to precisely and dynamically control emulation topologies and link characteristics, which will be further introduced in Section 4.

As shown in the upper left of Figure 2, virtual emulation node (VEN) is a Docker virtual machine which is hosted in physical emulation node. It is spawned from the operating system image where Network Time Protocol (NTP) service, custom network protocol software, and Puppet client service can be installed. In particular, Puppet client service can be utilized by virtual emulation nodes to receive control information from network emulator or physical emulation nodes.

Note that time synchronization is very essential for EmuStack. The DTN bundle protocol depends on absolute time to determine whether received packets are expired. Furthermore, EmuStack must ensure the experimental program in different virtual emulation nodes which can be exactly synchronously executed in the correct time sequences. Therefore, Chrony [29], an implementation of NTP [30], is installed in all nodes to provide the properly synchronizing services. In detail, network emulator is configured to reference accurate time servers while physical and virtual emulation nodes refer to network emulator. In our local area network (LAN) of EmuStack, the time synchronization precision reaches as high as 0.1 milliseconds, which meets the requirements for most emulation experiments.

4. Implementation

This section describes the details of EmuStack core modules (Neutron-Netem service and Neutron-Netem agent). Firstly, in order to sketch the outline of EmuStack implement, EmuStack emulation workflow is described in Section 4.1. Secondly, Sections 4.2, 4.3, and 4.4 present the details of emulation synchronous control, topology control, and customization of

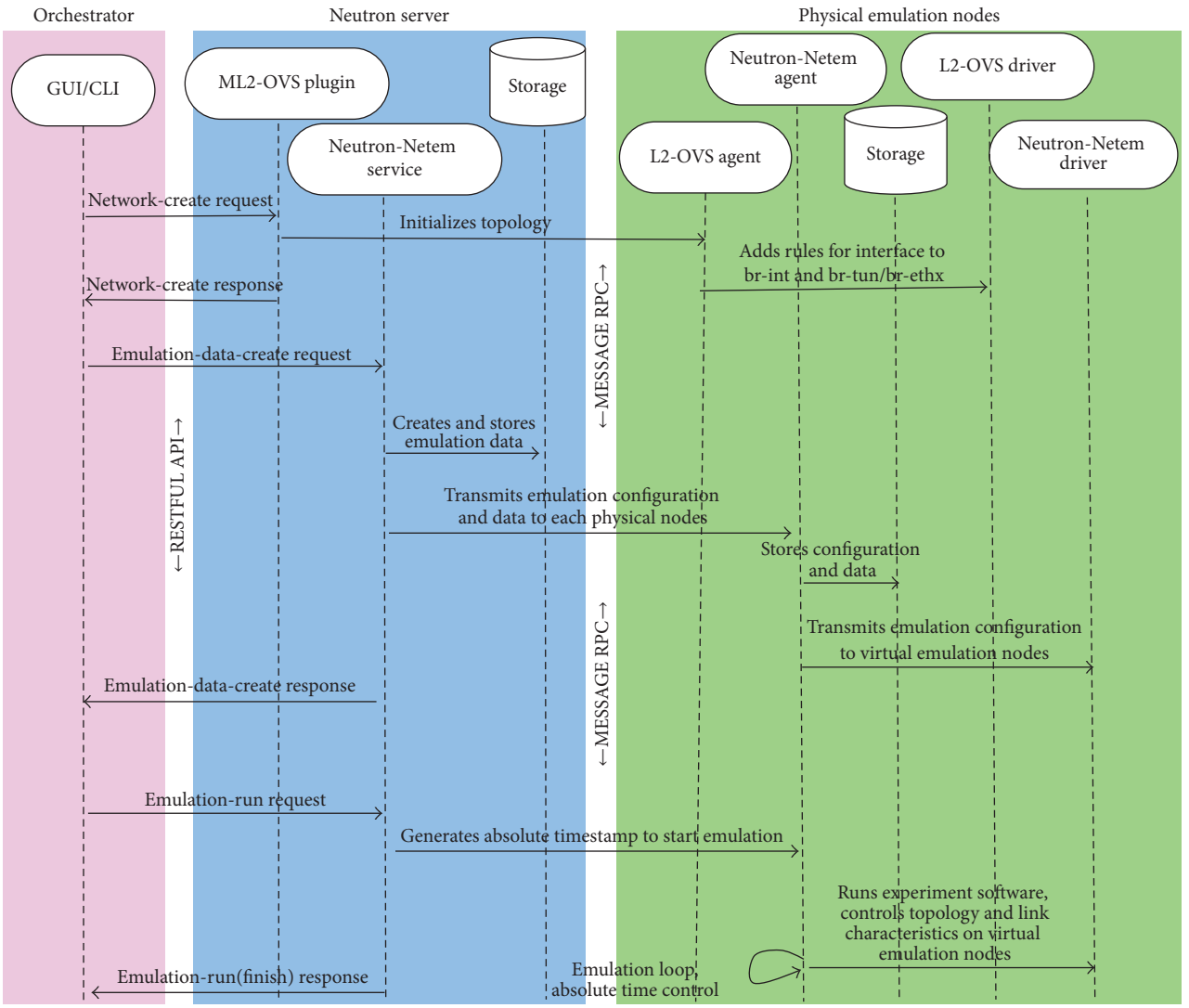


Figure 3: Process flow of emulation network.

link characteristics, respectively. Finally, the scalability and performance of EmuStack are discussed in Section 4.5.

4.1. Emulation Workflow. Before the beginning of emulation, we first create a virtual machine image, where special software and shell scripts should be installed to fulfill the specific experimental requirements. For example, you must install an SSH server (or Puppet client) into the image and ensure that it starts up on boot with the correct configuration, or you may install shell scripts to collect some experiment results. Next, we create virtual networks before launching virtual emulation nodes. Virtual networks are composed of two types of networks, namely, management network and emulation network. Management network is Neutron flat network in OpenStack, where all nodes (including virtual, physical emulation nodes, and network emulator) reside on the same network and no VLAN tags are created. Emulation network involves one or more private virtual networks. Moreover, one virtual emulation node could belong to either one or more virtual emulation networks. After creating virtual network, we launch a sufficient number of virtual emulation nodes and initialize virtual networks, right before running the emulation.

Unlike a simulator running in virtual time based on discrete event, EmuStack runs in real time and cannot pause a node's clock to pend for events. For a distributed real-time emulation platform, it is difficult to ensure that every control command can be executed synchronously in the different physical nodes due to the stochastic communication delay and background system load. In order to avoid communication delay, especially the control information transmission delay, EmuStack stores the control information in the local-storage before emulation starts to run.

We can now introduce the process flow of emulation network described in Figure 3. Note that the ML2-OVS plugin, L2-OVS agent, and L2-OVS driver are components of the core plugin in Neutron service. As with OpenStack, EmuStack firstly initializes emulation topology by launching instances together. Secondly, after a successful initialization, the orchestrator requests Neutron-Netem service to run

mobility module to create topology and link characteristics data. Meanwhile, in order to support the requirements of those who evaluate the same experimental protocol with different protocol parameters and the same model data, Neutron-Netem service stores the generated model data in the persistent storage. Thirdly, Neutron-Netem service dispatches the emulation data to each agent residing in every physical emulation node. The emulation data is split into different parts for each agent and every agent just only receives its own part and stores it. Relatively, every agent can transmit experimental configuring parameters to virtual emulation nodes by invoking Puppet server API. In each virtual emulation node, Puppet client works in kick-mode and starts to receive configuration (or command) once triggered by Neutron-Netem agent. Finally, after dispatching the emulation data, the orchestrator sends a request to Neutron-Netem service to start emulation; then Neutron-Netem service delivers an absolute timestamp to every agent. Once the staring time is up, agents will start to emulate the experiment, and therefore, the starting timestamp has to be a little (such as sixty seconds) larger than current timestamp, and that extra time is left for Neutron-Netem agents receiving the starting timestamp.

In EmuStack, Neutron-Netem service is organized into separate submodules such as storage and mobility modules. In particular, Neutron-Netem service provides a simple plugin mechanism to enable users to extend different mobility modules. Thus mobility modules can be individually built as researchers' own experimental purposes. The various mobility modules are intended to provide required realistic network emulation environment for different experimental network protocol development. Besides, Neutron-Netem service provides the inheritance mechanism that a mobility module can be developed based on the others. The primary functionality of a mobility module is to create data for dynamically controlling emulation topology and link characteristics. In Section 5, we will employ two mobility modules for DTN large file transmission experiment and the DTN routing protocol comparison experiment of Probabilistic Routing with Epidemic, respectively.

4.2. Synchronous Control. Algorithm 1 describes the synchronous control of Neutron-Netem agent. As shown in lines (2) to (4), Neutron-Netem agents all are asleep and synchronously start emulation once the starting time comes. The time synchronization accuracy depends on the sleeping time SLEEP_TIME and the NTP synchronization precision. Since the NTP synchronization precision is as high as 0.1 milliseconds in our platform environment, the synchronization accuracy is only up to SLEEP_TIME. In fact, the SLEEP_TIME is a trade-off between synchronization precision and system load. In practice, we set SLEEP_TIME to 100 (milliseconds) to satisfy the requirements of most experiment with lightweight CPU load.

With the coming of starting time, Algorithm 1 goes into the outer loop as shown in lines (11) to (23). This outer loop takes advantage of absolute time to control its cycles. As shown in line (13), LOOP_CYCLE (loop cycle) is an important parameter for this loop system. The topology and link characteristics are updated every LOOP_CYCLE. The control operation delay (lines (14) to (16)) plus sleeping time (lines (20) to (22)) is around equal to LOOP_CYCLE. However, due to system load and other unknown factors, the control operation delay may be larger than LOOP_CYCLE by accident; this will lead to synchronous control failure. To help users evaluate the fidelity of the experiments, this failure information all is logged (lines (17) to (19)). Besides, the exceeded time will force future cycles of the loop to reduce the sleeping time; this will enable platform to synchronize again. After the end of outer loop, Neutron-Netem agents kill all experiment processes to get ready for next experiment.

```
(1)  INIT protocol software
(2)  WHILE current_time < starting_time
(3)          sleep SLEEP_TIME miliseconds
(4)  END WHILE
(5)
(6)  INIT topology control
(7)  INIT link characteristics control
(8)  START state collection
(9)
(10) SET counter to 0
(11) WHILE counter < CONTROL_PERIOD
(12)    increment counter
(13)    next_time = start_time + \
                    LOOP_CYCLE * counter
(14)    control topology
(15)    control link characteristics
(16)    control protocol software
(17)    IF current_time – next_time > THRESHOLD
(18)       collect error log
(19)    END IF
(20)    WHILE current_time < next_time
(21)       sleep SLEEP_TIME miliseconds
(22)    END WHILE
(23) END WHILE
(22) KILL all experiment processes
```

Algorithm 1: Synchronous control on Neutron-Netem agent.

4.3. Controlling Topology. Figure 4 provides details on the controlling topology and link characteristics. As shown in the right of Figure 4, Neutron-Netem service delivers the control information to Neutron-Netem agents in advance. According to the received control information, Neutron-Netem agents invoke their driver to dynamically control the emulation experiment once the starting time is coming. In particular, as a part of this control information, the topology control information is described by connection matrix in EmuStack, as shown in Figure 5. In fact, a network topology, no matter how complex it is, can be represented by a connecting relationship between any two nodes. An example for a three nodes topology is shown in Figure 5, where "1" corresponds to connection between two nodes and "0" means disconnection.

In EmuStack, the connection matrix along with time sequences is generated by mobility module. According to connection matrix, Neutron-Netem agents periodically

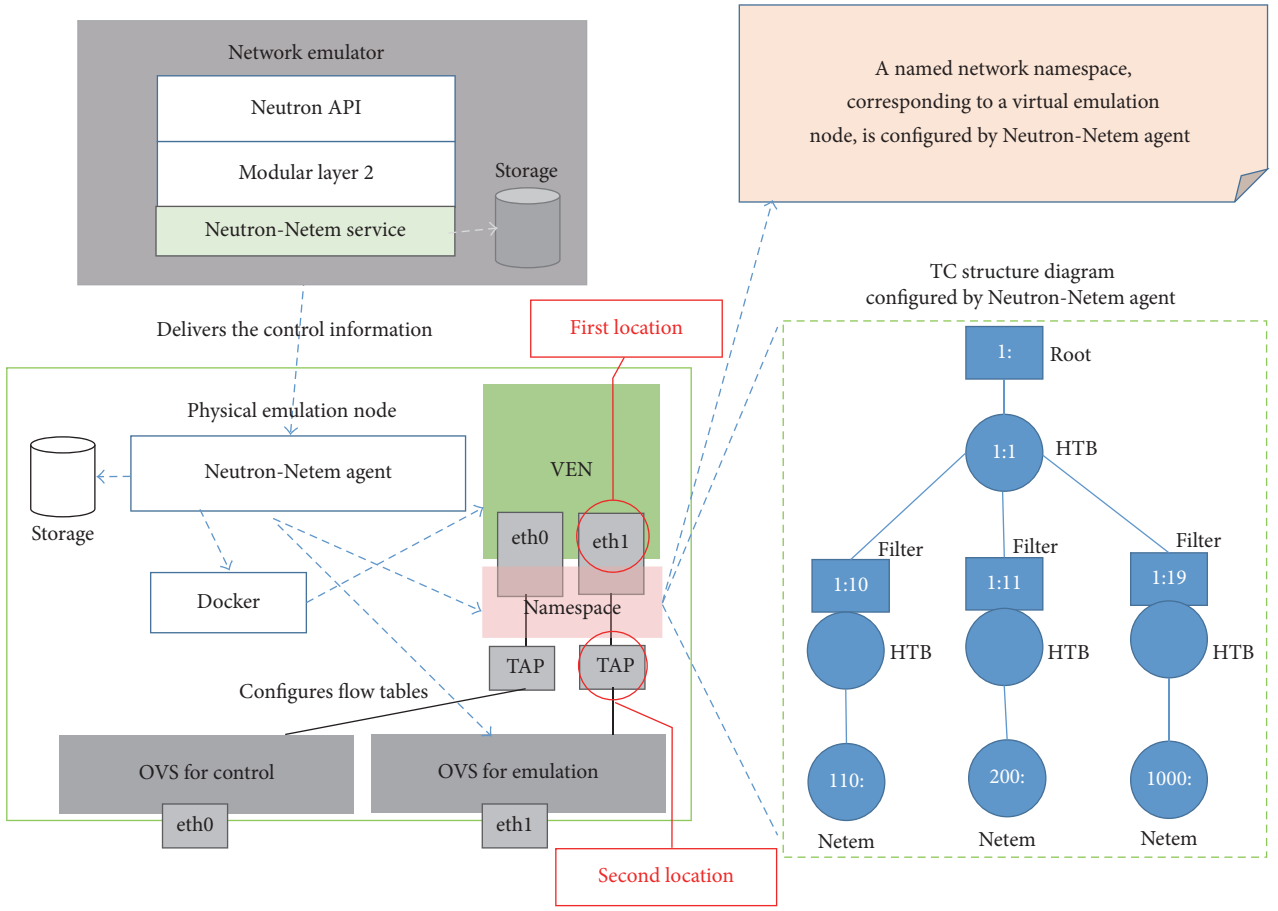


Figure 4: Topology and link characteristics control.

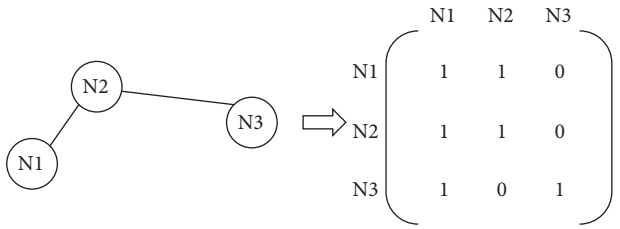


Figure 5: Simple topology and connection matrix.

invoke their drivers to dynamically change emulation topology during the emulation. There are two ways to dynamically controlling emulation topology: one is based on Open vSwitch and the other is to depend on iptables. Neutron-Netem agents can control virtual emulation topology by configuring flow tables on "OVS for emulation." Managing virtual emulation topology in this way is similar to how Neutron-Open vSwitch agent manages virtual topology in OpenStack, but Neutron-Netem agents can do these more efficiently and quickly. Meanwhile, Neutron-Netem agents can achieve higher synchronous precision since they have already store the emulation control information into local-storage, while Neutron-Open vSwitch needs to get this control information by Remote Procedure Call (RPC) services which take a long-term delay. Additionally, Neutron-Netem agents can dynamically control virtual emulation topology by configuring iptables entries in the special named network namespace. This namespace is corresponding to the virtual emulation node as shown in the top right of Figure 4. In the initial implement of EmuStack, the second way to control topology is implemented in Neutron-Netem agent driver, whose performance will be discussed in Section 4.5. As to the first method, we would take it into consideration in the future work.

4.4. Controlling Link Characteristics. In Linux, system offers a very rich set of tools for traffic control. The Traffic Control, TC, utility is one of the most famous tools. TC is good at shaping link characteristics which include link bandwidth, latency, jitter, packet loss, duplication, and reordering. Besides, it allows users to set queuing disciplines (QDiscs) within network namespace. There are two types of QDiscs in TC: one is classful queuing disciplines which have filters attached to them and allow traffic to be directed to particular classed queues or subqueues; the other is classless queuing disciplines which can be used as primary QDiscs or inside a leaf class of a classful QDiscs. As shown in the bottom right of Figure 4, Hierarchical Token Bucket (HTB) [31] is classful QDiscs, and Netem [32] is classless. In EmuStack,

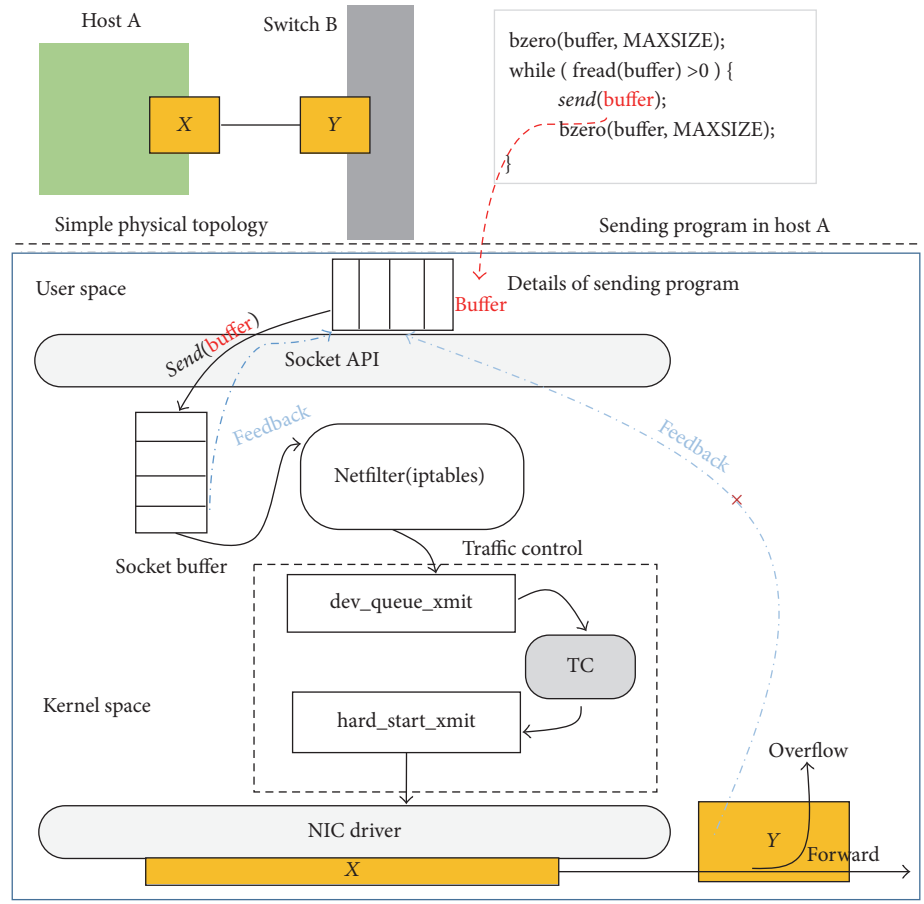


Figure 6: Rate-limiting difference between two locations.

Neutron-Netem agents use HTB to control link rate, attaching filters to HTB QDiscs to distinguish different virtual emulation links. Meanwhile, Netem is used inside HTB leaf classes to emulate variable delay, loss, reordering, and duplication.

In telecommunications, a link is a communication channel that connects two communicating devices (such as network interfaces); a media access control address (MAC address) is a unique identifier assigned to network interface for communications. Hence, in EmuStack, we can use source-destination MAC addresses to configure filter rules to distinguish different virtual emulation links. In particular, due to the high link asymmetry in most DTN experiments, EmuStack adopts the source-destination ordered pairs to distinguish the difference between uplink and downlink. Meanwhile, we elaborately design control policies since TC QDiscs are only good at shaping outgoing traffic. For example, assuming a link between node A and node B, for A, EmuStack handles A's uplink at one end of the link (on A) and controls A's downlink at the other end of the link (on B); then emulation data can be shaped bilaterally. In addition, EmuStack also can create one or more special intermediate virtual nodes for all virtual emulation nodes of the same physical emulation node to shape their downlink traffic.

We can limit link rate at both locations as shown at the middle of Figure 4. The two locations marked with red circles stand for two different network devices. The first location stands for network interfaces in virtual emulation nodes. All network interfaces are corresponding to those of named network namespaces. The second location represents TAP devices which are paired with those network interfaces mentioned above and attached to Open vSwitch ("OVS for emulation"). Limiting link rate at both locations is feasible, but there are some notable differences. Assuming experimental network protocols (such as UDP) do not have any congestion control algorithms, then any rate-limiting at the second location will lead to a large number of packet loss, but this will not happen at the first location. In most DTN experiments, rate-limiting leading to much packet loss probably is not what we want, and we mostly expect that rate-limiting and packet loss do not interfere with each other.

Figure 6 describes the rate-limiting difference between the two locations with a simple topology and a sending program. In this simple topology, X device is at the first location and Y device is at the second location. Assume that the sending program calls UDP socket API. When sending program sends application data, Linux kernel copies application data from user space buffer to socket buffer. If socket buffer

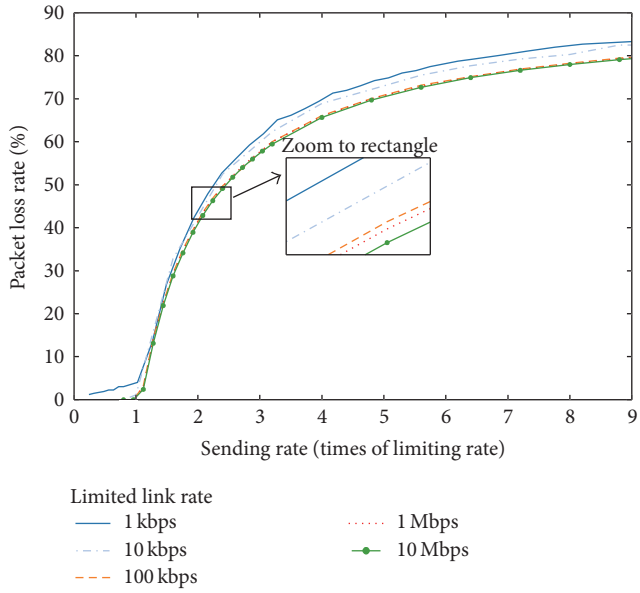


Figure 7: Feature of rate-limiting at the second location.

ever gets full, the blocking socket will put the program in sleep state until the socket buffer has enough space, or the nonblocking socket will return the error "Operation Would Block" immediately. Therefore, no matter which mode (blocking or nonblocking) the socket works in, sending program always receives a "feedback," and this prevents the socket buffer from overflowing as shown in Figure 6. These packets are delivered to QDiscs buffer to shape them and finally transmitted to link by NIC driver. In brief, TC QDiscs consume packets in socket buffer and clear socket buffer, and then sending program can send application data to socket buffer again. As a result, TC indirectly affects the transmission of sending program; a sufficient TC buffer and the feedback mechanism ensure packet loss does not to happen. As to rate-limiting at Y (the second location), since there is not feedback between Y and sending program, Y ingress buffer will overflow and drop most application data as shown in Figure 7.

Figure 7 presents the relationship between packet loss rate and sending rate for rate-limiting at the second location. As expected, when rate-limiting is fixed, the larger the sending rate (times), the more packets the link drops. Meanwhile, setting the sending rate as constant and HTB buffer as default, the larger the rate-limiting, the smaller the packet loss rate. For most DTN scenarios, this is not what we want to see except for testing congestion control algorithms. For example, assuming NIC bandwidth is 90 Mbps and rate-limiting is 10 Mbps, packet loss rate will be up to 80 percent.

In current EmuStack version, we implement all link characteristics control at the first location but only achieve rate-limiting function at the second location by configuring ingress policing rules in Open vSwitch (OVS for emulation). Although rate-limiting at the second location has been implemented in QoS (Quality of Service) plugin of OpenStack Neutron service, it is implemented in centralized model and the synchronous precision is too low. Hence we reimplement the function with the distributed model and obtain that higher synchronous precision is achieved.

4.5. Scalability and Performance. We deploy EmuStack in our experimental platform consisting of nine physical nodes. Each physical node is an identical Dell™ PowerEdge™ R720 2U rack server with one 2.4 GHz Intel Xeon E5-2609 processor (with 4 cores), 10 M of L3 cache per core, 32 GB RAM, and Broadcom 5720 Quad Port 1 GbE BASE-T. In particular, network emulator is integrated with four more Intel EXPI9402PT Dual Port NICs. All management network interfaces of nine physical nodes are interconnected by TP-LINK TL-SF1024D Ethernet switch. All emulation network interfaces of eight physical emulation nodes are linked to those of network emulator. The Ubuntu 14.04 LTS Linux distribution is installed on the all physical nodes and the NetworkManager service is not allowed to start up upon boot, since NetworkManager always repeatedly invokes the useless dhclient program and occupies an amazing number of CPU resources whenever EmuStack launches Docker containers. In addition, operating system kernel version is 3.19.0-31, iptables version is 1.4.21, iprouter2 version is ss131122, and Docker version is 1.10.1. Based on these platform environments, we analyze the emulation scalability and performance as follows.

Compute (CPU), memory (RAM), and network (NIC) are the three chief factors of EmuStack scalability. To make efficient use of CPU and RAM, EmuStack adopts Docker container as virtualization technology instead of kernel-based full virtualization solutions. Docker containers share the same operating system kernel so that they can consume fewer CPU and RAM resources. For example, our platform launches sixty containers on a single machine with about nine percent of CPU usage and ten percent of RAM usage, which serve as virtual emulation nodes which are installed with Ubuntu 14.04 LTS and start up with OpenSSH server and Puppet client. Additionally, in order to ensure that emulation network does not hit network bottleneck easily, EmuStack dispatches compute requests to as few as possible physical emulation nodes for the same experiment, so most virtual emulation nodes are interconnected by the internal bridge (OVS for emulation) and communications between them can consume the least bandwidth of physical emulation network. Meanwhile, all emulation network interfaces on the network emulator are attached to a Linux bridge to improve the bandwidth of physical emulation network. All of these enable EmuStack to support hundreds of nodes with nine physical nodes.

The major factor of EmuStack performance is the updating delay, the time consumed by changing experimental emulation topology or link characteristics for one time. In Algorithm 1, the updating delay determines the LOOP_CYCLE parameter presented on line (13). The minimum LOOP_CYCLE should be no less than the maximum updating delay; otherwise EmuStack will fails to synchronize. In addition, current EmuStack version employs iptables and TC utility to dynamically control the virtual emulation network, respectively. Hence the iptables and TC performance directly impact EmuStack performance and their processing

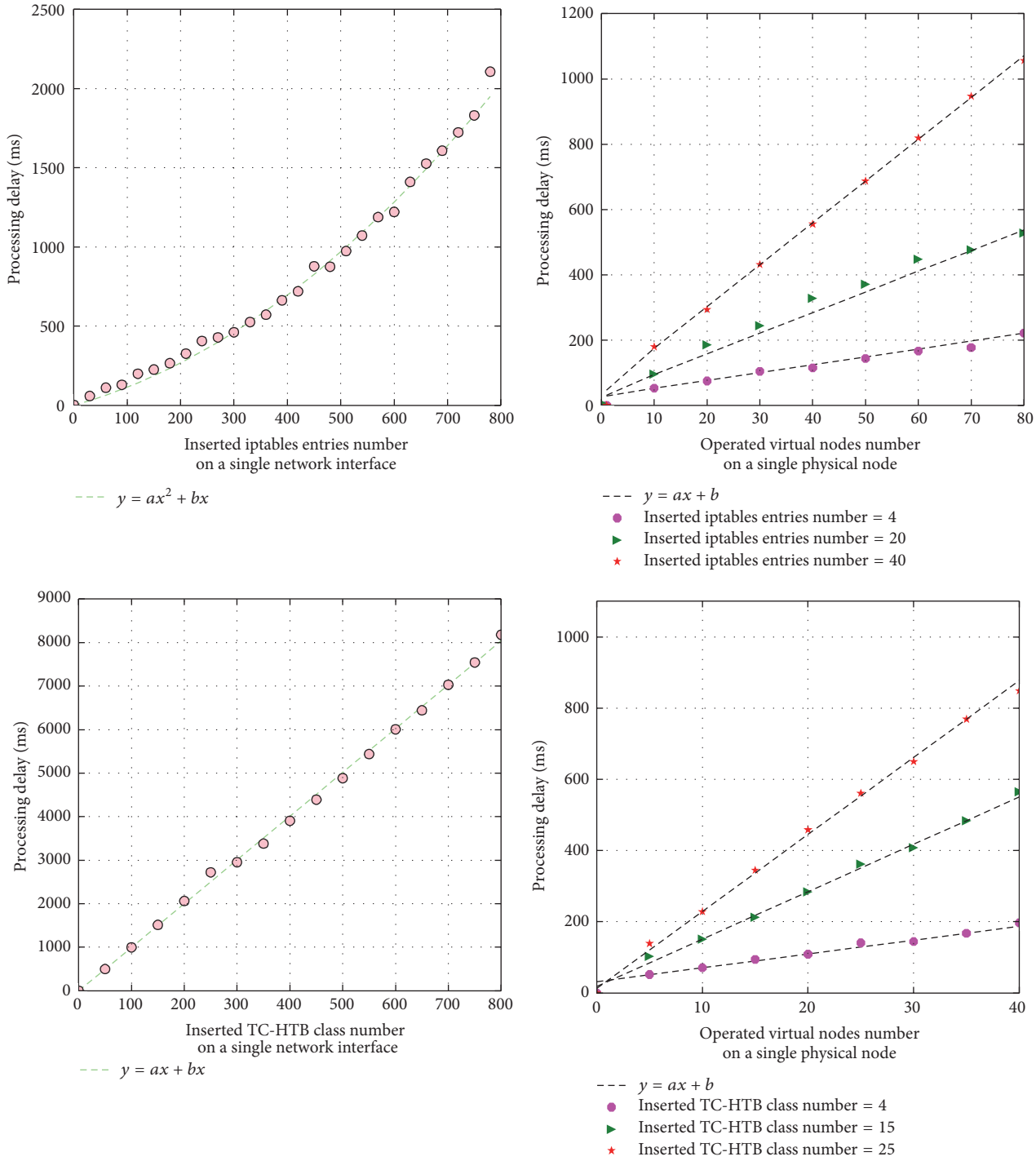


Figure 8: The average performance of iptables and TC.

delay has direct impact on the updating delay. The performance of iptables and TC is analyzed as follows.

Figure 8 shows the average performance of iptables and TC, where average performance stands for updating delay trend. The left of Figure 8 describes the performance for operating at a single network interface. Interesting enough, for iptables, the average processing delay can be represented by quadratic function of inserted entries number; for TC-HTB, the relationship between average processing delay and inserted entries number can be well described by a linear function. Hence EmuStack can estimate processing delay with both functions of law. The right shows the performance of concurrently operating multiple virtual nodes in a single physical node. The processing delay of iptables and TC all

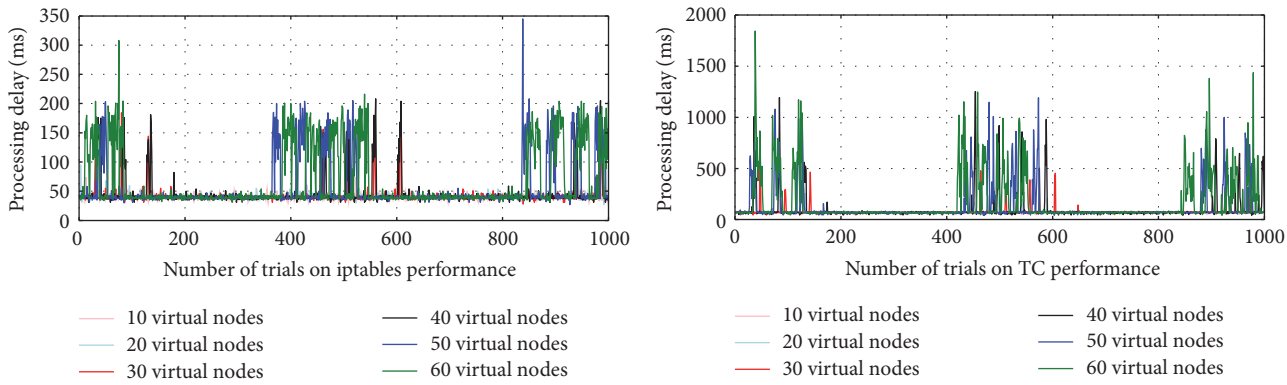


Figure 9: The real-time performance of iptables and TC.

grow linearly when the inserted iptables entries (or TC-HTB class) number is fixed, and this is influenced by serialization, contention, and system load.

Figure 9 shows the real-time performance where the processing delay is the time it takes to insert ten iptables entries (or TC-HTB classes) into a single network interface. The processing delay starts to fluctuate violently with the increasing number of virtual nodes in a single physical node (each virtual node has a network interface which is paired with TAP device in host namespace and linked to Open vSwitch). For example, when there are less than thirty virtual nodes in a single physical node, the processing delay remains stable throughout one thousand trials. However, when the number of virtual nodes increases up to sixty, the fluctuation scope gets wider, with an iptables maximum scope that reaches about 350 milliseconds. TC maximum scope is up to 1800 milliseconds which is five times more than that of iptables. Hence the updating delay of link characteristics (TC processing delay) is probably the most serious limitation in EmuStack.

By analyzing the feature of the real-time performance, we can estimate the maximum updating delay and obtain the minimum LOOP_CYCLE for a specific scale experiment in simple experimental environment (single user). However, it is hard to do that in complex experimental environment (multiuser), and this will raise a lot of complex problems such as virtual nodes orchestration. Because of the limited space, we do not get into details and have a deeper insight into such topic here, leaving this part to be discussed in the future work.

5. Experimental Evaluation

To evaluate and demonstrate EmuStack, this section reproduces key results from two published DTN experiments. One is the DTN large file transmission experiment that applies Low Earth Orbit (LEO) [33, 34] and the other one is the DTN routing protocol comparison experiment of Probabilistic Routing with Epidemic [35, 36]. The goal of the first experiment is to prove that results obtained on EmuStack can match with the results measured on hardware. The goal of the second experiment is to demonstrate that EmuStack can dynamically change a large-scale topology and precisely support a large-scale experiment.

5.1. Large File Transmission Using LEO Satellite. One type of LEO satellite is the remote sensing satellite. Generally, remote sensing satellite transfers a lot of sensing data to the ground station, and these data are usually in a large scale. For example, a single raw picture created by earth observation satellite is usually in a room of hundreds of megabytes (MB) or even more. Unfortunately, only about 10 minutes contact time is allowed for LEO when passing over a ground station in one orbital cycle. Additionally, the LEO transmission rate is low; taking the UK-DMC satellite [34] as an example, there is a downlink of 8.134 Mbps and uplink of 9600 bps. Therefore, it is almost impossible that LEO can transfer a large file to the ground during the period that it passes over a single ground station. Actually, three passes are needed to transfer the complete file to the ground as shown in Figure 10. During each pass, LEO transfers one segment of the total file to Earth Control Center via one Earth Gateway (GW), and once the job of transferring the complete image file is finished, it has been reassembled at the Earth Control Center.

To test whether the results obtained by EmuStack can match those of hardware, we created the experimental topology both in EmuStack and in real hardware as shown in Figure 10. The real hardware environment is built by seven physical nodes, where we utilize TC shell scripts to dynamically control the topology and link characteristics. All the parameters of real hardware that are related to the large file transmission are the same as those of EmuStack, which is described in the following passage.

To ensure reality of experiment process and data, firstly we use Satellite Tool Kit (STK) [37] to model the LEO link characteristics and topology as described in the Table 1. Based on the parameters generated by STK, we write this experimental mobility module in Neutron-Netem service. The mobility module can create the emulation topology and link control information according to the requirements of the large file transmission. Secondly, the OpenStack virtual machine image equipped with the DTN network protocol software ION-3.3.1 [38] is built. ION-3.3.1 uses CFDP [39] program to fragment and reassemble the 258 MB image file from LEO to Earth Control Center. CFDP is configured with 32 kilobytes (kB) bundle [40], 128 kB block of Licklider Transmission Protocol (LTP) [41], and Contact Graph Routing (CGR) [42] protocol. Additionally, it is worth noting that TC

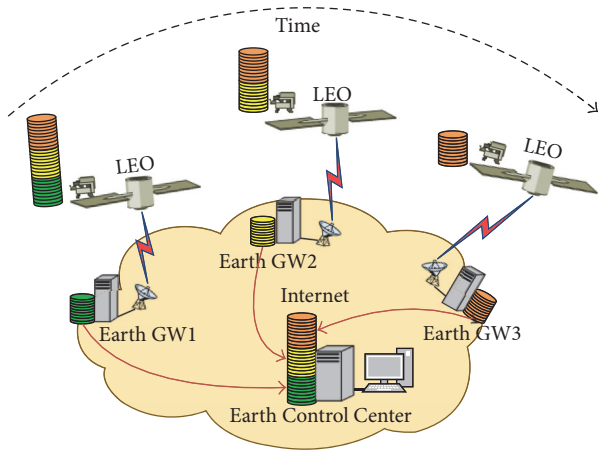


Figure 10: LEO block transmission scenario.

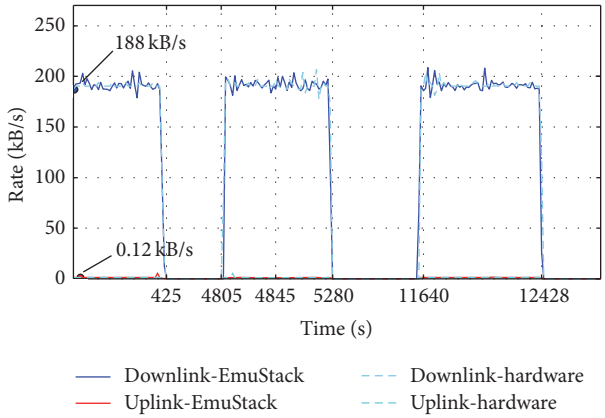


Figure 11: Downlink rate and uplink rate in the Earth Control Center.

Table 1: LEO block transmission scenario contact plan.

Contact or idle	Duration	Rate	Delay	Jitter	Packet loss rate
LEO → GW1	425 s	200 kB/s	15 ms	2 ms	0.3%
Idle period	73 min				
LEO → GW2	475 s	200 kB/s	16 ms	4 ms	1.1%
Idle period	106 min				
LEO → GW3	788 s	200 kB/s	13 ms	3 ms	0.8%

Netem delay is limited by the frequency (HZ) of the Linux system clock (the tick rate), and the system clock should run at 1000 HZ to allow Netem delays in increment of 1 millisecond.

Figure 11 shows the downlink rate and uplink rate in the Earth Control Center. Due to the effectiveness of ION scheduled, LTP starts a transmission as soon as the link is available. During the whole transmission, LEO first transmits about 79 MB block of image file to Earth Control Center via earth GW1. After about 73 minutes' disconnection, LEO secondly establishes the connection with earth GW2 and transmits another 79 MB block to Earth Control Center. Finally, with 106 minutes' break, LEO transmits the rest of image file to Earth Control Center via earth GW3. In this experiment, experimental results show that downlink unitization is high (about 94%), and the ratio between downlink rate and uplink rate is 1600 : 1. These results prove that DTN protocol family has a good support for intermittent and asymmetric links. Thus EmuStack can be employed to achieve significant results in advance of (or possibly without) setting up a hardware testbed. Meanwhile, since the results of EmuStack closely match with those of the hardware, it indicates that EmuStack has good support for experimental fidelity.

5.2. Comparison of PROPHET Routing with Epidemic. Vahdat and Becker present a routing protocol for DTN called Epidemic routing [36]. This routing protocol allows nodes to exchange summary vectors (an index of their own messages) and request messages which were not owned once they encounter each other. This means messages will spread through the network like an Epidemic, as long as buffer is large enough and the possibility exists. Lindgren et al. propose PROPHET, a Probabilistic Routing Protocol using History of Encounters and Transitivity [35]. The operation of PROPHET is similar to that of Epidemic routing. When two hosts meet, they exchange summary vectors which also contain delivery possibility. Relying on this predictability data, each node calculates the new delivery possibility, which is used to decide which messages to request from the other node.

To evaluate the ability of EmuStack precise control in large-scale experiment, we emulate the simulation experiment described in the PROPHET paper [35] and compare PROPHET with Epidemic in the community scenario. The community scenario consists of a 3000 ∗ 1500 m area and fifty-six virtual emulation nodes as shown in Figure 12. The area is split into twelve subareas: eleven communities (C1–C11) and one "gathering place (G)." Every community contains five nodes (the same color circles as shown in Figure 12): one fixed node acting as the gateway of the community and four mobile nodes; all nodes treat the community as their home community. The four mobile nodes of every community select a destination, move there with a speed between ten and thirty miles per second, pause there for a moment, and select a new destination and speed. The probabilities of different destinations are chosen according to the current location of mobile nodes. In the experiment, a warm-up period of 500 seconds is used to initialize protocols, and 3000 seconds is used to create and delivery massages, and another 8000 seconds is used for allowing more messages to be delivered.

In order to emulate the above community mobility scene in EmuStack, we develop the community mobility model into a mobility module. Before the beginning of experiment, we first create the experimental virtual image which is equipped with IBRDTN [9]. IBRDTN supports the Epidemic routing and PROPHET routing whose "link_request_interval" parameter is set as 1000 milliseconds since the community model is updated every second. The other model parameters are configured the same as those in PROPHET paper [35].

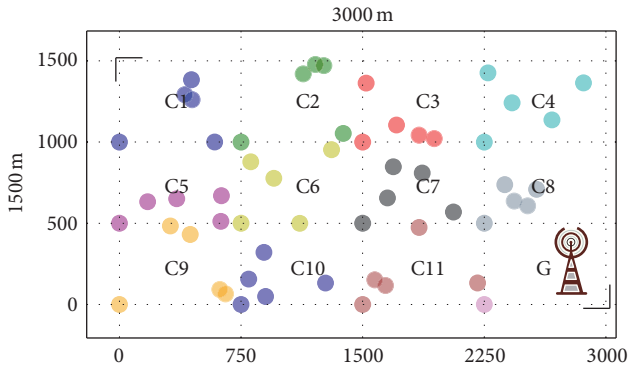


Figure 12: Community mobility model.

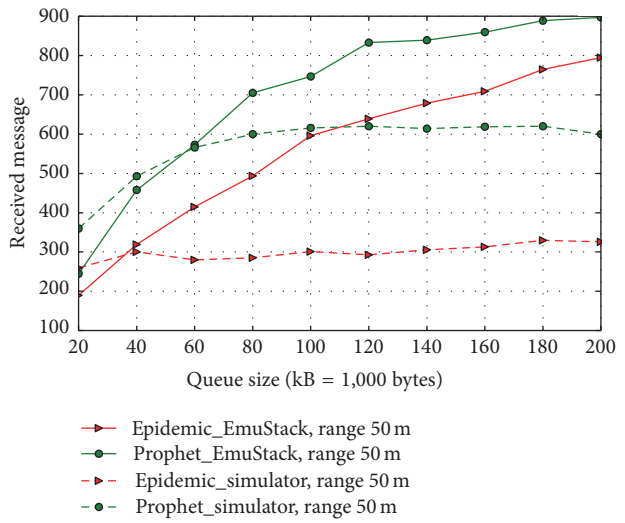


Figure 13: The average delivery rates in community scenario.

Note that LOOP_CYCLE is set as one second in the experiment. We attempt to dispatch the fifty-six virtual nodes to different numbers of physical nodes; then EmuStack performs the above experiment for several times with the different configurations of the IBRDTN "limit_storage" parameter (namely, the queue size). At the end of experiments, we check Neutron-Netem agents logs for synchronizing errors. We find that even though all the fifty-six virtual nodes are orchestrated into a single physical node, no synchronizing errors were thrown in EmuStack. This indicates the ability to precisely control large-scale experiment in EmuStack. We further discuss the details of experimental results in the following passage.

Figure 13 shows the average delivery rates in both EmuStack and the simulator described in the PROPHET paper (Hop count = 11). The Epidemic and PROPHET routing protocol show the similar performance in both EmuStack and the simulator. For example, with the increasing size of the queue, the number of messages which eventually reach destination goes up. It is obvious that they can be buffered for longer time and get more opportunities to be delivered successfully, since the larger queue size would enable more messages to be cached and less be dropped. Meanwhile, as

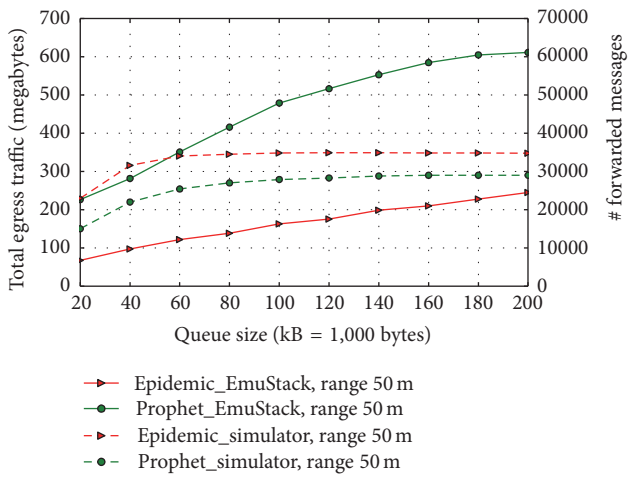


Figure 14: The consumption of the network resources in the community scenario. In the simulator, Lindgren utilizes the number of forwarded messages to indirectly evaluate the consumption; in EmuStack, we employ the value of the total egress traffic to directly measure the consumption.

shown in Figure 13, the PROPHET routing protocol has a much better performance compared with the Epidemic routing protocol in terms of the delivery rate, and the results of EmuStack matches with those of the simulator. All these results can demonstrate that both PROPHET and Epidemic routing protocols run normally in EmuStack. EmuStack can emulate the large-scale experiments.

Figure 14 presents the consumption of the network resources in the community scenario. In the simulator [35], Lindgren utilizes the number of forwarded messages that occur when nodes encounter each other to indirectly evaluate the consumption; in EmuStack, we employ the value of the total egress traffic to directly measure the consumption. The egress traffic is composed of the forwarded messages and routing overhead; hence it can achieve the more comprehensive evaluation for the consumption of network resources. As described in Figure 14, in EmuStack, PROPHET has a much higher network overhead than Epidemic, as opposed to that in the simulator. This is because the Epidemic routing protocol has been optimized by IBRDTN [9]. IBRDTN already has replaced the summary vectors of the basic Epidemic with the efficient Bloom-Filter mechanism and manages a purge vector as an extension of the Epidemic routing protocol which ensures the bundles delivered successfully to be deleted throughout the network. Therefore Epidemic can consume fewer network traffic than the origin PROPHET described in [35].

Finally, Figure 15 describes the average delivery delay in both EmuStack and the simulator. There are two ways of calculating the average delay. One way is by dividing the sum of the delay of the messages successfully delivered by the number of those (delay 1). The other way is by dividing the sum of the delay of all the messages successfully and unsuccessfully delivered by the number of those (delay 2). The delay of those unsuccessfully delivered messages is defined as

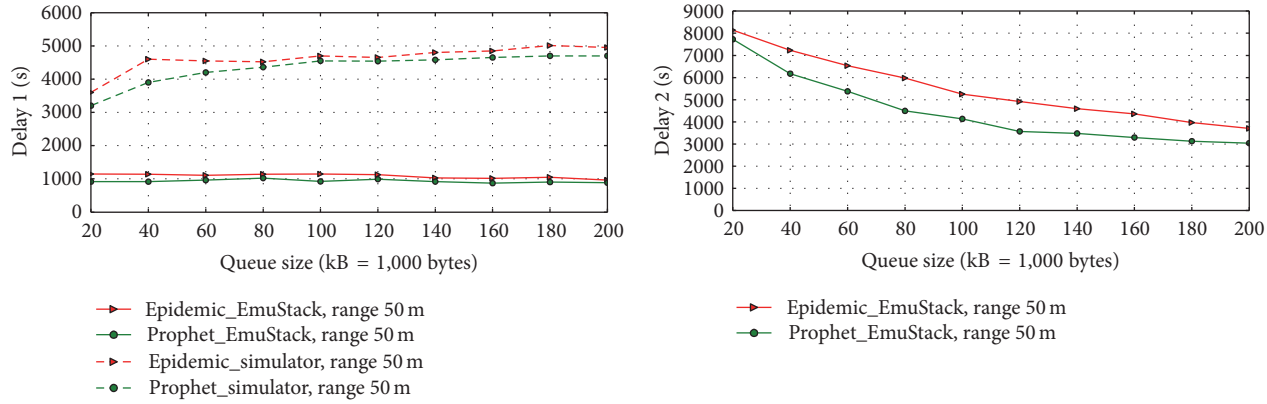


Figure 15: The average delivery delay in community scenario.

the value of subtracting the messages' sending time from the experimental ending time.

The delay 1 is utilized to evaluate the average delay of massages in [35]. As shown in the left of Figure 15, the value of the delay 1 fluctuates back and forth with the increase of queue size. As we all know, the larger queue can shorten the delivery delay of the messages which would be successfully delivered even if the queue is relatively small, and it can also enable messages which would be unsuccessfully delivered to reach their destination nodes, while the value of the delivery delay of these messages become larger compared with the origin zero. Therefore, there are increases and decreases in the delivery delay value, which result in the value of delay 1 fluctuating in small scope.

Due to the above phenomenon, we argue that the first way of calculating delivery delay may be unreasonable. Hence we attempt to evaluate the average delivery delay of the forwarded massages by the second way where we take the delivery delay of unsuccessfully delivered messages into consideration when calculating the average delivery delay. As shown in the right of Figure 15, with the increasing size of the queue, there is an obvious decrease in the average delivery delay (delay 2) for both routing protocols. It is intuitive that the value of delay 2 decreases since larger queue leads to more messages delivered successfully and quickly. In short, no matter which method is used to calculate the average delivery delay, PROPHET always has shorter delivery delay than Epidemic in both EmuStack and the simulator.

As we expected, all the above results demonstrate that EmuStack can reproduce the key results of the large-scale DTN experiment described in [35] and achieve more details of the experimental network protocols than the simulator, which is helpful for us to further improve the design of the experimental network protocol.

6. Conclusion

In this work, we present a real-time, distributed, and scalable emulation platform based on OpenStack for DTN. Firstly, we discuss hardware, software deployment, the design architecture, and implementation. Specially, we present the details of control of link characteristics and topology. Secondly, we analyze the platform scalability and performance. Finally, we evaluate and demonstrate the emulation platform with two classical DTN experiments.

In order to have a thorough evaluation, as a part of the future work, we will create more realistic mobility and link characteristic models to emulate more complex DTN experiments. Meanwhile, we will also evaluate these effects with different virtual computing, virtual network technologies and complex experimental environment (multiuser orchestration).

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by NSFC of China under Grant no. 61271202, NSAF of China under Grant no. U1530118, National High Technology of China ("863 program") under Grant no. 2015AA015702, and National Basic Research Program of China ("973 program") under Grant no. 2013CB329101.

References

[1] H. Zhang, P. Dong, W. Quan, and B. Hu, "Promoting efficient communications for high-speed railway using smart collaborative networking," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 92–97, 2015.

[2] W. Quan, C. Xu, J. Guan et al., "Social cooperation for information-centric multimedia streaming in highway VANETs," in *Proceedings of the IEEE 15th International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM '14)*, pp. 1–6, IEEE, Sydney, Australia, June 2014.

[3] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 27–34, ACM, August 2003.

[4] K. Fall, W. Hong, and S. Madden, "Custody transfer for reliable delivery in delay tolerant networks," IRB-TR 03-030, 2003.

[5] A. P. Silva, S. Burleigh, C. M. Hirata, and K. Obraczka, "A survey on congestion control for delay and disruption tolerant networks," *Ad Hoc Networks*, vol. 25, pp. 480–494, 2015.

[6] M. Liu, Y. Yang, and Z. Qin, "A survey of routing protocols and simulations in delay-tolerant networks," in *Wireless Algorithms, Systems, and Applications*, vol. 6843 of *Lecture Notes in Computer Science*, pp. 243–253, Springer, Berlin, Germany, 2011.

[7] Delay Tolerant Networking Research Group, DTNReference Implementation, March 2016, https://sites.google.com/site/dtnres-group/home/code.

[8] S. Burleigh, "Interplanetary overlay network an implementation of the DTN bundle protocol," in *Proceedings of the 2007 4th Annual IEEE Consumer Communications and Networking Conference (CCNC '07)*, pp. 222–226, IEEE, Las Vegas, Nev, USA, January 2007.

[9] S. Schildt, J. Morgenroth, W. B. Pöttner et al., "IBRDTN: a lightweight, modular and highly portable Bundle Protocol implementation," *Electronic Communications of the EASST*, vol. 37, 2011.

[10] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: a comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.

[11] J. A. Fraire and J. M. Finochietto, "Design challenges in contact plans for disruption-tolerant satellite networks," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 163–169, 2015.

[12] E. Koutsogiannis, S. Diamantopoulos, and V. Tsaoussidis, "A DTN testbed architecture," in *Proceedings of the International Conference on Ultra Modern Telecommunications & Workshops (ICUMT '09)*, pp. 1–2, St. Petersburg, Russia, October 2009.

[13] R. Beuran, S. Miwa, and Y. Shinoda, "Network emulation testbed for DTN applications and protocols," in *Proceedings of the 32nd IEEE Conference on Computer Communications (INFOCOM '13)*, pp. 3441–3446, Turin, Italy, April 2013.

[14] I. Komnios, I. Alexiadis, N. Bezirgiannidis et al., "SPICE testbed: a DTN testbed for satellite and space communications," in *Testbeds and Research Infrastructure: Development of Networks and Communities: 9th International ICST Conference, TridentCom 2014, Guangzhou, China, May 5–7, 2014, Revised Selected Papers*, vol. 137 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 205–215, Springer, Berlin, Germany, 2014.

[15] Docker, https://www.Docker.com/.

[16] "TC," http://www.lartc.org/lartc.html.

[17] OpenStack, http://www.openstack.org/.

[18] "Ceilometer," https://launchpad.net/ceilometer.

[19] H. Li, H. Zhou, H. Zhang et al., "EmuStack: an openstack-based DTN network emulation platform," in *Proceedings of the IEEE International Conference on Networking and Network Applications (NaNA '16)*, pp. 387–392, Hakodate, Japan, July 2016.

[20] M. Hibler, R. Ricci, L. Stoller et al., "Large-scale virtualization in the emulab network testbed," in *Proceedings of the USENIX Annual Technical Conference (USENIX '08)*, pp. 113–128, Boston, Mass, USA, June 2008.

[21] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Monterey, Calif, USA, October 2010.

[22] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible network experiments using container-based emulation," in *Proceedings of the 8th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT '12)*, pp. 253–264, ACM, Nice, France, December 2012.

[23] "LXC," https://linuxcontainers.org/.

[24] M. G. Xavier, M. V. Neves, F. D. Rossi, T. C. Ferreto, T. Lange, and C. A. F. De Rose, "Performance evaluation of container-based virtualization for high performance computing environments," in *Proceedings of the 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP '13)*, pp. 233–240, March 2013.

[25] "Keystone," https://wiki.openstack.org/wiki/Keystone.

[26] "Horizon," https://wiki.openstack.org/wiki/Horizon.

[27] "Glance," https://wiki.openstack.org/wiki/Glance.

[28] F. Song, D. Huang, H. Zhou, H. Zhang, and I. You, "An optimization-based scheme for efficient virtual machine placement," *International Journal of Parallel Programming*, vol. 42, no. 5, pp. 853–872, 2014.

[29] "Chrony," http://chrony.tuxfamily.org/.

[30] "NTP," http://www.ntp.org/.

[31] "HTB," http://luxik.cdi.cz/~devik/qos/htb/manual/userg.htm.

[32] NetEm, http://www.linuxfoundation.org/collaborate/workgroups/networking/netem.

[33] W. Ivancic, W. M. Eddy, L. Wood et al., "Delay/disruption-tolerant network testing using a LEO satellite," in *Proceedings of the NASA Earth Science Technology Conference (ESTC '08)*, University of Maryland, June 2008.

[34] C. Caini and R. Firrincieli, "Application of contact graph routing to LEO satellite DTN communications," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 3301–3305, IEEE, Ottawa, Canada, June 2012.

[35] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.

[36] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Tech. Rep. CS-20000, Duke University, 2000.

[37] "STK," http://www.agi.com/products/stk/.

[38] "ION-DTN," http://sourceforge.net/projects/ion-dtn/.

[39] S. C. Burleigh, "CFDP for interplanetary overlay network," *NASA. Tech Briefs*, vol. 35, no. 3, p. 36, 2011.

[40] K. L. Scott and S. Burleigh, Bundle protocol specification, 2007.

[41] R. Wang, S. C. Burleigh, P. Parikh, C.-J. Lin, and B. Sun, "Licklider transmission protocol (LTP)-based DTN for cislunar communications," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 359–368, 2011.

[42] S. Burleigh, Contact graph routing, 2010.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 8695102, 13 pages
http://dx.doi.org/10.1155/2016/8695102

Research Article

Diffusion Strategies for Distributed Kalman Filter with Dynamic Topologies in Virtualized Sensor Networks

Shujie Yang, Tao Huang, Jianfeng Guan, Yongping Xiong, and Mu Wang

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Shujie Yang; zongshi@bupt.edu.cn

Received 23 August 2016; Accepted 25 September 2016

Academic Editor: George Ghinea

Network virtualization has become pervasive and is used in many applications. Through the combination of network virtualization and wireless sensor networks, it can greatly improve the multiple applications of traditional wireless sensor networks. However, because of the dynamic reconfiguration of topologies in the physical layer of virtualized sensor networks (VSNs), it requires a mechanism to guarantee the accuracy of estimate values by sensors. In this paper, we focus on the distributed Kalman filter algorithm with dynamic topologies to support this requirement. As one strategy of distributed Kalman filter algorithms, diffusion Kalman filter algorithm has a better performance on the state estimation. However, the existing diffusion Kalman filter algorithms all focus on the fixed topologies. Considering the dynamic topologies in the physical layer of VSNs mentioned above, we present a diffusion Kalman filter algorithm with dynamic topologies (DKFdt). Then, we emphatically derive the theoretical expressions of the mean and mean-square performance. From the expressions, the feasibility of the algorithm is verified. Finally, simulations confirm that the proposed algorithm achieves a greatly improved performance as compared with a noncooperative manner.

1. Introduction

With the rising of Big Data analysis and Smart City Conception, a variety of mobile applications such as Multipath Data Transfer [1, 2], target tracking [3–5], environmental monitoring [6, 7], Video-On-Demand Services [8–10], and distributed data storage have reshaped our daily life. In this context, wireless sensor networks (WSNs) as a key technology or such applications have drawn a significant attention from both academia and industry [11–16]. However, traditional WSNs consist of large number of heterogeneous sensors having capability of sensing, computation, and wireless communication, which make them fail to share of infrastructure and support the large scale sensing coverage. Fortunately, the rapid development of network virtualization opens a new opportunity for wireless sensor networks and creates a new sensing paradigm named virtualized sensor networks (VSNs) [17–19]. The salient feature of VSNs is enabling heterogeneous WSNs to coexist on a shared physical substrate. This allows VSNs to be more scalable and flexible, and, therefore, VSNs have become a promising solution for large scale sensing task. Despite the unprecedented opportunities brought about by the VSNs, how to guarantee the sensing accuracy is a critical issue since mobile sensors in VSNs are highly dynamic and fully distributed without central control. To resolve this problem, we focus on the distributed Kalman filter algorithms, which is the combination of distributed algorithms and Kalman algorithms.

On the one hand, in distributed algorithms, a set of nodes can estimate the target state accurately through a cooperation manner. These nodes can be PCs, laptops, cell phones, sensors, and actuators [20]. Applications of distributed estimation algorithms are in several contexts, including wireless and sensor networks, where scalability, robustness, and low power consumption are desirable [21–23].

On the other hand, Kalman filter algorithms are one of the most popular methods for estimating the states of dynamic system from an incomplete and noisy measurement. As a recursion algorithm, Kalman filter has little requirements on calculation and memory space, which makes it more favorable in the real-time system application. Since it was proposed in 1960s, Kalman filter has been widely applied in

many fields, such as navigation, signal processing, control system, and information fusion [24].

As one strategy of distributed Kalman filter algorithms, diffusion Kalman filter algorithm has a better performance on the state estimation by diffusing information through a sequence of Kalman iterations and data-aggregation. In [25, 26], diffusion Kalman filtering has been introduced. According to them, a diffusion Kalman filter algorithm is comprised of the incremental update step and the diffusion update step. In the incremental update step, each node receives the observations from its neighbors and combines these observations to update its existing estimate to an intermediate value. In the diffusion update step, every node combines its neighbors intermediate estimates produced by the last step to update its own estimate. It is worth mentioning that all nodes perform these two steps similarly at the same time. The algorithm has an excellent performance in tracing a moving target as well as a good performance in convergence.

All of the literatures above assumed the links between nodes were ideal. However, in VSNs, the performance of the physical layer is strongly affected by the presence of such link state, where nodes and links may be subject to failure. These issues motivated the study of adaptive networks in dynamic scenarios. In [27], the author proposed a method to analyze the least mean-square (LMS) filter algorithm with changing topologies and normalized data. Similarly, a novel distributed affine projection algorithm (APA) with dynamic diffusion networks was presented in [28]. Although these algorithms give a good performance in dynamic topologies, they are all one kind of Wiener filtering. As we know, Wiener filtering required that the signal and noise must be a smooth process, which greatly restricts its application. And because of the limited shortage of these filters, they fail to obtain accurate measurement for moving targets. Considering these problems, we present a diffusion Kalman filter algorithm with dynamic topologies (DKFdt) to solve them. However, in order to analyze the rationality of the algorithm, we further need to consider the complexity and convergence of the algorithm under certain reasonable assumptions. Therefore, after proposing the new algorithm, our objective is to study whether the algorithm will have a good convergence and whether the performance of diffusion strategies in this scenario is still well. So in this article, we detailedly study the steady-state performance of the mean and the mean square of our algorithm based on link instabilities. All of these will be the basis of our future research into VSNs.

In comparison, we derive the models for the transient and steady-state behavior of the diffusion algorithm. From the results, we can obtain an interesting observation that although communication between nodes is limited to a small fraction, which is due to a probabilistic diffusion protocol, the mean-square performance does not present a significant decrease compared to a theory diffusion strategy.

The remainder of this paper is organized as follows. Section 2 lays the related work about our study and Section 3 describes the background of our research and lays the foundation of this article. The analysis of the DKFdt algorithm is presented in Section 4. In Section 5, we analyze the mean and mean-square performance of the estimate errors based on the proposed algorithms and then simulation is given in Section 6. The conclusion and future work are described in Section 7.

We summarize the key symbols used in this paper in Notation.

2. Related Work

2.1. Framework of VSNs. Wireless sensor networks (WSNs) have been used in many application domains (e.g., temperature monitoring, security, and trajectory tracking) and become more and more popular in the last few years. However, as WSNs consist of large number of heterogeneous sensor nodes having capability of sensing, computation, communication and so forth, multiple applications sharing the very same WSNs infrastructure become a problem to be solved. Virtualization is a technology that can potentially enable this sharing.

Actually, network virtualization may bring nothing new in terms of technical capabilities and theoretical performance, but it provides a way of organizing networks such that it is possible to overcome some of the practical issues in traditional Internet. Now, in the field of virtualized sensor networks, research mainly focuses on the design and analysis of the VSNs framework [17, 18, 29–31]. In this part, we analyze previous work and summarize overall VSNs architecture shown in Figure 1. As can be seen from it, the VSNs consist of three layers: physical layer, virtualization layer, and user layer.

Physical Layer (PL). This layer consists of large number of heterogeneous sensor nodes for different purposes of sensing the environment like temperature, sound, and trajectory tracking. It offers their resources and observations through programmable interfaces to virtualization layer.

Virtualization Layer (VL). According to the needs of the upper user layers, VL integrates information provided by the physical layers to format a new virtual topology and provide variety of services for end users. It can also create child VLs by partitioning its resources and act as a virtual PL by leasing those child networks to other VLs (Figure 1).

User Layer (UL). The user layer is very similar to the application layer of the traditional network; users will request resource and information from the VL according to their own needs and meanwhile measure the QOS of the whole network.

From the analysis above, we can see that information or estimate value provided by PL affects the whole network performance. And because of the dynamic and distribute properties in VSNs, we should focus on how to get the accuracy estimation value in PL under the dynamic and distributed scenario firstly.

2.2. Distributed Algorithms. We review related research on distributed algorithms about distributed estimate problems. In distributed algorithms, nodes with processing and learning abilities are linked together to solve distributed optimization and estimation in real time through cooperating with each other.

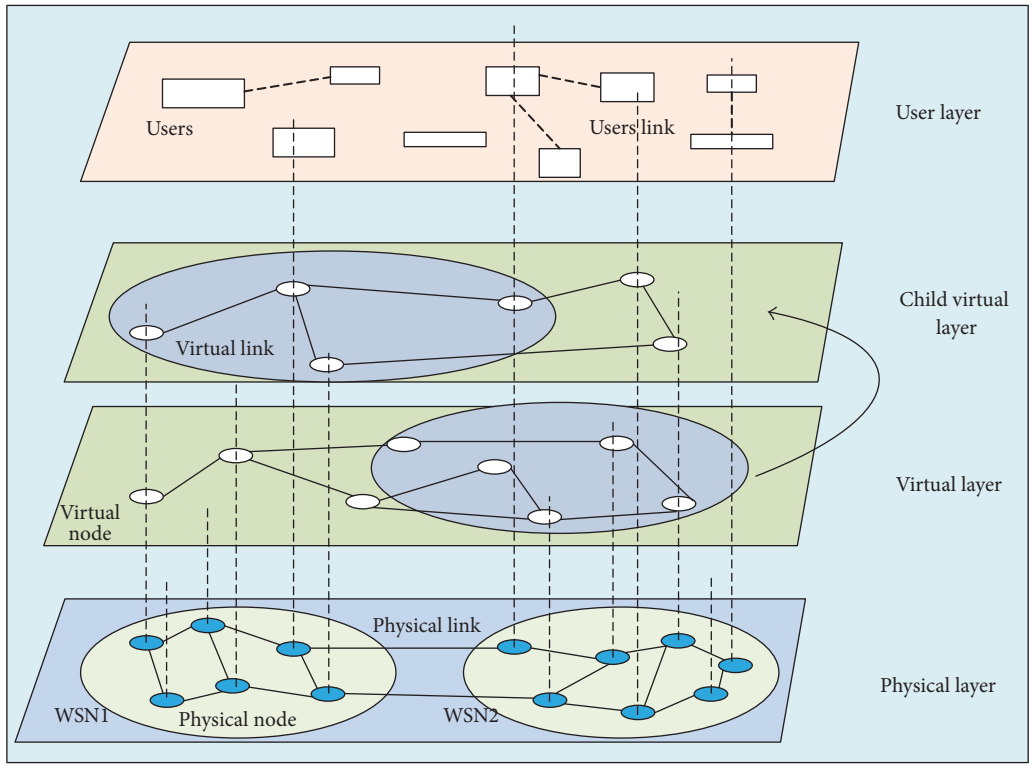


Figure 1: Framework of virtualized sensor network.

The existing distributed algorithms are mainly composed of consensus algorithms, gossipy algorithms, and diffusion algorithms.

Average consensus and gossip algorithms have been studied extensively in recent years, especially in distributed optimization [32, 33] and distributed estimation problems [34–36]. The consensus strategy was originally proposed in the statistics literature [37] and has since then been developed into an elegant procedure to enforce agreement among cooperating nodes. Consensus Kalman filer algorithm has been studied in [38–40] where it consists of two steps. At first step, nodes collect the measurements across the network and, at the second step, it will iterate enough over the collected data to attain agreement. Unfortunately, consensus strategy hinders the ability to perform real-time recursive estimation and adaptation when measurement data keep streaming in, which is needed in VSNs.

Besides, [41, 42] propose distributed Kalman filters where the agents communicate among themselves using the gossip protocol [34]. At every sensing time step in a gossip Kalman filter, there is a communication round between a single pair of connected sensors, chosen according to the gossip protocol, when these two sensors exchange their current state estimate and their current error covariance matrix provided by the local Kalman filter Riccati equation. Although gossip filters require very low communication bandwidth, their mean square deviation (MSD) is higher and their convergence rate is lower.

Considering problems mentioned above, in this paper, we focus on diffusion strategies, where nodes communicate with their neighbors only. There are plenty of researches on the diffusion strategies in the last more than ten years [25–28, 43–45]. The initial research is mainly focused on the application and convergence performance of some Wiener filtering algorithms in the fixed topology scenarios [43, 44, 46, 47]. In these articles, each node in the network has individual compute and sensor abilities. When nodes cooperate to estimate the performance of a target, each node will combine information from its neighbors firstly to update its own estimate to an intermediate value. After that, the node combines the intermediate value of its neighbors to obtain its update estimate again. Furthermore, in diffusion algorithms, the step-size parameters are not required to depend on the time index i and do not vanish as $i \to \infty$ compared with consensus algorithms, which implies networks with continuous adaptation and learning abilities. As the limitation of Wiener filter mentioned in Section 1, research for Kalman filter becomes increasing. In [25, 26, 45], diffusion mechanisms are used in Kalman filter, in which the problem of distributed Kalman filtering is studied. Nodes in these literatures are working directly in a linear dynamic system in a collaborative manner. The same with other diffusion algorithms, nodes in diffusion Kalman filtering algorithm communicate only with their neighbors, and the information is diffused across the network through a sequence of Kalman iterations and data-aggregations. Meanwhile, the mean and mean-square performance are analyzed to study the convergence of the diffusion Kalman filtering in the fix network topology.

In [27, 28], diffusion strategies are used in a changing topology which greatly expand the practicality in wireless

environment. Because of the instability of links, a dynamic topology model is given in these literatures, which provides a new way to study these problems. In this paper, considering dynamic topologies and distributed environment in physical layer of VSNs, we provide the diffusion Kalman filter algorithm in dynamic topologies and study the performance of its convergence through analyzing the value of mean and mean square. We will discuss this algorithm in detail in later sections.

3. System Model

In this section, we provide several fundamental models. First of all, the problem setup is described in Section 3.1. Then, in Section 3.2, the algorithm DKF (diffusion Kalman filter) given in [25] is briefly reviewed. Finally, to facilitate later analysis, the dynamic model is available in Section 3.3.

3.1. Problem Setup. In this part, we describe the dynamical system, local observations, and modeling assumptions, respectively, firstly and then give the well-known Kalman filter algorithm as the basis of the whole article.

Dynamical System. Consider tracking a moving object (assume its trajectory is an ellipse) in WSNs. We use x_i^b to define the state of this object with property b at time i, $b \in M$, of M properties (like position coordinates, velocity, and direction). To describe this problem well, we will introduce a discrete control process of the system. The system can use a linear stochastic difference equation by stack M variables in a position state vector $x_i = [x_i^1, \ldots, x_i^M]^T$, defined as

$$x_{i+1} = F_i x_i + G_i u_i + w_i, \tag{1}$$

where $F_i \in \mathbb{C}^{M\times M}$ is a sparse localized model matrix, $u_i = [u_i^1, \ldots, u_i^M] \in \mathbb{C}^M$ is the state noise vector, and $G_i \in \mathbb{C}^{M\times M}$ is the state noise matrix. The signal w_i is a deterministic input and is usually equal to zero.

Local Observations. We assume that the random field (1) is observed by N sensors, where each sensor observes only a few properties of the target. We use B_k to denote the number of properties observed by sensor k, $B_k \ll M$. Then, observations at sensor k can be represented by a linear model:

$$y_{k,i} = H_{k,i} x_i + v_{k,i}, \quad k = 1, \ldots, N, \tag{2}$$

where $y_{k,i}$ denotes measurements by sensor k at time i, $H_{k,i} \in \mathbb{C}^{B_k\times M}$ is the local observation matrix, and $v_{k,i} \in \mathbb{C}^{B_k}$ is the local observation noise to reflect the inaccuracy in measurements due to sensor precision and other unavoidable constraints.

We can get the global observation model by stacking the observations as follows:

$$y_i = \begin{bmatrix} y_{1,i} \\ \vdots \\ y_{N,i} \end{bmatrix}, \quad H_i = \begin{bmatrix} H_{1,i} \\ \vdots \\ H_{N,i} \end{bmatrix}, \quad v_i = \begin{bmatrix} v_{1,i} \\ \vdots \\ v_{N,i} \end{bmatrix}. \tag{3}$$

Then, the global observation matrix $y_i \in \mathbb{C}^{\sum_{k=1}^N B_k}$ is given by

$$y_i = H_i x_i + v_i. \tag{4}$$

Modeling assumptions are as follows.

Assumption 1. It is customary to assume that the noise signals u_i and v_i are temporally white and spatially independent with covariance matrices denoted as [28]

$$E \begin{bmatrix} u_i \\ v_i \end{bmatrix} \begin{bmatrix} u_i \\ v_j \end{bmatrix}^* = \begin{bmatrix} Q_i & 0 \\ 0 & R_i \end{bmatrix} \delta_{ij}, \tag{5}$$

where $*$ denotes conjugate transposition.

Assumption 2. We assume the initial state vector x_0, with zero mean and covariance matrix $\Pi_0 > 0$, is also temporally and spatially independent with the state and measurement noises.

Kalman Filter Algorithm. Based on the analysis above, let $\widehat{x}_{i|j}$ denote the linear minimum mean-square error estimate of x_i given observations up to and including time j. And let $P_{i|j}$ denote the covariance matrix of the estimation error $\widetilde{x}_{i|j} \triangleq x_i - \widehat{x}_{i|j}$. Therefore, the Kalman filter algorithm can start from $\widehat{x}_{0|-1} = 0$ and $P_{0|-1} = \Pi_0$ as the following equations [48]:

(1) Measurement update:

$$\begin{aligned} K_i &= P_{i|i-1} H_i^* \left(R_i + H_i P_{i|i-1} H_i^*\right)^{-1}, \\ \widehat{x}_{i|i} &= \widehat{x}_{i|i-1} + K_i \left[y_i - H_i \widehat{x}_{i|i-1}\right], \\ P_{i|i} &= P_{i|i-1} - K_i H_i P_{i|i-1}. \end{aligned} \tag{6}$$

(2) Time update:

$$\begin{aligned} \widehat{x}_{i+1|i} &= F_i x_{i|i} + u_i, \\ P_{i+1|i} &= F_i P_{i|i} F_i^* + G_i Q_i G_i^* \end{aligned} \tag{7}$$

in which K_i is the kalman filtering gain.

3.2. Algorithm DKF

Neighborhood Communication Model. In diffusion strategies, sensors exchange their current estimates of the object state

with their neighbors only to reduce communications costs. Formally, we model the communication structure with a simple, undirected, connected graph $(\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ is the set of N sensors and $\mathscr{E}$ is the set of allowable communication links among the sensors. The neighborhood of sensor k is defined by $\mathscr{N}_k \subseteq \mathscr{E}$; then, we can obtain the adjacency matrix Ω with entries $a_{k,l}$ as follows:

$$\Omega = \{a_{k,l}\} = \begin{cases} 1, & l \in \mathscr{N}_k \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

We assume that the diagonal elements of Ω are identically 1, indicating that a sensor k can always communicate with itself. At a particular instant, each sensor aggregates the observation information of its neighbors to have an accurate estimation to the target traction. However, due to the influence of the observation noise $v_{k,i}$, we provide a transfer coefficient for each neighbor to adjust the impact of noise. For definiteness, we assume the following transfer coefficient matrix, which is a left-random matrix and has a detailed description in literature [20]. Define the coefficient matrix $C = [c_{l,k}]$ with the properties:

$$\begin{aligned} &c_{l,k} \geqslant 0, \quad \sum_{l=1}^{N} c_{l,k} = 1, \\ &c_{l,k} = 0 \quad \text{if } l \notin \mathscr{N}_k \text{ for } k = 1, 2, \ldots, N. \end{aligned} \tag{9}$$

We call C the diffusion matrix, since it governs the diffusion process and plays an important role in the steady-state performance of the network. The entries in C represent the weights that are used by the diffusion algorithm to combine nearby estimates and have a variety of rules for its value. In this article, we will adopt Metropolis rule like [26] to express the set of nonnegative coefficients $c_{l,k}$. We use n_k to denote the degree of node k and we can get $c_{l,k}$ as follows:

$$\begin{aligned} c_{l,k} &= \frac{1}{\max(n_l, n_k)}, \quad \text{if } l \neq k \text{ are linked} \\ c_{l,k} &= 0, \quad \text{if } l, k \text{ not linked} \\ c_{k,k} &= 1 - \sum_{l \in \mathscr{N}_k / k} c_{l,k} \quad \text{if } l = k. \end{aligned} \tag{10}$$

The values of the weights $c_{l,k}$ may be chosen as shown in Figure 2.

The Diffusion Kalman Filter. In diffusion implementation, nodes cooperate with their direct neighbors only and diffuse the information across the entire network through a sequence of Kalman iterations and data-aggregation. In [25], diffusion Kalman filter algorithms have been proposed, which contain two steps in information exchange (local information exchange and global information exchange, resp.) and are summarized in Algorithm 1.

The algorithm has better adaptation and tracking abilities. However, in the wireless environment, the probability of link failure will increase significantly. We will discuss this situation detailedly in next section.

Consider the state-space model as in (1).
Start with $\widehat{x}_{k,0|-1} = Ex_0$
and $P_{k,0|-1} = \Pi_0$ for all k. For every time i, repeat:
Step 1. Incremental Update:

$$P_{k,i|i}^{-1} = P_{k,i|i-1}^{-1} + \sum_{l \in \mathscr{N}_k} H_{l,i}^T R_{l,i}^{-1} H_{l,i}$$

$$\phi_{k,i|i} = \widehat{x}_{k,i|i-1} + P_{k,i|i} \sum_{l \in \mathscr{N}_k} H_{l,i}^T R_{l,i}^{-1} \left(y_{l,i} - H_{l,i} \widehat{x}_{k,i|i-1}\right)$$

Step 2. Diffusion Update:

$$\widehat{x}_{k,i|i} = \sum_{l \in \mathscr{N}_k} c_{l,k} \phi_{l,i}$$

$$\widehat{x}_{k,i+1|i} = F_i \widehat{x}_{k,i|i}$$

$$P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + G_i Q_i G_i^*$$

Algorithm 1: The DKF algorithm in [25].

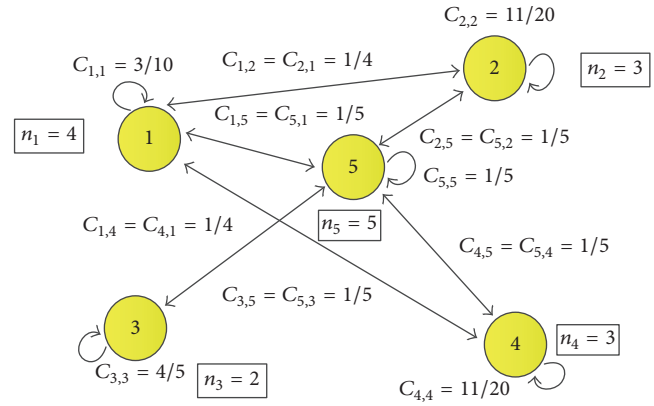


Figure 2: Values of the weights $c_{l,k}$ with Metropolis rule.

3.3. Dynamic Topology Scheme. We extend the diffusion Kalman filter to the dynamic topologies by assuming the links between any two points are dynamically linked. We assume that, at any time i, the link weight $c_{l,k}(i)$ (now randomly related to time i) will be either a nominal value $c_{l,k} = c_{k,l}$ with probability $p_{l,k} = p_{k,l}$ or zero with probability $q_{l,k}$:

$$c_{l,k}(i) = \begin{cases} c_{l,k} & \text{with } p_{lk} \\ 0 & \text{with } q_{lk} = 1 - p_{lk}. \end{cases} \tag{11}$$

For simplification, we assume an N-node network with a fixed topology. We adopt m_l to denote the maximum number of its links. Then, with different probabilities p_l, the network can be divided up to 2^{m_l} different subnetworks C_l. A simple example is shown in Figure 3, which describes a 3-node network with $m_l = 2$. For instance, the subnetwork C_2 happens with probability $p_2 = p_{12} q_{23}$.

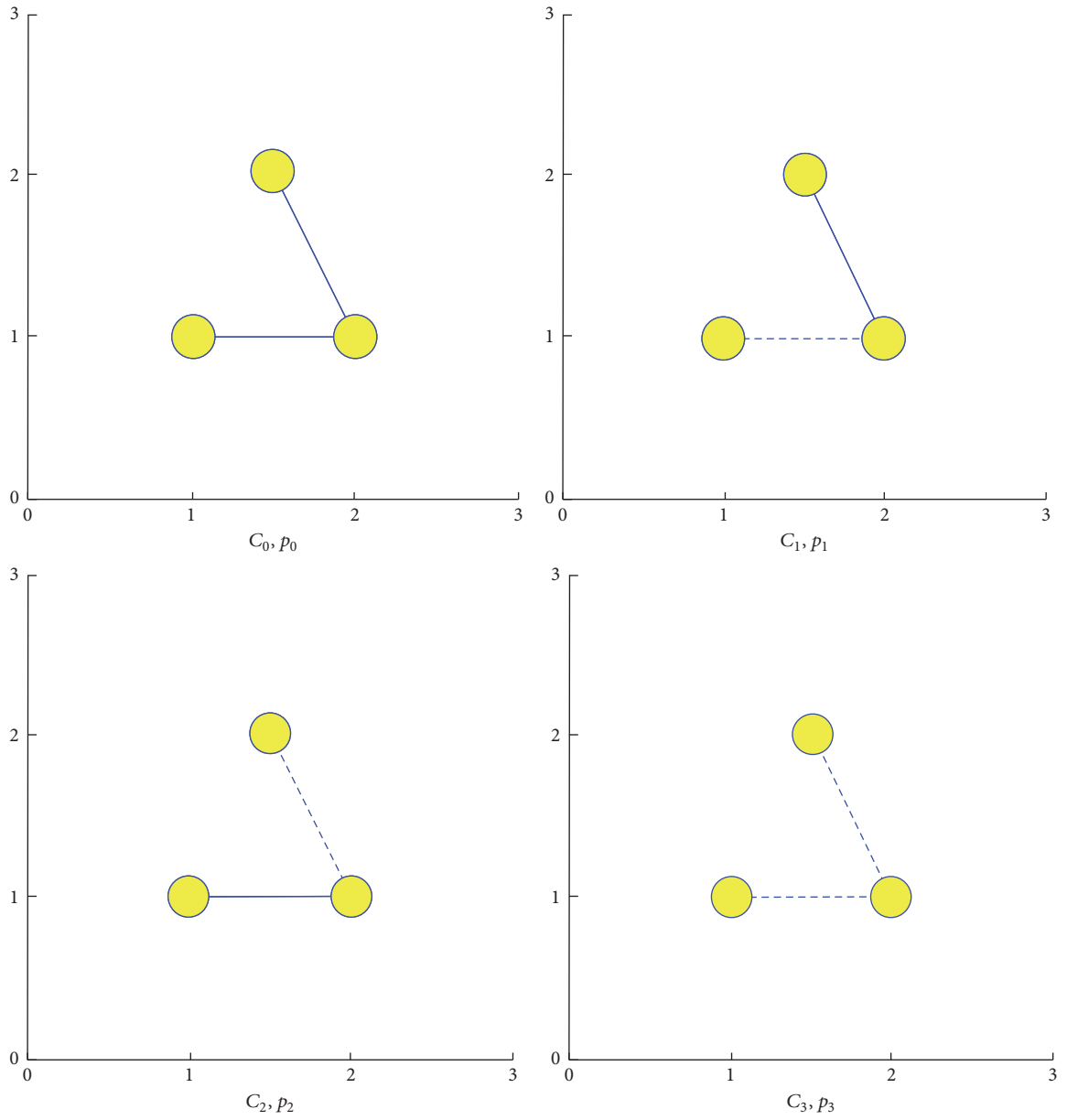


Figure 3: The subnetworks C_j with corresponding probability p_l.

In this manner, we can get the mean topology matrices $A = EA_i$ and $\mathscr{A} = E(A_i \odot A_i^*)$; namely,

$$A = \sum_{l=1}^{2^{m_l}} p_l A_l,$$
$$\mathscr{A} = \sum_{l=1}^{2^{m_l}} p_l \left(A_l \odot A_l^{*T}\right), \tag{12}$$

where $\odot$ denotes the block Kronecker product. And $A_i = C_i \otimes I_M$, in which $\otimes$ denotes Kronecker product, I_M denotes the identity matrix of size $M \times M$, and the definition of C_i is now related to time i with the form $C_i = [c_{l,k}(i)]$.

4. Analysis for DKFdt Algorithm

Based on the analysis above, we propose our diffusion Kalman filter with dynamic topologies (DKFdt) algorithm (see Algorithm 2). In our algorithm, we can see the definition of the neighbors of node k; $\mathcal{N}_{k,i}$ is now varied with time i. At the same time, the diffusion weight matrix $c_{l,k}(t)$ is also related to time compared with Algorithm 1.

Then, we divide the DKFdt algorithm into three steps. To begin with, node k collects the measurements $y_{l,i}$, the measurement matrices $H_{l,i}$, and the noise covariance matrices $R_{l,i}$ from its neighbors l at time i to update its intermediate estimate $\psi_{k,i}$. And other nodes in the whole network will do the same incremental update operations like node k.

Consider the state-space model as in (1).
Start with $\widehat{x}_{k,0|-1} = Ex_0$
and $P_{k,0|-1} = \Pi_0$ for all k. For every time i, repeat:
Step 1. Incremental Update:

$$K_{l,i} = P_{k,i|i-1}H_{l,i}^{*}\left(R_{l,i} + H_{l,i}P_{k,i|i-1}H_{l,i}^{*}\right)^{-1}$$

$$\widehat{\psi}_{k,i} = \widehat{x}_{k,i|i-1} + \sum_{l\in\mathcal{N}_{k,i}} K_{l,i}\left[y_{l,i} - H_{l,i}\widehat{x}_{k,i|i-1}\right]$$

$$P_{k,i|i} = P_{k,i|i-1} - \sum_{l\in\mathcal{N}_{k,i}} K_{l,i}H_{l,i}P_{k,i|i-1}$$

Step 2. Diffusion update with dynamic topologies:

$$\widehat{x}_{k,i|i} = \sum_{l\in\mathcal{N}_{k,i}} c_{l,k}(i)\widehat{\psi}_{l,i}$$

Step 3. Prediction update:

$$\widehat{x}_{k,i+1|i} = F_i\widehat{x}_{k,i|i}$$

$$P_{k,i+1|i} = F_iP_{k,i|i}F_i^{*} + G_iQ_iG_i^{*}$$

Algorithm 2: Diffusion Kalman filter with dynamic topologies (DKFdt).

Secondly, after all nodes complete the first step, they step into the diffusion update stage. In this stage, nodes combine the intermediate estimates produced in the first step through a coefficient matrix C_i.

The last step is the prediction update stage. The main objective of this step is to predict next state estimate values by using present values; for example, estimate value $\widehat{x}_{k,i+1|i}$ at time $i+1$ can be predicted by that at time i using the state model in (1); covariance matrix of the estimation error $P_{k,i+1|i}$ can be predicted by $P_{k,i|i}$ as well.

In the DKFdt algorithm, we combine the DKF algorithm and the dynamic topologies model together. Through this process, the network will be more close to the actual situation, where links between nodes are always instable. To prove the effectiveness of the algorithm, we need to ensure that the algorithm is convergent. So we will analyze its performance in mean and mean square to obtain its convergence behavior. These will be detailedly stated in the following sections.

5. Performance Analysis for Convergence

We now move on to examining the behavior of the DKFdt algorithm. Due to the random nature of the perturbations, it becomes necessary to evaluate the behavior of the algorithms on average. Therefore, we will study the convergence of the weight estimates in both the mean and mean-square performances.

At first, let $\widetilde{\psi}_{k,i} = x_i - \widehat{\psi}_{k,i}$ denote estimation error at the end of the incremental update. Let $\widetilde{x}_{k,i|i-1} = x_i - \widehat{x}_{k,i|i-1}$ denote estimation error at node k at the end of the diffusion update, where x_i is the real objective value. Then, we can obtain the following expressions:

$$\begin{aligned}\widetilde{\psi}_{k,i} &= x_i - \widehat{x}_{k,i|i-1} - \sum_{l\in\mathcal{N}_{k,i}} K_{l,i}\left[y_{l,i} - H_{l,i}\widehat{x}_{k,i|i-1}\right]\\ &= \widetilde{x}_{k,i|i-1} - \sum_{l\in\mathcal{N}_{k,i}} K_{l,i}\left(H_{l,i}\widetilde{x}_{k,i|i-1} + v_{l,i}\right)\\ &= \left(I - \sum_{l\in\mathcal{N}_{k,i}} K_{l,i}H_{l,i}\right)\widetilde{x}_{k,i|i-1} - \sum_{l\in\mathcal{N}_{k,i}} K_{l,i}v_{l,i},\end{aligned} \tag{13}$$

where the second equation follows (2):

$$\widetilde{x}_{k,i|i-1} = F_{i-1}\widetilde{x}_{k,i-1|i-1} + G_{i-1}n_{i-1}. \tag{14}$$

Combining (13) into the diffusion update step of Algorithm 2, we obtain

$$\begin{aligned}\widetilde{x}_{k,i|i} &= \sum_{l\in\mathcal{N}_{k,i}} c_{l,k}\widetilde{\psi}_{l,i}\\ &= \sum_{l\in\mathcal{N}_{k,i}} c_{l,k}\left[\left(I - \sum_{m\in\mathcal{N}_{l,i}} K_{m,i}H_{m,i}\right)\widetilde{x}_{l,i|i-1}\right.\\ &\quad\left. - \sum_{m\in\mathcal{N}_{l,i}} K_{m,i}v_{m,i}\right].\end{aligned} \tag{15}$$

5.1. Mean Performance. Assuming that the noise signals n_i and v_i are temporally white and spatially independent with each other, we obtain the following formulation when we take expectations of both sides of (14) and (15):

$$E\widetilde{x}_{k,i|i-1} = F_{i-1}E\widetilde{x}_{k,i-1|i-1} \tag{16}$$

$$E\widetilde{x}_{k,i|i} = E\left[\sum_{m\in\mathcal{N}_{l,i}} c_{l,k}\left(I - S_{l,i}\right)\widetilde{x}_{l,i|i-1}\right]. \tag{17}$$

Here, $S_{l,i} = \sum_{m\in\mathcal{N}_{l,i}} K_{m,i}H_{m,i}$.

Since $E\widetilde{x}_{k,0|-1} = 0$ and $E\widetilde{x}_{k,-1|-1} = 0$ as defined in Section 2, (17) indicates the mean performance of the DKFdt algorithm is unbiased.

5.2. Mean-Square Stability. To analyze mean-square stability, we will introduce filter mean square deviation (MSD) as the steady-state measure for node k [20]:

$$\mathrm{MSD}_{k,i} = E\left\|x_i - \widehat{x}_{k,i|i}\right\|^2. \tag{18}$$

Then, we give the definition of the global vectors as follows:

$$\begin{aligned}\overline{\mathcal{X}}_{i|i} &\triangleq \mathrm{col}\left\{\widetilde{x}_{1,i|i},\ldots,\widetilde{x}_{N,i|i}\right\},\\ \mathcal{K}_i &\triangleq \mathrm{diag}\left\{K_{1,i},\ldots,K_{N,i}\right\},\\ \mathcal{H}_i &\triangleq \mathrm{diag}\left\{H_{1,i},\ldots,H_{N,i}\right\},\\ \mathcal{S}_i &\triangleq \mathrm{diag}\left\{S_{1,i},\ldots,S_{N,i}\right\}.\end{aligned} \tag{19}$$

Consider the matrices $\mathcal{A}_i = C_i \otimes I_M$. We may now express (14) and (15) in a global form:

$$\overline{\mathcal{X}}_{i|i} = A_i^{T}\left[\mathcal{F}_i^{C}\overline{\mathcal{X}}_{i-1|i-1} + \mathcal{G}_i^{C}\left(\mathbf{1}\otimes n_{i-1}\right) - \mathcal{D}_i^{C}v_i\right]. \tag{20}$$

Here,

$$\begin{aligned}\mathcal{F}_i^{C} &\triangleq \left(I - \mathcal{S}_i\right)\left(I\otimes\mathcal{F}_{i-1}\right),\\ \mathcal{G}_i^{C} &\triangleq \left(I - \mathcal{S}_i\right)\left(I\otimes G_{i-1}\right),\\ \mathcal{D}_i^{C} &\triangleq \mathcal{K}_iR_i^{-1},\end{aligned} \tag{21}$$

where T denotes the transpose of matrix A_i, $R_i = Ev_i v_i^*$ is a block-diagonal matrix with v_i defined in (4), and $\mathbf{1}$ is the vector of size $N \times 1$ and whose entries are all equal to one.

Let $\mathscr{P}_{\widetilde{\mathscr{X}},i} = E\widetilde{\mathscr{X}}_{i|i}\widetilde{\mathscr{X}}_{i|i}^*$ denote the covariance matrix of $\widetilde{\mathscr{X}}_{i|i}$. Then, the recursion can be obtained by (21) as follows:

$$\mathscr{P}_{\widetilde{\mathscr{X}},i} = EA_i^T \left[\mathscr{F}_i^C \mathscr{P}_{\widetilde{\mathscr{X}},i-1} \mathscr{F}_i^{C*} + \mathscr{G}_i^C \left(\mathbf{1}\mathbf{1}^T \otimes Q_{i-1}\right) \mathscr{G}_i^{C*} + E\left(\mathscr{D}_i^C R_i \mathscr{D}_i^{C*}\right)\right] EA_i. \quad (22)$$

In order to analyze the mean-square stability, we also adopt the same assumption in [26] as follows.

Assumption 3. The matrices in model (1) are time-invariant; that is, the matrices F, G, H, R, and Q do not depend on i.

Assumption 4. A Kalman filter that uses data from a neighborhood converges for every neighborhood (see [49] for conditions on Kalman filter convergence).

Based on the assumptions, the matrix (21) can be denoted as the steady-state value as follows:

$$\begin{aligned} \mathscr{F}^C &\triangleq \lim_{i\to\infty} \mathscr{F}_i^C = (I - \mathscr{S})(I \otimes F), \\ \mathscr{G}^C &\triangleq \lim_{i\to\infty} \mathscr{G}_i^C = (I - \mathscr{S})(I \otimes G), \\ \mathscr{D}^C &\triangleq \lim_{i\to\infty} \mathscr{D}_i^C = \mathscr{K} R^{-1}. \end{aligned} \quad (23)$$

Then, (22) can be denoted as follows:

$$\mathscr{P}_{\widetilde{\mathscr{X}}} = EA_i^T \left[\mathscr{F}^C \mathscr{P}_{\widetilde{\mathscr{X}}} \mathscr{F}^{C*} + \mathscr{G}^C \left(\mathbf{1}\mathbf{1}^T \otimes Q\right) \mathscr{G}^{C*} + E\left(\mathscr{D}^C R \mathscr{D}^{C*}\right)\right] EA_i. \quad (24)$$

From (12) and the temporally and spatially independent assumptions on the state and measurement noises, (24) converges to the unique solution of the Lyapunov equation [50]:

$$\mathscr{P}_{\widetilde{\mathscr{X}},i} = \mathscr{F}\mathscr{P}_{\widetilde{\mathscr{X}},i-1}\mathscr{F}^* + \mathscr{G}\left(\mathbf{1}\mathbf{1}^T \otimes Q\right)\mathscr{G}^* + \mathscr{D}R\mathscr{D}^*. \quad (25)$$

Here,

$$\begin{aligned} \mathscr{F} &\triangleq A^T F_i^C = A^T (I - \mathscr{S}_i)(I \otimes \mathscr{F}_{i-1}), \\ \mathscr{G} &\triangleq A^T G_i^C = A^T (I - \mathscr{S}_i)(I \otimes G_{i-1}), \\ \mathscr{D} &\triangleq A^T E D_i^C = A^T \mathscr{K}_i R_i^{-1}. \end{aligned} \quad (26)$$

Now, based on the DKFdt algorithm, we can get the same solution from the analysis of the mean-square stability as the existing DKF algorithms. Then, we also express the solution by using the operator, which is obtained by stacking the columns of a matrix. Then, (22) can be written as

$$\operatorname{vec}\left(\mathscr{P}_{\widetilde{\mathscr{X}}}\right) = \left(I - \mathscr{F}^{*T} \otimes \mathscr{F}\right)^{-1} \cdot \operatorname{vec}\left[\mathscr{G}\left(\mathbf{1}\mathbf{1}^T \otimes Q\right)\mathscr{G}^* + \mathscr{D}R\mathscr{D}^*\right]. \quad (27)$$

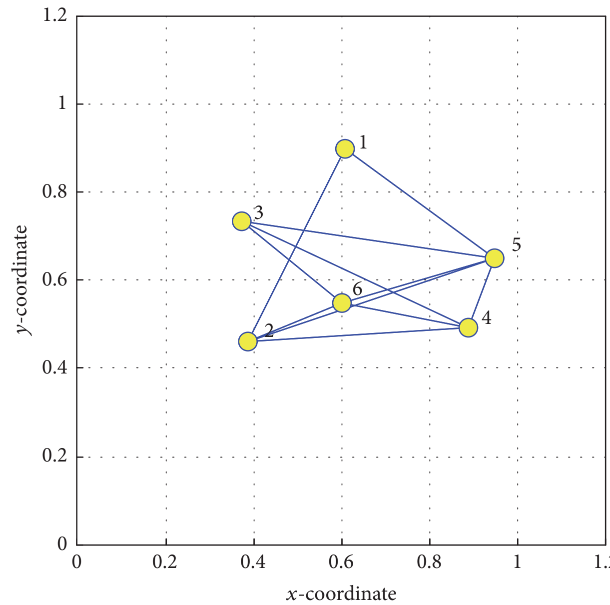


FIGURE 4: Network topology with $N = 6$.

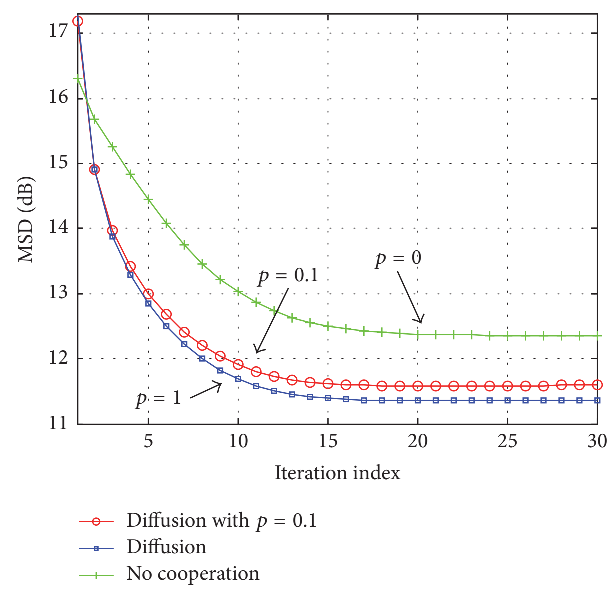


FIGURE 5: Global MSD performance for $p = 0$ (no cooperation), $p = 0.1$ (DKFdt), and $p = 1$ (standard diffusion) with $N = 6$.

Here, we use the property that

$$\operatorname{vec}(P\Sigma Q) = \left(Q^T \otimes P\right)\operatorname{vec}(\Sigma). \quad (28)$$

According to (27) and using the following equalities for arbitrary matrices $\{U, W, \sigma\}$ of compatible dimensions,

$$\operatorname{Tr}(\Sigma W) = \left[\operatorname{vec}\left(W^T\right)\right]^T \sigma, \quad (29)$$

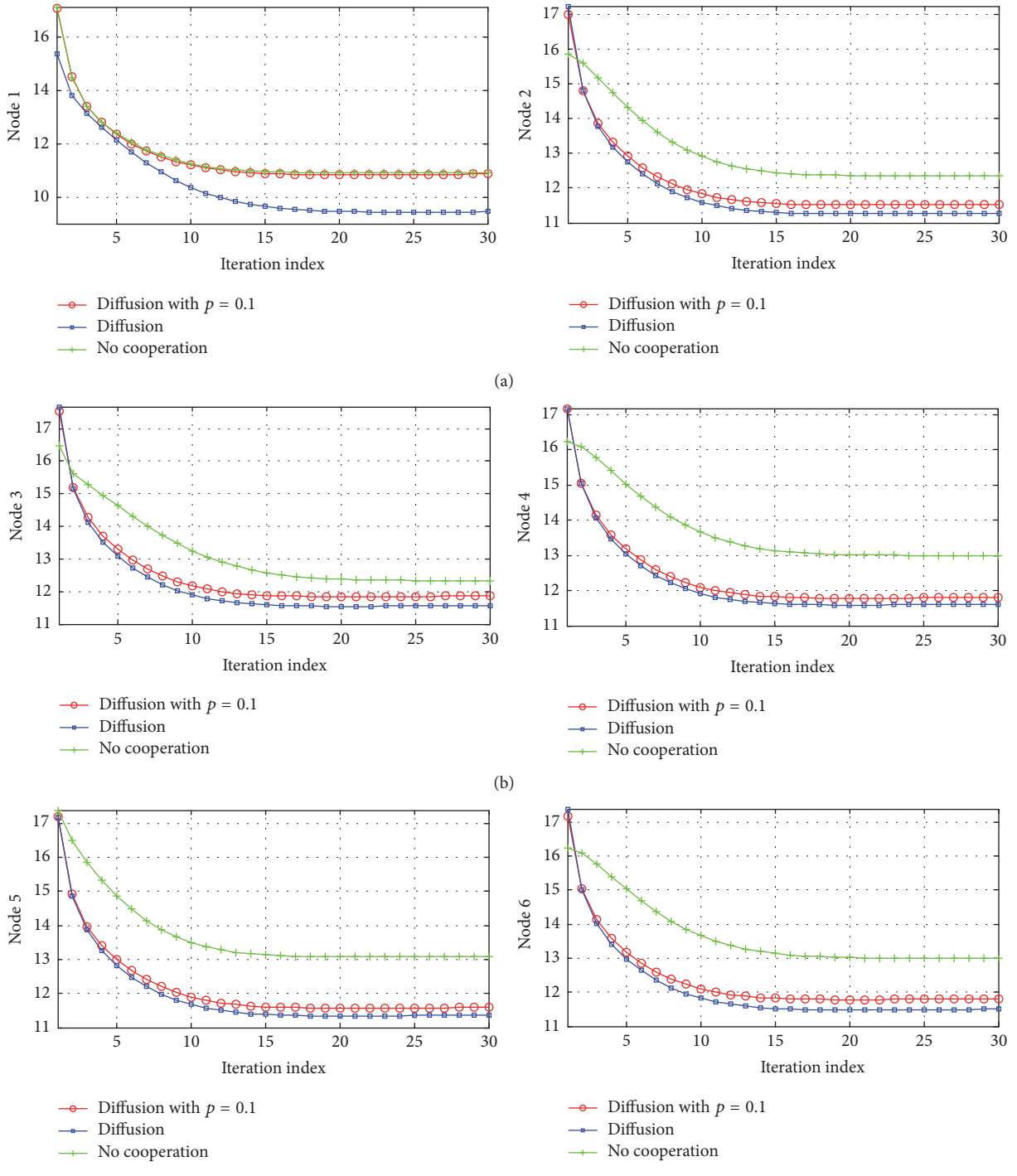


FIGURE 6: MSD per node (local).

the steady-state MSD at node k and the average steady-state MSD across the network may now be expressed as

$$\begin{aligned} \mathrm{MSD}_k &= \lim_{i\to\infty} E\left\|x_i - \widehat{x}_{k,i|i}\right\|^2 = \mathrm{Tr}\left(\mathscr{P}_{\widetilde{x}}\mathscr{I}_k\right), \\ \mathrm{MSD}^{\mathrm{ave}} &= \frac{1}{N}\mathrm{Tr}\left(\mathscr{P}_{\widetilde{x}}\right), \end{aligned} \tag{30}$$

where $\mathscr{I}_k = B_{N\times N} \otimes I_M$ and $B_{N\times N}$ is an $N \times N$ dimension matrix with all unit entries and I_M is the M dimension unit matrix.

Then, we can draw a conclusion that the DKFdt algorithm is unbiased and stability based on analysis of the mean and mean-square performance.

6. Number Evaluation

In this section, we give the simulation to illustrate the performance of the proposed algorithm. Consider the case that a set of sensors in WSNs attempts to track the trajectory of an ellipse object. We numerically evaluate the MSD performance of the DKFdt and compare it against the noncooperate Kalman filter (KF) and the diffusion Kalman filter (DKF) in [25]. To achieve this objective, we build a time-varying random system.

For simplicity, at first we use a small-scale network model to verify our algorithm. This network consists of six sensors with the topology shown in Figure 4, in which the connection lines between nodes only represent the information communication with two agents rather than the real link. At the same time, each sensor node can independently obtain the target status and communicate with its neighbors.

In our simulation, the matrices in (1) are

$$F = \begin{bmatrix} 0.992 & -0.1247 \\ 0.1247 & 0.992 \end{bmatrix},$$
$$Q = I_2,$$
$$G = \begin{bmatrix} 0.625 & 0 \\ 0 & 0.625 \end{bmatrix}, \quad (31)$$
$$R_{k,i} = 10\sqrt{k}, \quad k = 1, \ldots, N$$

and the initial conditions are

$$x_0 = (15, -10)^T,$$
$$P_0 = 10I_2. \quad (32)$$

Because the nodes take measures of the unknown position of the object, the measurement matrix $H_{k,i}$ is chosen to be either $H_{k,i} = H_x = (1, 0)$ or $H_{k,i} = H_y = (0, 1)$.

In Figure 5, we present the global MSD evolution for $p = 1$ (Algorithm 1), $p = 0.1$ (DKFdt, Algorithm 2), and $p = 0$ (without cooperation), where variable p represent the probability of the connection between any two points defined as $p_{k,l}$ or $p_{l,k}$ in Section 3.3. In the figure, the x-coordinate represents the iteration index and the y-coordinate represents the MSD value. From it, we can see that, with the passage of time, all the algorithm can converge to a limited range with different convergence speed and MSD values. Through comparison and analyzing the simulation among three different algorithms, we can conclude that when $p = 0.1$, which indicates the low utilization of the links, there is still a significant improvement over the situation without cooperation. These results confirm advantages of diffusion strategies for adaptive networks. Moreover, in order to embody the characteristics of our algorithm, we illustrate the performance of MSD of each node. As shown in Figure 6, although performance of convergence in each node is not different because of the effect of noise, they all can reach convergence through communication with each other.

In order to further illustrate the characteristics of our algorithm, we simulate another example in a larger network topology with $N = 20$ in Figure 7 and other parameters remain the same with that of six-point topology network. In Figure 8, we take the same simulation to illustrate the performance of MSD with different probability of link connections. We have observed that values of diffusion algorithm with $p = 0.1$ are closer with diffusion algorithm in reliable link (with $p = 1$) than that in six-point network, which reflect that collaboration among more sensors can reduce the effect on instability of links.

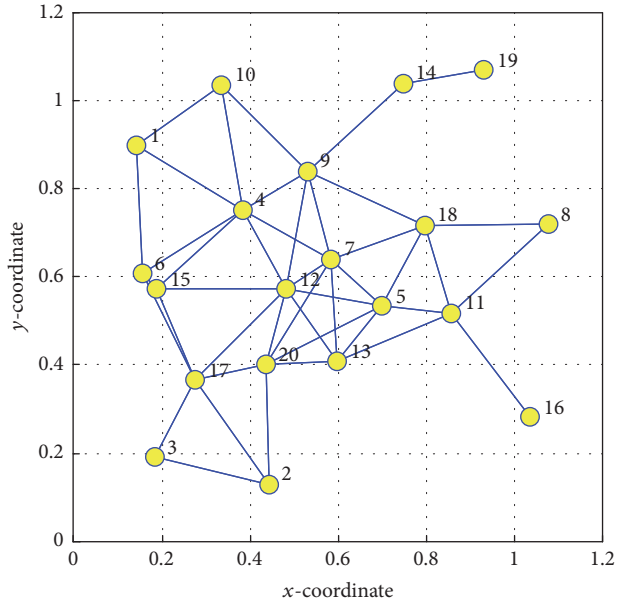


Figure 7: Network topology with $N = 20$.

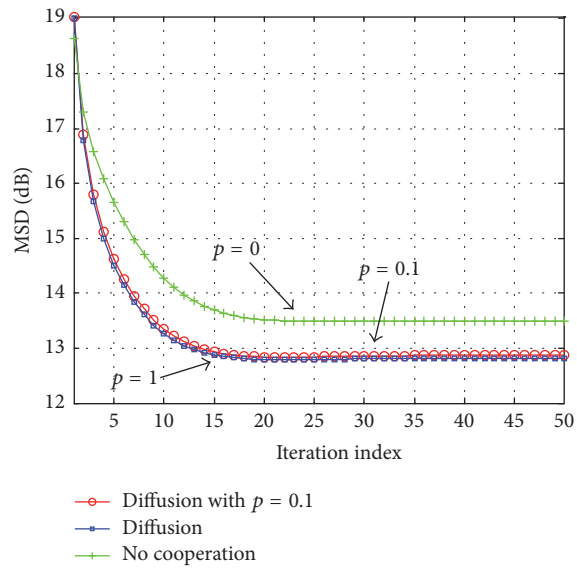


Figure 8: Global MSD performance for $p = 0$ (no cooperation), $p = 0.1$ (DKFdt), and $p = 1$ (standard diffusion) with $N = 20$.

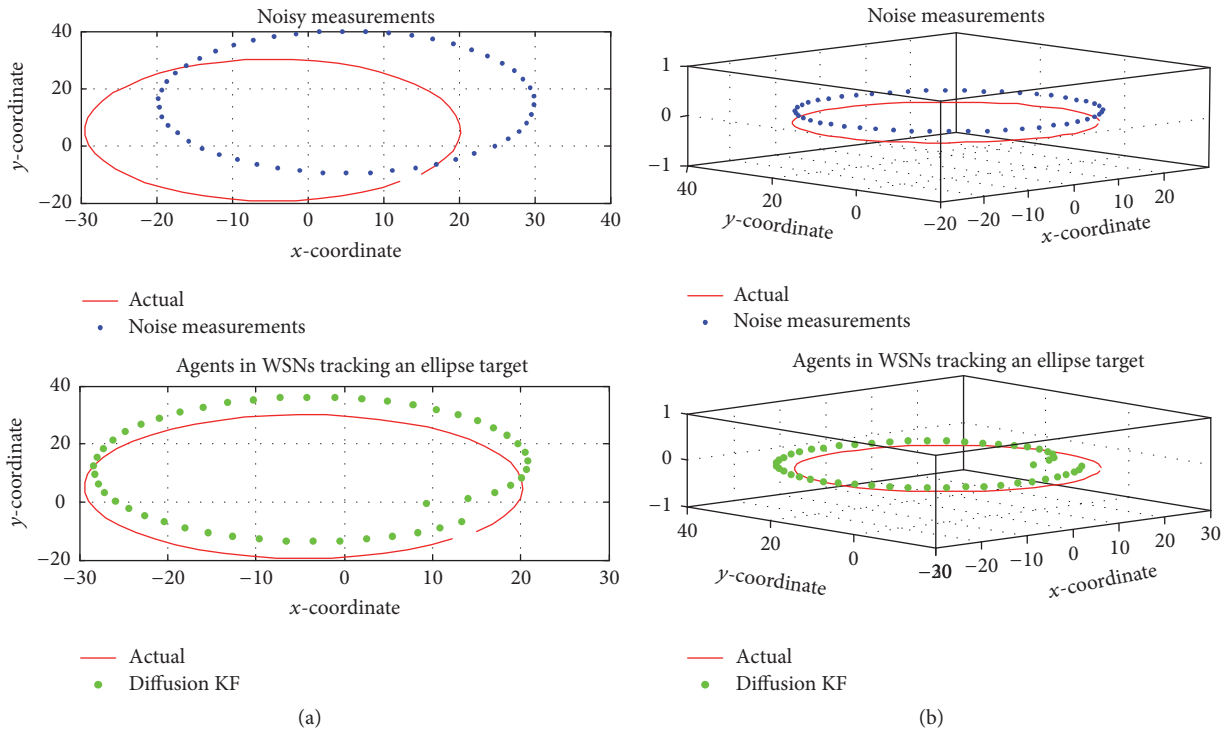


FIGURE 9: (a) Noisy measurement and tracking an ellipse target. (b) Noisy measurement and tacking the position of a moving object in 3D space in WSNs.

At last, we demonstrate the result of the nodes cooperating together to track the position of an ellipse object in Figure 9. And we show the result through 2D and 3D model, respectively. As we can see, the networks still have a good performance in estimating the position of the target through nodes diffusing information with each other continuously, although nodes are affected greatly by noises.

7. Conclusion and Further Work

In this paper, we provide a diffusion Kalman filter algorithm with dynamic topologies (DKFdt) to achieve this requirement in physical layer of VSNs. We also analyze the mean and the mean-square performance in dynamic scenarios. The result shows that the global estimation error can converge to a unique solution well. Through the simulation, our proposed diffusion strategy with cooperative method performs well in dynamic networks. It achieves an improved performance over the noncooperative strategy ($p = 0$) even at a high dynamic environment with low link probability ($p = 0.1$).

Although the algorithm is provided under the background of VSNs, its applications are not confined to this field. Based on what is mentioned above, we will further study other problems on the distributed Kalman filter algorithms and, at the same time, their applications in other fields are also the emphasis of our studies.

Notation

$\mathbb{R}$: The real domain
$\mathbb{C}$: The complex domain.

Key Symbols

x_i: State vector of object at time i
$y_{k,i}$: Observation value at sensor k at time i
F_i: Dynamical system matrix
$H_{k,i}$: Observation matrix at sensor k
Q_i: System noisy covariance
R_i: Observation noise covariance
K_i: Kalman filtering gain
$P_{i|j}$: The covariance matrix of the estimation error
ω: The adjacency matrix
C: The diffusion left-random matrix
C_i: The diffusion left-random matrix varied with time i
A: The mean topology matrix.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by the National Science and Technology Major Project (Grant no. 2015ZX03003002-002), in part by the National Natural Science Foundation of China (NSFC) under Grant nos. 61522103, 61372112, and 61232017, in part by the National Key Basic Research Program of China (973 Program) under Grant 2013CB329102, and in part by the Beijing Natural Science Foundation under Grant 4142037.

References

[1] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean, "CMT-QA: quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2193–2205, 2013.

[2] C. Xu, Z. Li, J. Li, H. Zhang, and G.-M. Muntean, "Cross-layer fairness-driven concurrent multipath video delivery over heterogeneous wireless networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 7, pp. 1175–1189, 2015.

[3] S. Barath Kumar and G. Myilsamy, "Multi-target tracking in mobility sensor networks using Ant Colony Optimization," in *Proceedings of the International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN '13)*, pp. 350–354, Tirunelveli, India, March 2013.

[4] M. Z. Bhuiyan, G. Wang, and A. V. Vasilakos, "Local area prediction-based mobile target tracking in wireless sensor networks," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 1968–1982, 2015.

[5] E. T. Yazdi, A. Moravejosharieh, and S. K. Ray, "Study of target tracking and handover in mobile wireless sensor network," in *Proceedings of the 28th International Conference on Information Networking (ICOIN '14)*, pp. 120–125, Phuket, Thailand, February 2014.

[6] C. Türkay, C. G. Aksakallı, K. Özmal, and H. K. Ekenel, "A mobile plant identification application for environmental monitoring," in *Proceedings of the 21st Signal Processing and Communications Applications Conference (SIU '13)*, pp. 1–4, April 2013.

[7] M. DArienzo, M. Iacono, S. Marrone, and R. Nardone, "Estimation of the energy consumption of mobile sensors in WSN environmental monitoring applications," in *Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA '13)*, pp. 1588–1593, IEEE, Barcelona, Spain, March 2013.

[8] C. Xu, F. Zhao, J. Guan, H. Zhang, and G.-M. Muntean, "QoE-driven user-centric vod services in urban multihomed P2P-based vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2273–2289, 2013.

[9] C. Xu, S. Jia, L. Zhong, H. Zhang, and G.-M. Muntean, "Ant-inspired mini-community-based solution for video-on-demand services in wireless mobile networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 322–335, 2014.

[10] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang, and G.-M. Muntean, "Performance-aware mobile community-based VoD streaming over vehicular Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1201–1217, 2015.

[11] D. P. Mishra, G. M. Asutkar, and S. S. Dorale, "An application of wireless sensor network in intelligent transportation system," in *Proceedings of the 6th International Conference on Emerging Trends in Engineering and Technology (ICETET '13)*, pp. 90–91, Maharashtra, India, December 2013.

[12] Z. Rasin, H. Hamzah, and M. S. M. Aras, "Application and evaluation of high power Zigbee based wireless sensor network in water irrigation control monitoring system," in *Proceedings of the IEEE Symposium on Industrial Electronics & Applications (ISIEA '09)*, vol. 2, pp. 548–551, Kuala Lumpur, Malaysia, October 2009.

[13] G. Wu, C. Wu, and S. Xu, "Application of wireless sensor network in the monitoring system of boiler," in *Proceedings of the 26th Chinese Control and Decision Conference (CCDC '14)*, pp. 1751–1754, IEEE, Changsha, China, May 2014.

[14] W. Jing and L. Tingting, "Application of wireless sensor network in Yangtze River basin water environment monitoring," in *Proceedings of the 27th Chinese Control and Decision Conference (CCDC '15)*, pp. 5981–5985, IEEE, Qingdaob, China, May 2015.

[15] S. Rajba, P. Raif, T. Rajba, and M. Mahmud, "Wireless sensor networks in application to patients health monitoring," in *Proceedings of the IEEE Symposium on Computational Intelligence in Healthcare and e-health (CICARE '13)*, pp. 94–98, Singapore, April 2013.

[16] N. M. Hassan, O. M. Olaniyi, A. Ahmed, and E. M. Dogo, "Wireless sensor networks for remote healthcare monitoring in Nigeria: challenges and way forward," in *Proceedings of the 2nd IEEE International Conference on Emerging and Sustainable Technologies for Power and ICT in a Developing Society (IEEE NIGERCON '13)*, pp. 182–187, Owerri, Nigeria, November 2013.

[17] C. Liang and F. R. Yu, "Wireless network virtualization: a survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 358–380, 2015.

[18] I. Khan, T. SudParis, F. Belqasmi, and R. Glitho, "Wireless sensor network virtualization: a survey," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 553–576, 2015.

[19] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: early architecture and research perspectives," *IEEE Network*, vol. 29, no. 3, pp. 104–112, 2015.

[20] M. S. E. Abadi and Z. Saffari, "Distributed estimation over an adaptive diffusion network based on the family of affine projection algorithms," in *Proceedings of the 2012 6th International Symposium on Telecommunications (IST '12)*, pp. 607–611, Tehran, Iran, November 2012.

[21] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[22] Y. J. Chun, M. O. Hasna, and A. Ghrayeb, "Adaptive network coding for spectrum sharing systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 639–654, 2015.

[23] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.

[24] Y. Huo, Z. Cai, W. Gong, and Q. Liu, "A new adaptive Kalman filter by combining evolutionary algorithm and fuzzy inference system," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '14)*, pp. 2893–2900, July 2014.

[25] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, 2010.

[26] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering: formulation and performance analysis," in *Proceedings of the Cognitive Information Processing*, Santorini, Greece, June 2008.

[27] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 3285–3288, IEEE, Las Vegas, Nev, USA, March-April 2008.

[28] L. Li and J. A. Chambers, "Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology," in *Proceedings of the IEEE/SP 15th Workshop on Statistical Signal Processing (SSP '09)*, pp. 757–760, Cardiff, UK, September 2009.

[29] I. Khan, F. Belqasmi, R. Glitho, and N. Crespi, "A multi-layer architecture for wireless sensor network virtualization," in

Proceedings of the 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC '13), pp. 1–4, IEEE, Dubai, UAE, April 2013.

[30] C. Delgado, J. R. Gallego, M. Canales, J. Ortin, S. Bousnina, and M. Cesana, "An optimization framework for resource allocation in virtual sensor networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '15)*, pp. 1–7, IEEE, San Diego, Calif, USA, December 2015.

[31] I. Khan, "Design and analysis of virtualization framework for wireless sensor networks," in *Proceedings of the IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '13)*, pp. 1–2, IEEE, Madrid, Spain, June 2013.

[32] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[33] A. Nedic and A. Ozdaglar, "Cooperative distributed multi-agent optimization," in *Convex Optimization in Signal Processing and Communications*, Y. Eldar and D. Palomar, Eds., pp. 340–386, University of Cambridge, Cambridge, UK, 2009.

[34] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[35] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: fundamental limits and tradeoffs," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 674–690, 2011.

[36] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

[37] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[38] H. Long, Z. Qu, X. Fan, and S. Liu, "Distributed extended Kalman filter based on consensus filter for wireless sensor network," in *Proceedings of the 10th World Congress on Intelligent Control and Automation (WCICA '12)*, pp. 4315–4319, Beijing, China, July 2012.

[39] R. Grasso, P. Braca, S. Fortunati, and F. Gini, "Distributed underwater glider network with consensus kalman filter for environmental field estimation," in *Proceedings of the OCEANS*, pp. 1–7, Genova, Italy, May 2015.

[40] R. Olfati-Saber, "Kalman-Consensus filter: optimality, stability, and performance," in *Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 7036–7042, Chinghai, China, 2009.

[41] S. Kar and J. M. F. Moura, "Gossip and distributed Kalman filtering: weak consensus under weak detectability," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1766–1784, 2011.

[42] D. Li, S. Kar, J. M. F. Moura, H. V. Poor, and S. Cui, "Distributed Kalman filtering over massive data sets: analysis through large deviations of random Riccati equations," *IEEE Transactions on Information Theory*, vol. 61, no. 3, pp. 1351–1372, 2015.

[43] R. Abdolee and B. Champagne, "Diffusion LMS strategies in sensor networks with noisy input data," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 3–14, 2016.

[44] S. Vlaski, L. Vandenberghe, and H. Ali, "Diffusion stochastic optimization with non-smooth regularizers," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '16)*, pp. 4149–4153, March 2016.

[45] F. S. Cattivelli and A. H. Sayed, "Diffusion mechanisms for fixed-point distributed Kalman smoothing," in *Proceedings of the 16th European Signal Processing Conference (EUSIPCO '08)*, pp. 1–4, Lausanne, Switzerland, August 2008.

[46] Y. Wang, W. P. Tay, and W. Hu, "An energy-efficient diffusion strategy over adaptive networks," in *Proceedings of the 10th International Conference on Information, Communications and Signal Processing (ICICS '15)*, pp. 1–5, Singapore, December 2015.

[47] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, 2015.

[48] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 5492–5498, New Orleans, La, USA, December 2007.

[49] F. S. Cattivelli and A. H. Sayed, "Diffusion mechanisms for fixed point distributed Kalman smoothing," in *Proceedings of the 2008 16th European Signal Processing Conference (EUSIPCO '08)*, Lausanne, Switzerland, August 2008.

[50] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, New Jersey, NJ, USA, 2000.

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 5279425, 12 pages
http://dx.doi.org/10.1155/2016/5279425

Research Article

A Novel Mobile Video Community Discovery Scheme Using Ontology-Based Semantical Interest Capture

Ruiling Zhang[1,2] and Shengwu Xiong[1]

[1]*School of Computer Science and Technology, Wuhan University of Technology, Wuhan, Hubei 430070, China*
[2]*Academy of Information Technology, Luoyang Normal University, Luoyang 471934, China*

Correspondence should be addressed to Ruiling Zhang; ruilingzhang@163.com

Received 1 July 2016; Accepted 14 September 2016

Academic Editor: Gabriel-Miro Muntean

Leveraging network virtualization technologies, the community-based video systems rely on the measurement of common interests to define and steady relationship between community members, which promotes video sharing performance and improves scalability community structure. In this paper, we propose a novel mobile Video Community discovery scheme using ontology-based semantical interest capture (VCOSI). An ontology-based semantical extension approach is proposed, which describes video content and measures video similarity according to video key word selection methods. In order to reduce the calculation load of video similarity, VCOSI designs a prefix-filtering-based estimation algorithm to decrease energy consumption of mobile nodes. VCOSI further proposes a member relationship estimate method to construct scalable and resilient node communities, which promotes video sharing capacity of video systems with the flexible and economic community maintenance. Extensive tests show how VCOSI obtains better performance results in comparison with other state-of-the-art solutions.

1. Introduction

The video streaming services are very popular multimedia applications in the Internet (e.g., the video traffic is now more than two-thirds of the network traffic) [1]. The convenient access of video content via mobile smart devices further promotes the development of video applications with the help of mobile networking technologies and increasing wireless bandwidth [2]. The huge traffic demand brings heavy load for the video systems, which requires scalable architecture and high-efficiency resource sharing to support large-scale deployment. Peer-to-Peer (P2P)/Mobile Peer-to-Peer (MP2P) technologies construct virtual networks in terms of resource demand of (mobile) users to manage and distribute resources [3–7]. The continuous increase in the number of nodes in video systems not only triggers huge traffic demand for video content but also increases the maintenance cost of overlay networks due to dynamic playback state of nodes. However, the traditional P2P-based video systems make use of the request-response relationship between nodes to construct overlay networks. The fragile relationship between nodes is one of the main reasons for the frequent disconnection of logical links between them. Because the predefined relationship neglects the investigation for the resource supply and demand, it is difficult to achieve high-efficiency resource sharing and low-cost maintenance for overlay networks. This leads to the low quality of service (QoS) such as long startup delay and low system scalability.

Inspired by the investigation of relationship between nodes in social networks, the virtual community-based video systems rely on the measurement results of common interests for the video content to define relationship between community members [8–11]. As Figure 1 shows, the whole overlay network is divided into multiple communities composed of nodes with similar interests. The similar interests not only steady the logical link between community members to reduce maintenance cost for overlay networks, but also achieve efficient resource sharing to promote QoS of video systems. A key issue is how to accurately estimate the similarity of interests between nodes. The videos watched by users can reflect user interests; namely, the two users which watch the same or similar videos can be considered as having

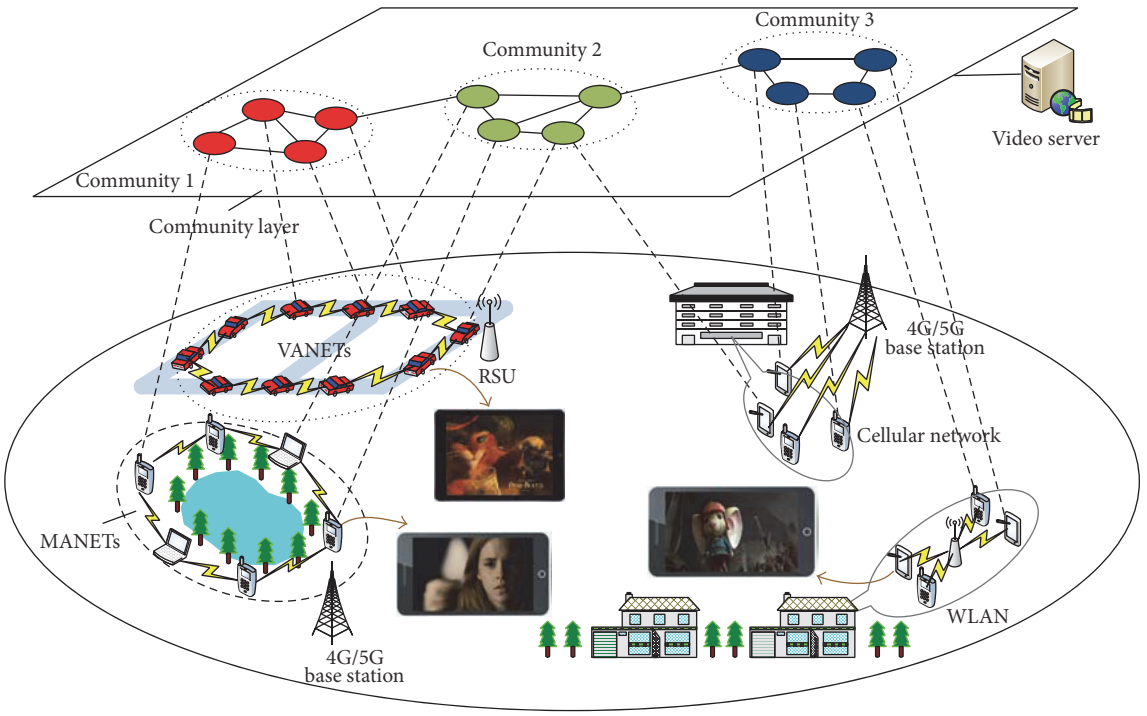


Figure 1: Multimedia streaming services in wireless mobile networks.

common interests. The video similarity estimation in the traditional methods only investigates file name of videos in historical playback traces of users. The file name difficultly reflects the whole video content, which cannot ensure estimation accuracy of interest similarity. The low accuracy of estimation results leads to fragile logical link between community members, which still triggers the frequent construction of community structure.

The description of video content is a key factor for the measurement accuracy of video similarity, which usually includes video name, title, type, actors, director, and abstract. The expanded video information is described by the short text. However, the noise words and valid data sparseness in the short text bring negative influence for the estimation accuracy, such as advertisement words in video abstract. A key issue is how to build a rich semantic-based description space for video content to efficiently promote estimation accuracy of video similarity. On the other hand, the expanded description brings more video information, which constructs the high-dimensional similarity measurement space composed of massive feature words. The video similarity calculation with high time complexity consumes large amount of energy of mobile nodes. Another key issue is how to design a light-duty estimation method to reduce calculation load of mobile nodes.

In this paper, we propose a novel mobile Video Community discovery scheme using ontology-based semantical interest capture (VCOSI). VCOSI employs an ontology-based semantical extension approach to describe video content and adopts three classification methods to select key words in the extended description of video content. VCOSI designs an estimation method to calculate similarity of video content by making use of the selected key words. In order to reduce calculation complexity of similarity, VCOSI further proposes a prefix-filtering-based estimation algorithm to reduce the calculation load of mobile nodes. Based on the estimation results of video similarity, VCOSI measures closeness levels of common interests between nodes and builds scalable and resilient community structure, which promotes video sharing capacity of video systems. Simulation results show how VCOSI achieves much better performance results in comparison with other state-of-the-art solutions.

2. Related Work

The traditional P2P/MP2P-based video systems mainly employ structured and unstructured overlay networks to distribute video resources. The systems based on structured overlay networks, such as tree and Chord, can achieve fast resource location but need to consume large number of resources of bandwidth and computation to maintain overlay networks due to dynamic node state, which results in low scalability. The systems based on structured overlay networks do not handle real-time variation of node state, which obtains high scalability. However, the flooding-based resource lookup method wastes massive network bandwidth, which leads to network congestion and high startup delay.

The community-based video systems group the nodes into multiple communities in terms of closeness levels of relationship between them. Each community undertakes the tasks of state maintenance of intracommunity members; namely, the maintenance cost for the whole overlay networks

is distributed into multiple communities, which improves system scalability. Moreover, the tight relationship enables logical links between community members to become more stable, which further reduces maintenance cost of community structure. On the other hand, the community members store similar video resources and request similar video content with high probability in the future, so the video request of members is quickly responded by other intracommunity members, which reduces lookup delay.

SocialTube measures relationship between nodes by the investigation of number of watched videos [8]. The more the watched same videos are, the more similar the interests between nodes are. The nodes with similar interests form a community. The source nodes push the interested videos to the community members. However, it is difficult to obtain accurate measurement results of similarity by making use of number of watched videos to estimate interest similarity. This results in the low video push success rate and fragile logical link between nodes. SPOON also relies on investigating file name vector in historical playback traces to capture common interests between nodes [9]. The mobile nodes with similar interests are grouped into communities. The designed role assignment method distributes the maintenance cost for the community structure into multiple community members, which balances the load of mobile nodes and improves the community scalability. However, the interest similarity measurement based on file name difficultly ensures the estimation accuracy of common interests between nodes.

In order to address the problems of low estimation accuracy in traditional methods, some common interest capture methods expand the description of video content (e.g., video name, title, type, actors, director, and abstract). The expanded description enriches video information and is denoted by the short text, which increases the measurement accuracy of similarity between videos. However, the short text includes large number of noise words, which severely influences measurement accuracy of video similarity. The existing studies make use of semantic dictionary, feature extension, and topic model to address the problems of noise words.

(1) Semantic Dictionaries. The methods based on semantic dictionaries, such as WordNet and HowNet, can accurately estimate similarity of word pairs by the conversion from short text to word pairs. Mihalcea et al. [12] considered the weighted average values of similarities among word pairs as the similarities of short text. Li et al. [13] combine semantic and sequence similarity to calculate similarity of short text. Based on the work in [13], Islam and Inkpen [14] make use of the weighted sum of string similarity, semantic similarity, and sequence similarity to calculate similarity of short text. However, the semantic dictionary always is incomplete due to the continuous occurrence of new words, which cannot ensure high-accuracy similarity measurement results of short text.

(2) Feature Extension. Sahami and Heilman employed a similarity kernel function to estimate the short text similarity by making use of the search engine to extend features of short text [15]. Wang et al. [16] and Yuan [17] proposed a mining algorithm based on association rule to extract association relationship between features included in training and testing sets, which further obtains the extended features corresponding to the words. Genc et al. built the mapping relationship between short text and Wikipedia pages to calculate similarities of short text [18]. Wang et al. [19] proposed a unified framework to expand short texts by making use of the convolutional neural network, which can address the problems of sparsity of short texts and semantic sensitivity. However, the above methods still do not avoid the negative influence caused by noisy words.

(3) Topic Model. Quan et al. make use of topic model based on the Latent Dirichlet Allocation (LDA) to calculate similarity of short text, which improves measurement accuracy based on the vector space model (VSM) model [20]. Phan et al. employed a similarity estimation method of short text based on the probability distribution of document topic [21]. Zhang and Zhong collected large-scale external data to build the topic model according to the LDA, which enables word topics to enrich feature representations of short text [22]. Vo and Ock employed a LDA-based method to discover hidden topic from universal datasets (e.g., Computer Science Bibliography), Lecture Notes in Computer Science book series (LNCS), and Wikipedia [23].

In addition to the above methods, there are many studies related to similarity measurement of short text, such as deep leaning based methods [24–27], Earth Mover's Distance-(EMD-) based method [28] and multilevel sentence similarity calculation [29].

3. VCOSI Detailed Design

3.1. Video Similarity Measure Based on Semantic Feature Extension. The noisy words in short text corresponding to video information do not reflect real video content, which results in severely negative influence for the estimation accuracy of video similarity. Therefore, it is essential to eliminate noisy words and select key words in short texts corresponding to video content (the key words can embody real video content). The video information mainly includes title, actor, director, and tag and is denoted by short text. For instance, if the two videos have many cooccurrence words in title, actor, and director of videos, they may have similar content (e.g., the TV series). The tags are the artificial sign to represent valuable information such as category and plot themes of videos. The above video information is used to recognize the key words; namely, *isTitleTerm*, *isActorTerm*, *isDirectorTerm*, and *isTag* are defined as the features of key words. Moreover, we also consider position and frequency of word as the features of key words. For instance, if a word has high frequency of occurrence in title and tag of video, it may carry important information for video content and is considered as the key words. Therefore, *termPOS*, *termFreq*, and *locToTitle* also are defined as the features of key words. Table 1 lists name and description of features of key words.

The key words disperse in short text, so the selection of key words is considered as the classification of words. In order to accurately identify the key words, we choose three kinds of classification methods including decision tree, SVM, and

Table 1: Features of key words of video content.

ID	Feature name	Note
*fea*1	*isTitleTerm*	Video title word
*fea*2	*isActorTerm*	Video actor word
*fea*3	*isDirectorTerm*	Video director word
*fea*4	*isTag*	Video tag word
*fea*5	*termPOS*	Part of speech of a word
*fea*6	*termFreq*	Word frequency in video description
*fea*7	*locToTitle*	Location in video description

Table 2: Accuracy rate of different classification methods.

Features	Classification methods		
	Decision tree	SVM	KNN
All features	45.16%	62.90%	67.74%
*fea*1–*fea*6	48.39%	66.13%	**82.26**%
*fea*1–*fea*5	48.39%	**74.19**%	58.06%
*fea*1, *fea*5, *fea*6, *fea*7	45.16%	62.90%	66.13%
*fea*6	46.77%	66.13%	53.23%

KNN. Table 2 lists the accuracy rate of decision tree, SVM, and KNN. Obviously, KNN and SVM have higher accuracy rate than that of decision tree. For instance, the maximum values of KNN and SVM are 88.26% and 74.19%, respectively. Because KNN has the best accuracy rate among decision tree, SVM, and KNN in the range from *fea*1 to *fea*6, we use KNN based on the range from *fea*1 to *fea*6 to select the key words in the following experiments. Because KNN is a classical classification method, the classification process is no longer described in detail.

On the other hand, if the short text corresponding to video information only includes small number of valuable key words, the selected key words also cannot reflect the whole video content. We employ an ontology-based feature extension method to enrich key words because the vector space model does not deal with the synonym, hypernym, and hyponym in key words. The ontology is constructed by domain experts, which includes rich semantic information based on the specific background. The text label, comments, instances, properties, and relations between other concepts are used to denote a concept in the ontology. We make use of the information of ontology to extend the key words of video content. Let R_k be the set of key words of a video v_k; namely, $R_k = \{r_1, r_2, \ldots, r_i, \ldots, r_m\}$. Let $F_e(r_i)$ denote the set of extended words of a key word r_i based on the built ontology information, which is defined as

$$F_e(r_i) = F_{\text{label}}(r_i) \cup F_{\text{structure}}(r_i) \cup F_{\text{property}}(r_i) \cup F_{\text{instance}}(r_i), \tag{1}$$

where $F_{\text{label}}(r_i)$ is the set of literal features and includes key words of text label and comments of r_i; $F_{\text{structure}}(r_i)$ is the set of structure features and includes key words in ancestor of concept r_i; $F_{\text{property}}(r_i)$ is the set of property features and includes key words in the properties of concept r_i; $F_{\text{instance}}(r_i)$ is the set of instance features and includes key words in the instances of concept r_i. For example, a video, called "angry birds," has the four feature sets corresponding to the key word "birds"; namely, F_{label}(birds) = {bird, warm-blooded, egg-laying, vertebrates, characterized, by, feathers, and, forelimbs, modified, as, wings}, $F_{\text{structure}}$(birds) = {entity, physical, entity, object, whole, living, thing, organism, animal, chordate, vertebrate}, F_{property}(birds) = {warm-blooded, egg-laying}, and F_{instance}(birds) = {Red, Chuck, Bomb, Matilda, Mighty Eagle}. Obviously, the ontology-based semantic description of video content efficiently enriches key words to reduce the probability of erroneous judgement.

The extended words of all items in R_k form a new set $R_e = F_e(r_1) \cup F_e(r_2), \ldots, F_e(r_{m-1}) \cup F_e(r_m)$. R_e can be used to measure the similarity between videos. The traditional Tversky similarity model relies on common and distinctive features between two objects to estimate similarity. However, the users may focus more on common features in the process of similarity estimation. Therefore, we improve the Tversky similarity model and make use of the set R_e to estimate the content similarity between any two videos v_i and v_j according to the following equation.

$$\text{Sim}(v_i, v_j) = \text{Sim}(R_e(v_i), R_e(v_j)) = \frac{\alpha \left| R_e(v_i) \cap R_e(v_j) \right|}{\alpha \left| R_e(v_i) \cap R_e(v_j) \right| + \beta \left| R_e(v_i) - R_e(v_j) \right| + (1-\beta) \left| R_e(v_j) - R_e(v_i) \right|}, \tag{2}$$

where α and β are the weight factors added by us, which are used to adjust common and different features. For instance, we assume that $R_e(v_i)$ is a subset of $R_e(v_j)$ ($R_e(v_i) \subset R_e(v_j)$) and the size of $R_e(v_i)$ is much smaller than that of $R_e(v_j)$ ($|R_e(v_i)| \ll R_e(v_j)$). The similarity value of $R_e(v_i)$ and $R_e(v_j)$ based on the Jaccard similarity measure is very small; namely, $R_e(v_i)$ and $R_e(v_j)$ should be similar to some extent. Therefore, we use (2) to increase the similarity between $R_e(v_i)$ and $R_e(v_j)$ by the adjustment of α and β. For instance, we assume that $R_e(v_i) = \{w_1, w_2, w_4\}$, $R_e(v_j) = \{w_2, w_3, w_5, \ldots, w_{m-1}, w_m\}$, and $|R_e(v_i) \cap R_e(v_j)| = |\{w_2\}| = 1$, $|R_e(v_i) - R_e(v_j)| = |\{w_1, w_4\}| = 2$, and $|R_e(v_j) - R_e(v_i)| = |\{w_3, w_5, \ldots, w_{m-1}, w_m\}| = 50$. The Jaccard similarity of $R_e(v_i)$ and $R_e(v_j)$ is defined as Jaccard$(R_e(v_i), R_e(v_j)) = 1/(1 + 2 + 50) = 0.018$. According to (2), The similarity of $R_e(v_i)$ and $R_e(v_j)$ is defined as Sim$(R_e(v_i), R_e(v_j)) = (\alpha * 1)/(\alpha * 1 + \beta * 2 + (1 - \beta) * 50) = 0.057$, where α is set to 1 and β is set to 0.7. Obviously, Sim$(R_e(v_i), R_e(v_j))$ > Jaccard$(R_e(v_i), R_e(v_j))$.

3.2. Prefix-Filtering-Based Similarity Estimation Algorithm. The extension of key words increases the available information for video content to improve measurement accuracy of video similarity. However, the increase in the extension scale of key words results in high computation complexity due to large number of duplicated comparisons in the process of similarity estimation. The high computation load consumes large amount of energy of mobile nodes. The prefix filtering technique based on the Jaccard similarity in [30] can efficiently reduce computation complexity. Because the video similarity $\mathrm{Sim}(v_i, v_j)$ in (2) is different with the Jaccard similarity, we derive the relationship of the Jaccard similarity and $\mathrm{Sim}(v_i, v_j)$ and convert $\mathrm{Sim}(v_i, v_j)$ to the Jaccard similarity. Further, we design a bloom-filter-based duplicated comparison free method, which reduces the computation load by prevention of duplicated comparisons.

3.2.1. Prefix Filtering Based on Derived Jaccard Similarity Threshold

Lemma 1. *When $\alpha + \beta \le 1$ and $\beta \le 1/2$, if $J(v_i, v_j) < ((1-\beta)/\alpha)\epsilon$, then $Sim(v_i, v_j) < \epsilon$, where $J(v_i, v_j)$ is the Jaccard similarity between v_i and v_j:*

$$J\left(v_i, v_j\right) = \frac{\left|F\left(v_i\right) \cap F\left(v_j\right)\right|}{\left|F\left(v_i\right) \cap F\left(v_j\right)\right| + \left|F\left(v_i\right) - F\left(v_j\right)\right| + \left|F\left(v_j\right) - F\left(v_i\right)\right|}, \tag{3}$$

where ϵ is the similarity threshold. If the similarity $Sim(v_i, v_j)$ between v_i and v_j is smaller than ϵ, v_i and v_j are not similar.

Proof. For simplicity, let x, y, and z represent $|F(v_i) \cap F(v_j)|$, $|F(v_i) - F(v_j)|$, and $|F(v_j) - F(v_i)|$, respectively. The similarity between v_i and v_j can be redescribed as $\mathrm{Sim}(v_i, v_j) = \alpha x/(\alpha x + \beta y + (1-\beta)z)$.

$$\begin{aligned} &\because \alpha + \beta \le 1 \\ &\therefore \alpha \le 1 - \beta \\ &\because \beta \le \frac{1}{2} \\ &\therefore \beta \le 1 - \beta. \end{aligned} \tag{4}$$

So we can get

$$\begin{aligned} \mathrm{Sim}\left(v_i, v_j\right) &= \frac{\alpha x}{\alpha x + \beta y + (1-\beta) z} \\ &\le \frac{\alpha x}{(1-\beta) x + (1-\beta) y + (1-\beta) z} \\ &= \frac{\alpha}{1-\beta} \frac{x}{x+y+z} = \frac{\alpha}{1-\beta} J\left(v_i, v_j\right) \end{aligned} \tag{5}$$

$$\because J\left(v_i, v_j\right) < \frac{1-\beta}{\alpha}\epsilon \Longrightarrow$$

$$\mathrm{Sim}\left(v_i, v_j\right) < \frac{\alpha}{1-\beta} J\left(v_i, v_j\right) = \frac{\alpha}{1-\beta} \frac{1-\beta}{\alpha}\epsilon = \epsilon.$$

□

Lemma 2. *When $\alpha + \beta > 1$ and $\alpha < \beta$, if $J(v_i, v_j) < (\alpha/\beta)\epsilon$, then $Sim(v_i, v_j) < \epsilon$.*

Proof. For simplicity, we still use x, y, and z to represent $|F(v_i) \cap F(v_j)|$, $|F(v_i) - F(v_j)|$, and $|F(v_j) - F(v_i)|$, respectively. The similarity between v_i and v_j can be redescribed as $\mathrm{Sim}(v_i, v_j) = \alpha x/(\alpha x + \beta y + (1-\beta)z)$.

$$\begin{aligned} &\because \alpha + \beta > 1, \\ &\alpha < \beta \\ &\therefore 1 < 2\beta \Longrightarrow \\ &\beta > \frac{1}{2} \Longrightarrow \\ &1 - \beta < \beta. \end{aligned} \tag{6}$$

So we can get

$$\begin{aligned} \mathrm{Sim}\left(v_i, v_j\right) &= \frac{\alpha x}{\alpha x + \beta y + (1-\beta) z} < \frac{\alpha x}{\beta x + \beta y + \beta z} \\ &= \frac{\alpha}{\beta} \frac{x}{x+y+z} = \frac{\alpha}{\beta} J\left(v_i, v_j\right) \\ &\because J\left(v_i, v_j\right) < \frac{\beta}{\alpha}\epsilon \Longrightarrow \\ &\mathrm{Sim}\left(v_i, v_j\right) < \frac{\alpha}{\beta} J\left(v_i, v_j\right) < \frac{\alpha}{\beta}\frac{\beta}{\alpha}\epsilon = \epsilon. \end{aligned} \tag{7}$$

□

3.2.2. Video Similarity Estimation. Let $\mathrm{VT}(n_i)$ and $\mathrm{VT}(n_j)$ denote the set of videos watched by the two nodes n_i and n_j. The extended feature set of each video in $\mathrm{VT}(n_i)$ and $\mathrm{VT}(n_j)$ can be obtained according to (1), recorded as S and T, respectively (line 4). In order to improve the filtering effect, we construct a global ordering $\mathcal{O}$ to investigate the feature frequency (line 5). The corresponding Jaccard similarity threshold ϵ' can be calculated according to Lemmas 1 and 2. For instance, when $\alpha + \beta \le 1$ and $\beta \le 1/2$, $\epsilon' = (1-\beta)/\alpha$ (line 6). According to the principle of the prefix filtering scheme [30], we can obtain the prefix of each $s_a \in S$, recorded as $\mathrm{Pre}(s_a)$. $\mathrm{Pre}(s_a)$ refers to the first $|\mathrm{Pre}(s_a)|$ items of s_a, $|\mathrm{Pre}(s_a)| = |s_a| - \lceil \epsilon' |s_a| \rceil + 1$. For instance, we assume that $s_a = \{A, C, D, B, G, K\}$, $\epsilon' = 0.45$; then $|\mathrm{Pre}(s_a)| = 3$, $\mathrm{Pre}(s_a) = \{A, C, D\}$. The inverted index I_S based on all prefixes in S is constructed (line 7). The indexed terms in I_S are unique terms of prefixes in S, where each indexed term refers to a list composed of all videos whose prefixes contain the corresponding indexed term. In the following stage, s_p may be possible similar to $t_b \in T$ by the inverted index I_S (line 8–19). For each item $t_b \in T$, we firstly figure out the prefix $\mathrm{Pre}(t_b)$. For each term $w_k \in \mathrm{Pre}(t_b)$, we obtain the corresponding inverted index list I_{w_k} in I_S. By traversing all items in I_{w_k}, for each $s_p \in I_{w_k}$, the similarity of t_b and s_p can be calculated. If $\mathrm{Sim}(t_b, s_p) \ge \epsilon$, t_b and s_p are similar. We can add $\langle t_b, s_p \rangle$ to the similar concept set Θ. The pseudocode of the process of prefix-filtering-based video similarity estimation is detailed in Algorithm 1.

(1) **Input**: two video sets VT(n_i) and VT(n_j); similarity threshold ϵ; α and β;
(2) **Output**: similar video pairs set Θ;
(3) $\Theta \leftarrow \emptyset$;
(4) gets extended feature set of each video from VT(n_i) and VT(n_j), recorded as S and T, $S = \{s_1, s_2, \ldots, s_m\}$, $T = \{t_1, t_2, \ldots, t_n\}$;
(5) gets global ordering $\mathcal{O}$ according to term frequency;
(6) gets Jaccard similarity threshold ϵ' according to the parameters α, β and ϵ.
//refer to Lemmas 1 and 2
(7) constructs inverted index I_S based on the prefix of s_a, $s_a \in S$, $a \in [1, m]$;
(8) **for each** t_b of T, $b \in [1, n]$
(9) gets prefix of t_b, recorded as Pre(t_b);
(10) **for each** term $w_k \in$ Pre(t_b)
(11) gets according inverted index list I_{w_k};
(12) **for each** $s_p \in I_{w_k}$
(13) calculates similarity of t_b and s_p;
(14) **if** (Sim(t_b, s_p) $\geq \epsilon$)
(15) $\Theta \leftarrow \Theta \cup \langle t_b, s_p \rangle$;
(16) **end if**
(17) **end for**
(18) **end for**
(19) **end for**

Algorithm 1: Prefix-filtering-based video similarity estimation.

3.3. Construction and Maintenance of Video Communities. The mobile nodes joined into systems make use of the watched video similarity to calculate the similarity levels of interests. Let VT(n_i) = $\{(v_a, t_a), \ldots, (v_h, t_h)\}$ and VT(n_j) = $\{(v_c, t_c), \ldots, (v_m, t_m)\}$ denote historical playback traces of n_i and n_j, respectively. v_c and t_c are the video ID and time of playing video c for n_j. n_i and n_j exchange the information of VT(n_i) and VT(n_j) with each other and calculate the similarity of VT(n_i) and VT(n_j). For instance, n_i calculates similarity value $S(v_a, v_c)$ between v_a and v_c and mean value $\bar{t}_{ac}$ of playback time ratio $t_a = l_a/L_a$ and $t_c = l_c/L_c$, namely, two-tuple $ST_{ac} = (S(v_a, v_c), \bar{t}_{ac})$. l_a and L_a are watched time and length of v_a, respectively; l_c and L_c are watched time and length of v_c, respectively. n_i can obtain a two-tuple set ST of all videos in VT(n_i) and VT(n_j). We use the Least Square Method (LSM) [31] to estimate the correlation coefficient R_{ij} of items in ST according to the following equation:

$$R_{ij} = \frac{\sum_{e=1}^{k} \left|S_e - \bar{S}_{ij}\right| \times \left|\bar{t}_e - \bar{t}_{ij}\right|}{\sqrt{\sum_{e=1}^{k} \left|S_e - \bar{S}_{ij}\right|^2} \times \sqrt{\sum_{e=1}^{k} \left|\bar{t}_e - \bar{t}_{ij}\right|^2}}, \quad (8)$$

where k is the number of items in ST; S_e and $\bar{t}_e$ are video similarity value and average playback time ratio of items in ST, respectively; $\bar{S}_{ij}$ and $\bar{t}_{ij}$ are the mean value of similarity and playback time ratio among all videos in VT(n_i) and VT(n_j), respectively. R_{ij} is considered as the interest similarity value between n_i and n_j. If $R_{ij} > \bar{S}_{ij}$, n_i and n_j have similar interest. We introduce the two discovery methods of community members, as follows.

(1) The nodes make use of the message exchange to find the nodes with similar interests in one-hop neighbor nodes. For instance, n_i broadcasts the messages including the own playback trace to all one-hop neighbor nodes. If the neighbor nodes of n_i do not join any community, they estimate the interest similarity value with n_i. If there are similar interests between them, they return the acknowledgment messages to n_i. n_i and the neighbor nodes with similar interests form a community or the neighbor nodes join into the community corresponding to n_i.

(2) If n_i sends a request message to a supplier n_j of v_k or receives a request message from n_j where the message includes the playback trace of n_j, n_i calculates the interest similarity value with n_j. If n_i and n_j have similar interest and n_j does not join any community, n_i invites n_j to join the community to n_i by the message exchange.

If the target node n_j invited by n_i has joined other communities, n_j decides whether to join the community to n_i. Let R_{uj} denote the interest similarity value between n_j and a corresponding node n_u in current community. If $R_{uj} > R_{ij}$, n_j stays in current community; otherwise, if $R_{uj} < R_{ij}$, n_j quits current community and joins new community to n_i. Obviously, the movement of nodes between communities leads to dynamic community structure and increases the load of community maintenance but also can be considered as a continuous optimization process of community structure. In other words, the nodes always seek the community which has the most similar interest value.

Except for the enter and departure of nodes, the community member role assignment and the community structure maintenance are very important for the video sharing performance. Due to the variation of community structure caused by the enter and departure of nodes, the community members need to maintain community structure and the maintenance load is assigned to the specific members. The real-time and

random variation of community structure bring huge maintenance load. In order to ensure community scalability, the maintenance load should be distributed to multiple members. Because the members join communities by the invitation with each other, there are the logical links among members. Therefore, the community members form an undirected and connected graph $G = (E, V)$. By the collection of video lookup-related communication paths between nodes in G, there is the member n_h which has the most occurrence frequency in all communication paths of a community. The high-frequency occurrence in communication paths means that n_h participates in most of message forwarding of video lookup. n_h's neighbor nodes in G become the broker nodes of community and n_h becomes the head node of all broker nodes.

Because all paths include n_h in G, the paths are divided into multiple subpaths which do not have n_h. The neighbor nodes of n_h are responsible for maintaining the state of other members in the paths including them. n_h acts as the community interface to contact other communities. If a neighbor node n_s of n_h quits the system or current community, n_h reselects a neighbor node of n_s as the broker member. If n_h leaves current system or current community, the member which has the highest occurrence frequency among all neighbor nodes in n_h becomes new head node. All broker nodes exchange the information of community members during a period of time.

When a node n_i in a community requests a video v_k, it sends a request message to the contacted broker node n_p. If n_p is aware of the supplier cached v_k, n_p directly forwards the request message to the supplier. Otherwise, if n_p does not store the information of suppliers cached v_k, n_p forwards the request message to the head node n_h. n_h makes use of the maintained interface information between communities to help n_i search v_k. If n_i quits the system or joins other communities, n_i needs to send a message to inform n_p. n_p removes information of n_i from local member list. If a mobile node or a member in other communities joins current community, the inviter sends a message containing the information of new member to the corresponding broker node. The latter updates local member list.

4. Testing and Test Results Analysis

4.1. Testing Topology and Scenarios

4.1.1. Testing Scenarios of Video Similarity. The common interest of users is the key factor for the tight levels between community members, which determines the stability of community structure. We firstly compare the measurement performance of video similarity between the proposed ontology-based semantical estimation method, Semantic Feature Extension based Similarity Measure (SFE*b*SM) and Semantic Feature Extension based Similarity Measure using Key Word Selection (SFE*b*SMKS), and the two classical methods, Jaccard Similarity Mesaure using FileName only (JVSMfn) and Jaccard Similarity Mesaure using all features (JVSMaf).

Because there is no existing benchmark for video similarity estimation, we construct an annotated dataset composed of 4000 videos selected from the internet. The attributes of videos include video title, video description, video type, tags, directors, and actors. We invite 100 student volunteers to estimate the similarity value for 5000 pairs of videos according to the video attributes where the similarity value range is between 0 and 1. The similarity value can be any real number with two decimal places between 0 and 1. Table 3 shows a part of evaluation results of volunteer for the 5000 video pairs.

Table 3: 5000 video pairs with volunteer ratings.

Video A	Video B	Volunteer ratings
video 1	video 2	0.12
video 1	video 3	0.32
video 4	video 10	0.52
video 4	video 29	0.31
video 10	video 21	0.71
video 10	video 28	0.65
video 100	video 201	0.54
video 100	video 324	0.16
video 124	video 401	0.62
video 135	video 552	0.87
video 351	video 601	0.36
video 361	video 701	0.33
...	...	...

4.1.2. Testing Scenarios of Video Sharing. We make use of the estimation results of video similarity to build the communities and further compare the video sharing performance of VCOSI with a state-of-the-art solution SPOON [9].

VCOSI and SPOON were modeled and implemented in NS-2. 500 mobile nodes are deployed in a wireless mobile network whose area is set to $1000 \times 1000\ m^2$. The mobile speed range of nodes is in the range [1, 20] m/s. The simulation time is set to 500 s. The signal range of mobile nodes is set to 200 m. The transmission protocol of video data is TCP and the wireless routing protocol is DSR. The bandwidth of mobile nodes and server is set to 10 Mb/s and 20 Mb/s, respectively. The server acts as initial provider of video resources. The sufficient bandwidth of server not only avoids the overload caused by large-scale access due to low resource supply capacity in overlay network during initial simulation but also promotes the resource distribution. The mobile nodes request video and act as core network in the built wireless mobile environment. The settings of bandwidth of mobile nodes avoid severe congestion caused by large-scale request, which does not result in super-long congestion recovery time. The transmission rate of video data is set to 128 kb/s. The two solutions employ the same network environment. Moreover, the number of video files requested by all nodes is set to 20 and the length of each file is 180 s.

We created 300 historical playback traces for the 300 mobile nodes where the historical playback traces have video structure and different attribute values. Further, we also generated 300 playback logs for the 300 mobile nodes where the playback logs include the ID and time of watched videos. After the nodes join the system, they play videos in terms of

the playback logs. Once the nodes finish the playback for the videos in terms of the defined playback time, they continue to request new videos according to the playback logs. The surplus 200 mobile nodes do not join the system and are responsible for forwarding request messages and video data. We further describe the behaviours of 300 mobile nodes which join the system. Before the implementation of simulation, we assign the 100 historical playback traces for 100 mobile nodes. We make use of the proposed community construction method based on the measurement results of interest similarity to group the 100 mobile nodes into multiple communities. The members in the built communities request videos following Poisson distribution in terms of the playback logs and provide video resources for the request nodes. After the beginning of simulation, 100 mobile nodes join the system and request videos following Poisson distribution from the simulation time $t = 200$ s to $t = 500$ s. After the 100 nodes join the system, they join the corresponding communities in terms of the interest similarity values. In VCOSI, the surplus 100 mobile nodes do not actively request videos and join the system. Once they receive the invitation from the community members, they join the corresponding communities and request and play videos according to the assigned playback logs. In SPOON, the surplus 100 mobile nodes are randomly assigned the time of joining the system; namely, they randomly join the system from $t = 0$ s to $t = 500$ s according to the assigned time and play videos according to the playback logs.

We define a random mobility model for all mobile nodes. The mobile nodes move from current location to target location in terms of the specific speed. After the mobile nodes arrive at the target location, the mobile nodes move to new target location in terms of new speed. The speed and initial and target location of mobile nodes are randomly assigned.

4.2. Performance Evaluation of Video Similarity Measurement

4.2.1. Estimate Accuracy of Video Similarity. We compare accuracy of similarity estimation of four methods: SFE*b*SM, SFE*b*SMKS, JVSMfn, and JVSMaf. Based on the similarity measurement results of video pairs by student volunteers, we firstly calculate the Pearson correlation values according to the following:

$$r = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2} \sqrt{\sum_{i=1}^{n} \left(Y_i - \overline{Y}\right)^2}}. \quad (9)$$

We make use of the calculated results of the Pearson correlation to further evaluate the accuracy of the proposed similarity measures and compare with other approaches.

Figure 2 shows the correlation coefficient with different α and β. The correlation coefficient is higher with $\alpha \in [1.0, 1.8]$ and $\beta \in [0.5, 0.8]$, which means that the common features are more important than the different features for the similarity measure methods. Moreover, when $\alpha = 1.5$ and $\beta = 0.5$, the correlation coefficient reaches the maximum value 0.919. Therefore, the values of $\alpha = 1.5$ and $\beta = 0.5$ are 1.5 and 0.5 in the following experiments, respectively.

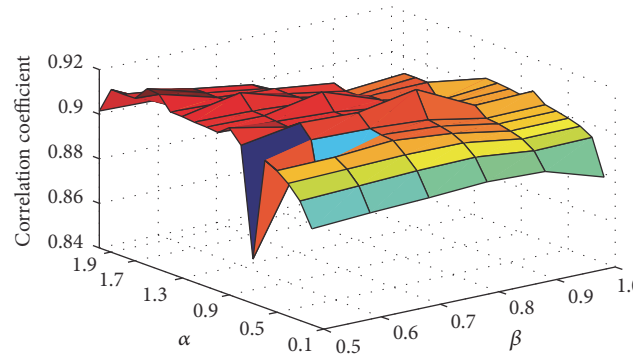


Figure 2: Correlation coefficient versus α, β.

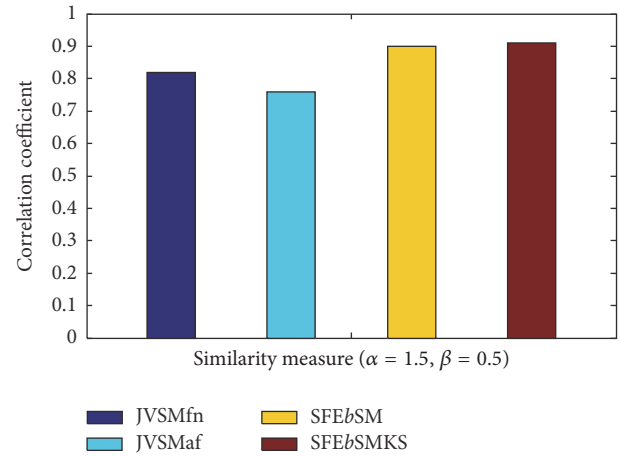


Figure 3: Correlation coefficient versus different similarity measure.

Figure 3 shows the measurement results of correlation coefficient of the four methods by using WordNet. Although JVSMfn only uses video name to estimate similarity between videos and JVSMaf investigates multiple features, JVSMaf cannot efficiently eliminate noisy words in the features. Therefore, the measure results of JVSMaf are lower than those of JVSMfn. SFE*b*SMKS and SFE*b*SM employ the semantic feature extension method to estimate the similarity of video content; they can obtain higher similarity accuracy than JVSMaf and JVSMfn by efficient elimination of noisy words and extraction of key words. Moreover, SFE*b*SMKS rely on the selection of key words based on the KNN classification to obtain better extraction effect of noisy words than SFE*b*SM, so that the estimation accuracy of SFE*b*SMKS is higher than that of SFE*b*SM.

4.2.2. Run Time. The execution time of similarity estimation is defined as the run time, in order to compare efficiency of the proposed prefix-filtering-based algorithm (Prefix) with the brute force method (Bruteforce).

As Figure 4 shows, the two curves corresponding to Prefix and Bruteforce have the same rise trend with the increase in the data size. The green curve of Prefix is lower than the blue curve of Bruteforce. The increment and peak value of Prefix are less than those of Bruteforce.

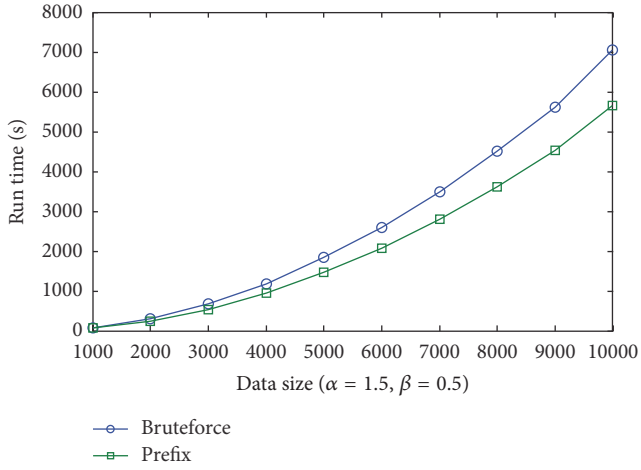


Figure 4: Run time with different data size.

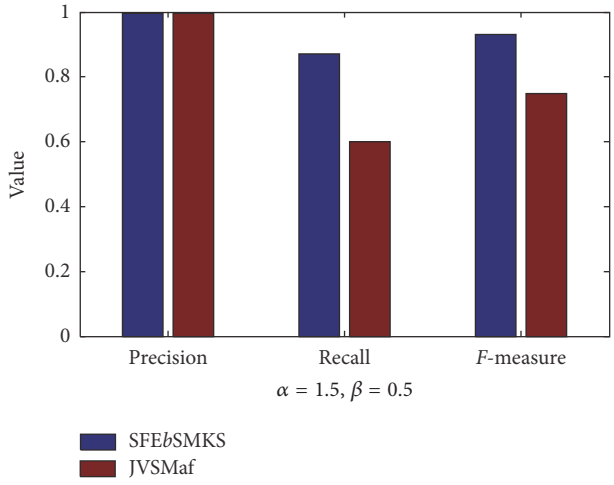


Figure 5: Precision, recall, and F-measure with different similarity measure.

Prefix makes use of the prefix filtering method to decrease the large number of repeated comparisons. In the other words, Prefix eliminates more repeated video pairs than those of Bruteforce. The computation complexity of Prefix is lower than that of Bruteforce, so that the run time of Prefix is less than that of Bruteforce.

4.2.3. Precision, Recall, and F-Measure. Figure 5 shows the precision, recall, and F-measure of SFEbSMKS and JVSMaf with $\alpha = 1.5$ and $\beta = 0.5$. The precision results of SFEbSMKS and JVSMaf are 0.997 and 0.996, respectively. While the recall and F-measure of SFEbSMKS are higher than those of JVSMaf, obviously, the precision, recall, and F-measure of SFEbSMKS are better than those of JVSMaf. SFEbSMKS depends on the semantic feature extension and the key word selection to obtain better performance of similarity measure than JVSMaf.

4.3. Performance Evaluation of Video Sharing. The performance of VCOSI is compared with that of SPOON in terms of average startup delay, PSNR, and maintenance overhead, respectively.

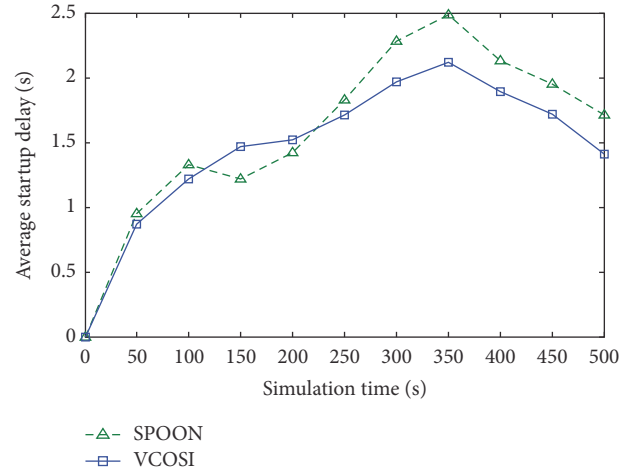


Figure 6: Average startup delay against simulation time.

4.3.1. Average Startup Delay. We use the time span between sending request message and receiving first video data to denote the startup delay.

Figure 6 shows the mean values of collected startup delay for the results of VCOSI and SPOON during a time interval $T = 50$ s. As Figure 6 shows, SPOON's green curve has the two processes of rise and fall with the whole simulation time. The green curve firstly increases from $t = 0$ s to $t = 100$ s and slightly falls from $t = 100$ s to $t = 150$ s. It continues to quickly rise from $t = 150$ s to $t = 350$ s and quickly decreases from $t = 350$ s to $t = 500$ s. The blue curve of VCOSI keeps the rise trend with the fluctuation from $t = 0$ s to $t = 350$ s and quickly falls from $t = 350$ s to $t = 500$ s. VCOSI's blue curve is higher than that of SPOON from $t = 0$ s to $t = 200$ s, but VCOSI's results are less than those of SPOON from $t = 250$ s to $t = 500$ s.

Figure 7 includes the mean values of collected startup delay for the results of VCOSI and SPOON in the process of every 30 request nodes requesting videos. As Figure 7 shows, SPOON's green curve has fast increase with the fluctuation in the process of simulation. Although VCOSI's results also quickly rise, the increment of VCOSI's results is less than that of SPOON.

SPOON estimates the relationship between community members according to the similarity of file name of watched video (SPOON does not describe the measurement method of communication frequency between nodes). Although the file name reflects main content of videos, there is the information loss for the video content with rich-content. It is difficult to accurately denote the interest similarity relationship between members for the measurement method of similar interest in SPOON. This leads to fragile relationship between members and low sharing performance of system. For instance, once there are the different interests between members, the request messages for video content difficultly are quickly responded by other intracommunity members.

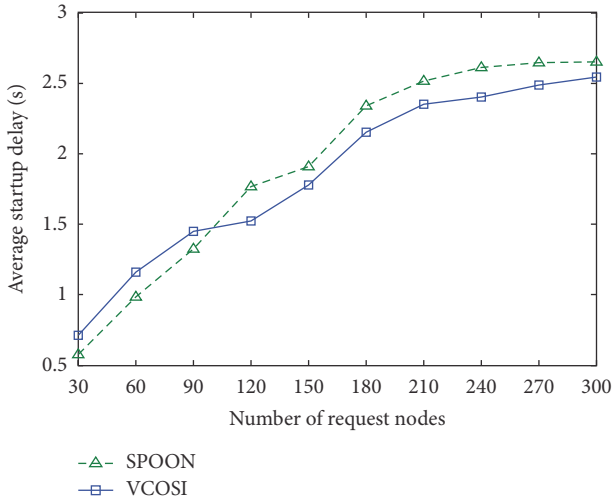


Figure 7: Average startup delay against number of request nodes.

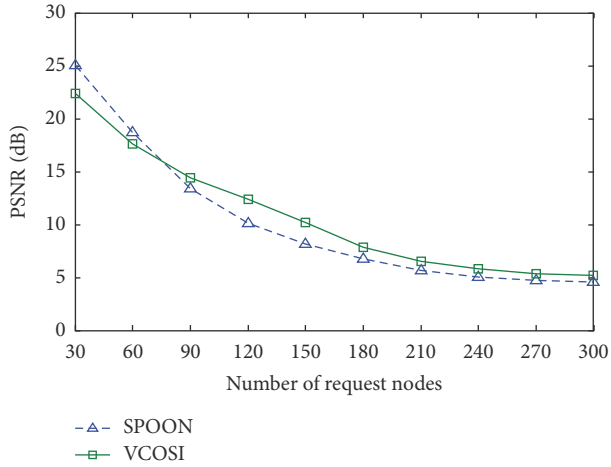


Figure 8: PSNR against number of request nodes.

The low video lookup success rate increases the lookup number, which increases the lookup delay. Because SPOON does not consider the mobility of members in the process of community construction, the transmission performance of video data is easily subjected by the influence of node mobility. The fast increase in the number of request nodes leads to quick rise of startup delay. VCOSI estimates the interest similarity between members by making use of ontology-based semantical interest capture to measure the extensional video information, which obtains the accurate measurement results of interest similarity. VCOSI further obtains high video lookup success rate; namely, the more video request can be responded by intracommunity members, which reduces the lookup delay. On the other hand, the community members make use of the message exchange to invite the one-hop neighbor nodes to join the communities, which reduces the geographical distance between video requesters and suppliers. The transmission delay of video data may be reduced. The two solutions VCOSI and SPOON experience slight network congestion (from $t = 200$ s to $t = 350$ s) with increasing number of nodes requesting videos. The community members in VCOSI need to invite the mobile nodes to join the communities, so VCOSI's results are higher than those of SPOON from $t = 0$ s to $t = 100$ s. However, after the mobile nodes join the communities, VCOSI's results are better than those of SPOON from $t = 250$ s to $t = 500$ s.

4.3.2. Peak Signal to Noise Ratio (PSNR). We use the PSNR of each video streaming to show the watched video quality according to the following equation which is defined in [32]:

$$\text{PSNR} = 20 \times \log_{10}\left(\frac{\text{MAX\_Bitrate}}{\sqrt{(\text{EXP\_Thr} - \text{CRT\_Thr})^2}}\right), \quad (10)$$

where MAX_Bitrate is the transmission rate of video data and is set to 128 k/s; EXP_Thr and CRT_Thr are expected and real throughput, respectively. The value of EXP_Thr of each video streaming should be equal to the value of MAX_Bitrate. The value of CRT_Thr is real throughput of video streaming received by each node.

Figure 8 shows the mean values of PSNR for video streaming of each node corresponding to VCOSI and SPOON with increasing number of request nodes. As Figure 8 shows, the two curves corresponding to the results of VCOSI and SPOON have the same fall trend. The curve of VCOSI is lower than that of SPOON when the number of request nodes increases from 30 to 60. When the number of request nodes increases from 60 to 300, VCOSI's curve is higher than that of SPOON.

In SPOON, the measurement of relationship between community members mainly investigates the interest similarity based on the similarity calculation results of file name. The influence of node mobility cannot be avoided in the process of video data transmission due to the neglect for the investigation of mobility similarity. The increase in the number of request nodes brings the high demand for the network traffic, which results in the network congestion. The high packet loss rate caused by congestion severely influences the amount of video data received by nodes; namely, the decrease in the real average throughput of each node makes the values of SPOON's PSNR quickly fall. In VCOSI, the community members need to exchange the playback traces to find the potential community members with similar interests. The increasing number of request nodes, including nodes passively receiving invitation and nodes actively joining the system, brings high traffic demand. The high traffic of VCOSI also results in the increase in the number of lost packets, so that the values of PSNR of VCOSI are less than those of SPOON when the number of request nodes increases from 30 to 60. With the continuous increase in the number of request nodes, the network congestion also leads to the large number of packet losses and the fast decrease of VCOSI's PSNR. However, the negative influence levels of VCOSI caused by congestion are lower than those of SPOON (the curve of VCOSI's PSNR is higher than that of SPOON when the number of request nodes increases from 60 to 300). This is as the invited community members include the one-hop neighbor nodes of

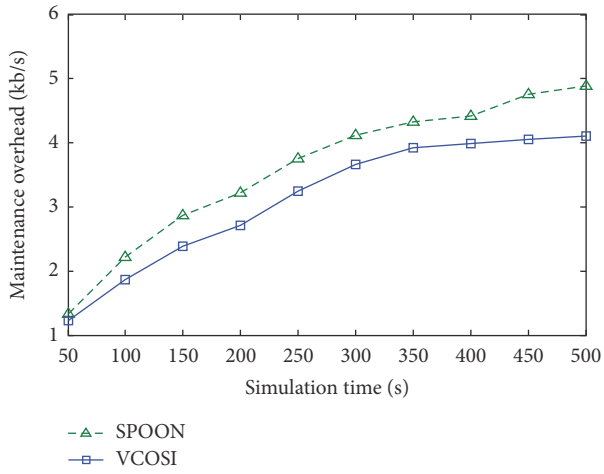


Figure 9: Maintenance overhead against simulation time.

inviters. The close geographical distance between inviters and invited nodes reduces the influence from the node mobility and the risk of packet loss.

4.3.3. Maintenance Overhead. The maintenance of member state, handling request message, and interaction between communities need to consume network bandwidth. The usage amount of bandwidth for maintaining the above information is considered as the maintenance overhead.

Figure 9 shows the maintenance overhead of VCOSI and SPOON during a time interval $T = 50$ s. As Figure 9 shows, the two curves of VCOSI and SPOON experience the rise with the whole simulation time. VCOSI's blue curve is lower than that of SPOON and the increment of VCOSI's results gradually decreases. The increment of SPOON's results still keeps the increasing trend.

SPOON relies on the coordinators and ambassadors in communities to maintain the member state, handle request messages, and interact with other communities. The increase in the number of members results in the large amount of consumption of bandwidth in order to maintain the logical links between members. Moreover, the increase in the number of members also brings more request messages for video content, so that the members need to consume large amount of bandwidth to handle the massive request messages. On the other hand, the inaccurate interest similarity measurement leads to fragile relationship between members. The fragile relationship between members brings severely negative influence for stability of community structure; namely, the community members continuously leave current community and join into new community. The reconstruction of community structure results in huge consumption of bandwidth. The low lookup success rate also causes repetitive search, which increases the bandwidth consumption. Therefore, SPOON needs to consume massive bandwidth to maintain the overlay network. VCOSI has stable community structure relative to SPOON because of the accurate interest similarity measurement between members. The consumption of bandwidth from the maintenance cost of community structure keeps low growth with increasing number of members. The high lookup success rate reduces the number of lookup messages, which reduces the bandwidth use. VCOSI has lower maintenance overhead than SPOON. Moreover, VCOSI distributes the maintenance overhead to multiple broker nodes, which does not cause the overload of broker nodes.

5. Conclusion

In this paper, we propose a novel mobile Video Community discovery scheme using ontology-based semantical interest capture (VCOSI) in order to enhance the stability and scalability of overlay networks in the process of network virtualization. VCOSI proposes an ontology-based semantical extension approach, which achieves distinct and precise description for video content and accurately measures the similarity between videos. In order to reduce the calculation load of mobile nodes, VCOSI employs a prefix-filtering-based estimation algorithm to reduce the comparison times between videos. VCOSI makes use of the similarity for watched videos to estimate the interest similarity level between nodes, construct node communities, and economically maintain community structure. The simulation results show how VCOSI has lower startup delay, higher video quality, and lower maintenance overhead than SPOON.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants nos. 61272015 and 61050004, the Basic and Frontier Technology Research Project of Henan province (142300410303), and the Science and Technology Plan Projects (Openness & Cooperation) of Henan province (152106000048).

References

[1] Y. Zhou, L. Chen, C. Yang, and D. M. Chiu, "Video popularity dynamics and its implication for replication," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1273–1285, 2015.

[2] J. Otwani, A. Agarwal, and A. K. Jagannatham, "Optimal scalable video scheduling policies for real-time single- and multiuser wireless video networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2424–2435, 2015.

[3] C. Yang, Y. Zhou, L. Chen, T. Z. J. Fu, and D. M. Chiu, "Turbocharged video distribution via P2P," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 2, pp. 287–299, 2015.

[4] H. Park and M. V. D. Schaar, "Coalition-based resource reciprocation strategies for P2P multimedia broadcasting," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 557–567, 2008.

[5] S. Jia, C. Xu, J. Guan, H. Zhang, and G.-M. Muntean, "A novel cooperative content fetching-based strategy to increase the quality of video delivery to mobile users in wireless networks,"

IEEE Transactions on Broadcasting, vol. 60, no. 2, pp. 370–384, 2014.

[6] C. Xu, F. Zhao, J. Guan, H. Zhang, and G.-M. Muntean, "QoE-driven user-centric vod services in urban multihomed P2P-based vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2273–2289, 2013.

[7] C.-L. Chang, W.-M. Chen, and C.-H. Hung, "Reliable consideration of P2P-based VoD system with interleaved video frame distribution," *IEEE Systems Journal*, vol. 8, no. 1, pp. 304–312, 2014.

[8] H. Shen, Z. Li, Y. Lin, and J. Li, "Socialtube: P2P-assisted video sharing in online social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2428–2440, 2014.

[9] K. Chen, H. Shen, and H. Zhang, "Leveraging social networks for P2P content-based file sharing in disconnected MANETs," *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 235–249, 2014.

[10] C. Xu, S. Jia, L. Zhong, and G.-M. Muntean, "Socially aware mobile peer-to-peer communications for community multimedia streaming services," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 150–156, 2015.

[11] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang, and G.-M. Muntean, "Performance-aware mobile community-based VoD streaming over vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1201–1217, 2015.

[12] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI '06)*, pp. 775–780, Boston, Mass, USA, July 2006.

[13] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 1138–1150, 2006.

[14] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 2, article 10, 2008.

[15] M. Sahami and T. D. Heilman, "A web-based kernel function for measuring the similarity of short text snippets," in *Proceedings of the 15th International Conference on World Wide Web (WWW '06)*, pp. 377–386, 2006.

[16] X. Wang, X. Fan, and J. Zhao, "Method of Chinese short text classification based on feature extension," *Journal of Computer Applications*, vol. 29, no. 3, pp. 843–845, March 2009.

[17] M. Yuan, "Feature extension for short text categorization using frequent term sets," *Procedia Computer Science*, vol. 31, pp. 663–670, 2014.

[18] Y. Genc, Y. Sakamoto, and J. Nickerson, "Discovering context: classifying tweets through a semantic transform based on wikipedia," in *Foundations of Augmented Cognition. Directing the Future of Adaptive Systems: 6th International Conference, FAC 2011, Held as Part of HCI International 2011, Orlando, FL, USA, July 9–14, 2011. Proceedings*, vol. 6780 of *Lecture Notes in Computer Science*, pp. 484–492, Springer, Berlin, Germany, 2011.

[19] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, and H. Hao, "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification," *Neurocomputing*, vol. 174, pp. 806–814, 2016.

[20] X. Quan, G. Liu, Z. Lu, X. Ni, and L. Wenyin, "Short text similarity based on probabilistic topics," *Knowledge and Information Systems*, vol. 25, no. 3, pp. 473–491, 2010.

[21] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, pp. 91–100, Beijing, China, April 2008.

[22] H. Zhang and G. Zhong, "Improving short text classification by learning vector representations of both words and hidden topics," *Knowledge-Based Systems*, vol. 102, pp. 76–86, 2016.

[23] D.-T. Vo and C.-Y. Ock, "Learning to classify short text from scientific documents using topic models with various types of knowledge," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1684–1698, 2015.

[24] M. Wang, Z. Lu, H. Li, and Q. Liu, "Syntax-based deep matching of short texts," https://arxiv.org/abs/1503.02427.

[25] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *Proceedings of the 38th International ACM SIGIR conference on research and development in Information Retrieval (SIGIR '15)*, pp. 373–382, Santiago, Chile, 2015.

[26] Z. Lu and H. Li, "A deep architecture for matching short texts," in *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pp. 1367–1375, 2013.

[27] Z. Wang, H. Mi, and A. Ittycheriah, "Semi-supervised clustering for short text via deep representation learning," 2016, http://arxiv.org/abs/1602.06797.

[28] X. Wan, "A novel document similarity measure based on earth mover's distance," *Information Sciences*, vol. 177, no. 18, pp. 3718–3730, 2007.

[29] H. Huang, Z. Chen, X. Zhang, and K. Zhang, "Approach of large-scale sentence similarity computation," *Journal of Chinese Information Processing*, vol. 20, no. 1, pp. 47–52, 2006.

[30] R. Bayardo, Y. Ma, and R. Srikant, "Scaling up all pairs similarity search," in *Proceedings of the 16th ACM International Conference on World Wide Web (WWW '07)*, New York, NY, USA, 2007.

[31] B. Yang and C. A. Balanis, "Least square method to optimize the coefficients of complex finite-difference space stencils," *IEEE Antennas and Wireless Propagation Letters*, vol. 5, no. 1, pp. 450–453, 2006.

[32] S.-B. Lee, G.-M. Muntean, and A. F. Smeaton, "Performance-aware replication of distributed pre-recorded IPTV content," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 516–526, 2009.