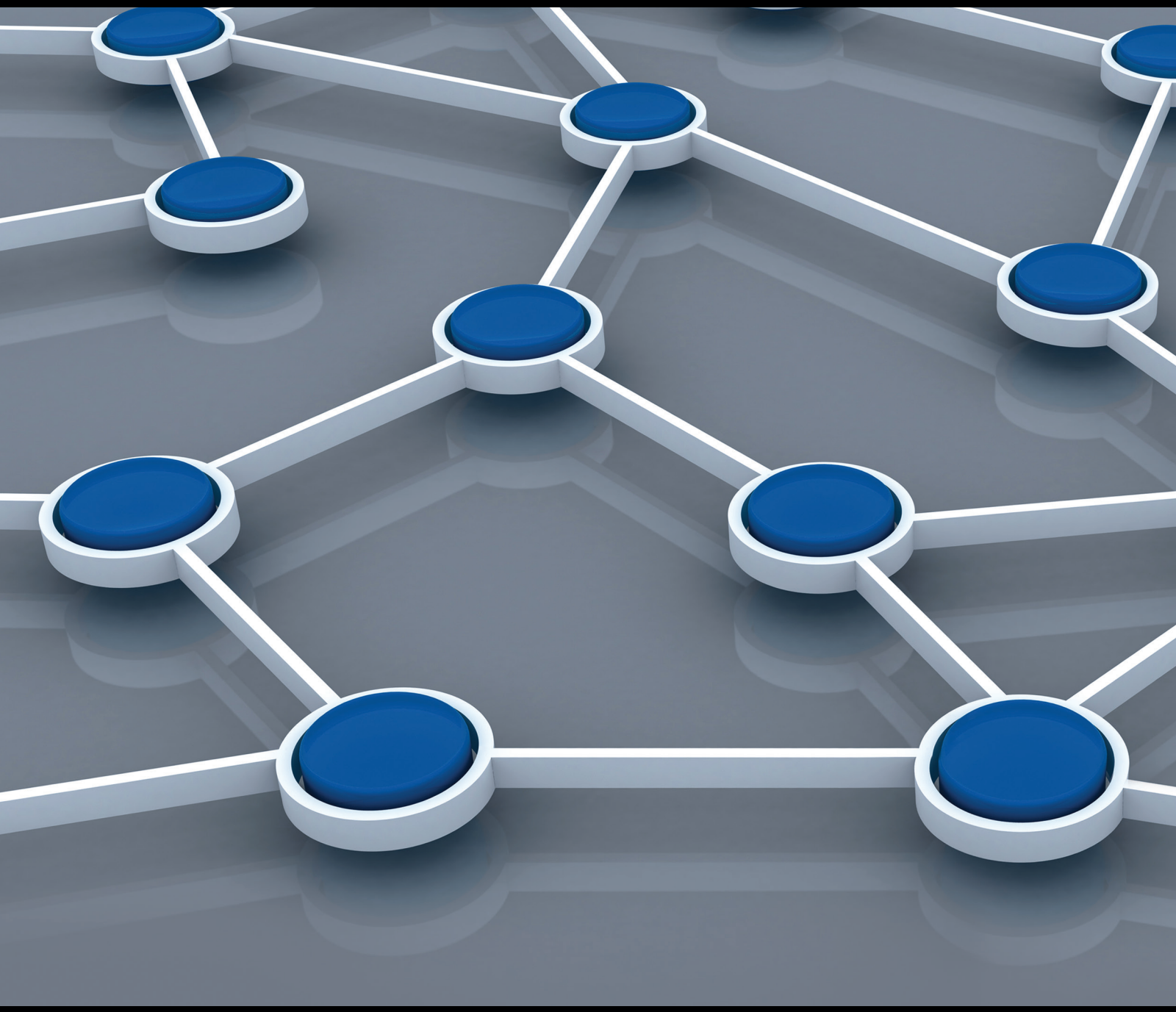
International Journal of Distributed Sensor Networks

Green and Friendly Communication for Sensor Networks

Guest Editors: Changqiao Xu, Gabriel-Miro Muntean, Liang Zhou, and Xiaohong Jiang



Green and Friendly Communication for Sensor Networks

International Journal of Distributed Sensor Networks

Green and Friendly Communication for Sensor Networks

Guest Editors: Changqiao Xu, Gabriel-Miro Muntean, Liang Zhou, and Xiaohong Jiang

Editorial Board

Contents

Green and Friendly Communication for Sensor Networks, Changqiao Xu, Gabriel-Miro Muntean, Liang Zhou, and Xiaohong Jiang
Volume 2015, Article ID 968167, 2 pages

Cloud-Assisted Scalable Video Delivery Solution over Mobile Ad Hoc Networks, Lujie Zhong and Shijie Jia
Volume 2015, Article ID 205106, 10 pages

ECMTADR: Energy Conservative Multitier Architecture with Data Reduction for Cluster-Based Wireless Sensor Networks, Taner Cevik
Volume 2015, Article ID 236354, 11 pages

Intelligent Transmission Power Allocation for Distributed Beamforming in Wireless Sensor Networks, Sungmoon Chung and Inwhee Joe
Volume 2015, Article ID 510516, 10 pages

Shared MPR Sets for Moderately Dense Wireless Multihop Networks, Teruaki Kitasuka and Shigeaki Tagashira
Volume 2015, Article ID 486023, 11 pages

Modeling MAC Protocol Based on Frame Slotted Aloha for Low Energy Critical Infrastructure Sensor Networks, Niamat Ullah, Kifayat Ullah, S. M. Riazul Islam, Pervaiz Khan, Sana Ullah, and Kyung Sup Kwak
Volume 2015, Article ID 701418, 11 pages

An Adaptive Routing Protocol Based on QoS and Vehicular Density in Urban VANETs, Yongmei Sun, Shuyun Luo, Qijin Dai, and Yuefeng Ji
Volume 2015, Article ID 631092, 13 pages

A Novel Interest Detection-Based Video Dissemination Algorithm under Flash Crowd in Mobile Ad Hoc Networks, Shijie Jia, Shengli Jiang, Yuanchen Li, Xihu Zhi, and Mu Wang
Volume 2015, Article ID 239267, 10 pages

A Novel Energy-Efficient Reception Method Based on Random Network Coding in Cooperative Wireless Sensor Networks, Yulun Cheng and Longxiang Yang
Volume 2015, Article ID 238386, 13 pages

Security Trade-Off and Energy Efficiency Analysis in Wireless Sensor Networks, Damian Rusinek, Bogdan Ksiezopolski, and Adam Wierzbicki
Volume 2015, Article ID 943475, 17 pages

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 968167, 2 pages
http://dx.doi.org/10.1155/2015/968167

Editorial

Green and Friendly Communication for Sensor Networks

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

[1]*School of Network Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*School of Electronic Engineering, The RINCE Institute, Dublin City University, Research and Engineering Building, Glasnevin, Dublin 9, Ireland*
[3]*School of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210000, China*
[4]*School of Systems Information Science, Future University Hakodate, Hakodate 041-8655, Japan*

Correspondence should be addressed to Changqiao Xu; cqxu@bupt.edu.cn

Received 29 April 2015; Accepted 29 April 2015

Recent advances of various sensor technologies, such as the Internet of Things, sensor cloud, underwater sensor, and healthcare sensor, have moved us toward the era of worldwide sensor networks. People can sense and collect necessary sensory data anytime and anywhere. However, efficient resource utilization in terms of energy consumption, spectrum allocation, routing selection, and so forth is still a big challenge in the sensor networks research area. Designing "green" sensor networks, the next generation of wireless sensor networks, has become a matter of paramount importance. On the other hand, the lack of cooperation among sensors not only affects the quality of communication, but also results in unbalance of resource utilization, which further reduces the robustness of the sensor system. "Friendly" cooperation among sensors, such as information sharing, spectrum/energy awareness, routing adaptation, and data caching, enables providing potential benefits for optimizing and balancing the resource usage, hence improving the lifetime of the entire sensor network. Therefore, green and friendly communication becomes the utmost important and promising avenue for the future sensor network research.

17 paper submissions were received. After thorough and meticulous reviews, followed by extensive discussions among the guest editors.

Among the accepted papers, there are two articles addressing the integration of data communications coming from wireless sensor networks with the purpose of achieving energy-saving. In "ECMTADR: Energy Conservative Multitier Architecture with Data Reduction for Cluster-Based Wireless Sensor Networks," T. Cevik proposes a sophisticated architecture comprising data reduction, load balance, and topology control for data communication. N. Ullah et al., in "Modeling MAC Protocol Based on Framed Slotted Aloha for Low Energy Critical Infrastructure Sensor Networks," analyze a MAC protocol for low energy critical infrastructure monitoring (LECIM) networks and propose a framed slotted aloha based MAC for LECIM using linear increasing contention window size to reduce the packet drop probability.

Cooperation is one of the most effective methods to improve the performance and robustness of routing algorithms, particularly when working under wireless sensor networks. The paper by Y. Cheng and L. Yang entitled "A Novel Energy-Efficient Reception Method Based on Random Network Coding in Cooperative Wireless Sensor Networks" presents an opportunistic reception algorithm for energy-efficient transmission in cooperative WSNs. In "Shared MPR Sets for Moderately Dense Wireless Multihop Networks," T. Kitasuka and S. Tagashira propose a method for achieving more efficient multipoint relay selection in moderately dense wireless multihop networks than the conventional multipoint relay selection. In "Intelligent Transmission Power Allocation for Distributed Beamforming in Wireless Sensor Networks," S. Chung and I. Joe propose an Intelligent Transmission Power Allocation algorithm to guarantee the required channel capacity considering dynamic channel statement, number

of cooperating source nodes, and distance between the average source nodes and destination. Y. Sun et al., in "An Adaptive Routing Protocol Based on QoS and Vehicular Density in Urban VANETs," explore an adaptive routing protocol based on QoS and vehicular density in urban VANET environments.

The trade-off between security and energy efficiency for wireless sensors networks is addressed in the paper by D. Rusinek et al. entitled "Security Trade-Off and Energy Efficiency Analysis in Wireless Sensor Networks." The authors propose an energy analysis module for the quality of protection modeling language by means of which one can analyze the influence of various security levels on the energy consumption of a protocol. Furthermore, an advanced communication module is proposed as an extension of the quality of protection modeling language, which enhances the abilities to analyze complex wireless sensor networks.

High efficient resource dissemination and delivery are also an important issue for the green and friendly communications in wireless sensor networks. The paper by S. Jia et al. entitled "A Novel Interest Detection-Based Video Dissemination Algorithm under Flash Crowd in Mobile Ad Hoc Networks" proposes a novel interest detection-based video dissemination algorithm under flash crowd in mobile ad hoc networks. In "Cloud-Assisted Scalable Video Delivery Solution over Mobile Ad Hoc Networks," L. Zhong and S. Jia present a novel Cloud-Assisted Scalable Video Delivery Solution over mobile ad hoc networks.

All these works are mature and present detailed research proposals, good testing results, and interesting result analyses. It is hoped that the audience will appreciate them and the readers will have pleasant reading.

Acknowledgments

Thanks are due to all the authors for submitting their works to this special issue. The guest editors would like to thank all the reviewers for their hard work and for all their suggestions and comments, leading to improvements in the quality of the accepted papers. They hope these papers will represent a useful starting point and stimulus for further research in the sensor technologies area.

Changqiao Xu
Gabriel-Miro Muntean
Liang Zhou
Xiaohong Jiang

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 205106, 10 pages
http://dx.doi.org/10.1155/2015/205106

Research Article

Cloud-Assisted Scalable Video Delivery Solution over Mobile Ad Hoc Networks

Lujie Zhong[1] and Shijie Jia[2]

[1]*Information Engineering College, Capital Normal University, Beijing 100048, China*
[2]*Academy of Information Technology, Luoyang Normal University, Luoyang 471022, China*

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

Received 14 October 2014; Accepted 17 December 2014

Academic Editor: Liang Zhou

Cloud computing is promising avenue for supporting high-performance and interactive streaming service. Making use of the clouds to flexibly increase the scale of video service system and provide fast search are key determinants for scalable mobile video streaming in order to ensure smooth playback experience. In this paper, we propose a novel Cloud-assisted Scalable Video Delivery Solution over mobile ad hoc networks (CSVD). CSVD makes use of the clouds to share responsibility for the load of resource management of media server, which supports fast resource searching and enhances system scalability. CSVD designs a new estimation model of resource maintenance scale in terms of quality of service- (QoS-) oriented dynamic balance between supply and demand, in order to economically use the clouds. A novel supplier scheduling algorithm that assigns resource suppliers for fast responding user request in terms of their load and serving capacity is proposed. Extensive tests show how CSVD achieves much better performance results in comparison with other state-of-the-art solutions.

1. Introduction

The mobile ad hoc networks (MANETs) are the significant network technologies for the next generation Internet, which have extensive application areas [1–3]. The new wireless communication protocol such as IEEE 802.11 can meet high bandwidth requirement of multimedia services to enable provision of rich visual content for the mobile users in MANETs [4]. Video streaming is a significant one of the multimedia services, which provides rich content in multiple network environment such as MANETs, VANETs, and wireless sensor networks (WSN) [5–10]. P2P technologies are well known for supporting large-scale video streaming system deployment [11–15]. However, provision of P2P-based high-quality video streaming service with efficient content sharing over MANETs is a challenging issue. Due to limited capacities of energy and storage, the mobile nodes only cache relatively short video clip, so as to frequently replace the data in the playback buffer for watching desired content. Searching requested video content from fragmentary distributed resources in P2P networks leads to long start delay and high-cost network bandwidth, which cannot ensure smooth playback and meet the demand of green communication. On the other hand, the pursuit of popular video content results in high load of system due to process request and schedule resources, reducing system scalability. Therefore, a light-duty solution which efficiently maintains and schedules resources carried by the mobile nodes and supports fast search for video content should be considered for video streaming service in MANETs.

Numerous researchers have shown great interest in high-efficiency resource sharing for video streaming system in wireless networks. For instance, QUVoD in [16], a Chord-based video sharing solution over VANETs, groups peers into a chained Chord structure in 4G networks in terms of the similarity of stored video chunks, which can achieve reliable supply and fast location of video resources. SURFNet in [17] is a tree-based video sharing solution in which the peers with long online times are grouped into an AVL tree and connect with an attached holder-chain whose items have similar video content. However, with increasing number of nodes, the high maintenance cost for structured topology (Chord/tree) limits the scalability of these solutions. For uncertain blowout of user access for video resources, enlarging the scale of server

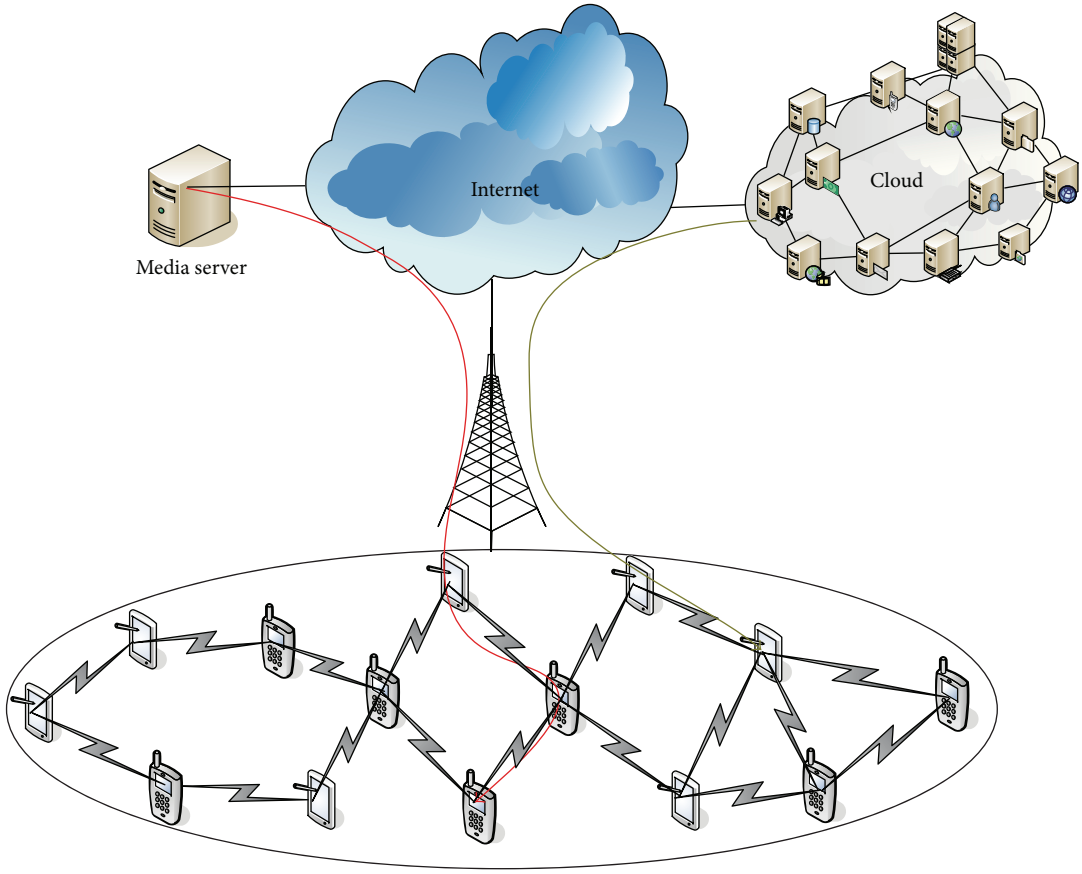


Figure 1: CSVD architecture.

cluster increases the cost of system deployment. SPOON in [18] is a community-based file sharing solution over MANETs. SPOON groups the peers into multiple communities in terms of the interest similarity, which can achieve high efficiency of content sharing. However, the stability of community structure relies on the capacities of community coordinator and determines system performance and maintenance cost of communities. Making use of the server to compensate insufficient bandwidth further limits the system scale.

Recently, cloud computing has become the most popular computing paradigm [19]. Providing on-demand server resources to users relies on the shared pool of servers in datacenters [20]. With rapid growth of mobile devices usage, mobile cloud streaming becomes a promising avenue for supporting high-performance and interactive streaming service [21, 22]. The most of cloud-based multimedia systems use the clouds to compensate insufficient capacities of upload bandwidth, computing and storage in the systems. However, the random request behavior of users for the video content leads to the increase in the complexity of cloud usage and the cost of interactivity between server and clouds. Moreover, making use of the clouds to meet the requirement of bandwidth brings expensive monetary cost.

In this paper, we propose *a novel Cloud-assisted Scalable Video Delivery solution over mobile ad hoc networks (CSVD)*. As Figure 1 shows, the clouds assist the media server to manage the hotspot resources to address the large-scale intensive request when the server cannot meet the demand of users for video resources. A novel estimation model of resource maintenance scale based on QoS-oriented dynamic balance between supply and demand is proposed, reducing monetary cost from rental resources in the clouds. A novel supplier scheduling mechanism that schedules suppliers in terms of their load and estimated processing capacity is proposed, reducing the lookup delay and balancing the load of suppliers. Simulation results show how CSVD achieves much better performance results in comparison with other state-of-the-art solutions.

2. Related Work

There have been numerous studies on P2P-based management and lookup optimization of resources for video streaming services in recent years.

The solutions based on structured content distribution topology are well known for resource lookup efficiency. For instance, QUVoD in [16] employs a group-based Chord structure to uniformly distribute the video content, where the peers which store similar content in the Chord overlay form a group. By using this structure, the request for sequential video

chunks can be addressed in a group, reducing chunk seeking traffic and balancing peer load. However, the increase in the number of nodes leads to high system load for maintaining the Chord overlay and limits QUVoD's scalability. SURFNet in [17] groups stable peers which have long online time and store superchunk-level video content into an AVL tree. A holder-chain which is composed of peers with similar video content is attached to a peer in the AVL tree in terms of the similarity of stored content. SURFNet makes use of the tree-based overlay topology to obtain nearly constant and logarithmic lookup time for seeking in a video stream or between different videos. The maintenance cost of peers relies on the stability of the AVL tree. However, the long online time cannot ensure the state stability of nodes in the tree. Therefore, with the increase in the number of peers, the maintenance of the tree structure also increases very much so that the system scalability and lookup efficiency are highly restricted. The structured overlay such as QUVoD and SURFNet can achieve fast lookup of resources and make full use of peers' upload bandwidth, but the high maintenance cost limits the system's scale and wastes network bandwidth.

The proposed mesh-based solutions with an unstructured topology have high system scalability. For instance, the authors of [23] proposed a mesh-based P2P streaming solution. Each peer selects the nodes as its neighbors according to different predefined policies. Because mutual contact between these nodes form a random graph, the system can perceive dynamic distribution process of video chunk and utilize created cluster of large-bandwidth peers to address intensive request of hotspot resources. Chang and Huang [24] proposed a mesh-based interleaved video frame distribution scheme to support user interactivity. However, the resource search in the mesh-based solutions employs gossip scheme which does not support fast resource lookup. The low performance of resource lookup does not ensure smooth playback. Moreover, the dissemination of gossip messages consumes mass network bandwidth.

Recently, some P2P file sharing solutions based on virtual communities have been proposed. For instance, SPOON [18] groups mobile nodes into a community in terms of common interest and frequent interaction between users. SPOON designs a role assignment for the community members to handle the file lookup both intracommunity and intercommunity and an file searching scheme for high-efficiency resource search in terms of user interest. However, SPOON makes use of files stored to estimate the interest similarity between nodes, which does not obtain high accuracy of interest similarity. The fragile community structure results in increasing maintenance cost of community members and a negative influence on file search efficiency. The mobility of nodes is not mentioned in SPOON so that the dynamic geographical distance between community members brings negative influence for content delivery. *C5* [25] groups the peers which request the same content and are near to each other into a community and collaboratively fetches content. The community members use a WLAN to deliver local resources with other members. Making use of WLAN interfaces to communicate with internal members can improve the delivery efficiency. However, the deployment environment of *C5* relies on the premise that a number of mobile nodes have close location with each other during a long period and subscribe the same content, so *C5* is difficult to be implemented in mobile networks. The increase in the community scale introduces the high maintenance cost for community members, so the capacities of community coordinator become the bottleneck of system scalability.

Inspired by community-based file sharing solutions, the community-based video streaming systems have attracted increasing research interests from various researchers. For instance, AMCV in [26] proposed a mini-community-based video sharing scheme in wireless mobile networks. AMCV groups the peers into a community in terms of the similarity of requested video chunks and uses an ant colony optimization-based community communication strategy that dynamically bridges communities to support fast search for resources. The interest-based peer groups can ensure high accuracy for predicting the resource demand of users to reduce the start delay, however, because the broker nodes in the communities need to maintain information of community members and handle the request messages from internal or external members. With increasing number of peers, AMCV's scale relies on the capacities of broker nodes; namely, the broker nodes in the communities cannot bear high load for the management of community members so as to limit system's scalability. PMCV in [27] proposed a novel performance-aware mobile community-based video delivery system over vehicular ad hoc networks. PMCV employs a mobile community detection scheme to group peers into a mobile community in terms of the similarity of playback and movement behavior between users. This scheme can obtain high stability of community structure and efficient video delivery. Moreover, PMCV makes use of a community member management mechanism to achieve high efficiency of resource lookup and low maintenance cost.

3. CSVD Detailed Design

3.1. Media Server. The media server stores several video resources to provide original video data for all mobile nodes in MANETs; namely, a video files set is $S_{\text{video}} = (v_1, v_2, \ldots, v_n)$. When a mobile node n_i joins the system or a system member requests a new video, it sends a request message to the server. The server uses a system member list to record the information of these request nodes, namely, $S_{\text{member}} = ((n_1, \text{VID}, tp_1), (n_2, \text{VID}, tp_2), \ldots, (n_m, \text{VID}, tp_m))$, whose items include the node ID, requested video ID, and timestamp. After the server receives the request message, it selects a supplier which has the minimum value of requested timestamp with the requester, ensuring stable logical link between supplier and requester and reducing the number of repetitive request messages. Moreover, in order to balance the load between suppliers and enhance the video delivery efficiency, the server considers the serving capacities and moving behavior of suppliers. The supplier scheduling algorithm is detailed in Section 3.3. When the requester receives the return message containing the information of the supplier, it connects with the supplier and fetches video content. If any

member leaves the system, it sends the quit message to the server. After the server receives a quit message of the system members, it removes the member information from S_{member}.

With increasing number of maintained members and request messages, the server cannot shoulder the load of maintaining node state and processing messages. The server requires the clouds to maintain the state of members which have the video content of high-frequency access. When the system members or mobile nodes request a popular video file, their request messages are redirected to the clouds which are responsible for assigning the appropriate supplier for the requesters.

3.2. Maintenance Scale. Renting cloud resources to maintain and schedule the video resources stored by the nodes in P2P networks brings monetary cost. In order to economically use clouds, the scale of maintained members should be kept within appropriate level in terms of the balance between supply and demand. Due to the variation of cached video content, the maintenance scale is dynamically regulated in terms of the resource requirements change. Let $S_{v_i} \Leftrightarrow (n_a, n_b, \ldots, n_t)$ be the set of system members which cache v_i, where the items in S_{v_i} are considered as candidate suppliers (CSs). The CSs receive the request message forwarded by the clouds and deliver the video content for the requesters. Let $\text{NR}(n_k)$ be the number of request messages which are received by a member n_k carrying v_i during a period time T_{n_k}. The request arrival rate (RAR) of n_k is defined as

$$\lambda_{n_k} = \frac{\text{NR}(n_k)}{T_{n_k}}, \tag{1}$$

where λ_{n_k} is the number of request messages received by n_k in unit time. The sum of RAR of all items in S_{v_i} is obtained according to

$$\sum_{c=1}^{|S_{v_i}|} \lambda_{n_c} = \lambda, \tag{2}$$

where $|S_{v_i}|$ returns the number of items in S_{v_i} and λ denotes the number of request messages received by the clouds in unit time and meets the Poisson distribution [20]. Because there is the mutual independence relationship between RAR of items in S_{v_i}, RAR meets the Poisson distribution. When n_k receives a request message, the event it connects with the request node and successfully delivers to the video data is considered as n_k handling the request message. The time of handling a request message can be defined as

$$T_{n_k}^{(p)} = (1 + \alpha)\, T_{n_k}^{(l)} + T_{n_k}^{(t)}, \quad \alpha > 0, \tag{3}$$

where α is a degeneration factor of denoting decrease in the handling capacities such as decreasing energy. With the decrease in the handling capacities of nodes, the value of $T_{n_k}^{(p)}$ increases. $T_{n_k}^{(l)}$ is the time of local handling request message and $T_{n_k}^{(t)}$ is the time of successful delivery of first video data. Let $\text{NH}(n_k)$ be the number of request messages which are handled by n_k during a period of time T_{n_k}. The request processing rate of n_k can be obtained according to

$$\mu_{n_k} = \frac{\text{NH}(n_k)}{T_{n_k}}, \tag{4}$$

where μ_{n_k} does not meet a specific distribution due to the unreliable link in the mobile environment and different performance of mobile devices. Each CS handles the request message according to the first-come-first-service rule. In order to ensure high QoS of system, the relationship between λ_{n_k} and μ_{n_k} needs to meet the following equation:

$$\min_{n_k \in S_{v_i}} \left\{ u_{n_k} - \lambda_{n_k} \right\} > 0. \tag{5}$$

The stay delay of request message in the message queue of supplier is defined as $T_{n_k}^{(s)} = T_{n_k}^{(l)} + T_{n_k}^{(w)}$, where $T_{n_k}^{(w)}$ is the delay of request message waiting to be processed. If the request processing rate is higher than the request arrival rate, $T_{n_k}^{(w)}$ is 0 s. The suppliers can fast handle the request message and deliver requested video data. Otherwise, if $T_{n_k}^{(w)} > 0$, the suppliers need to handle other request messages so that the increase in the startup delay of request nodes leads to the low level of user experience. In order to ensure the quality of user experience, if $T_{n_k}^{(w)} > 0$, the clouds need to maintain more resources to meet the demand of request nodes. Because the random quit of nodes results in decreasing the number of items in S_{v_i}, the scale of items in S_{v_i} should meet the following rule.

Rule 1. If the decreased number of items in S_{v_i} is greater than the increment of items in S_{v_i} in unit time or the request processing rate and request arrival rate of all items in S_{v_i} cannot meet (5), the clouds enlarge the scale of S_{v_i}.

As we know, once the request nodes receive the video data from the suppliers, they cache the video content and are considered as CSs. When the clouds need to enlarge the scale of S_{v_i}, these request nodes which have reliable state should preferentially be added into S_{v_i}. If the clouds cannot obtain more members (e.g., mass nodes request other video contents or quit the system), the clouds provide the video streaming for the request nodes.

3.3. Supplier Scheduling Mechanism. As Figure 2 shows, the supplier scheduling mechanism architecture relies on the serving capacity of CSs and the similarity of moving behavior between CSs and requester to select appropriate supplier. (1) Serving capacity of CSs: each CS makes use of the length of handling queue to calculate the expectation time of handling a request message and delivering video data. This expectation time is considered as the serving capacity of CS; namely, the low time of handling and delivery denotes strong serving capacity. (2) Similarity of moving behavior between CSs and requester: the CSs report the information of serving time and moving trace. The clouds/server use(s) the similarity of moving trace of CSs and requester as weight value of serving

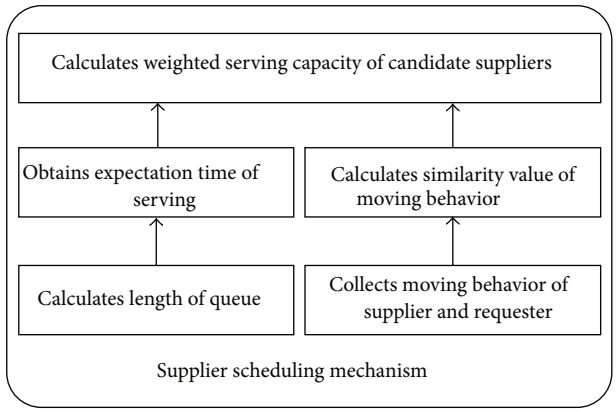


Figure 2: Supplier scheduling mechanism architecture.

capacity of CSs. The measurement of weighted serving capacity can ensure high efficiency of handling request message and delivering video data.

When the clouds/server receive(s) the request messages of members, it selects the appropriate CSs from S_{v_i}. This is because each mobile device is limited by the energy, bandwidth, computation, and storage. The clouds/server not only forward(s) these request messages in terms of the capacities of mobile devices, but also balance(s) the load between CSs. Each CS handles the request message according to the first-come-first-service rule; namely, the process of handling request message meets the $M/G/1$ queuing model. When the request message of a member is forwarded to n_k, in terms of Pollaczek-Khintchine (P-K) formula, the number of items in the handling queue of n_k can be defined as

$$NQ\left(n_k\right) = \rho_{v_i} + \frac{\rho_{v_i}^2 + \lambda_{v_i}^2 \sigma_{v_i}}{2\left(1 - \rho_{v_i}\right)}, \tag{6}$$

where σ_{c_i} is the variance of time of n_k handling received request messages during T_{n_k}. ρ_{v_i} denotes the time of n_k handling the request messages and is defined as

$$\rho_{v_i} = \lambda_{v_i} E\left(T_{n_k}^{(p)}\right)_{v_i}, \tag{7}$$

where $E(T_{n_k}^{(p)})_{v_i}$ is the expectation value of time of n_k processing all request messages during T_{n_k}. We make use of Little formula [28] to obtain the average residence time of request message in the queue of message processing of n_k according to

$$W_s = \frac{L_s}{\lambda_{v_i}} = E\left(T_{n_k}^{(p)}\right)_{v_i} + \frac{\lambda_{v_i} E\left(T_{n_k}^{(p)}\right)_{v_i}^2 + \lambda_{v_i}^2 \sigma_{v_i}}{2\left(1 - \lambda_{v_i} E\left(T_{n_k}^{(p)}\right)_{v_i}^2\right)}. \tag{8}$$

Further, the expectation time of serving a requester is obtained according to

$$T_{n_k}^{(s)} = W_s + E\left(T_{n_k}^{(d)}\right)_{v_i}, \tag{9}$$

where $T_{n_k}^{(d)}$ is the time of n_k delivering first data of c_i, $E(T_{n_k}^{(d)})_{v_i}$ is the expectation value of $T_{n_k}^{(d)}$, and $T_{n_k}^{(s)}$ also is considered as the serving capacity of n_k. The clouds can obtain a set of serving capacities of all items in S_{v_i}; namely, $S_{\mathrm{SC}} = (T_{n_a}^{(s)}, T_{n_b}^{(s)}, \ldots, T_{n_t}^{(s)})$. Moreover, we consider the similarity of moving trace between supplier and requester and transform (9) to (10):

$$\mathrm{SC}_{n_k} = \frac{1}{T_{n_k}^{(s)} + \cos\left(w_{n_k}\right) + 1}, \tag{10}$$

where w_{n_k} is a similarity of moving trace of nodes. Each node n_i periodically updates the information of one-hop neighbor nodes and considers them as encountered nodes; namely, $L_e = (n_1, n_2, \ldots, n_k)$. Let f_k denote the number of encounter of n_i and n_k. The mean value of encounter number of all nodes in L_e is defined as

$$\overline{f} = \frac{\sum_{c=1}^{k} f_c}{k}. \tag{11}$$

If $f_k > \overline{f}$, n_k is a frequently encountered node of n_i. n_i extracts the information of frequently encountered nodes and constructs a vector $mt_i = (n_a, n_b, \ldots, n_v)$ to denote moving trace. The similarity value of moving trace between n_i and n_k is defined as

$$w_{n_k} = \frac{\left|mt_i \cap mt_k\right|}{\max\left[\left|mt_i\right|, \left|mt_k\right|\right]}. \tag{12}$$

The new system members which request a video content have not served other nodes; namely, $T^{(s)} = 0$. The item in S_{v_i} with $\max\{\mathrm{SC}_{n_a}, \mathrm{SC}_{n_b}, \ldots, \mathrm{SC}_{n_t}\}$ is considered as a CS which has the strongest serving capacity and is selected as the supplier. Because $T^{(s)}$ of new system members is 0, the values of their SCs are less. The probability of new system members becoming suppliers is higher than other members, which can balance the load of system members.

3.4. Model Implementation. In the process of scheduling the request, the clouds/server need(s) to balance the load of CSs. The clouds/server remove(s) the CSs whase surplus bandwidth cannot meet the playback rate or request processing rate (RPR) is lower than the request arrival rate from S_{v_i}. If the removed CSs recover sufficient bandwidth and RPR, they are added into S_{v_i}. When the clouds/server receive(s) a request message r_x, they/it select(s) a CS n_k with $\max\{\mathrm{SC}_{n_a}, \mathrm{SC}_{n_b}, \ldots, \mathrm{SC}_{n_t}\}$ as the supplier. The clouds/server make(s) the decision of adding the requester into S_{v_i} in terms of Rule 1. In the process of handling each request message, n_k records receive the message number and the time of processing each message during a period time T_{n_k}, namely, $\mathrm{NR}(n_k)$, $\mathrm{NH}(n_k)$, $T_{n_k}^{(l)}$, and $T_{n_k}^{(t)}$. n_k further obtains the values of λ_{n_k}, μ_{n_k}, and $T_{n_k}^{(p)}$. After n_k finishes the delivery of video data for the requester, n_k records the total time $T_{n_k}^{(d)}$ of data transmission and sends a message containing λ_{n_k}, μ_{n_k}, $T_{n_k}^{(p)}$, $T_{n_k}^{(d)}$ and moving trace to the clouds/server. The clouds/server use(s) these parameter values to update the $T_{n_k}^{(s)}$ of n_k. The clouds/server make(s) use of moving traces of requester

(1) receives request message of requester n_u;
(2) **for** ($i = 0$; $i < t$; i++)
(3) **if** bandwidth and RPR of $S_{v_j}[i]$ meet the demand
(4) calculates serving expectation time of $S_{v_j}[i]$ by (9);
(5) calculates moving similarity of $S_{v_j}[i]$ and n_u by (11);
(6) obtains measurement value SC_i of capacity of $S_{v_j}[i]$;
(7) adds SC_i into result set R;
(8) **end if**
(9) **end for**
(10) forwards message to supplier n_v with minimum in R;
(11) n_v returns response message to n_u;
(12) n_v sends video data to n_u;

Algorithm 1: Search process of video file v_j.

and CSs to calculate similarity of moving behavior between them. By investigating the similarity of moving behavior and serving capacity of CS, the system can achieve fast response for the request and high-efficiency delivery of video data, ensuring smooth playback experience. The pseudocode of the process of resource search is detailed in Algorithm 1 whose computation complexity is $O(n)$.

4. Testing and Test Results Analysis

We investigate the performance of the proposed CSVD in comparison with SPOON [18], a state-of-the-art P2P-based file sharing solution. The number of video files is 100 and the length of each file is 60 s. CSVD was modeled and implemented in NS-2, as described in the previous sections.

4.1. Testing Topology and Scenarios. Table 1 lists some NS-2 simulation parameters of the MANET for the two solutions. We created 100 user viewing logs; namely, each user randomly accesses 20 video files and the viewing period time is set to random value. Moreover, when the users have watched a video in terms of the random playback period time, they continue to request new video file. 100 mobile nodes play video file following 100 logs and uniformly join the system following the Poisson distribution from 0 s to 360 s. After the nodes arrive at the target location, they continue to move to new target location in new assigned speed. In SPOON, 100 nodes randomly store 20 files with different keywords before they join system and the number of intersection of their local files is 100. The requested files corresponding to 100 logs are not included in their local files. We define some parameter value for SPOON: $\pi = 1$, $\Omega = 1$, $S_{\max} = 1$, $h_1 = 30$, $h_2 = 30$, and $T_G = 1$.

4.2. Performance Evaluation. The performance of CSVD is compared with that of SPOON in terms of average file seek delay (AFSD), packet loss rate (PLR), throughput, video quality, and overlay maintenance cost, respectively.

(1) AFSD. The difference value between the time when a node requests a video file and the time when the node receives first video data is considered as the file seek delay. The mean value of the cumulative sum of delay values during a time interval denotes AFSD.

As Figure 3 shows, SPOON's AFSD curve experiences slow increase from $t = 300$ s to $t = 540$ s after fast decrease from $t = 60$ s to $t = 240$ s with the delay range between 2.6 s and 3.9 s. SPOON's curve finally decreases to roughly 3.6 s from $t = 540$ s to $t = 600$ s. CSVD's curve shows slow rise with slight fluctuation, which slowly decreases to 1.4 s from $t = 120$ s to $t = 180$ s. The curve has a slow increase from $t = 210$ s to $t = 510$ s and decreases to roughly 2 s at $t = 600$ s, which maintains relatively low level (the values are roughly 35% lower than those of SPOON).

Table 1: Simulation parameter setting for MANET.

Parameters	Values
Area	800×800 m^2
Channel	Channel/WirelessChannel
Network interface	Phy/WirelessPhyExt
MAC interface	Mac/802_11
Number of mobile nodes	400
Number of mobile nodes playing video	100
Mobile speed range of nodes	[0, 30] m/s
Simulation time	600 s
Signal range of mobile nodes	200 m
Default distance between server and nodes	6 hops
Default distance between clouds and nodes	6 hops
Transmission protocol	UDP
Wireless routing protocol	DSR
Bandwidth of server	20 Mb/s
Bandwidth of mobile nodes	10 Mb/s
Transmission rate of video data	128 kb/s
Travel direction of mobile nodes	Random
Pause time of mobile nodes	0 s

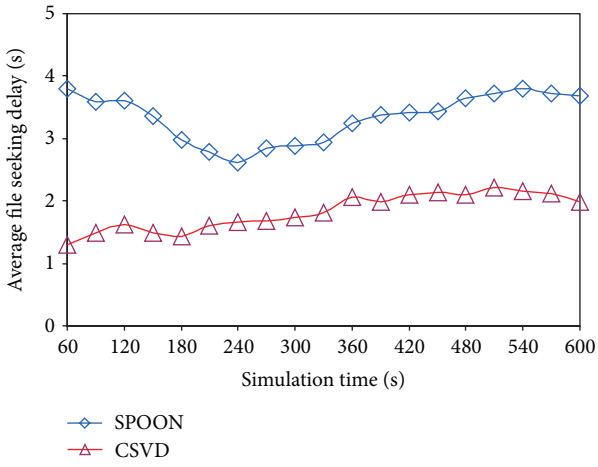


Figure 3: AFSD against simulation time.

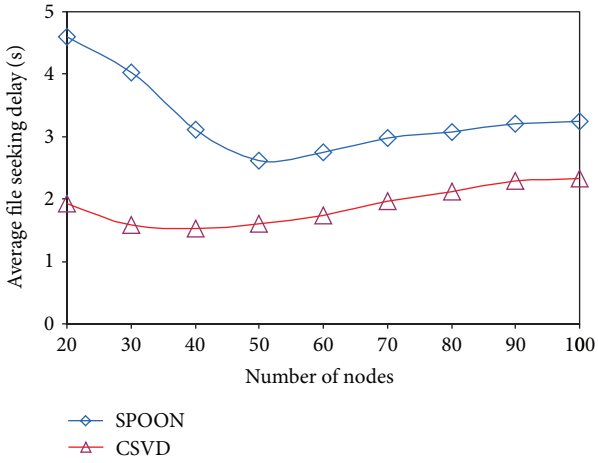


Figure 4: AFSD against number of nodes.

Figure 4 presents the AFSD variation with increasing number of nodes which have joined the system. The blue curve corresponding to the SPOON's results has both higher values and larger fluctuation with the increase in the number of nodes (the values are between 2.6 s and 4.6 s). The CSVD results, illustrated with red curve, have values between 1.5 s and 2.3 s, with lower variations than SPOON's results.

Initially, the number of members in communities is relatively less, so SPOON only uses the interest-oriented routing algorithm (IRA) to search the video files from foreign communities. The request messages are continually forwarded between the mobile nodes, which leads to the high AFSD. With the increase in the number of nodes, the community members require the community coordinators to search desired files. If the requested files are in the intracommunity, the intracommunity search speeds up the process of resource location, so SPOON's AFSD values can fast decrease and keep the relatively low level. However, when the members do not fetch the requested files from the intracommunity, they rely on the community ambassadors to search the resources from foreign communities. Therefore,

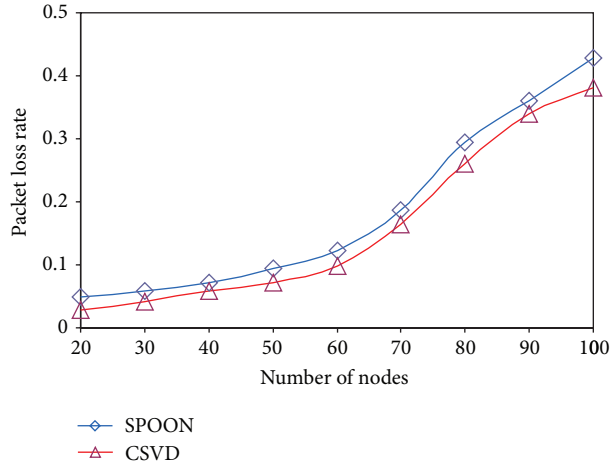


Figure 5: PLR against number of nodes.

SPOON's AFSD values show fast rise with increasing number of search messages. Moreover, SPOON does not consider the mobility of nodes, so that the dynamic change of geographical location between the requesters and the suppliers brings negative influence for the delay of data delivery. In CSVD, the server and clouds are responsible for handling the request messages and scheduling the available resources for the requesters, so the fast response at the server and clouds side reduces the delay of video file seeking. Moreover, CSVD investigates the similarity of moving trace between requesters and suppliers. The stable mobility between requesters and suppliers enhances the efficiency of video data delivery. On average CSVD's results are better than those of SPOON.

Packet Loss Rate (PLR). The ratio between the number of packets lost in the process of video data transmission and the total number of packets of video data sent is defined as PLR.

As Figure 5 shows, the curves corresponding to CSVD and SPOON show a rise trend with increasing number of nodes. The results of CSVD and SPOON maintain low levels when the number of nodes increases to 60 and represent fast rise from 70 to 100. However CSVD's PLR is roughly 20% better than the values associated with SPOON.

Figure 6 shows the variation of PLR values with the increase in mobility speed of mobile nodes in MANETs. The CSVD results have both low values and slight increase from [0, 5] to (25, 30] and are between 0.15 and 0.22. The blue bars corresponding to the SPOON results maintain high levels and are between 0.16 and 0.24. The CSVD PLR values are roughly 10% lower than the results of SPOON.

Small scale system members only consume the small number of network bandwidth, so the PLR curves of two systems keep slight rise. With increasing number of system members, the members require more network bandwidth so that the network congestion results in high PLR. On the other hand, the low mobility of mobile nodes brings slight variation in the PLR due to the slow change in the geographical distance. With the increase in the mobility of mobile nodes, the fast variation of geographical distance between requesters

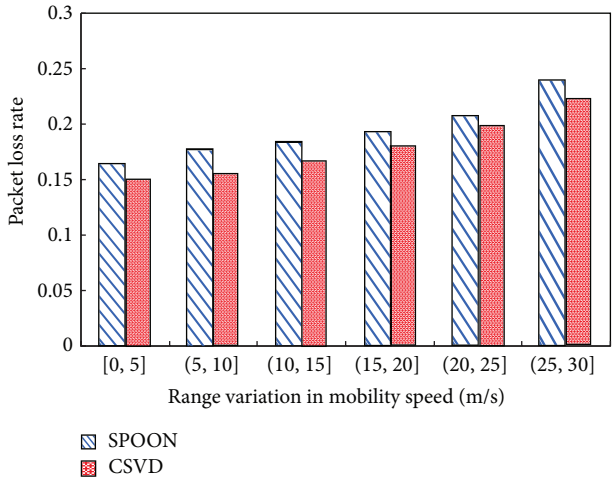


Figure 6: PLR against range variation in the mobility speed of mobile nodes.

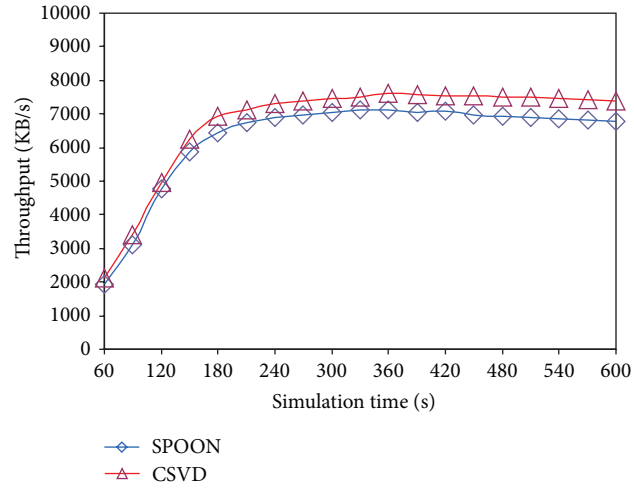


Figure 7: Throughput against simulation time.

and suppliers leads to high probability of packet loss. The community coordinators and ambassadors in SPOON do not consider the mobility of nodes in the process of the assignment of suppliers for the requesters. The efficiency of data delivery in SPOON is influenced by the increase in the geographical distance between requesters and suppliers; namely, the communication with long distance increases the probabilities of wireless link break and packet loss. In CSVD, the server and clouds match the similarity between requesters and suppliers and assign the suppliers which have the most similar moving behavior with requesters to provide requested video streaming. The stable geographical distance between them ensures high transmission performance such as low delay and reduced PLR. Therefore, CSVD PLR is kept at low level.

Average Throughput. The total number of packets received in the overlay during a certain time period divided by the length of this time period is defined as the average throughput.

Figure 7 shows the variation of average throughput of SPOON and CSVD with increasing simulation time. The curves corresponding to two systems show similar rise trajectory; namely, they experience a fast rise from $t = 60$ s to $t = 180$ s and a decreasing trend from $t = 210$ s to $t = 600$ s. The increment of CSVD results is higher than that of SPOON and the peak value of SPOON's curve at 360 s is larger than that of CSVD.

The mobile nodes request the video content following a Poisson distribution from $t = 0$ s to $t = 360$ s. The increase in the number of nodes leads to numerous transmitted video data packets in network. The two systems have fast rise trend from $t = 0$ s to $t = 180$ s and slow increase from $t = 210$ s to $t = 360$ s and reach peak values at $t = 360$ s. When the data traffic of transmitting video content is larger than the bandwidth provided by the network, numerous data packets are discarded, reducing the throughput. In SPOON, the delivery without considering mobility of requesters and suppliers is subjected to serious negative influence; namely, the communication quality in the transmission path decreases due to the network congestion. The values of SPOON throughput maintains low level. In CSVD, the requesters receive video data from the suppliers assigned by the server and clouds in terms of similarity of their moving behavior. This ensures that the delivery performance of CSVD is positive, including high throughput, low PLR, and low delay. Therefore, the throughput values of CSVD are larger than those of SPOON.

Video Quality. The peak signal-to-noise ratio (PSNR) [29] is used to denote the video quality, measured in decibels (dB), and is estimated according to (13):

$$\mathrm{PSNR} = 20 \cdot \log_{10}\left(\frac{\mathrm{MAX\_Bit}}{\sqrt{(\mathrm{EXP\_Thr} - \mathrm{CRT\_Thr})^2}}\right), \quad (13)$$

where EXP_Thr is the average throughput expected from the delivery of the video content, MAX_Bit is the average bitrate of the video stream as resulted from the encoding process, and CRT_Thr denotes the actual throughput measured during delivery. MAX_Bit and EXP_Thr are 128 kb/s in terms of simulation settings, respectively. We calculate PSNR of single video streaming corresponding to every node according to the throughput with increasing number of nodes.

Figure 8 shows the average video quality of single video streaming corresponding to each node with increasing number of the nodes. The results of SPOON and CSVD show the fall trend. The red bars corresponding to CSVD's results have the range [8, 30] and are roughly 10% higher than SPOON. The blue bars corresponding to SPOON's results have a fast decrease from the peak value 26 dB to a minimum of 7 dB.

PSNR reflects the video quality perceived by users. The delivery of video content relies on the retransmission of mobile nodes in MANETs. The small number of system members do not consume too many bandwidths of the forwarding nodes, so that the PLR and delay maintain low level (PSNR also keeps high level). With increasing number of nodes,

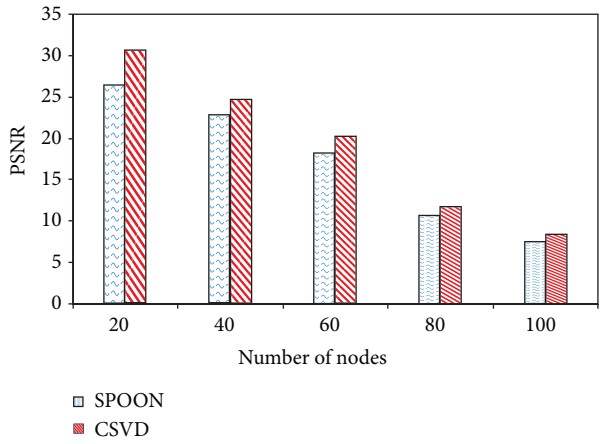


Figure 8: PSNR against number of nodes.

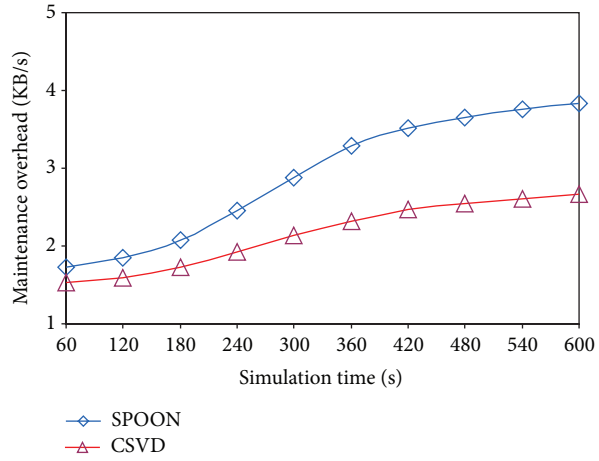


Figure 9: Maintenance overhead against simulation time.

the requirement of high bandwidth for the video streaming consumes the network bandwidth and triggers the network congestion. At the moment, the high PLR and low throughput bring low PSNR of single video streaming corresponding to each node. SPOON neglects the mobility of requesters and suppliers, so that the transmission performance of video data is subjected to serious influence from network congestion. In CSVD, the requesters and suppliers have similar moving behavior by the assignment of the server and clouds. The high-efficiency delivery of video data can obtain, respectively, low PLR. The PSNR values of CSVD are better than those of SPOON.

Maintenance Overhead. The average bandwidth which is used by the sent messages for maintaining the overlay topology is considered as the maintenance overhead.

As Figure 9 shows, the overlay maintenance overhead values of two systems have similar changing trend with increasing number of mobile nodes. SPOON's results fast increase from $t = 240$ s to $t = 600$ s after a slow rise from $t = 60$ s to $t = 240$ s. The curve corresponding to CSVD maintains a slow increasing trend and has a low increment, with values roughly 30% lower than those of SPOON.

The nodes in SPOON are grouped into multiple communities in terms of the interest. The community coordinators are responsible for maintaining the state and stored resources of members and handling the request messages of files. Although SPOON employs a periodical state maintenance mechanism, the increasing scale of maintained nodes leads to high load for the coordinators. Mass messages of state maintenance and files request consume a lot of energy and bandwidth of coordinators, so the capacities of coordinators become the bottleneck of system scalability. Moreover, the frequent exchange of members' state messages and broadcast messages of coordinators' replacement increase the maintenance cost of communities. SPOON's maintenance overhead values fast increase with increasing number of nodes. Unlike SPOON (all nodes are maintained by the coordinators), the server and clouds in CSVD only maintain the playback state of nodes, reducing the number of exchanging messages. The clouds also dynamically regulate the number of maintained nodes in terms of the balance between supply and demand. Therefore, CSVD's maintenance overhead for the overlay topology keeps lower level than that of SPOON.

5. Conclusion

In this paper, we propose a novel Cloud-assisted Scalable Video Delivery solution over mobile ad hoc networks (CSVD). CSVD improves the scalability of light-duty video streaming system with the help of the clouds by maintaining the peers which carry hotspot resources in P2P networks to ensure smooth playback experience of users. The estimation model of resource maintenance scale can regulate the utilization of cloud resources in terms of dynamic balance between supply and demand. The supplier scheduling mechanism can assign resource suppliers in terms of their load and serving capacity. The results show how CSVD ensures lower average file seek delay, lower packet loss rate, higher throughput, higher video quality, and lower overlay maintenance cost than SPOON.

Conflict of 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 61303053, in part by the Project of Beijing Municipal Commission of Education under Grant KM201510028016, and in part by the Beijing Natural Science Foundation under Grant 4142037.

References

[1] M. Qi and J. Hong, "AAPP: an anycast based AODV routing protocol for peer to peer services in MANET," *International Journal of Distributed Sensor Networks*, vol. 5, no. 1, p. 48, 2009.

[2] S. Jia, C. Xu, G.-M. Muntean, J. Guan, and H. Zhang, "Cross-layer and one-hop neighbour-assisted video sharing solution in mobile Ad Hoc networks," *China Communications*, vol. 10, no. 6, pp. 111–126, 2013.

[3] Q. Guan, F. R. Yu, S. Jiang, V. C. M. Leung, and H. Mehrvar, "Topology control in mobile Ad Hoc networks with cooperative communications," *IEEE Wireless Communications*, vol. 19, no. 2, pp. 74–79, 2012.

[4] S. Jin and S. Choi, "A seamless handoff with multiple radios in IEEE 802.11 WLANs," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 3, pp. 1408–1418, 2014.

[5] 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.

[6] L. Zhou, Z. Yang, Y. Wen, and 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.

[7] C. Xu, Z. Li, J. Li, H. Zhang, and G.-M. Muntean, "Cross-layer fairness-driven concurrent multipath video delivery over heterogenous wireless networks," *IEEE Transactions on Circuits and Systems for Video Technology*, no. 99, 2014.

[8] W. Zhang, Z. Li, and Q. Zheng, "SAMP: supporting multi-source heterogeneity in mobile P2P IPTV system," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 4, pp. 772–778, 2013.

[9] M. A. Hoque, M. Siekkinen, and J. K. Nurminen, "Energy efficient multimedia streaming to mobile devices-a survey," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 579–597, 2014.

[10] 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.

[11] Y. Chen, B. Zhang, C. Chen, and D. M. Chiu, "Performance modeling and evaluation of peer-to-peer live streaming systems under flash crowds," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 2428–2440, 2014.

[12] L. Zhou, Y. Zhang, K. Song, W. Jing, and A. V. Vasilakos, "Distributed media services in P2P-based vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 692–703, 2011.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] D. Wang and C. K. Yeo, "Superchunk-based efficient search in P2P-VoD system," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 376–387, 2011.

[18] 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.

[19] F. Liu, S. Shen, B. Li, and H. Jin, "Cinematic-quality VoD in a p2p storage cloud: design, implementation and measurements," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 214–226, 2013.

[20] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. M. Lau, "CloudMedia: when cloud on demand meets video on demand," in *Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS '11)*, pp. 268–277, July 2011.

[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] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 870–883, 2013.

[23] A. P. C. Da Silva, E. Leonardi, M. Mellia, and M. Meo, "Chunk distribution in mesh-based large-scale P2P streaming systems: a fluid approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 3, pp. 451–463, 2011.

[24] C.-L. Chang and S.-P. Huang, "The interleaved video frame distribution for P2P-based VoD system with VCR functionality," *Computer Networks*, vol. 56, no. 5, pp. 1525–1537, 2012.

[25] L. Tu and C.-M. Huang, "Collaborative content fetching using MAC layer multicast in wireless mobile networks," *IEEE Transactions on Broadcasting*, vol. 57, no. 3, pp. 695–706, 2011.

[26] 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.

[27] 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*, 2014.

[28] F. J. Beutler, "Mean sojourn times in Markov queueing networks: little's formula revisited," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 233–241, 1983.

[29] 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.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 236354, 11 pages
http://dx.doi.org/10.1155/2015/236354

Research Article

ECMTADR: Energy Conservative Multitier Architecture with Data Reduction for Cluster-Based Wireless Sensor Networks

Taner Cevik

Department of Computer Engineering, Fatih University, 34500 Istanbul, Turkey

Correspondence should be addressed to Taner Cevik; tcevik@fatih.edu.tr

Received 21 January 2015; Revised 19 March 2015; Accepted 26 March 2015

Academic Editor: Changqiao Xu

Regarding the energy shortage problem of wireless sensor networks (WSNs), various schemes and protocols are proposed to prolong the network lifetime. Data communication is undoubtedly the most important determinant for this energy scarce network type. In this paper, we propose a sophisticated architecture comprising data reduction, load balance, and topology control. Data reduction is ensured by the parameter spatial correlation proximity range (SCPR) that can be adjusted statically at the setup phase or dynamically revised depending on the necessities in the network. Four-layer virtual architecture is applied for implementing topology control. Furthermore, network area is partitioned into fixed-size hexagonal clusters. Depending on the regions in which the clusters take place, cluster heads (CHs) are elected from the respective subregions of the clusters. Load balance is achieved by considering residual energies and distances to the sink during both CH election and data transmission stages. Aggregated data in each cluster is transmitted towards the sink by using a load balancing single-hop intercluster routing protocol instead of direct transmission as offered in LEACH. Simulation results demonstrate that the proposed architecture ECMTADR shows almost 50% better performance in terms of energy conservation and network lifetime when it is compared to LEACH and HEED.

1. Introduction

Recent technological developments in areas such as microelectronic, signal processing, and communication protocols have enabled sensor nodes to be produced cheaply with extremely small sizes which led wireless sensor networks to be deployed, self-organized, and activated rapidly with conceivable maintenance costs. This, in turn, provided to WSNs a broad range of application areas including industry, military, agriculture, health care, and sports [1–4]. Fortunately, it has recently become possible to perform dangerous and time-consuming jobs for humankind by computers in a very short time with higher accuracy and lower costs than before [5].

The mandate of WSNs consists of three stages: gathering data from the environment, in-processing (optional), and transmitting raw data (results) to the data collection center (sink). Obviously, a sensor node consists of four subunits: sensing subunit, processing subunit, communication subunit, and a power source. Among these subunits, the radio communication subsystem is the primary energy consumer of a sensor node [6]. The amount of energy consumed by the radio communication subunit during transmission of one bit is equal to the amount of energy consumed during transaction of 1000–10000 bits inside the processing subunit [7].

In order to achieve low cost operation and minimal maintenance, sensor nodes are produced in very small sizes which begot the fundamental challenge to overcome, that is, energy constraint [8]. The protocols and architectures developed for traditional wireless ad hoc networks cannot be applied to WSNs. Staff employed in traditional wireless networks is not as energy scarce as the ones that operate in WSNs. Therefore, the energy constraint problem is not the primary determinant during protocol and architecture description. Regarding the type of the application, hundreds, even thousands, of sensor nodes are scattered to the environment [9]. In many cases, substituting the energy depleted node for the new one can be dangerous, infeasible, time-consuming, costly, or impossible [10]. Thus, depending on the type of the application, WSNs should maintain durability long enough without any human

intervention after the initial deployment stage [11]. General acceptance for network lifetime definition is the time that the first node depletes energy. Thus, WSN-specific software and architecture design is crucial in order to maximize network lifetime [12].

Due to the aforementioned reasons, energy efficiency has received a great deal of attention from both academy and industry and since communication is the determinant energy consumer, the vast majority of the efforts have been devoted to improving longevity of sensor networks [13]. In this context, clustering is one of the most promising approaches among the ideas that have been proposed so far. Nodes are virtually organized into groups called clusters centrally or in a distributed manner. Energy consumption increases exponentially in proportion to the increase in the amount of data transmitted on the network. Thus, the point is to reduce the number of messages conveyed, thereby eliminating data redundancy [14]. By assigning some tasks to a group of nodes, not to just a single node, helps the total load to be shared. For event-based applications, an event can be recognized and the same information may be obtained by multiple nodes. Instead of all nodes in the sensing range sending their obtained data, it is sufficient for just a single node or groups of nodes to relay data towards the sink. In each cluster, a single node is elected as a cluster head (CH). CHs take the responsibility for gathering data from other plain staff and relaying the aggregated data to the sink. Aggregated data can be filtered, processed, or directly transmitted in the raw state to the sink that is application-specific and can be left to the discretion of the software embedded in the CH.

In this paper, we present a sophisticated cluster-based architecture regarding energy efficiency at each stage of lifetime such as CH election and intercluster routing. Nodes are organized into hexagonal fixed-size clusters statically. Each cluster is also partitioned into six equal-sized cells. Furthermore, a higher layer virtual segmentation of the topology is performed for preventing redundant retransmissions. Depending on the region in which a cluster falls, CHs are elected among the nodes falling into the cells facing the sink. CH selection is performed at the beginning of each round depending on both residual energy levels and the distances of the nodes to the sink. At the beginning of each round, nodes gather data from the environment. However, not all of them send their data to their respective CHs. A parameter called SCPR is considered. Each node starts a timer at the beginning of each data collection period. The timer of the node with a larger residual energy level expires earlier. And other nodes located in the SCPR of that node refrain from sending their data to the CH. By this means, redundant energy consumption and bandwidth invasion are prevented. During the interrouting process again energy levels of the next-hop candidates are considered.

The rest of the paper is organized as follows. Section 2 reviews existing protocols related to clustering in WSNs. Sections 3-4 provide an overview of the system model and the proposed scheme ECMTADR, respectively. Section 5 presents the simulation-based evaluation of the proposed architecture. Finally, Section 6 concludes the paper and provides an outline of future directions.

2. Motivation

Depending on the application type, hundreds and even thousands of nodes are utilized in a WSN. Particularly, for application areas where periodical data acquisition is required, an excessive amount of data transmission occurs in the network. As mentioned in the previous section, to operate at low costs for a satisfactory lifetime, sensor nodes of very small sizes are produced. An ultimate outcome is that a limited amount of resources such as antenna and power supply can be embedded in this restricted node body. The primary vital reflection of this resource constraint is the limited communication capacity.

Though WSNs are a type of ad hoc network, conventional methods, protocols, and architectures utilized for classical wireless networks cannot be employed in WSNs. WSNs are generally utilized for gathering scalar data from the environment. The size of the acquired data mostly does not exceed 50 bytes, whereas traditional ad hoc networks cover much larger data packets in greater amounts. Furthermore, sensor nodes are equipped with radios that can transfer messages at most 250 kbps. More competent radios entail additional energy expenditure. Thus, in order not to violate energy conservation, low capacity devices are preferred, which, in turn, causes low bandwidth. Therefore, in order to accomplish the assigned task at reasonable costs for satisfactory periods with sufficient rates, degradation in the quantity of transmitted data is essential.

In this context, a number of solutions are proposed thus far. The most two promising ones among these suggestions are data aggregation and reduction. Data reduction is performed locally which is called in-node processing. On the other hand, depending on the application type, data reduction may not always be possible. Another promising alternative solution is configuring the network hierarchically rather than utilizing a flat network topology. Clustering has been trailed by a broad range of research activities in order to achieve energy efficiency and network scalability. Sensor nodes are organized into groups called clusters. Ordinarily, one of the nodes in the cluster is elected as CH. Other nodes are called plain nodes. Plain nodes gather data from the environment and transmit it to their corresponding CH. After receiving data from plain nodes, CH endeavors to relay the aggregated data to the data collection center that is called a sink. Hence, just a single node in each cluster is responsible for the transfer of data to the sink which yields efficient common resource sharing that is wireless medium.

Several research activities have focused on cluster-based WSNs. Three remarkable issues about cluster-based WSNs are as follows.

(i) Clustering algorithm, which defines the methodology of dividing the network into clusters virtually. Clustering may be done statically at the beginning during setup phase and dynamically at regular intervals or when a certain condition occurs.

(ii) Cluster head election algorithm, which identifies the method of electing the optimum CH candidate. For the purpose of sharing the burden of holding

the responsibility of supervising a cluster of nodes, CH election should be made periodically or with regard to occurrence of a condition.

(iii) Clustering routing algorithm, which describes the way of conveying aggregated messages to the sink. Two challenges to be overcome are intracluster routing and intercluster routing which define the methodology of conveying data from plain nodes to CHs and from CHs to the sink, respectively.

The most prominent of the relevant studies are briefly discussed in [15–21].

3. System Model

3.1. Network Model. There are n nodes (set of nodes $\rightarrow |V| = n$) deployed over a square shaped region randomly. A single sink with unconstrained energy supply is positioned at the center of the topology. Network can be represented by a graph model $G = (V, E)$. E denotes the set of links connecting the nodes in the network. Link between two nodes u and v ($u, v \in V$) is represented by ($e \rightarrow (u, v)$, $\{e \in E \mid d(u, v) <= r\}$). $d(u, v)$ which is also equal to the length ($\|u, v\|$) of the link (u, v) denotes the Euclidean distance between the nodes u and v. r represents the coverage radius of the radio which is the maximum distance that a bit of data can be transmitted with a reasonable SINR value. Links are bidirectional and half-duplex. That is, each link between two nodes represents a single communication channel on which one-way transmission can occur at a time as shown in Figure 1.

Propagation delay and other parameters such as SINR and loss rate for links (channels) $e(u, v) = e(v, u)$ are assumed to be the same for both directions which, in turn, results in

$$t_2 - t_1 = t_3 - t_2$$
$$E_{\text{snd}}(u, v) = E_{\text{snd}}(v, u). \tag{1}$$

Each node owns a unique id number assigned in a distributed manner at the setup phase. Uniqueness is achieved by constituting the id number by the relative (x, y) coordinates of the node in the topology. Each node is assumed to know its relative position. That can be achieved by equipping each node with a low power GPS device or by means of applying special signal processing techniques [22].

3.2. Radio Model. In this work, we use the first-order radio model as utilized by many works proposed in the literature [23, 24]. The amount of energy consumed during send and receive operations is given below:

$$E_{\text{snd}}(l, d) = E_{\text{snd-elec}}(l) + E_{\text{snd-amp}}(l, d), \tag{2}$$

where $E_{\text{snd-elec}}$ and $E_{\text{snd-amp}}$ represent the energies dissipated during running the transmitter circuit and amplification for achieving a reasonable SINR value, respectively:

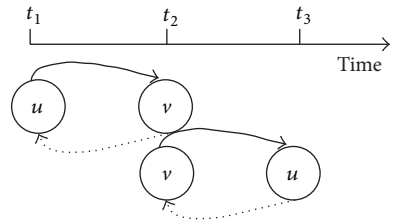


Figure 1: Symmetric half-duplex link.

$$E_{\text{snd}}(l, d) = \begin{cases} (l * E_{\text{elec}}) + (l * \varepsilon_{\text{fs}} * d^2) & d < d_0 \\ (l * E_{\text{elec}}) + (l * \varepsilon_{\text{mp}} * d^4) & d \geq d_0 \end{cases} \tag{3}$$

$$E_{\text{rcv}} = l * E_{\text{elec}} \tag{4}$$

$$d_0 = \sqrt{\frac{\varepsilon_{\text{fs}}}{\varepsilon_{\text{mp}}}}. \tag{5}$$

Equations (3)-(4) identify the energy dissipated by the nodes during transmit and receive operations. In the equations, l denotes the number of bits to be transmitted and d represents the Euclidean distance between the communicating pairs. As is known, if the distance between two nodes is below the threshold distance (5) which is denoted by d_0, the energy consumed by the sender node is calculated according to the first part of (3). Otherwise, the energy consumed by the sender node is calculated according to the second part. It is obvious that if the distance between the communicating pairs increases, the amount of energy consumed rises exponentially.

4. ECMTADR

In this section we briefly clarify the details of our proposed architecture ECMTADR. The first section describes the clustering method employed. Subsequent sections introduce the methodologies suggested for neighborhood definition, data reduction, CH election, intracluster communication, and intercluster communication processes, respectively.

4.1. Clustering. The network area is configured into a four-level hierarchy. In the first step, topology is zoned into 8 regions as depicted in Figure 2. During CH election stage, nodes will only be selected among the nodes appearing in the corresponding sectors depending on the zone that the cluster locates.

Subsequently, topology is sliced into tiers from the inside out. Afterwards, each tier is again virtually partitioned into fixed-size hexagonal cells as done in recent cellular communication. The number of clusters that appear at each tier is given in the following:

$$\text{numOfCls}_{(\text{tierNo})} = \text{tierNo} * 6 \mid 0 \leq \text{tierNo} \leq \text{NumOfTiers}. \tag{6}$$

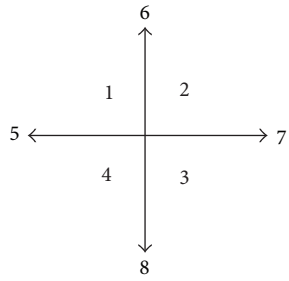


Figure 2: Network area zoning.

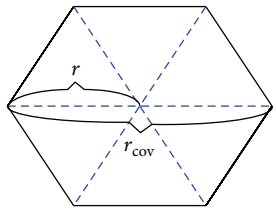


Figure 3: Cluster diameter and radio coverage.

In (6), tierNo and NumOfTiers denote the corresponding tier number and the total number of tiers present in the network, respectively.

Any two nodes deployed in a cluster should be able to communicate directly with each other. Thus, the distance between the two most distant points of the cell which represents the diameter ($2r$) of the hexagon is adjusted so as to be equal to the coverage radius of the radio (r_{cov}) as clarified as follows (Figure 3):

$$r_{cov} = 2r. \tag{7}$$

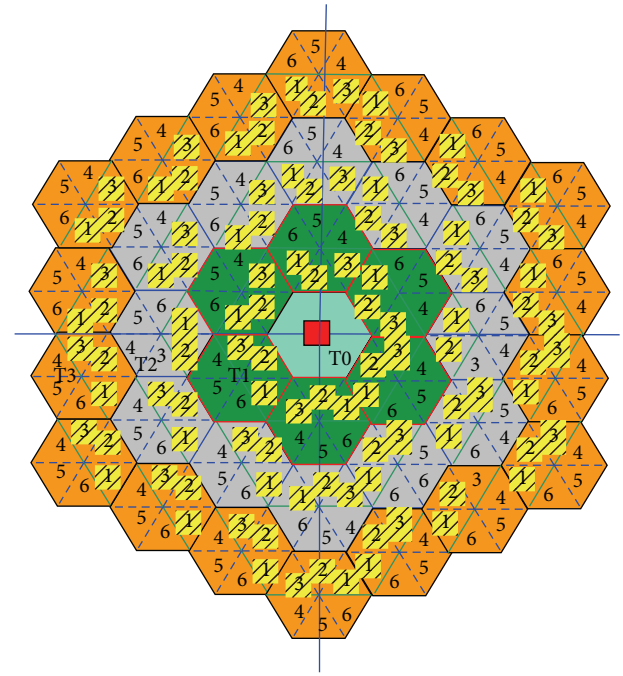
The final stage of the configuration is parceling out each cell into six equal sectors. As was mentioned above, CH election in each cluster is performed according to at which region the cluster takes place. In order to prevent redundant retransmission back towards the sink, nodes deployed at shaded sectors can be CH candidates. As an example, considering region 1, if a node locating in one of the sectors (4, 5, 6) is elected as the CH, after the data aggregation process, data has to be transmitted in the opposite direction backward to the sink which ultimately results in a redundant energy dissipation. Therefore, in region 1, a node among the nodes that belong to one of the sectors (1, 2, 3) is selected as the CH.

All the above-mentioned hierarchical configurations of the network area and the CH candidate pattern for each zone are illustrated in Figure 4.

The sink is positioned at the center of the topology with the coordinates ($x = 0$, $y = 0$). Cluster central point calculation and assignment of the nodes to the corresponding clusters are performed as follows.

(i) (x, y) coordinates of the cluster CLS0 are defined for each tier as follows:

$$x_0 = 0, \qquad y_0 = (\text{TierNo} * 2) * r. \tag{8}$$



T0: tier0 T2: tier2
T1: tier1 T3: tier3

Figure 4: Hierarchical configuration of the network area.

(ii) Rotate the center point of the previous cluster through angle α clockwise (Figure 5). The value of angle α is calculated according to

$$\alpha = \frac{2\pi}{(\text{TierNo} * 6)}. \tag{9}$$

Rotation of a point through angle α about the origin is performed according to

$$\begin{aligned} x_i &= (x_{i-1} * \cos\alpha) - (y_{i-1} * \sin\alpha) \\ y_i &= (x_{i-1} * \sin\alpha) + (y_{i-1} * \cos\alpha). \end{aligned} \tag{10}$$

Algorithmic representation of the cluster center definition procedure is presented in Algorithm 1.

Assignment of the nodes to the clusters is done at the setup phase in a distributed manner. The sink is responsible for virtual parcellation of the network area into clusters. The relative geographical coordinates of the center point of each cluster are calculated by the sink and then broadcast to the network over the common communication channel. This broadcast message includes the id, (x, y) coordinates of the central point, and the channel that will be used in the cluster (Figure 6). Since the nodes are assumed to know their relative geographical coordinates, each of them selects the cluster with the closest central point.

4.2. Neighborhood Definition. Following the process of identifying the corresponding cluster, each node should inform

```
ClusterCenterCalculation(){
  for (i ← 1 to NumOfTiers) do
    Tiers[i].Cls[0].x = 0
    Tiers[j].Cls[0].y = (i ∗ 2 ∗ r)
    for (j ← 1 to (i ∗ 6)) do
      Tiers[i].Cls[j].x = (Tiers[i].Cls[j − 1].x ∗ cos α) − (Tiers[i].Cls[j − 1].y ∗ sin α)
      Tiers[i].Cls[j].y = (Tiers[i].Cls[j − 1].x ∗ sin α) + (Tiers[i].Cls[j − 1].y ∗ cos α)
    end for
  end for
}
```

Algorithm 1: Clusters' central point calculation.

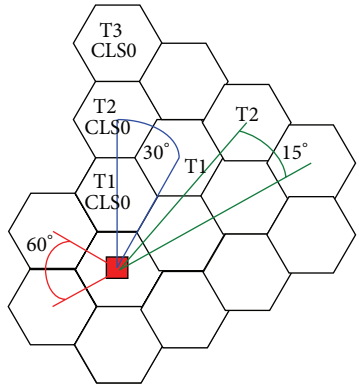


Figure 5: Central point rotation pattern for each tier.

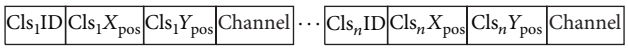


Figure 6: Structure of the broadcast message announcing the coordinates and the assigned channel number of each cluster.

other nodes that belong to the same cluster about its existence and relation to the cluster that it selects. Since multiple nodes will try to attain to the common transmission medium at the same time, in order to prevent multiaccess collision, each node starts a timer. Only after the expiration of its timer, a node can make an attempt to transmit its announcement message. Each node defines a timer value different from that of the other nodes. Since two nodes possess different (x, y) coordinates, it is not possible for two timers to expire at the same time. By the time a node hears the announcement of another node, it records the values included in the message to its neighborhood table which is consulted during intercluster routing stage.

4.3. Data Gathering and Reduction. The major concern during communication protocol and architecture definition for WSNs is energy efficiency. One promising solution is the reduction of data that is transmitted in the system. Depending on the type and the aim of the application, not all of the nodes deployed in a particular proximity have to gather data and convey it to the CH. Among the nodes in a particular area, the one with the highest residual energy level gathers data

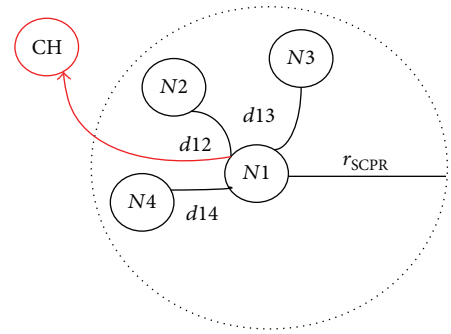


Figure 7: Sample scenario.

and first transmits it to the corresponding CH. Other nodes overhearing this message that appears in the proximity range abort their transmission attempt. This determinant is called the spatial correlation proximity range (SCPR) and will be denoted by r_{SCPR}.

In the scenario depicted in Figure 7, the only data transmitter is $N1$, where

(i) $d12$, $d13$, and $d14$ denote the distances between the nodes;

(ii) $r_{\text{SCPR}} \geq d12, d13, d14$;

(iii) $\text{ResEng}_{N1} > \text{ResEng}_{N2}, \text{ResEng}_{N3}, \text{ResEng}_{N4}$.

4.4. CH Election. With the aim of providing fair load balance in the clusters, the cluster leadership task should be assigned dynamically to different nodes. During this dynamic assignment process, different parameters can be considered, such as proximity to the sink or residual energy level. In this study, we suggest a cost factor (costCH) which is calculated for each node in every cluster at the beginning of each cycle. Each node calculates its own costCH and lack of the knowledge about the other nodes' cost values. In many research studies, nodes inform each other about the values of such parameters. However, that will cause collision and transmission errors, which, in turn, results in redundant energy consumption. Therefore, without any necessity of any communication, each node starts a timer internally. The value of each node's timer depends on the cost factor calculated by the node. Since each node runs the same algorithm, they will get the results in

the same direction proportional with their cost factors. The timer of the node with the largest cost factor expires earliest and accesses to the common communication medium first. In this way, it makes its announcement of being the CH of the corresponding cluster at the present cycle. Other nodes of the cluster receive the message and stop the operation of CH calculation and assign the announcing node as their CH. The cost factor and timer value calculations are performed as follows:

$$\begin{aligned} \text{intraCost} &= \sum_{n=1}^{\text{InClsNumOfNodes}} d_{\text{node}(n)} \\ \text{costCH} &= \text{ResEng} * d_{\text{sink}}{}^{-1} * \text{intraCost}^{-1} \\ \text{timer} &= \text{costCH}^{-\beta}, \end{aligned} \tag{11}$$

where

(i) ResEng and d_{sink} denote the residual energy level and distance of the node to the sink, respectively.

(ii) intraCost is the cost belonging to a node and calculated by adding the distances between the node and the remaining nodes in the cluster.

(iii) InClsNumOfNodes represents the number of nodes in the corresponding cluster.

(iv) d_{node} is the distance between the concerned node and the node in the same cluster.

(v) β stands for the parameter used during simulations in order to scale the cost value into the time domain.

Equation (11) clearly identify that a node with a higher residual energy level and closer to the sink, at the same time with a closer proximity to other nodes in the cluster, has a higher probability for being the CH. Flow chart of the proposed CH election method is depicted in Figure 8.

4.5. Intracluster Communication. In order to prevent collision during intracluster transmission, nodes apply an 802.11-type CSMA mechanism. Since every node can receive the transmission of any other node in the cluster, when a node overhears the transmission of another node directed toward the CH, it refrains from accessing to the common transmission medium.

4.6. Intercluster Communication. CHs aggregate the data arriving from the plain nodes and transmit it towards the sink. If the sink is in the coverage, aggregated data is transmitted directly to the sink. Otherwise, an intermediate relay node belonging to one of the clusters on the way is selected as the next hop. During next-hop selection, firstly the residual energy levels of the candidate nodes are considered. The node with the highest energy level is selected as the relay node. In the situation of equality, the proximity to the sink is taken into account. Among the nodes with equal residual energies, the one closer to the sink is selected as the relay node.

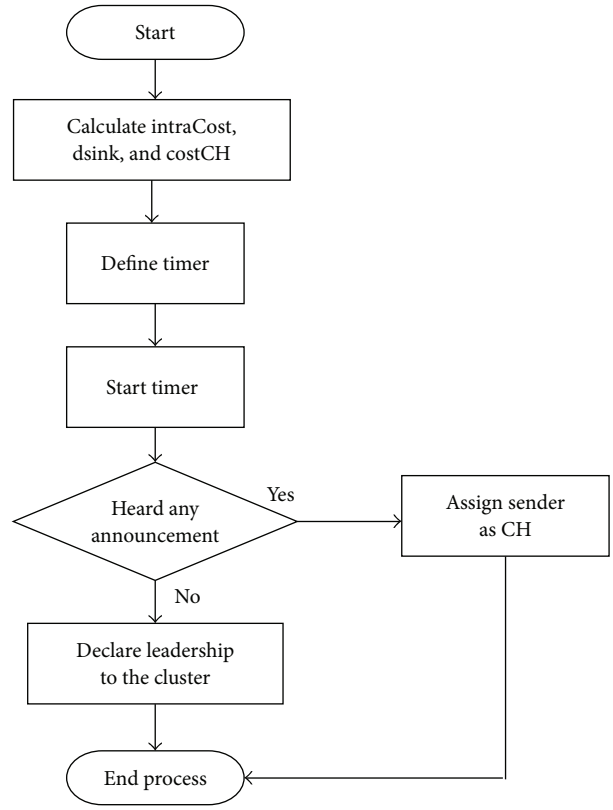


Figure 8: Flowchart of the CH election process.

4.7. Energy Consumption Model. As mentioned in the previous sections, the primary energy exhausting unit of a sensor node is the communication subunit. Thus, the energy dissipated by the other staff is ignored. Nodes are classified as CH or plain. The amount of energy consumed by the communication subunit changes depending on the situation whether it becomes the CH or not in each cycle:

$$E_n = E_{\text{CH}} * \rho_{\text{CH}} + E_P * (1 - \rho_{\text{CH}}), \tag{12}$$

where E_n is the total amount of energy consumed for the node n in a single cycle. The energy dissipated by a node when it is elected as the CH is denoted by E_{CH}. ρ_{CH} represents the probability for a node to be elected as the CH in its cluster. If a node is not elected as the CH, then it serves as a plain node. E_P denotes the energy consumed if the node n is a plain node:

$$E_{\text{CH}} = E_{\text{rcv-intra}} + E_{\text{Inter-Snd}} + E_{\text{Inter-Rrelay}}. \tag{13}$$

The amount of energy consumed by a CH is denoted by E_{CH} in (13). When a node is elected as the CH, it aggregates the data sent by the plain nodes in its cluster. Energy dissipated by a CH during this aggregation process is shown with $E_{\text{rcv-intra}}$ and clarified in (14). $E_{\text{Inter-Snd}}$ is the energy consumed during internally transmitting the aggregated data towards the sink and identified in (15). Lastly, a node can be selected as a relay node for sending the aggregated data incoming from a cluster positioning at an outer tier. The amount of energy

consumed as a relay node is denoted by $E_{\text{Inter-Rrelay}}$ and expressed in (16)–(18):

$$E_{\text{rcv-intra}} = \sum_{n=1}^{\text{InClsNumOfNodes}} (l * E_{\text{elec}}) \quad (14)$$

$$E_{\text{Inter-Snd}} = (l * \text{InClsNumOfNodes}) \left(E_{\text{elec}} + \left(\varepsilon_{\text{fs}} * d^2\right)\right) \quad (15)$$

$$E_{\text{Inter-Relay}} = E_{\text{Inter-Relay\_Rcv}} + E_{\text{Inter-Relay\_Snd}} \quad (16)$$

$$E_{\text{Inter-Rcv}} = \rho_{\text{relay}} * \left(l_{\text{InterRelay}} * E_{\text{elec}}\right) \quad (17)$$

$$E_{\text{Inter-Snd}} = \rho_{\text{relay}} * l_{\text{InterRelay}} * \left(E_{\text{elec}} + \left(\varepsilon_{\text{fs}} * d^2\right)\right). \quad (18)$$

The energy consumed by a node when it becomes a plain node is denoted by E_P calculated according to the model given in (3). This time, d, represents the distance between the plain node and the sender CH.

5. Performance Evaluation

In this section, we evaluate the key performance metrics of the ECMTADR protocol via extensive simulations. The proposed approach is compared with LEACH [25, 26] and HEED [27]. Performance assessment is made in terms of energy consumption, delay, and lifetime.

The metrics that are evaluated during simulations are given in the following.

(i) *End-to-end delay* is the time elapsed between the time that the first cycle starts and the time that the last cycle ends.

(ii) *TotalEnergy* is the total energy consumed in the network by all sensor nodes.

(iii) *Lifetime* is the time when the first node depletes energy.

(iv) *MaxEnergyConsumed* is the amount of energy dissipated by the node which is the one that has the lowest residual energy level.

The major factors that affect these metrics are the number of tiers, the number of nodes, the cell radius, and SCPR. As a result of detailed simulations, the way these factors affect the metrics described above is obtained.

Simulations are performed for 100 rounds on a 500 m * 500 m square network area. Events occur randomly in the network without depending on any constraint. A sink equipped with a single half-duplex transceiver is positioned at the center. It is assumed that there are n nonoverlapping channels comprising a common data channel and $(n-1)$ data channels. Each sensor node is equipped with a single half-duplex radio with transmission rate of 250 Kbps.

Following the values defined in the literature for the parameters expressed in Section 3, the values given in Table 1 are considered during energy consumption calculations. All sensors are considered to be identical with an initial residual energy of 3 J. The transmission range of each sensor node is set to 100 meters. The energy spent in transmitting a bit over a 1 meter distance is taken as 10 pJ/bit/m^2 and the energy spent in receiving a bit is set to 50 nJ/bit.

Table 1: Simulation parameters.

Radio transmission data rate	250 Kbps
d_0 (threshold distance)	85 m
R_0 (coverage radius)	100 m
E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
E_{mp}	0.0013 pJ/bit/m^4
$E_{\text{Residual-Initial}}$	3 J

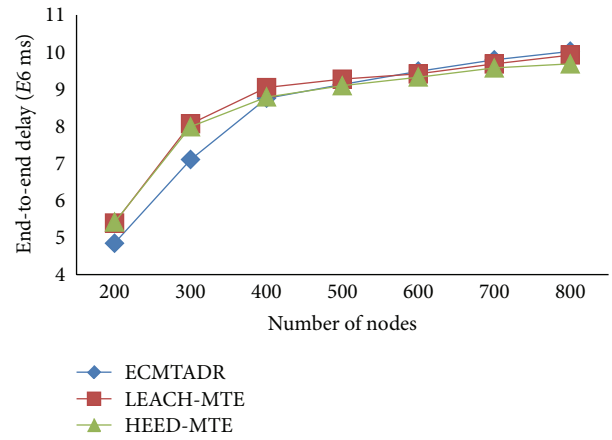


Figure 9: Effect of the number of tiers on end-to-end delay.

5.1. Node Density. In this study, end-to-end delay refers to the time that elapses between the time that the simulation first starts and the time when the last cycle ends. There are two alternatives for the definition of the term duration here. 100-round simulation is performed for each parameter-metric pair evaluation. The first definition for the term duration is that if a node exhausts energy before the completion of the 100th cycle, duration becomes the lifetime of the network. Otherwise, if none of the sensor nodes depletes energy until the end of the last cycle, duration is the time at which the last cycle ends.

Node density is a crucial determinant for energy consumption. Thus, in this study, the impact of node density on different performance metrics of the network is analyzed briefly. In this study, the major concern is to prolong the network lifetime to the utmost; therefore, the delay is disregarded. As obviously depicted in Figure 9, all of the three methods perform nearly identically in terms of end-to-end delay. A sharper increase in delay occurs up to a threshold density. However, when the density reaches the specified level, a smoother incline is at stake.

Figure 10 represents the change in energy consumption regarding the variation in node density. Increase in node density accordingly leads to an increase in the amount of data carried in the network, which, in turn, induces energy

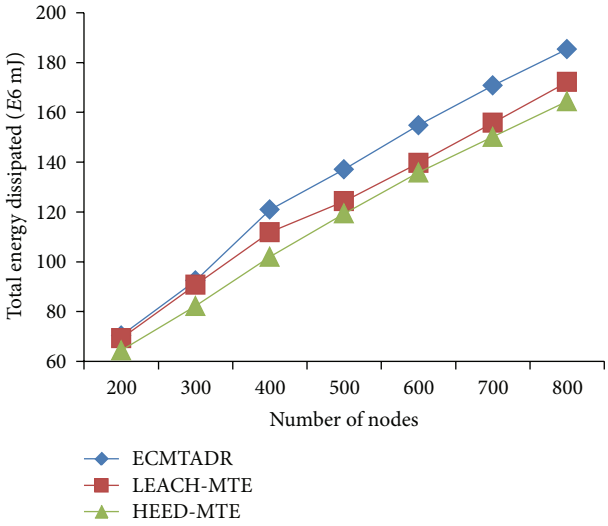


Figure 10: Total energy consumed in the network.

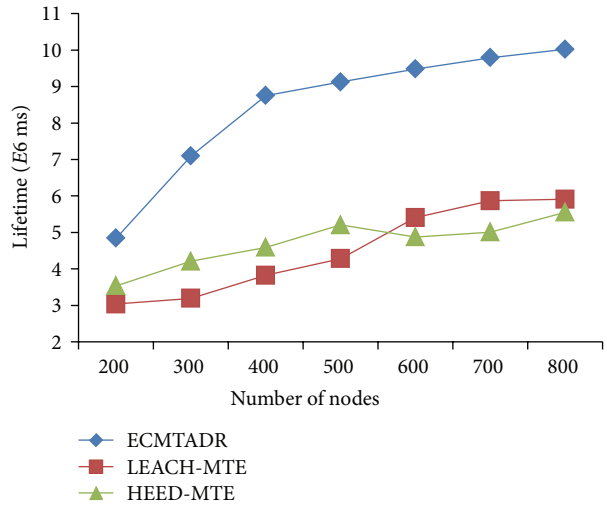


Figure 11: Impact of the node density on the network lifetime.

dissipation at the same time. Figure 10 clarifies that LEACH and HEED outperform ECMTADR in terms of the total amount of energy consumed in the network. However, the decisive factor while developing an architecture or protocol for WSNs is the lifetime of a single node which actually stands for the lifetime of the network. One way of extending the network lifetime is achieving load balance among the staff as much as possible. In this study, a maximum number of nodes collaborate possibly during data transmission towards the sink. Therefore, though it may seem that much more energy is dissipated totally, the individual performance lies exactly in the opposite sense which is clearly identified in Figure 11. The reason of the deviations encountered is the random deployment of the nodes at the beginning of each simulation. That is, the topology of a 500-node network and one with a 600 node is not identical.

MTE above refers to the "minimum transmission energy" routing method. ECMTADR shows nearly 50% better performance in terms of network lifetime. Figure 11 proves that, by increasing the number of the staff in the network, a better load distribution is achieved. Fair load distribution inherently prolongs the network lifetime. Although the increase in the population seems to induce a rise in energy consumption cumulatively, in the individual sense that will eventuate with network lifetime prolongation.

5.2. Cluster Size and Radio Coverage. Cluster size has also crucial impact in terms of delay and energy efficiency. During the design stage, it is intended for all nodes in a cluster to communicate directly with each other. Therefore, each cell diameter is set to be equal to the radio coverage. As the coverage radius increases clusters also expands. Thus, the number of nodes per cluster density increases. In this way, especially during intercluster communication, if an intermediate relay node covers the distance, it will prefer to relay the data directly to the sink which ultimately reduces delay. The impact of cluster size on delay is depicted in Figure 12.

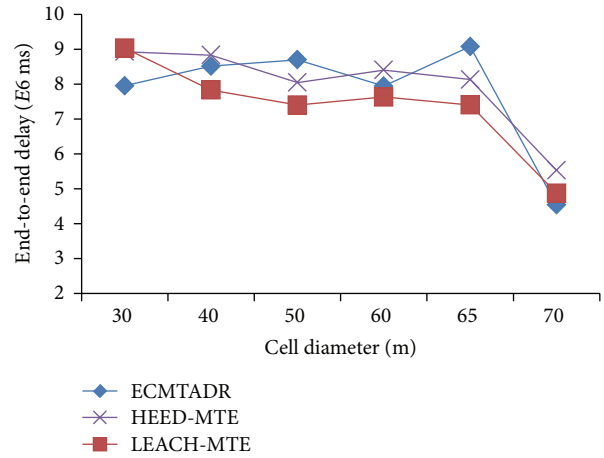


Figure 12: Impact of the cluster size on the network lifetime.

As mentioned before, our methodology does not have any claim on the improvement of the delay. Therefore, as can be seen from the graph, all three methods perform nearly the same.

In terms of total energy consumption in the network, LEACH and HEED are seen to outperform ECMTADR as depicted in Figure 13. However, as expressed previously, the point is reducing the individual energy consumption which ultimately delays node failure. As obviously identified in Figure 14, the amount of the energy consumed by a node in ECMTADR is lower when compared with LEACH and HEED. Up to a threshold value for the cluster size and radio coverage, individual energy consumption value stays constant. However, after the threshold value, a trend in the vertical direction is observed. That is because the distance between the plain nodes and their CHs and also between the CHs and intermediate relay nodes increase.

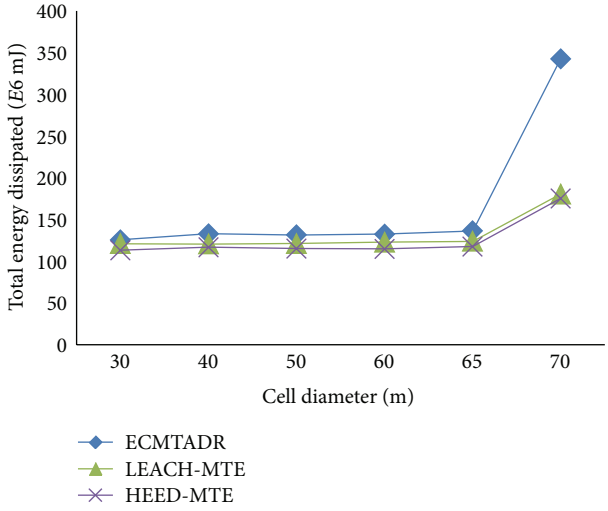


Figure 13: Total energy consumed in the network.

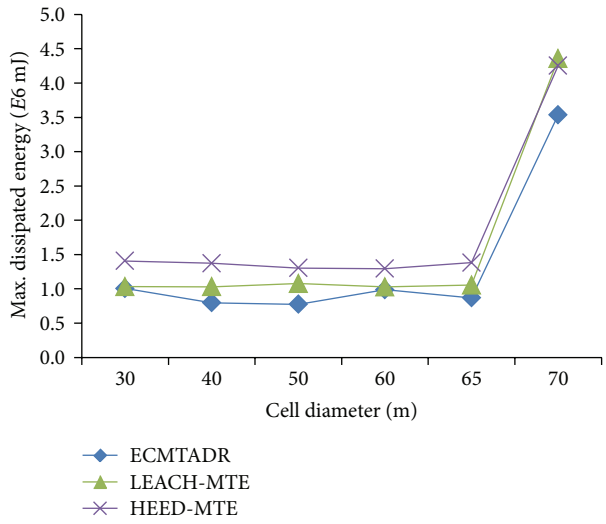


Figure 14: Maximum energy dissipated individually.

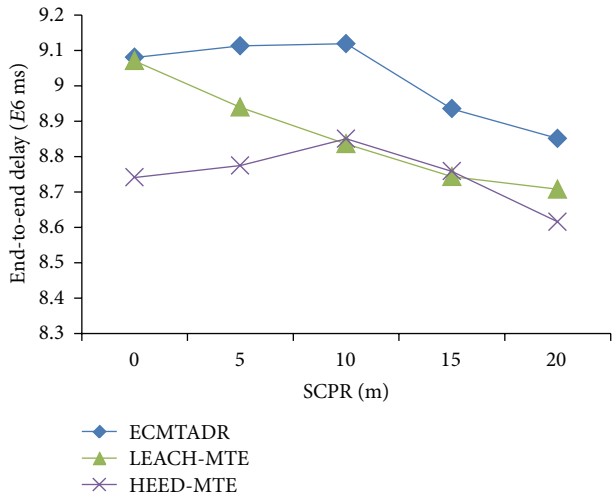


Figure 15: Impact of SCPR on delay.

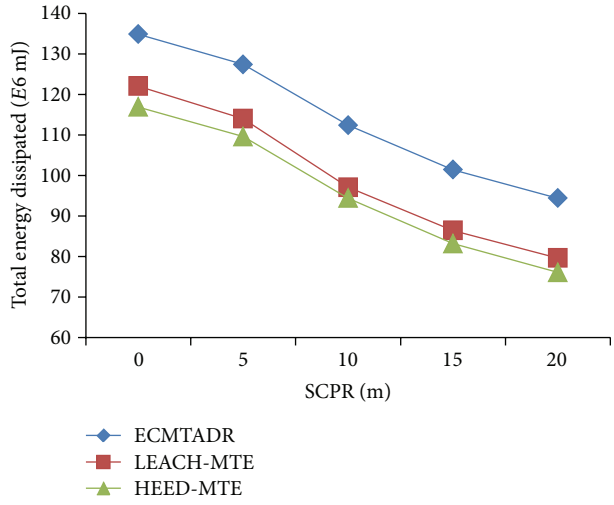


Figure 16: Impact of SCPR on total energy consumption.

5.3. SCPR. The major energy consuming component of a sensor node is the communication subunit. Therefore, most of the effort focuses on the challenge of reducing communication energy expenditure. Some of the studies have focused on developing energy-efficient protocols and architectures. Besides, some of them aimed to reduce the amount of data transmitted in the network. One way of achieving data reduction is reducing the amount of emerged data in each cluster. That refers to assigning the task of data gathering to certain nodes, not all of them. In this study, a parameter with the name spatial correlation proximity range (SCPR) is suggested in order to reduce the amount of data created. Nodes with higher energies are charged with data gathering. When a node is assigned the task of data gathering, other nodes located in the area with the radius of SCPR do not perform any data collection from the environment. Depending on the type of the application requirements, the value of SCPR can vary.

Lower data transmission inevitably achieves improvement in delay as shown in Figure 15. Moreover, by reducing the amount of data carried in the network, substantial energy saving is achieved as represented in Figures 16-17.

6. Conclusions

In this paper, we propose a sophisticated architecture called ECMTADR that comprises data reduction, load balance, and topology control. Two important solutions that have been suggested for energy conservation are clustering and data reduction. Both of these methodologies intend to mitigate the challenge of redundant energy consumption induced by the greedy communication subunit. ECMTADR establishes a four-layer architecture virtually. In first place, the network area is partitioned into eight regions. Above the regionalized architecture, sensor nodes are grouped into fixed-size hexagonal clusters at which CHs are elected periodically depending

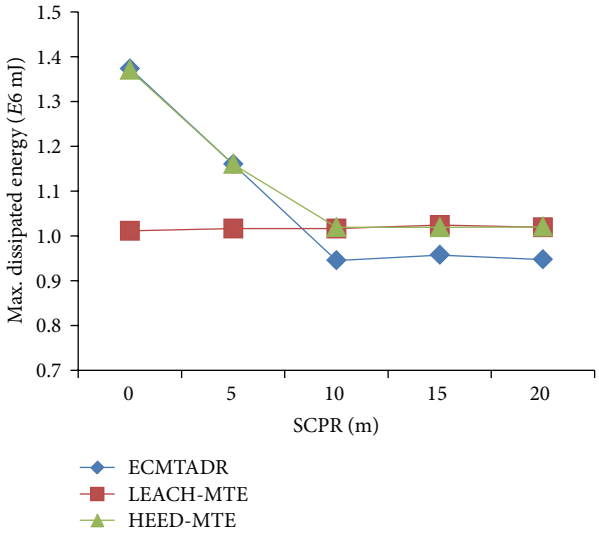


Figure 17: Impact of SCPR on individual energy consumption.

on their residual energy levels, distances to the sink, and other nodes deployed in their clusters. In this way, a fair energy load balance is achieved. Depending on the type of application, data generation can be restrained. In each cluster, only the nodes with higher energy levels can gather data. Other nodes that are positioned in the SCPR range of those nodes do not need to proceed during that stage. Besides, each cluster is again virtually divided into triangular sectors. The reason for sectoring each cluster is to prevent redundant backward transmission. In this way, unnecessary energy expenditure is also prevented. Extensive simulations are performed by considering major parameters that have serious impact on network lifetime and delay. Simulation results clarify that ECMTADR performs better than LEACH and HEED in terms of energy efficiency and lifetime.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] Z. Chen, Y. Xiao, X. Li, and R. Li, "A clustering protocol for wireless sensor networks based on energy potential field," *The Scientific World Journal*, vol. 2013, 7 pages, 2013.

[3] M. Al Ameen, S. M. R. Islam, and K. Kwak, "Energy saving mechanisms for MAC protocols in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2010, Article ID 163413, 16 pages, 2010.

[4] V. Coşkun, E. Cayirci, A. Levi, and S. Sancak, "Quarantine region scheme to mitigate spam attacks in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 1074–1086, 2006.

[5] T. Çevik and A. H. Zaim, "A multichannel cross-layer architecture for multimedia sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 457045, 11 pages, 2013.

[6] D. Dardari, A. Conti, C. Buratti, and R. Verdone, "Mathematical evaluation of environmental monitoring estimation error through energy-efficient wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 7, pp. 790–802, 2007.

[7] F. Zhao, J. Liu, L. Guibas, and J. Reich, "Collaborative signal and information processing: an information-directed approach," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1199–1209, 2003.

[8] T. Çevik, A. H. Zaim, and D. Yiltaş, "Localized power-aware routing with an energy-efficient pipelined wakeup schedule for wireless sensor networks," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 20, no. 6, pp. 964–978, 2012.

[9] O. D. Incel, "A survey on multi-channel communication in wireless sensor networks," *Computer Networks*, vol. 55, no. 13, pp. 3081–3099, 2011.

[10] A. Rao, M. Akbar, N. Javaid, S. N. Mohammad, and S. Sarfraz, "AM-DisCNT: angular multi-hop distance based circular network transmission protocol for WSNs," in *Proceedings of the IEEE 8th International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA '13)*, pp. 29–35, IEEE, Compiègne, France, October 2013.

[11] R. Soua and P. Minet, "Multichannel assignment protocols in wireless sensor networks: a comprehensive survey," *Pervasive and Mobile Computing*, vol. 16, pp. 2–21, 2014.

[12] T. Çevik and A. H. Zaim, "EETBR: energy efficient token-based routing for wireless sensor networks," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 21, no. 2, pp. 513–526, 2013.

[13] M. H. Alsharif, R. Nordin, and M. Ismail, "Survey of green radio communications networks: techniques and recent advances," *Journal of Computer Networks and Communications*, vol. 2013, Article ID 453893, 13 pages, 2013.

[14] N. Amini, A. Vahdatpour, W. Xu, M. Gerla, and M. Sarrafzadeh, "Cluster size optimization in sensor networks with decentralized cluster-based protocols," *Computer Communications*, vol. 35, no. 2, pp. 207–220, 2012.

[15] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[16] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, no. 8, pp. 11113–11153, 2012.

[17] L. M. C. Arboleda and N. Nasser, "Comparison of clustering algorithms and protocols for wireless sensor networks," in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE '06)*, pp. 1787–1792, Ottawa, Canada, May 2006.

[18] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[19] O. Boyinbode, H. Le, A. Mbogho, M. Takizawa, and R. Poliah, "A survey on clustering algorithms for wireless sensor networks," in *Proceedings of 13th International Conference on Network-Based Information Systems*, pp. 358–364, Takayama, Japan, September 2010.

[20] C. Jiang, D. Yuan, and Y. Zhao, "Towards clustering algorithms in wireless sensor networks—a survey," in *Proceedings of the*

IEEE Wireless Communications and Networking Conference (WCNC '09), pp. 1–6, Budapest, Hungary, April 2009.

[21] B. P. Deosarkar, N. S. Yadav, and R. P. Yadav, "Clusterhead selection in clustering algorithms for wireless sensor networks: a survey," in *Proceedings of the International Conference on Computing, Communication and Networking (ICCCN '08)*, pp. 1–8, St. Thomas, Virgin Islands, USA, December 2008.

[22] Y. Wang, W.-Z. Song, W. Wang, X.-Y. Li, and T. A. Dahlberg, "LEARN: localized energy aware restricted neighborhood routing for ad hoc networks," in *Proceedings of the 3rd Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Reston, Va, USA, September 2006.

[23] Q. Nadeem, M. B. Rasheed, N. Javaid, Z. A. Khan, Y. Maqsood, and A. Din, "M-GEAR: gateway-based energy-aware multi-hop routing protocol for WSNs," in *Proceedings of the IEEE 8th International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA '13)*, pp. 164–169, Compiegne, France, October 2013.

[24] N. Amjad, M. M. Sandhu, S. H. Ahmed et al., "DREEM-ME: distributed regional energy efficient multi-hop routing protocol based on maximum energy with mobile sink in WSNs," *Journal of Basic and Applied Scientific Research*, vol. 4, no. 1, pp. 289–306, 2013.

[25] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, January 2000.

[26] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[27] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 510516, 10 pages
http://dx.doi.org/10.1155/2015/510516

Research Article

Intelligent Transmission Power Allocation for Distributed Beamforming in Wireless Sensor Networks

Sungmoon Chung[1] and Inwhee Joe[2]

[1]*Department of Electronics Computer Engineering, Hanyang University, 17 Haengdang-dong, Seongdong-gu, Seoul, Republic of Korea*
[2]*Division of Computer Science & Engineering, Hanyang University, 17 Haengdang-dong, Seongdong-gu, Seoul, Republic of Korea*

Correspondence should be addressed to Inwhee Joe; iwjoe@hanyang.ac.kr

Received 18 December 2014; Revised 25 March 2015; Accepted 29 March 2015

Academic Editor: Changqiao Xu

Distributed beamforming can significantly improve the reliability of the link and the capacity and the coverage of wireless networks. Using a subset number of nodes from a network of sensors, they collectively transmit a common message to an intended destination. In distributed beamforming, the maximum channel capacity could be changed according to the number of cooperating source nodes and the distance (between the average source nodes and destination). Therefore, the scheme is necessary to guarantee the required channel capacity. However, it is difficult to adapt the practical environment due to signal fading, interference, and low quality of sensor nodes in WSNs. Therefore, we studied about the channel characteristic and required transmission power according to the number of cooperating nodes and the distance theoretically to overcome these problems. As a result, we propose an Intelligent Transmission Power Allocation (ITPA) algorithm to guarantee the required channel capacity considering dynamic channel statement, the number of cooperating source nodes, and the distance between the average source nodes and destination with simplicity computation. In addition, ITPA distinguishes noise data (using an exponential weighted received power average) from the estimated original data. From that the system can satisfy requirements of the user without wasting power by itself.

1. Introduction

The techniques of the ad hoc wireless sensor networks have quickly emerged as an interesting research topic due to recent advancements in both size and power performance. It is now possible to cover large networks by relatively small devices distributed over the large area with limited power and coverage and guarantee low power to spare for long haul links [1–4]. A lot of researches have been done in an effort to improve the capacity, coverage, and reliability to transfer data from the individual nodes in a network to the final destination [5–9]. In particular signal fading and interference are among the major problems encountered in wireless sensor communications. In an effort to further improve and optimize utilization in wireless sensor network, the use of distributed beamforming has been studied as a method for nodes to collaborate in their transmissions.

In wireless communication systems, distributed beamforming is defined as a technique that cooperating source nodes transmit a common radio frequency signal over an antenna with aligning the phases of its transmission; after propagation, the received signals combine constructively at the destination [10]. The concept of distributed beamforming is shown in Figure 1.

Conventional transmit beamforming scheme can be emulated in distributed environment using a network of cooperative single-antenna source nodes. In case the source nodes agree on a common message, transmit it simultaneously, synchronize their carrier frequencies, and control their carrier phases to combine constructively.

In transmit beamforming systems, it is known that distributed beamforming has advantages compared to single-antenna transmission. It can achieve increased $N2$-fold rate or increased power efficiency (an N-fold decrease for a fixed desired received power). Therefore gain can be presented by formula (1); see [11]:

$$P_R = \left(N * \sqrt{a * P_T}^{2}\right) = a * N^2 * P_T. \quad (1)$$

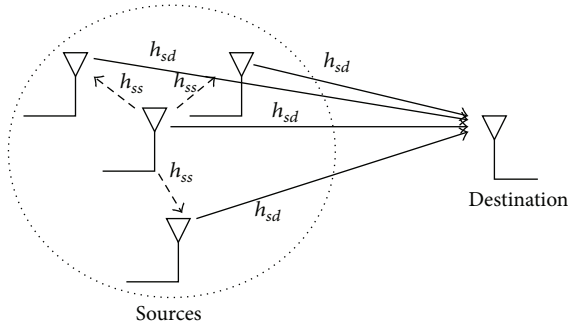


Figure 1: Distributed beamforming system.

P_R denotes the received power, a denotes identical attenuation, P_T denotes identical transmission power, and N denotes the number of the cooperating sources. From formula (1), channel capacity can be represented by formula (2); see [11]:

$$C = W * \log\left(1 + \frac{\left(a * N^2 * P_R\right)}{\left(W * N_0\right)}\right), \tag{2}$$

where W is the bandwidth and N_0 is the one-sided power spectral density.

In general, each node of wireless sensor networks is power limited hardware. The source, which broadcasts common messages, consumes power more rapidly than the other source nodes in the distributed beamforming system. Furthermore, there are various unexpected situations that make source nodes inoperable in practical environments. In case parts of source nodes cannot transmit signals, the data rate of the system is decreased. In particular, it causes problems where the system supports multimedia services. Therefore, it is very useful if the distributed beamforming system is able to adapt to the changeable environment and satisfy requirements of the system by itself.

2. Related Works

The potential of cooperative communication [4, 12] has been shown as a promising technique that can significantly improve the coverage, link reliability, and the capacity of wireless networks. In one kind of the cooperative communication scheme, a group of cooperative nodes can emulate an antenna array by transmitting a common message signal from a source. In addition, timing synchronization and distributed carrier synchronization should proceed [10]. The transmissions of multiple sources are focused in the direction of intended destination which are combined coherently at the destination. This cooperative communication scheme, referred to as distributed beamforming, is studied in a lot of researches.

There are a number of challenges to improve the feasible availability of distributed beamforming in the practical networks. Detailed protocols must be designed for both cooperating sources and communication between the sources and the destination. Until recently, a lot of schemes for information sharing, timing synchronization, carrier frequency synchronization, and carrier phase alignment are proposed. There is, of course, a trade-off between the gain and the overhead to implement these schemes.

Also, power control is one of the essential research topics to increase the throughput and reduce the interference in distributed beamforming [13, 14]. Jing and Jafarkhani [15] studied controlling the power resource at each relay in order to maximize the signal-to-noise ratio at the destination. In [15], each relay does not transmit at its maximum power to achieve the maximum SNR depending on its own bidirectional channels and other relays' channels. The study provides the condition for determining the optimal transmission power at each relay. Also, there are statistical approaches. Havary-Nassab et al. proposed distributed beamforming with second-order statistics of the channel state information in [16]. Multiuser multirelay approaches were also considered and studied in [17–23]. Krishna et al. [17] proposed relay strategy to minimize the mean-square error between the source and destination. D. H. Nguyen and H. H. Nguyen [18] proposed optimal power allocation for multiuser multirelay networks with full channel state information. Recently, the approach of two-way relay networks has also been the focus of several studies. In [19–23], the researchers deal with the optimal relay selection and user power control for two-way relay networks where two end-users exchange information through multiple relays.

However, a large amount of signaling overhead is required when each node needs computation of channel statement continually. These information sharing processes increase the system complexity. In addition, previous schemes [15–29] which are based on channel statement information cannot achieve performance improvement largely because the estimated channel state has noise generally in the practical environment [30]. In the wireless sensor networks, sensor data which are measured at the destination are subject to several different sources of errors. Generally, these sources of errors can be classified as either systematic errors (bias) or random errors (noise). We are particularly interested in decreasing the effect of these errors on sensor readings since they may seriously affect the distributed beamforming schemes which are based on the channel statement information.

As mentioned above, distributed beamforming is useful to upload multimedia such as image/video data or summaries of sensor data gathered over days or even months. It is important to guarantee the required data rate for these applications where performance of data rate is critical. If the system cannot guarantee the required data rate in these applications, the system is useless. In addition, the excessive improvement transmission power to guarantee the required data rate causes waste power. We have to recognize the WSNs are power constrained networks. However, previous studies for power allocation of distributed beamforming did not consider degradation of channel capacity when the number of cooperating source nodes is decreased. Moreover, there are no literatures to apply the concept of self-optimization for the distributed beamforming systems.

Therefore, we propose an Intelligent Transmission Power Allocation (ITPA) algorithm which is able to control the transmission power considering the changeable channel

state, number of cooperating nodes, and the distance between the average source nodes and destination. Also ITPA is able to decrease the effect of errors on sensor readings efficiently. Thus, ITPA not only guarantees the required channel capacity, but also improves the network life span time largely by a suitable power allocation.

3. System Model

We consider the scenario where sensors are deployed on the ground with low-power single-antenna. We also assume the environment is LOS (line of sight) between sensor fields and destination. In addition, we assume all the deployed nodes have the same power and each node knows the number of source nodes that are cooperative before they transmit common information to the destination. One of the sources (master source) collects data such as image/video and broadcasts to the source nodes (slave source). After timing synchronization and distributed carrier synchronization, cooperating source nodes transmit these data to the destination by their distributed antenna which uses beamforming. Thus, it would enable uploading multimedia such as image/video data or summaries of sensor data gathered over days or even months. This application also can be modified easily in other interesting applications as low-power soldier radios in battlefield communication or monitoring rural and disaster environment where longer range might be required.

It is known that the $N2$-fold power gain is achieved from distributed beamforming. However there is also a path loss in terms of wireless propagation. Using the Friis formula for free-space propagation, the received signal power of distributed beamforming at the destination considering that N source nodes are deployed can be presented as follows:

$$P_R = g_i * N^2 * P_T * G_T * G_R * \frac{\lambda^2}{(4\pi R)^2}, \tag{3}$$

where P_T and P_R are the transmit and received power, g_i $(0 < g_i \leq 1)$ is undefined attenuation parameter, G_T and G_R are the directivity gains of the transmit and receiving antennas, R is the range between the average source nodes and destination, and λ is the carrier wavelength.

As well known, signal-to-noise ratio (often abbreviated SNR or S/N) is a measure used in communication theory that indicates the level of a desired signal to the level of background noise. Thus, we can define the SNR at the receiver as $\text{SNR} = P_{\text{received signal}}/P_{\text{noise}}$.

$P_{\text{received signal}}$ denotes signal power at the receiver and P_{noise} denotes noise power at the receiver.

From the SNR at the receiver, SNR (signal-to-noise ratio) of the destination can be represented as follows:

$$\text{SNR}_{\text{dB}} = 10\log_{10}\left(g_i * N^2 * P_T * G_T * G_R * \frac{\lambda^2/(4\pi R)^2}{P_{\text{noise}}}\right). \tag{4}$$

Consider a distributed beamforming system with one master source node, $(N-1)$ slave source nodes, and one destination node as shown in Figure 2.

We consider a distributed deployment with N nodes (including master node) which seek to collaboratively transmit a common baseband message signal $m(t) = m_I(t) + jm_Q(t)$. And $x_i(t)$ denotes RF signal transmitted by node i. Generally, the channel gain from transmitter i to the receiver can be represented as a complex scalar h_i.

Also, we assume that channel model of the system is Rician channel model, so that we can denote the channel model by

$$h_i = g_i e^{j\theta_i}, \tag{5}$$

where g_i is attenuation and θ_i is phase constant of channel. Each path can be modeled as circular Gaussian random variable using Central Limit theorem when the number of channels is multiple.

The circular Gaussian variable has the following form:

$$z = w_1 + jw_2. \tag{6}$$

The magnitude $|z|$ follows Rician probability distribution as

$$P_z(z) = \frac{z}{\sigma^2} e^{-(z^2+c^2)/2\sigma^2} I_0\left(\frac{zc}{\sigma^2}\right). \tag{7}$$

And Rician factor K is defined as follows:

$$K = \frac{c^2}{2\sigma^2}, \tag{8}$$

where K is the Rician factor, defined as a ratio of the specular component power c^2 and scattering component power $2\sigma^2$.

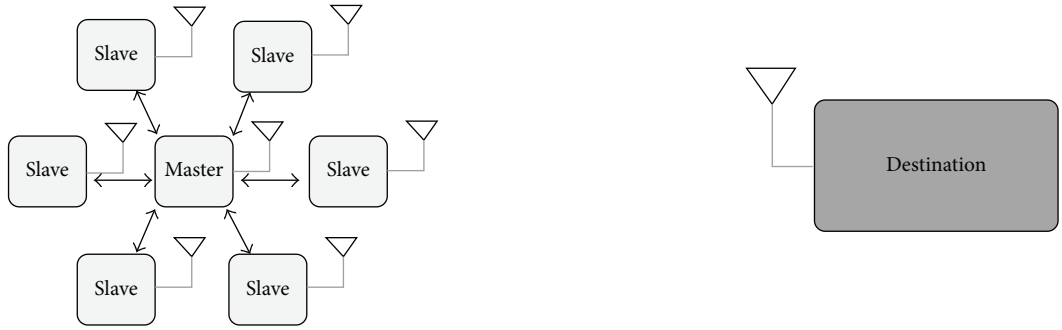
In our system model (which is called master-slave open-loop carrier synchronization system), master node sends a common reference signal $c_0(t) = \cos(2\pi f_0 t)$ to synchronize the oscillators on each of the source nodes, so that the signal transmitted by node i is $x_i(t) = \Re(m(t)\exp(j2\pi f_c t - j\theta_i))$, so that the overall received signal at the destination is as follows:

$$\begin{aligned} r(t) &= \sum_{i=1}^{N} \Re\left(h_i m(t) \exp\left(j2\pi f_c t - j\theta_i\right)\right) \\ &= \rho \Re\left(m(t) \exp\left(j2\pi f_c t\right)\right) \\ &\equiv \rho\left(m_I(t)\cos\left(2\pi f_c t\right) - m_Q(t)\sin\left(2\pi f_c t\right)\right). \end{aligned} \tag{9}$$

As formula (9), we show that the factor of ρ represents the distributed beamforming gain and the effect of constructive interference between the signals from each transmitter.

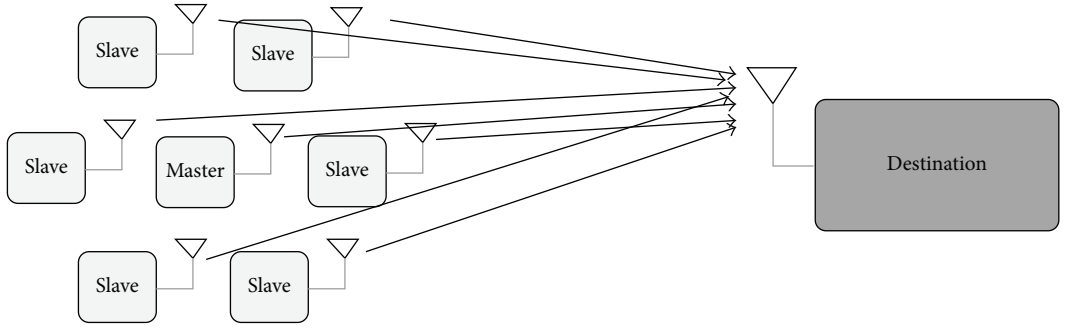
It is important to construct the signal $x_i(t)$ from each transmitter so that all the nodes are frequency locked and the phase of $x_i(t)$ is chosen to precisely cancel the effect of the channel phase θ_i. In our system model, the sources need to use the reference signal $c_o(t)$ at frequency f_0 to synthesize a carrier signal for beamforming at a completely different frequency f_c. We use the time-slot model for synchronization as shown in Figure 3.

Training signals which include the reference signal $c_o(t)$ and other feedback signals carrying information about the channel phase are periodically retransmitted every T_{slot} time in the short duration T_{est}. The duty cycle of the training



Master-slave frequency synchronization and phase synchronization among sources

Destination broadcasts beacon and sources estimate channel phase



Sources transmit as a distributed beamformer

FIGURE 2: Phase of distributed beamforming system.

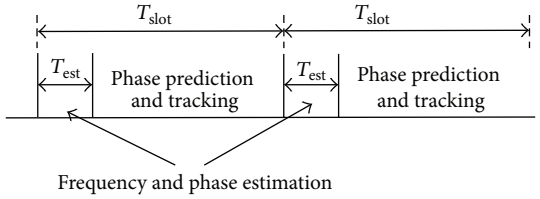


FIGURE 3: Time-slot model for synchronization.

process $\tau = T_{\text{est}}/T_{\text{slot}}$ can be quite small. The sources use the training signal in each time-slot to update their estimate of the phase $\Delta\phi_i$ and frequency offsets Δf_i.

During the "broadcast" phase, where a signal X is sent by the master source node, the received signal at slave source nodes $y_{S_m S_s}$ and destination y_{SD} can be written as

$$y_{S_m S_s} = \sqrt{E_{S_m S_s}} h_{S_m S_s} x + n_{S_m S_s},$$

$$y_{SD} = \sum_{i=1}^{N} \sqrt{\frac{\left(2^{C_R/W_0} - 1\right) * P_{\text{noise}} * (4\pi R)^2}{a * N^2 * G_T * G_r * \lambda^2}} h_{SD} X$$

$$+ n_{SD} = \left[\sqrt{\frac{\left(2^{C_R/W_0} - 1\right) * P_{\text{noise}} * (4\pi R)^2}{a * N^2 * G_T * G_r * \lambda^2}} \right.$$

$$\cdots \sqrt{\frac{\left(2^{C_R/W_0}-1\right) * P_{\text{noise}} * (4\pi R)^2}{a * N^2 * G_T * G_r * \lambda^2}} \Bigg] \begin{bmatrix} h_{S_1 D} \\ \vdots \\ h_{S_N D} \end{bmatrix} X$$

$$+ n_{SD} \equiv \sum_{i=1}^{N} \Re \left(h_i m(t) \exp\left(j2\pi f_c t - j\theta_i \right) \right), \tag{10}$$

where $h_{S_m S_s}$ denotes the channel response from the master source to the slave source nodes and h_{SD} denotes the channel response from cooperating source nodes to the destination node, and n_{SS} and n_{SD} are the observation noise at each link. C_R denotes required channel capacity of the system and W_0 denotes bandwidth per channel.

In our study environment, Johnson-Nyquist noise and receiver noise have a lot of parts in the noise. Therefore, we focus on both of them. Johnson-Nyquist noise can be represented as

$$P_{\text{noise}} = k_B * T * W, \tag{11}$$

where P_{noise} denotes the thermal noise power, k_B denotes Boltzmann's constant, and T denotes the resistor's absolute temperature.

By using formula (9), noise power in dBm at the temperature T is presented as

$$P_{\text{dBm}} = 10 * \log_{10}\left(k_B * T * W * 1000\right). \tag{12}$$

Consider the common scenario of estimated room temperature ($T = 300$ K) as

$$P_{\text{dBm}} = -174 + 10 * \log_{10}(W). \tag{13}$$

Therefore, Johnson-Nyquist noise has a power spectral density of 174 dBm/Hz at 300 Kelvin.

Also, we consider receiver noise which is caused by components in a radio frequency (RF) signal chain, which can give us

$$\text{Noise Figure (dB)} = 10 * \log_{10}\left(\frac{\text{Practical noise power}}{\text{Johnson-Nyquist noise power}}\right). \tag{14}$$

From formula (14), we can calculate practical noise power at the receiver.

4. Intelligent Transmission Power Allocation Algorithm

Algorithm 1 shows our proposed Intelligent Transmission Power Allocation algorithm. Note that we assume our distributed beamforming system synchronizes the sources by open-loop master-slave synchronization, one source node plays a role as the master, and the remaining source nodes play role as slaves. Through our power allocation algorithm, we are able to allocate suitable transmission power according to the environment when sensor nodes are deployed considering the maximum transmission power of the nodes. Also source nodes are able to communicate continually with the destination when some of source nodes are inoperable. Furthermore, it can overcome the problem that master node consumes power rapidly by changing the master periodically according to the residual energy level. As a result, it can improve the network life span time.

We estimated RSSI (received signal strength indicator) by practical wireless sensor nodes on the LOS (line of sight).

Figure 4 shows the result of the test. It is shown that there are noisy data due to various reasons, such as the low quality of sensor nodes and random effect of external environments [24]. How to remove the noisy data and achieve clean data is the key issue for the proposed Intelligent Transmission Power Allocation algorithm since our Intelligent Transmission Power Allocation algorithm is based on periodic estimated environment parameters. We proposed the scheme to get the clean received power data. Thereby we achieved the representative received power value from EWP_{avg} (exponential weighted received power average). Thus, we can remove the noisy data from the estimated received power data. Also we can much weight to the recent estimated received power based on EWP_{avg}.

Step 1. Compute R_{avg} and its σ. Consider

$$R_{\text{avg}} = \frac{R_{t-1} + R_{t-2} + \cdots R_{t-(k+1)} + R_{t-k}}{k},$$
$$\sigma = \sqrt{\frac{\sum_{n=t-1}^{t-k} \left(R_n - R_{\text{avg}}\right)^2}{k}}. \tag{15}$$

Step 2. Get an estimated sample R_t.

Step 3. Calculate $\sqrt{|R_{\text{avg}} - R_t|^2}$.

Step 4. If $\sqrt{|R_{\text{avg}} - R_t|^2} \geq \beta * \sigma$, R_t is rejected and go to Step 1.

Step 5. If $\sqrt{|R_{\text{avg}} - R_t|^2} < \beta * \sigma$, calculate EWP_{avg}:

$$\text{EWP}_{\text{avg}} = \alpha * \left(R_{t-1} + (1-\alpha) * R_{t-2} + (1-\alpha)^2 * R_{t-3} + \cdots + (1-\alpha)^k * R_{t-(k+1)}\right) + (1-\alpha)^{k+1} * R_{t-k}. \tag{16}$$

Step 6. Calculate the parameter value g_i:

$$g_i = \frac{\text{EWP}_{\text{avg}} * (4\pi R)^2}{N^2 * P_T * G_T * G_R * \lambda^2}. \tag{17}$$

R_t is estimated received power at time t, α is the weight value for the recent data, and β is the parameter which distinguishes noise data from the estimated original data. Therefore the affordable range of EWP_{avg} is $[R_{\text{avg}} - \sigma * \beta, R_{\text{avg}} + \sigma * \beta]$. From

INIT
Input *required channel capacity* (C_R), *bandwidth per channel* (W), *distance* (R), *transmit antenna gain* (G_T), *receive antenna gain* (G_R), *noise power* (P_{noise}), *undefined attenuation parameter* (g_i)
$\alpha = \dfrac{(2^{C_R/W} - 1) * 16\pi^2 * R^2 * P_{\text{noise}}}{g_i * G_T * G_R * \lambda^2}$
Count = 0
WHILE Count = 0 **DO**
Heartbeat = (Node ID ∪ Residual Energy level)
Broadcast its Heartbeat
Determine the number of neighbor active nodes, N = the number of neighbor active nodes
To be the Master node itself
FOR *M = 1 to Number of active node − 1* **DO**
IF *(Master Node_residual energy level < M_residual energy level)* **THEN**
Mater node = M
END IF
END FOR
Set to be Slave node except node M
$\omega = \left[\dfrac{1}{(N+1)^2} * \alpha\right]$
IF ω > *[Maximum transmission power]* **THEN**
Transmission power = [Maximum transmission power]
ELSE
Transmission power = $\left[\dfrac{1}{(N+1)^2} * \alpha\right]$
END IF
Count = *count for next Heartbeat*
WHILE *Count > 0* **DO**
Master send its data to the slave nodes
Source nodes transmit as a distributed beamformer
Subtract 1 to Count
END WHILE
END WHILE

Algorithm 1: Intelligent Transmission Power Allocation algorithm pseudocode.

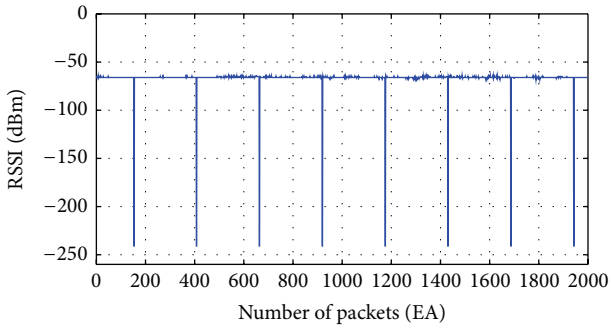


Figure 4: Estimated RSSI at the receiver.

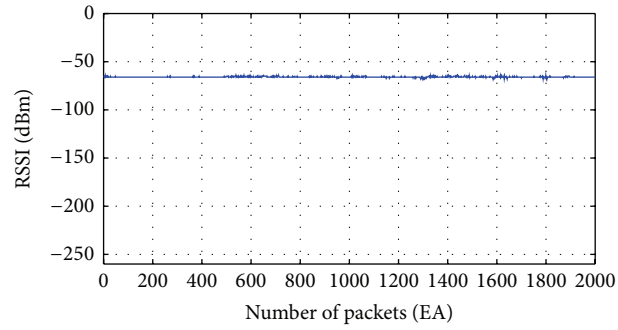


Figure 5: Estimated RSSI with EWP_{avg}.

the proposed scheme, we can achieve the reliable parameter value g_i which indicates undefined attenuation parameter. It can help Intelligent Transmission Power Allocation algorithm to adapt to the changeable wireless channel states.

Figure 5 shows the received power data after removing the noisy data. Our proposed scheme is able to eliminate the noise and more weightage to the present estimated data efficiently. As shown by the result, we are able to achieve clean received power data using our proposed scheme.

5. Performance Evaluation

For the specific numerical performance evaluation, the circuit-related parameters need to be defined first. Our performance evaluation parameters are shown in Table 1.

Table 1: Performance evaluation parameters.

Parameter	Value
Carrier frequency	3 GHz
Channel model	Rician fading
Rician factor	−40 dB~15 dB
Bandwidth	10 MHz
Transmit antenna gain	2 dBi
Receiving antenna gain	10 dBi
Transmission power	−10 dBm
Attenuation parameter	0.7~1
Temperature	300 Kelvin
Noise figure	6 dB
Number of source nodes	1~20
Distance	1000 m, 2000 m, 3000 m
Energy per source node	$4 * 10^6$ mJ
Current consumption (transmit mode: $P = -10$ dBm)	11 mA
Current consumption (receiving mode)	5 mA
Current consumption (idle mode)	426 μA
Transmission period	30 seconds

For generality, we compared our proposed ITPA with other distributed beamforming schemes, MAX-SINR [24, 25, 29], MMSE [26, 27, 29], Weighted MMSE [26, 28, 29], which do not consider degradation of channel capacity when the number of cooperating source nodes is decreased, to evaluate the transmission power, channel capacity, and network life span time performance.

As shown in Figure 6, SNR versus number of source nodes, if parts of cooperating source nodes cannot transmit signals, SNR is decreased regardless of schemes. Thus, it causes degradation of channel capacity because $N2$-fold power gain is decreased when the number of cooperating source nodes is decreased. In addition, it is also shown that acquired channel capacity is decreased exponentially when the distance between the average source nodes and destination increases, because of path loss in terms of wireless propagation. It is essential that each source node increases transmission power to compensate the degradation of power gain to guarantee the required channel capacity considering the number of cooperating source nodes and the distance between the average source nodes and destination.

Figure 7 shows the transmission power versus source nodes that guarantee the channel capacity (50 Mbps) of the system. ITPA required transmission power about −18.62 dBm at 1000 m, −12.50 dBm at 2000 m, and −9.38 dBm at 3000 m, respectively, to guarantee the channel capacity when the number of cooperating source nodes is 20. Each source node should increase transmission power cooperatively to compensate degradation of channel capacity, as shown in Figure 6, considering the maximum transmission power (0 dBm), when the number of cooperating source nodes is decreased. Since allocating over the maximum transmission

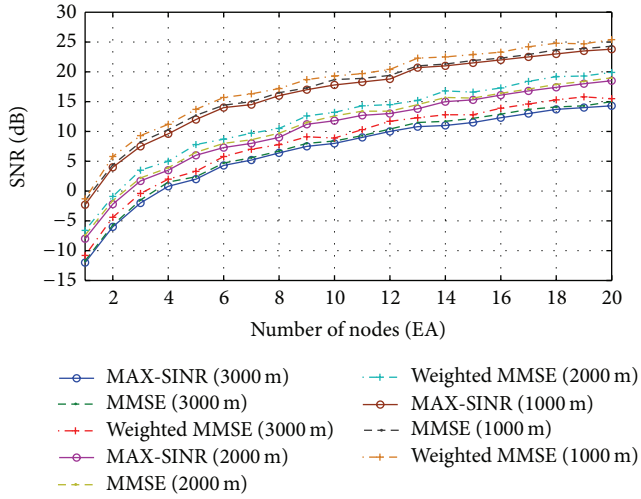


Figure 6: SNR versus number of source nodes.

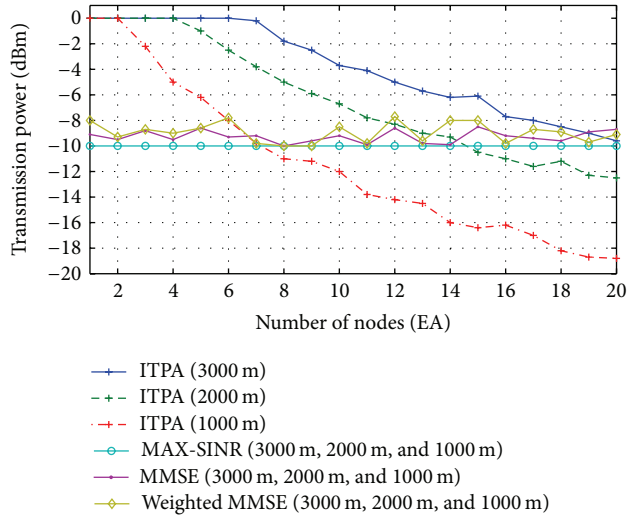


Figure 7: Transmission power versus number of source nodes.

power is, physically, impossible. And an excessive increase in transmission power leads to the performance degradation of network life span time. ITPA algorithm avoids these problems and allocates an appropriate transmission power to the source nodes based on its computation. On the other hand, the system with previous distributed beamforming schemes is not capable of power control considering the number of cooperating source nodes. MAX-SINR does not provide power control. Thus, it just transmits by default transmission power (−10 dBm). MMSE provides power control which only adapts channel statement. And Weighted MMSE also provides power control and it computes additionally for sum rate objectivity compared with MMSE. Therefore, Weighted MMSE could improve the channel capacity performance compared with MMSE. However, as mentioned above, these previous schemes have limitations, since they cannot adapt to the number of cooperating source nodes and the distance between the average source nodes and destination.

Table 2: Minimum number of source nodes for the required channel capacity.

	1000 m				2000 m				3000 m			
	ITPA	MAX-SINR	MMSE	Weighted MMSE	ITPA	MAX-SINR	MMSE	Weighted MMSE	ITPA	MAX-SINR	MMSE	Weighted MMSE
10 Mbps	1	1	1	1	1	3	3	3	2	4	4	4
30 Mbps	2	4	4	3	3	8	7	7	4	11	11	9
50 Mbps	3	7	7	6	5	14	12	12	7	21	21	18
70 Mbps	5	14	13	13	9	28	28	26	14	30	29	27

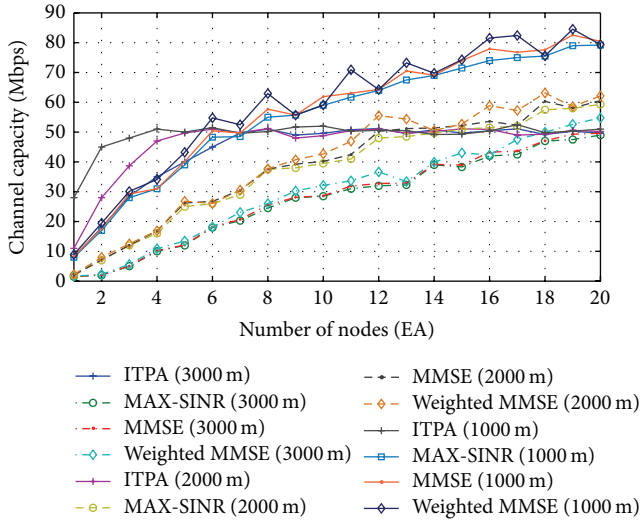


Figure 8: Channel capacity versus number of source nodes.

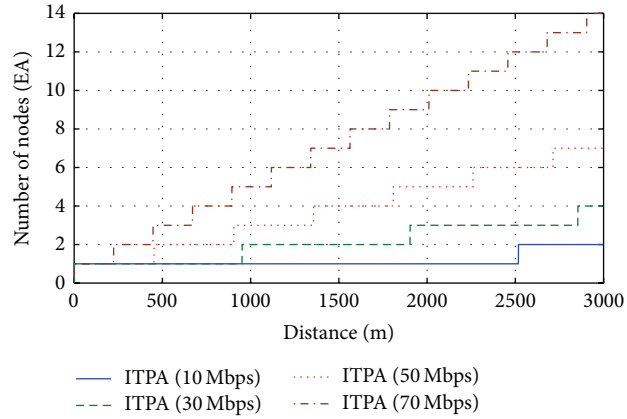


Figure 9: ITPA's minimum number of source nodes versus distance.

Figure 8 shows the channel capacity versus the number of source nodes. Channel capacity is decreased rapidly with previous schemes (MAX-SINR, MMSE, and Weighted MMSE) when the number of cooperating source nodes is decreased. On the other hand, the channel capacity is guaranteed (50 Mbps) with ITPA since ITPA can control transmission power of each source node according to the number of the cooperating source nodes and the distance between the average source nodes and destination. However ITPA also cannot guarantee the channel capacity when the required transmission power is over the maximum transmission power. In the section where the number of source nodes is 0~6, as shown in Figure 8, we can find this problem. Therefore, it is important to study the minimum number of source nodes for the required channel capacity when ITPA is used.

As shown in Table 2, ITPA guarantees the required channel capacity with the fewer source nodes than the other previous beamforming schemes (MAX-SINR, MMSE, and Weighted MMSE). At 1000 m, the number of cooperating source nodes, that ITPA needs, is 1, 2, 3, and 5, to guarantee the channel capacity of 10 Mbps, 30 Mbps, 50 Mbps, and 70 Mbps, respectively. And at 2000 m, the number of cooperating source nodes is 1, 3, 5, and 9, to guarantee the channel capacity of 10 Mbps, 30 Mbps, 50 Mbps, and 70 Mbps, respectively. At 3000 m, the number of cooperating source nodes is 2, 4, 7, and 14, to guarantee the channel capacity of 10 Mbps, 30 Mbps, 50 Mbps, and 70 Mbps, respectively. The minimum number of source nodes is increased when the distance between the average source nodes and destination is increased or the required channel capacity is increased, since each source node cannot transmit over its maximum transmission power.

Figure 9 shows ITPA's minimum number of cooperating source nodes versus distance to guarantee the channel capacity. As mentioned above, each node has a limit of maximum transmission power in the practical nodes of WSNs. Thus we studied the minimum number of cooperating source nodes when the maximum transmission power is 0 dBm, in theory. The minimum number of cooperating source nodes is different according to the required channel capacity and the distance between the average source nodes and destination. From that result, we are able to improve ITPA feasibility in practical WSNs.

Figure 10 shows the network life span time versus the number of nodes when the required channel capacity is 50 Mbps. We assume the environment where the number of power constrained sources is 20 at first. Our proposed ITPA algorithm has more networks life span time compared with previous schemes (MAX-SINR, MMSE, and Weighted MMSE) at 1000 m, 2000 m, since ITPA can allocate suitable transmission power (under −10 dBm) to guarantee the required channel capacity considering the number of cooperating source nodes and the distance between the average source nodes and destination. Moreover ITPA changes the master node periodically based on the residual energy of

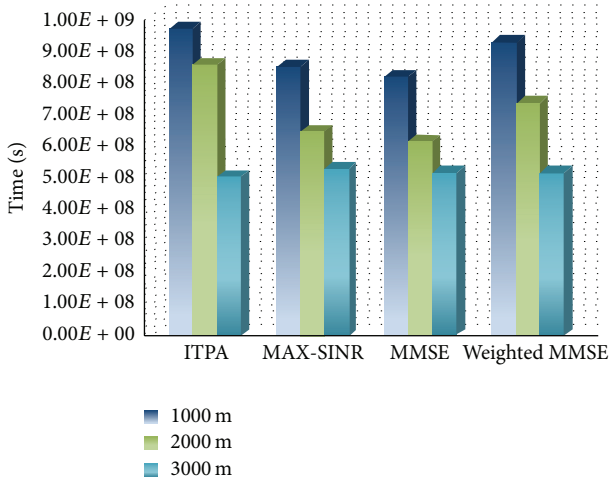


Figure 10: Network life span time versus number of nodes.

the source nodes. Thus, it can decrease the transmission power and save the energy of cooperating source nodes. At 3000 m, it is impossible theoretically to guarantee the channel capacity with default transmission power (−10 dBm) where the maximum number of sources is 20. However the other schemes just allocate around −10 dBm transmission power. On the other hand, ITPA allocates the transmission power which is over the default transmission power. As a result, ITPA consumes little more power than without ITPA at 3000 m. However the other schemes cannot guarantee the required channel capacity. Moreover ITPA has more network life span time than the other schemes if there are the unexpected situations that make source nodes inoperable.

6. Conclusion

Distributed beamforming can achieve the transmission gain by emulating an antenna array of a source. However, there are problems because of the source nodes which are deployed in distributed manner. We focused on the channel capacity of the system which is decreased when the parts of source nodes cannot transmit signal. It causes a critical problem when the system supports multimedia services. Therefore, we propose ITPA algorithm considering the changeable channel state, the number of cooperating source nodes, and the distance between the average source nodes and destination. ITPA can allocate suitable transmission power to the source nodes by adapting the changeable environment by itself. Thus, it can guarantee the required channel capacity and improve the network life span time largely. It will be useful in the practical environment where the distributed beamforming systems should be operated by the sources themselves.

Conflict of Interests

The authors declared that there was no conflict of interests regarding this paper.

References

[1] J. Uher, T. A. Wysocki, and B. J. Wysocki, "Review of distributed beamforming," *Journal of Telecommunications and Information Technology*, vol. 1, no. 1, pp. 78–88, 2011.

[2] D. Cui, L. Shu, J. Niu, and L. Wang, "Wireless sensor networks based research issues in large-scale petrochemical industries," *Sensor Letters*, vol. 11, no. 9, pp. 1675–1680, 2013.

[3] A. Chaganty, G. R. Murthy, N. Chilamkurti, and S. Rho, "A novel levelled and sectored based hybrid protocol for wireless sensor networks," *Sensor Letters*, vol. 11, no. 9, pp. 1715–1720, 2013.

[4] I. Joe and S. Chung, "The distance-power consumption trade-off with the optimal number of relays for cooperative Wireless Sensor Networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 9, no. 2, pp. 104–110, 2012.

[5] S. Zhang, J. Xue, C. Hu, and Y. Wang, "Binary spray and wait routing based on the remaining life time of message in wireless sensor networks," *Sensor Letters*, vol. 11, no. 9, pp. 1586–1590, 2013.

[6] N. Kumar and J. Kim, "An advanced energy efficient data dissemination for heterogeneous wireless sensor networks," *Sensor Letters*, vol. 11, no. 9, pp. 1771–1778, 2013.

[7] G.-J. Li, X.-N. Zhou, J. Li, and J.-H. Zhu, "Distributed targets tracking with dynamic power optimization for wireless sensor networks," *Sensor Letters*, vol. 11, no. 5, pp. 910–917, 2013.

[8] L. Guo, B. Wang, C. Gao, W. Xiong, and H. Gao, "The comprehensive energy-Routing protocol based on distributed cluster optimization in wireless sensor networks," *Sensor Letters*, vol. 11, no. 5, pp. 966–973, 2013.

[9] J.-P. Sheu, C.-C. Chang, and W.-S. Yang, "A distributed wireless sensor network testbed with energy consumption estimation," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 6, no. 2, pp. 63–74, 2010.

[10] R. Mudumbai, D. R. Brown III, U. Madhow, and H. V. Poor, "Distributed transmit beamforming: challenges and recent progress," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 102–110, 2009.

[11] Y.-S. Tu and G. J. Pottie, "Coherent cooperative transmission from multiple adjacent antennas to a distant stationary antenna through AWGN channels," in *Proceedings of the 55th Vehicular Technology Conference*, pp. 130–134, Birmingham, Ala, USA, May 2002.

[12] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[13] H. Son, S. Lee, S.-C. Kim, and Y.-S. Shin, "Soft load balancing over heterogeneous wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2632–2638, 2008.

[14] H. Lee, H. Son, and S. Lee, "Semisoft handover gain analysis over OFDM-based broadband systems," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1443–1453, 2009.

[15] Y. Jing and H. Jafarkhani, "Network beamforming using relays with perfect channel information," *IEEE Transactions on Information Theory*, vol. 55, no. 6, pp. 2499–2517, 2009.

[16] V. Havary-Nassab, S. ShahbazPanahi, A. Grami, and Z.-Q. Luo, "Distributed beamforming for relay networks based on second-order statistics of the channel state information," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4306–4316, 2008.

[17] R. Krishna, Z. Xiong, and S. Lambotharan, "A cooperative MMSE relay strategy for wireless sensor networks," *IEEE Signal Processing Letters*, vol. 15, pp. 549–552, 2008.

[18] D. H. Nguyen and H. H. Nguyen, "Power allocation in wireless multiuser multi-relay networks with distributed beamforming," *IET Communications*, vol. 5, no. 14, pp. 2040–2051, 2011.

[19] M. Zeng, R. Zhang, and S. Cui, "On design of collaborative beamforming for two-way relay networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2284–2295, 2011.

[20] S. Talwar, Y. Jing, and S. Shahbazpanahi, "Joint relay selection and power allocation for two-way relay networks," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp. 91–94, 2011.

[21] S. Atapattu, Y. Jing, H. Jiang, and C. Tellambura, "Relay selection schemes and performance analysis approximations for two-way networks," *IEEE Transactions on Communications*, vol. 61, no. 3, pp. 987–998, 2013.

[22] Y. Jing and S. ShahbazPanahi, "Max-min optimal joint power control and distributed beamforming for two-way relay networks under per-node power constraints," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6576–6589, 2012.

[23] S. Shahbazpanahi and M. Dong, "Achievable rate region under joint distributed beamforming and power allocation for two-way relay networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, pp. 4026–4037, 2012.

[24] K. Gomadam, V. R. Cadambe, and S. A. Jafar, "Approaching the capacity of wireless networks through distributed interference alignment," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 1–6, New Orleans, Lo, USA, December 2008.

[25] S. Stanczak, M. Kaliszan, and N. Bambos, "Admission control for power-controlled wireless networks under general interference functions," in *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers*, pp. 900–904, Pacific Grove, Calif, USA, October 2008.

[26] S. S. Christensen, R. Agarwal, E. de Carvalho, and J. M. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4792–4799, 2008.

[27] H. Shen, B. Li, M. Tao, and X. Wang, "MSE-based transceiver designs for the MIMO interference channel," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3480–3489, 2010.

[28] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[29] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, "Comparison of distributed beamforming algorithms for MIMO interference networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3476–3489, 2013.

[30] E. Elnahrawy and B. Nath, "Cleaning and querying noisy sensors," in *Proceedings of the International Conference on Wireless Sensor Networks and Applications*, pp. 78–87, San Diego, Calif, USA, September 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 486023, 11 pages
http://dx.doi.org/10.1155/2015/486023

Research Article

Shared MPR Sets for Moderately Dense Wireless Multihop Networks

Teruaki Kitasuka[1] and Shigeaki Tagashira[2]

[1]*Graduate School of Science and Technology, Kumamoto University, Kumamoto 860-8555, Japan*
[2]*Faculty of Informatics, Kansai University, Osaka 569-1095, Japan*

Correspondence should be addressed to Teruaki Kitasuka; kitasuka@cs.kumamoto-u.ac.jp

Received 19 December 2014; Revised 11 April 2015; Accepted 18 April 2015

Academic Editor: Changqiao Xu

Multipoint relays (MPRs) are used for flooding topology control messages and finding the shortest paths for unicast communications in the optimized link state routing protocol (OLSR). In this paper, we propose a method for achieving more efficient MPR selection in moderately dense wireless multihop networks (including sensor networks) than the conventional MPR selection. First, we analyze moderately dense networks to show that a node close to the two-hop border has little probability of being a two-hop neighbor. Second, we explain that there is a chance of the node's MPRs being shared with its neighbors. To maximize this chance, we propose using shared MPR sets. These sets minimize the MPR ratio, which is defined as the number of nodes selected as MPRs by at least one neighbor divided by the total number of nodes in the network. Simulations are used to confirm the efficiency of using shared MPR sets. A centralized heuristic algorithm shows an MPR ratio redundancy in moderately dense networks that is about 10% of that obtained through conventional MPR selection.

1. Introduction

Multipoint relays (MPRs) [1, 2] support the efficient flooding of topology control (TC) messages in the optimized link state routing protocol (OLSR) [3] and are used to find the shortest path to any pair of nodes for unicast communications in OLSR. Although OLSR is designed as a routing protocol for mobile ad hoc networks (MANETs), it can also be used for sensor networks. OLSR is a proactive routing protocol on which each node regularly exchanges topology information with other nodes.

MPR is the key concept used in OLSR. Each node selects a subset of its neighbors as its MPR set. According to the RFC 7181 [4], the MPR set is selected to satisfy two properties: (1) *if a node v sends a message and that message is successfully forwarded by all MPRs of v, then all 2-hop neighbors of v will receive that message* and (2) *keeping the MPR set small ensures that the overhead of the protocol is kept at a minimum*.

The nodes that have been selected as MPRs have two roles: generating and forwarding TC messages into the entire network and acting as routers. Each TC message generated by a node w advertises the link state information between itself and a node v during v's selection of w as one of its MPRs. In contrast, non-MPR nodes (none of which is selected as an MPR by a neighbor) do not generate or forward TC messages except to enable an OLSR redundancy option called `TC_REDUNDANCY`. Thus, it is implied that non-MPR nodes act as edge nodes; that is, they do not act as routers. The path constructed with a sequence of links between an MPR and its selector is the shortest path.

We analyze the density of MPRs in moderately dense wireless multihop networks including sensor networks and present two issues that arise with conventional MPR selection: high MPR density and redundancy. First, the high MPR density issue implies that many TC messages are generated and flooded. Wu et al. [5] have already proven the asymptotic property that the average number of MPRs in a finite region is infinite when the network is extremely dense. However, we analyze moderately dense networks for additional insight into their nonasymptotic properties. Our concern is the distance from a node to its two-hop neighbor.

The second issue—redundancy of the conventional MPR selection—causes redundancy in dense networks because it minimizes the number of MPRs selected by each node using a fully distributed manner. Therefore, we propose a concept called "shared MPR sets" to reduce routing overhead. Through a simulation with a heuristic shared MPR selection algorithm, which is not distributed, we confirm the redundancy of conventional MPR selection in moderately dense networks.

For sensor networks, it is important to conserve the spectrum and energy. A way to achieve this is to improve the efficiency of OLSR by reducing TC messages. Our goal is to explain the following properties of moderately dense wireless multihop networks.

(i) Each node has a chance to share its MPRs with its neighbors (Section 3).

(ii) If MPR sharing is achieved well, OLSR's routing overhead is reduced (Section 4).

In this paper, we only discuss the existence of shared MPR sets with a centralized algorithm (the effectiveness of this method is shown through simulations); we do not discuss a feasible algorithm for finding shared MPR sets in a distributed manner. A preliminary version of this study appeared in [6]. Developing an efficient distributed algorithm is our main goal for future work. Several distributed heuristic algorithms have been proposed by Yamada et al. [7], Maccari and Lo Cigno [8], and ourselves [9]. However, the efficiency of those proposed algorithms has not yet been fully analyzed.

The rest of the paper is organized as follows. Related work is described in Section 2; this includes studies regarding OLSR, MPRs, and connected dominating sets (CDSs). In Section 3, we analyze the distance from a node to its two-hop neighbor, which is associated with conventional MPR selection, and introduce the concepts of MPR sharing and shared MPR sets. In Section 4, we provide a heuristic algorithm for shared MPR selection and provide the simulation results for dense wireless multihop networks. Based on those results, we confirm that routing overhead is reduced when shared MPR sets are used. Finally, we conclude this paper in Section 6.

2. Related Work

2.1. OLSR. OLSR [3] is a proactive routing protocol for MANETs. OLSR maintains neighborhood information and topology information in each node and can find the shortest path between any pair of nodes with relatively less control traffic.

We briefly review these two types of information and two types of messages. We denote a node by v, w, or u. Regarding neighborhood information, node v stores the partial two-hop information $G_{P2}(v) = (N(v) \cup N_2(v), E_{P2}(v))$, MPRs $M(v)$, and MPR selectors $M^{-1}(v)$. The node sets $N(v)$ and $N_2(v)$ are sets of one-hop neighbors and strict two-hop neighbors, respectively. The edge set $E_{P2}(v)$ is a set of symmetric links between one-hop neighbors and two-hop neighbors. Note that $E_{P2}(v)$ does not contain links between strict two-hop neighbors.

Neighborhood information is updated when node v receives a HELLO message. HELLO messages are broadcasted by all nodes periodically but are never forwarded. When node v receives a HELLO message from node w, v recognizes that w is a one-hop neighbor of v and adds w to $N(v)$.

The HELLO message of w includes one-hop neighbors $N(w)$ and MPRs $M(w)$. When v receives a HELLO message of w, for each node $u \in N(w)$, v adds u and (w, u) in $N_2(v)$ and $E_{P2}(v)$, respectively. Node v also adds w in $M^{-1}(v)$ if $v \in M(w)$. Afterward, node v computes its MPRs $M(v)$ by using the MPR selection algorithm [3] with updated $N(v)$, $N_2(v)$, and $E_{P2}(v)$ (The MPR selection algorithm is described in Section 2.2). Note that if node $u \in N_2(v)$ is also in $N(v)$, then node u is removed from $N_2(v)$. After several HELLO messages are exchanged, the neighborhood information satisfies

$$
\begin{aligned}
&N(v) = \{w \mid d(v, w) < r\} \\
&N_2(v) = \{u \mid u \in N(w), w \in N(v), u \notin N(v), u \neq v\} \\
&E_{P2}(v) = \{(w, u) \mid w \in N(v), u \in N(w)\} \\
&M(v) \\
&\quad \subseteq N(v) \\
&\text{such that } {}^{\forall}u \in N_2(v), \ {}^{\exists}w \in M(v), \ (w, u) \in E_{P2}(v) \\
&M^{-1}(v) = \{w \mid v \in M(w)\}.
\end{aligned} \tag{1}
$$

Note that $d(v, w)$ is the distance between v and w and r is the radio range.

The topology information for each node, denoted by u, is a directed subgraph of the network G. The directed subgraph, denoted by $G'(u) = (V'(u), E'(u))$, includes all reachable nodes and partial directed links. $G'(u)$ is generated on the basis of the received TC messages. A TC message is periodically generated by every MPR, denoted by w, and includes $M^{-1}(w)$. $M^{-1}(w)$ is called an MPR selector set. TC messages are flooded into the entire network, and each of it informs the links between w and each of $M^{-1}(w)$. When node u receives a TC message generated by w, for each $v \in M^{-1}(w)$, node u adds v and (w, v) in $V'(u)$ and $E'(u)$, respectively. Node u also adds w in $V'(u)$.

The routing table of node u is constructed using $G_{P2}(u)$ of the neighborhood information and $G'(u)$ of the topology information. By using these two types of information, node u can find the shortest path to any other node in the network for unicast communications.

OLSRv2 [4] has already released in April 2014. Therefore, the discussion here is applicable to OLSRv2. In OLSRv2, nodes can freely interoperate regardless of whether they use the same MPR selection algorithm; an example algorithm for calculating MPRs is available in appendix B of OLSRv2 [4]. OLSRv2 defines two MPR sets (flooding MPR and routing MPR) and adopts neighborhood discovery protocol (NHDP) [10] to acquire neighborhood information.

2.2. Multipoint Relays. Multipoint relaying has been proposed by Qayyum et al. [1, 2] for efficiently flooding broadcast messages in mobile wireless networks. The concept of multipoint relaying is to reduce the number of duplicate retransmissions while forwarding a broadcast message.

Each node v selects a small subset of its neighbors $N(v)$ as MPRs $M(v)$. When node v transmits a broadcast message generated either by itself or another node, each node $w \in M(v)$ retransmits the message only once; other neighbors do not retransmit it.

In MANETs, for proactive and reactive protocols, flooding is used to find a path to the destination. OLSR [3] also adopts the use of MPRs to disseminate TC messages.

Qayyum et al. [2] analyze MPRs and propose a heuristic MPR set selection algorithm. They prove that the following MPR problem is NP-complete: *given a network (i.e., the set of one-hop neighbors for each node), a node v of the network, and an integer k, is there a multipoint relay set for v of size less than k?*

The heuristic algorithm proposed by Qayyum et al. provides a near-optimal MPR set. They prove that the MPR set computed by that heuristic contains at most $\log n$ times more nodes than the optimal MPR set [2], where n is the number of nodes in the network. The input of the heuristic is the partial two-hop information $G_{P2}(v)$, and the output of the heuristic is the MPR set $M(v)$. The heuristic algorithm is stated as follows:

(1) Start with an empty MPR set $M(v)$.

(2) First, select as MPRs those one-hop neighbors in $N(v)$ that are only neighbors of some node in $N_2(v)$; add these one-hop neighbors to $M(v)$.

(3) While there still exists some node in $N_2(v)$ that is not covered by $M(v)$, one has the following:

  (a) For each node in $N(v)$ that is not in $M(v)$, compute the number of nodes that it covers among the uncovered nodes in $N_2(v)$.

  (b) Add the node of $N(v)$ in $M(v)$ for which this number is maximum.

Jacquet et al. [11] have analyzed OLSR MPR flooding in two network models: the random graph model and the random unit disk graph model. These two models are used for indoor and outdoor networks, respectively. In the two-dimensional random unit disk graph model, Jacquet et al. prove that the average size of the MPR sets tends to be smaller than $3\pi(n'/3\pi)^{1/3}$, where n' is defined by (3) in Section 3.1 below.

Busson et al. [12] have analyzed the conventional MPR selection in random unit disk graphs. They did not analyze sharing MPRs with neighbors; however, they do show that approximately 75% of MPR sets are selected in step 2 of the heuristic algorithm above, which implies that only the remaining 25% have a chance of sharing MPRs with their neighbors. The starting point of their analysis also uses (4), shown in Section 3.2 below. Our analysis in Section 3.2 is similar to theirs; however, unlike their research, we go on to analyze MPR sharing (Sections 3.3 and 3.4).

The motivation behind Maccari and Lo Cigno's research [8] is the same as ours. They describe the size of the global MPR set $M_G = \bigcup_{v \in V} M(v)$ as the objective function that needs to be minimized; this objective function also appears in [6]. They have carefully revised the MPR selection algorithm and addressed implementation issues associated with that algorithm. Furthermore, they propose using the selector set tie breaker (SSTB) distributed strategy to minimize M_G. Unlike us, their strategy for finding shared MPR sets is to use distributed algorithms; we use the centralized algorithm instead. However, we believe that the size of the global MPR set, which is calculated by the centralized algorithm, can be referred to as a type of numerical lower bound.

2.3. Connected Dominating Set. A connected dominating set (CDS) is a subset of nodes. Each node in a CDS has a path only through other nodes in the CDS, and every node in the network has at least one node in the CDS as a neighbor.

A small CDS is another candidate to reduce the number of forwarding nodes during the flooding process [13]. Instead of MPRs, a CDS can be used to flood messages to the entire network, and the nodes that are not there in CDS do not need to relay messages to flood. A small CDS will have less control traffic than MPRs. Furthermore, if the shortest path for every node is not a mandatory routing requirement, a small CDS can also be used for routing instead of MPRs.

A CDS does not have the (1) property of MPR sets that is described in Section 1. In other words, the use of MPRs guarantees the shortest path between any pair of nodes; CDS does not guarantee it. Furthermore, the time and message complexities of CDS schemes are slightly higher than those of MPR schemes. Perhaps for these reasons, unfortunately, CDSs are not currently employed in major MANET routing protocols.

Wu et al. [5] have explained several extensions designed to generate smaller CDSs using complete two-hop information. The complete two-hop information of node v is denoted by $G_{C2}(v) = (N(v) \cup N_2(v), E_{C2}(v))$. The difference between $G_{C2}(v)$ and $G_{P2}(v)$ is the sets of links. $E_{C2}(v)$ includes all links in $E_{P2}(v)$ as well as links between any pair of two-hop neighbors $N_2(v)$. Consider

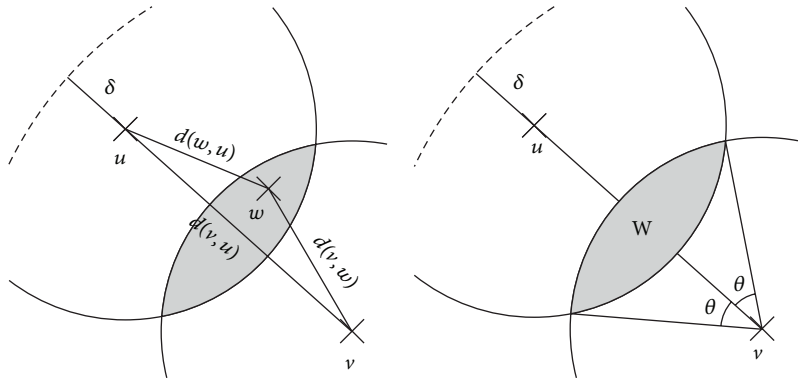
$$E_{C2} = E_{P2} \cup \{(w_1, w_2) \mid w_1, w_2 \in N_2(v), (w_1, w_2) \in V\}. \quad (2)$$

To construct $G_{C2}(v)$ in v, node v has to receive one-hop neighbor information $N(u)$ from each of its two-hop neighbor $u \in N_2(v)$. Because node u's HELLO messages have to reach all of u's two-hop neighbors, it implies that constructing G_{C2} requires higher communication overhead than constructing G_{P2}.

Wu et al. have proved that the extended MPR has a constant local approximation ratio rather than the logarithmic local ratio associated with the original MPR [5].

3. Analysis of MPRs in Dense Networks

3.1. Notation. We model a network as a unit disk graph $G = (V, E)$, where V is a set of nodes in the network and E is



(a) Node u is a two-hop neighbor of v, if w covers u

(b) Node u is not a two-hop neighbor of v if there is no node $\in W$

Figure 1: Determining whether u is a two-hop neighbor of v.

a set of available links between nodes. Each node knows partial information of G by receiving HELLO and TC messages. The total number of nodes in the network is denoted by n $(= |V|)$. For simplicity, we assume that the transmission range r of each node is uniform. There is an edge $(v, w) \in E$ if and only if $d(v, w) < r$, where the function $d(\,)$ is a distance between two nodes. A region $D(v, x)$ is defined as a disk with radius x that centers v. $D(v, r)$ is called a unit disk of v. A node in the unit disk $D(v, r)$ is called a one-hop neighbor of v.

Our analysis assumes that nodes are placed uniformly in a two-dimensional region. The expected number of nodes in a one-hop region is denoted by n'. For example, if the region is a square of side R and the number of nodes in the region is n, then

$$n' = \pi r^2 \frac{n}{R \times R}. \tag{3}$$

We use the term "density" (defined as n') instead of the number of nodes in a unit square, that is, n/R^2, for convenience.

3.2. Distance to Two-Hop Neighbors. Suppose that there are two nodes v and u such that $d(v, u) = 2r - \delta \le 2r$, as shown in Figure 1(a). We discuss the condition where u is a two-hop neighbor of v. Obviously, node u is a two-hop neighbor of v if and only if there is at least one node w that satisfies $d(v, w) \le r$ and $d(w, u) \le r$. In other words, $d(v, u) \le 2r$ is a necessary but insufficient condition for u being a two-hop neighbor of v.

The region W is defined as $W = D(v, r) \cap D(u, r)$, as shown in Figure 1(b). Node u is a two-hop neighbor of v if and only if there is at least one node $w \in W$. The size of region W is denoted by $S(W)$ and expressed as follows:

$$\begin{aligned} S(W) &= 2\left(\pi r^2 \cdot \frac{2\theta}{2\pi} - r^2 \sin\theta \cos\theta\right) \\ &= r^2 (2\theta - \sin 2\theta), \end{aligned} \tag{4}$$

where $\theta = \theta(\delta) = \arccos((2r - \delta)/2r)$. We define a probability function $p(\delta, n')$ that expresses the probability that u is a two-hop neighbor of v when u is $2r - \delta$ away from v. Suppose that node w is a two-hop neighbor of v. The probability that w is not in the region W is $(\pi r^2 - S(W))/\pi r^2$. Node u is a two-hop neighbor of v if there is at least one one-hop neighbor of v in W. Therefore, in uniformly distributed networks, we approximate

$$p(\delta, n') = 1 - \left(\frac{\pi r^2 - S(W)}{\pi r^2}\right)^{n'}, \tag{5}$$

where n' is the average number of a node's one-hop neighbors.

We try to expect the number of two-hop neighbors of v in various densities. First, we denote the expected number of nodes in $D(v, 2r)$ by $f(2r) = \pi(2r)^2 \cdot n'/\pi r^2 = 4n'$. Note that $f(2r)$ includes the number of one-hop neighbors in $D(v, r)$. We suppose that $f'(x) = 2n'x/r^2$ is the derivative of $f(x)$. We denote the expected number of two-hop neighbors of v by $E[|N_2(v)|]$. We derive $E[|N_2(v)|]$ using a function $g(x, n')$, which is the expected number of two-hop neighbors of v in a disk $D(v, x)$ with the radius x $(r \le x \le 2r)$ and the density n'. The function $g(x, n')$ is derived using $f'(x)$ and $p(\delta, n')$. Consider

$$E\left[|N_2(v)|\right] = g(2r, n') \tag{6}$$

$$g(x, n') = \int_r^x f'(y) \cdot p(2r - y, n')\, dy. \tag{7}$$

Figure 2 shows the numerical results of $g(x, n')$ with various radii x and densities $n' = \infty$, 80, 40, 20, 10, and 5. From these results, we confirm that nodes close to the border of $D(v, 2r)$ are rarely a two-hop neighbor of v under moderately high-density conditions.

To ensure the rareness of two-hop neighbors close to the border of $D(v, 2r)$ in moderately dense networks, Table 1 shows the radius x of a disk covering $q = 70\%$–95% of two-hop neighbors. The radius x satisfies $(q/100)g(2r, n') = g(x, n')$ for each q and n'. We conclude that, even when density $n' = 80$ (a very high density where the expected number of two-hop neighbors is 214), 95% of the two-hop neighbors (nearly 203 nodes) have $\delta > 0.10r = (2 - 1.90)r$,

Table 1: The radius x of a disk that covers q% of two-hop neighbors on various density n'.

q	$n' = \infty$	80	40	20	10	5
70%	1.76r	1.69r	1.66r	1.61r	1.55r	1.51r
80%	1.84r	1.77r	1.74r	1.69r	1.64r	1.59r
90%	1.92r	1.85r	1.82r	1.78r	1.74r	1.70r
95%	1.96r	1.90r	1.87r	1.84r	1.81r	1.78r

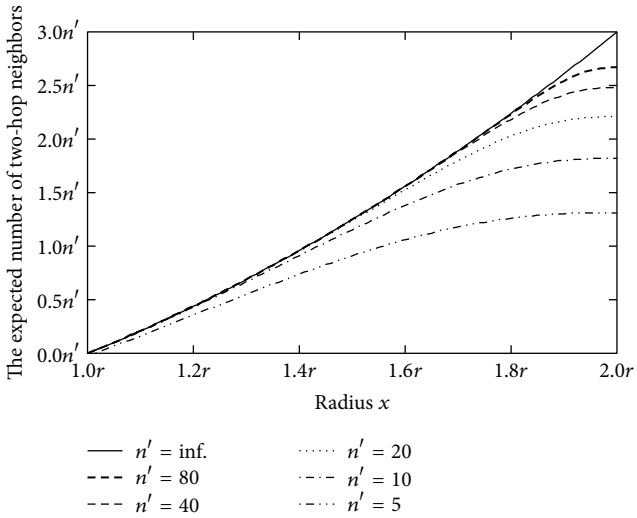


Figure 2: The expected number of two-hop neighbors in a disk of radius x.

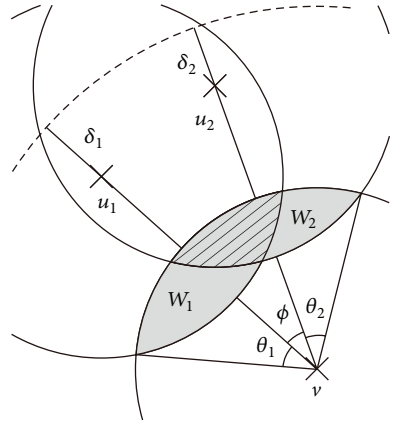


Figure 3: Covering two two-hop neighbors.

$\theta > 0.32 = 18^\circ$. In a moderately dense network, such as $n' = 20$ (where the expected number of two-hop neighbors is 44), 90% of two-hop neighbors (40 nodes) have $\delta > 0.22, \theta > 0.47 = 27^\circ$, and 70% of two-hop neighbors (31 nodes) have $\delta > 0.39, \theta > 0.64 = 36^\circ$. From this observation, we expect that some small number of MPRs can cover all of a node's two-hop neighbors in moderately dense wireless multihop networks. This expectation is confirmed by the simulation illustrated in Figure 7 of Section 4.

3.3. Conventional MPR Selection of a Node. Here, we review OLSR's conventional MPR selection in which any node selects its MPR set independently of its neighbors' MPR sets. Understanding the symmetric property of this conventional selection will be helpful in understanding how MPRs can be shared as described in Section 3.4.

Suppose that there are two nodes u_1 and u_2 that satisfy $d(v, u_1) = 2r - \delta_1$ and $d(v, u_2) = 2r - \delta_2$ and that they are node v's two-hop neighbors, as shown in Figure 3. We assume that u_1 and u_2 are closer to each other than to other v's two-hop neighbors.

The regions W_1 and W_2 are defined similar to W in Section 3.2. Consider

$$W_1 = D(v, r) \cap D(u_1, r)$$
$$W_2 = D(v, r) \cap D(u_2, r), \quad (8)$$

where u_1 (or u_2) is a two-hop neighbor of v if and only if there is at least one node $w_1 \in W_1$ (or $w_2 \in W_2$). The angles θ_1 and θ_2 are defined as $\theta_1 = \theta(\delta_1)$ and $\theta_2 = \theta(\delta_2)$, respectively.

The following discussion assumes that there are multiple nodes in W_1 and W_2; this avoids the case of having only one node in W_1 and W_2, in which case, the nodes must be added to MPR set $M(v)$ in step 2 of the heuristic described in Section 2.2.

If there is at least one node w in $W_1 \cap W_2$, which is the hatched region in Figure 3, then u_1 and u_2 can be covered by single node w ($= w_1 = w_2$). If there is no node in $W_1 \cap W_2$, there must be at least one node $w_1 \in W_1 \cap \overline{W_2}$ and the other node must be $w_2 \in \overline{W_1} \cap W_2$ to cover u_1 and u_2, respectively.

We denote $\angle u_1 v u_2$ by ϕ and suppose that $\delta_1 \geq \delta_2$; that is, $\theta_1 \geq \theta_2$. The inclusion relation between W_1 and W_2 is divided into the following three cases, where $S(\,)$ is the size of the region:

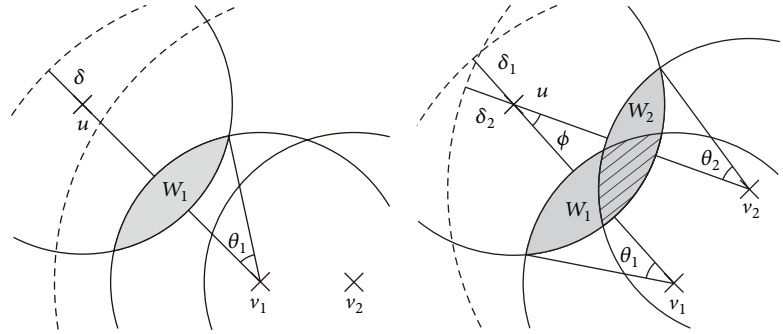
(1) $\phi > \theta_1 + \theta_2$; that is, $S(W_1 \cap W_2) = 0$.

(2) $\theta_1 + \theta_2 \geq \phi$ and $\phi > \theta_1 - \theta_2$, as shown in Figure 3; that is, $S(W_1 \cap W_2) > 0$.

(3) $\theta_1 + \theta_2 \geq \phi$ and $\theta_1 - \theta_2 \geq \phi$; that is, $W_2 \subseteq W_1$.

In the first case, two nodes $w_1 \in W_1$ and $w_2 \in W_2$ are selected as MPRs of v to cover u_1 and u_2, respectively. Two nodes w_1 and w_2 are selected independent of each other.

In the second case, if there is at least one node $w \in W_1 \cap W_2$, the optimal or heuristic MPR selection algorithm selects w as an MPR of v. However, this selection depends on the existence of other two-hop neighbors. The nodes in v's MPR set are either a single node w or two nodes $w_1 \in W_1 \cap \overline{W_2}$ and $w_2 \in W_2 \cap \overline{W_1}$.



(a) u is a two-hop neighbor of v_1 but not a two-hop neighbor of v_2

(b) u is a two-hop neighbor of both v_1 and v_2

Figure 4: Covering two two-hop neighbors.

In the third case, if the optimal or heuristic MPR selection algorithm selects a node $w_2 \in W_2$ as an MPR to cover u_2, then node w_2 also covers u_1.

The number of MPRs of node v changes by selecting the second case. Each iteration of step 3 in the heuristic greedily adds a node in $M(v)$; thus, the algorithm covers the maximum number of uncovered nodes in $N_2(v)$.

3.4. Sharing an MPR with a Neighbor. Considering the symmetric property of conventional MPR selection (see Section 3.3), we explain how sharing MPRs with a neighbor is possible. Suppose that there are two nodes v_1 and v_2, each of which is a one-hop neighbor of the other. Sharing MPRs means that v_1 and v_2 select the same node as their MPR.

First, we discuss the condition in which sharing MPRs is not allowed. As shown in Figure 4(a), if v_1 has a two-hop neighbor u such that u is not a two-hop neighbor of v_2, then v_2 has no one-hop neighbor in W_1. In this condition, v_1 and v_2 cannot share an MPR to cover u.

However, if there is a node u such that u is a two-hop neighbor of both v_1 and v_2, as shown in Figure 4(b), and there is at least one node $w \in W_1 \cap W_2$, then v_1 and v_2 can select a node w as an MPR to cover u, thus sharing an MPR.

The sharable condition discussed above is symmetrical to the second case of conventional MPR selection, which is described in Section 3.3. However, the optimal and heuristic MPR selection described in Section 2.2 does not consider such sharing.

Note that the two-hop coverage of v_1 and v_2's MPR set is maintained regardless of whether the MPR is shared between them. The MPR is still guaranteed to construct the shortest path between any pair of nodes in the network, in contrast to CDSs, which do not guarantee the shortest path.

3.5. The Proposed Shared MPR Sets. The concept of shared MPR sets was first introduced by Yamada et al. [7]. They call the concept an MPR selection "redundancy," which is defined as follows: *for a combination of MPR sets of all nodes, if the number of MPRs in the network is greater than that of other combinations of MPR sets, the combination of MPR sets is redundant.*

We define the MPR ratio to measure the degree of sharing. The ratio for shared MPR selection will be less than that for conventional MPR selection. We also define the number of MPRs per node to compare the average size of each node's MPR sets; this number will be constant for shared and conventional MPR selections. The MPR ratio and number of MPRs per node are defined as follows:

$$\text{the MPR ratio} = \frac{\left|\bigcup_{v \in V} M(v)\right|}{n},$$
$$\text{the number of MPRs per node} = \frac{\sum_{v \in V} |M(v)|}{n}, \tag{9}$$

where V is the set of nodes in the network, $M(v)$ is an MPR set of node v, and n is the number of nodes in the network (i.e., $n = |V|$). The MPR ratio shows the number of nodes selected as MPRs by at least one neighbor in the network. If the ratio is 1, then all nodes are selected as MPRs. The number of MPRs per node shows the average number of MPRs selected by a node.

To compute $M(v)$ for all nodes, we define a bipartite graph $\mathcal{G} = (\mathcal{N} \cup \mathcal{N}_2, \mathcal{E})$, where $\mathcal{N}$ is the union of one-hop neighbor sets $N(v)$ of all $v \in V$ and $\mathcal{N}_2$ is the target pairs set derived by $N_2(v)$ of all $v \in V$. Consider

$$\mathcal{N} = \bigcup_{v \in V} N(v) \tag{10}$$

$$\mathcal{N}_2 = \{(v, u) \mid v \in V, u \in N_2(v)\} \tag{11}$$

$$\mathcal{E} = \{(w, p) \mid w \in \mathcal{N}, p = (v, u) \in \mathcal{N}_2, w \in N(v), u \in N(w)\}, \tag{12}$$

where $(w, p) = (w, (v, u)) \in \mathcal{E}$ means that there are two links (v, w) and (w, u); $\mathcal{N}$ and $\mathcal{N}_2$ satisfy the conditions of $|\mathcal{N}| \leq n$ and $|\mathcal{N}_2| = \Sigma_{v \in V} |N_2(v)|$.

To achieve the smallest MPR ratio, we need to find the smallest subset of $\mathcal{N}$ that covers all of $\mathcal{N}_2$. We define the coverage that $w \in \mathcal{N}$ covers as $(v, u) \in \mathcal{N}_2$ if and only if $w \in N(v)$ and $u \in N(w)$; that is, $(w, p) \in \mathcal{E}$. In conventional MPR selection, the coverage is defined for each node v as $w \in N(v)$ covers $u \in N_2(v)$ if and only if $u \in N(w)$; that is, $(w, u) \in E_{P2}$.

Let $\mathcal{M}$ be a subset of $\mathcal{N}$. The set $\mathcal{M}$ is called a global MPR set if a subset $\mathcal{M}$ covers all pairs of $\mathcal{N}_2$. For every node $v \in V$, each node $u \in N_2(v)$ has one or more one-hop neighbors in a global MPR set. Then, $N(v) \cap \mathcal{M}$ can be a candidate MPR set of v. Each node can select $N(v) \cap \mathcal{M}$ or its subset as the shared MPR set. The MPR ratio is given by $|\mathcal{M}|/n$.

The computational complexity of finding the shared MPR sets that minimize the MPR ratio is expected to be NP-complete, because the basic structure of the problem is the same as the MPR problem [1] described in Section 2.2. Then, we use a heuristic shared MPR selection algorithm, which is discussed in the next section.

4. Experiments of Sharing MPRs with Neighbors

4.1. A Heuristic Shared MPR Selection Algorithm. We use the following algorithm in our experiment to show that shared MPR sets can reduce the routing overhead, especially the number of TC messages; however, note that because it is a centralized algorithm, it is not directly applicable to OLSR or other MANET routing protocols.

The algorithm adopts the greedy heuristic proposed by Qayyum et al. [1]. The primary difference between it and the conventional OLSR algorithm is that this algorithm runs on a whole network (i.e., is nondistributed) rather than on each node. Only the final step (4) runs on each node.

The input of this algorithm is $\mathcal{G}$ as defined in Section 3.5, and the output is the MPR sets $M(v)$ for all nodes. The heuristic algorithm is as follows:

(1) Start with an empty global MPR set $\mathcal{M}$.

(2) First, select, as global MPRs, those nodes in $\mathcal{N}$ that are the only neighbors of pairs in $\mathcal{N}_2$, and add these nodes to $\mathcal{M}$.

(3) While there exists at least one pair in $\mathcal{N}_2$ that is not covered by $\mathcal{M}$, one has the following:

  (a) For each node in $\mathcal{N}$ that is not in $\mathcal{M}$, compute the number of pairs that it covers among the uncovered pairs in $\mathcal{N}_2$.

  (b) Add the node of $\mathcal{N}$ in $\mathcal{M}$ for which this number is maximum.

(4) For each node v, run the heuristic described in Section 2.2 to compute $M(v)$. However, $N(v) \cap \mathcal{M}$ is used instead of $N(v)$ as the heuristic's input. The heuristic outputs the MPR set $M(v)$ for each v.

4.2. Metrics and Method. We use five metrics to explain our simulation results. The first two metrics concern the number of MPRs: the MPR ratio and the number of MPRs per node (described in (9)).

The remaining three metrics concern the routing protocol's communication overhead: the number of TC messages, the number of OLSR packets, and the total size of OLSR packets (in bytes). These three metrics are measured at the data link layer using a simulator log and are normalized by dividing the number of nodes and the simulation duration. Note that we count TC messages that are generated by a node and are forwarded by other nodes.

Moreover, to clearly show the redundancy of conventional MPR selection, the MPR ratio and number of TC messages in conventional MPR and shared MPR selections are compared. We define the MPR redundancy of conventional MPR selection as

$$\frac{\left|\bigcup_{v\in V} M(v)\right| \text{ of conventional MPR selection}}{\left|\bigcup_{v\in V} M(v)\right| \text{ of shared MPR selection}} - 1. \quad (13)$$

We define the TC message redundancy of conventional MPR selection as

$$\frac{\#\text{ of TC messages of conventional MPR selection}}{\#\text{ of TC messages of shared MPR selection}} - 1. \quad (14)$$

We use the ns-2 simulator (ver. 2.29) with UM-OLSR v0.8.8 [14]. The simulation is set so that each node has an IEEE 802.11 interface, the transmission range is $r = 250$ meters, and all nodes are distributed randomly in a region of 2000×2000 meters^2 (i.e., an $8r \times 8r$ region); the HELLO and TC message intervals are set to 2 and 5 s, respectively, and all nodes are set to `will_default` willingness. The simulation for each scenario runs for 100 s, and the number of messages is counted during the last 80 s of each simulation.

For the simulation, we assume that nodes do not move. We show the results of conventional MPR selection adopted in OLSR as well as shared MPR selection (which are described in Section 4.1). The number of nodes n in the network varies from 20 to 300. For each number of nodes, fifty different node topologies are simulated; the average results are shown in Section 4.3. Note that some topologies with small numbers of nodes are not fully connected graphs.

Regarding communication overhead, we describe how OLSR is modified to evaluate the heuristic shared MPR selection algorithm. Communication overhead caused by aggregating $\mathcal{G}$ to a virtual server and informing the global MPR set $\mathcal{M}$ from the virtual server is ignored. If it is included, the last two metrics (the number of OLSR packets and the total size of OLSR packets (in bytes)) increase, but the other three do not change. To suppress the increase in the first two metrics, we need to consider using a distributed algorithm; this is the most important goal of our future study. In our simulation, there is no difference in the number of HELLO messages when conventional MPR selection in OLSR is used and when shared MPR selection is used; in other words, HELLO messages are used to create partial two-hop neighbor information $G_{P2}(v)$ for each node v, and MPR set $M(v)$ is broadcasted to the neighbors of v using HELLO messages.

We assume that the wireless multihop network is moderately dense (meaning that each node has about 5 to 10 one-hop neighbors). Statistically, the number of one-hop neighbors for each node, except for those close to the border of the $8r \times 8r$ region, is $n' = \pi \times 250^2 \times n/(2000 \times 2000)$ as defined in (3). The number of strict two-hop neighbors is expected to be $E[|N_2(v)|]$ based on (6). In networks of 300

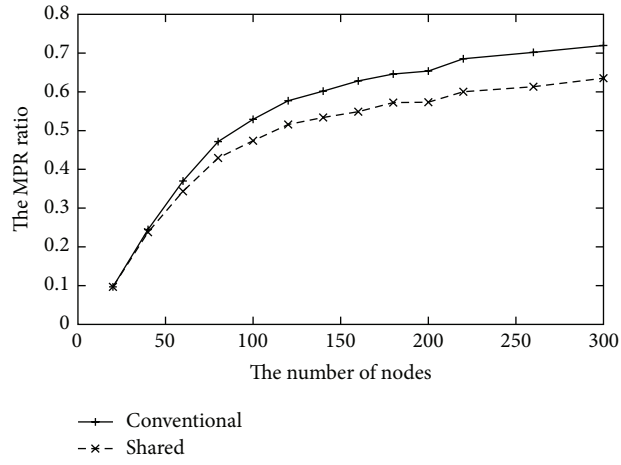


Figure 5: MPR ratio.

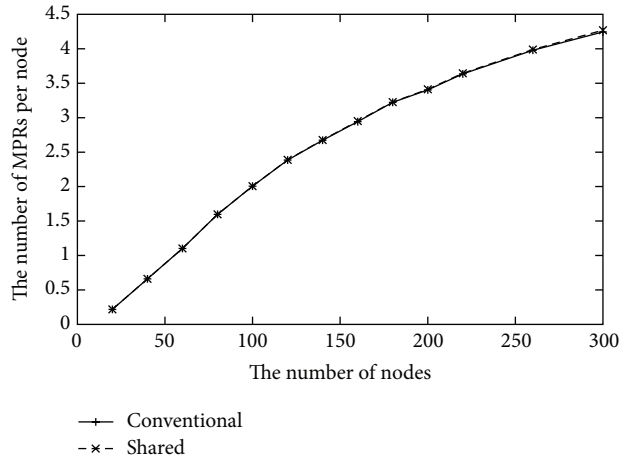


Figure 7: Number of MPRs per node.

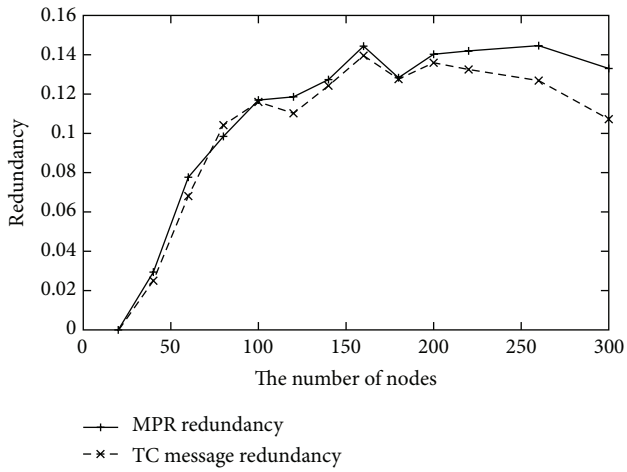


Figure 6: MPR and TC message redundancy.

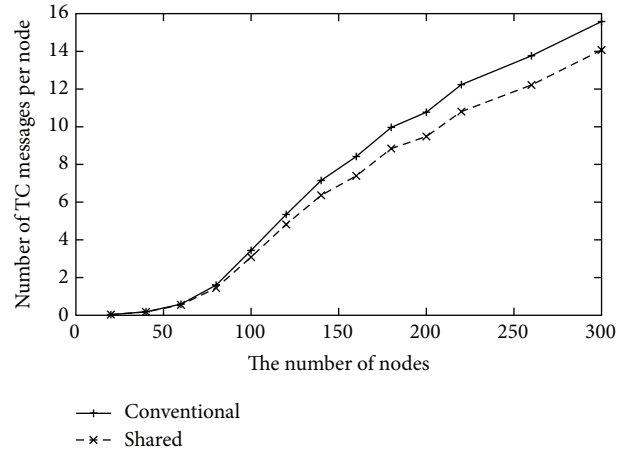


Figure 8: Number of TC messages per node per second.

nodes, there are statistically 14.7 one-hop neighbors and 34.2 strict two-hop neighbors for each node.

4.3. Simulation Results. Figures 5–10 show the simulation results for each metric. In each figure (except Figure 6), the results of conventional MPR selection and shared MPR selection are shown with the labels "Conventional" and "Shared," respectively.

The plot in Figure 5 shows that the MPR ratio for conventional MPR selection increases as the number of nodes increases (which is also discussed in Section 3.2). Wu et al. [5] have proved that this ratio will eventually increase to 1; however, the speed of increase shown in Figure 5 is rather slow.

By comparing the MPR ratios of conventional MPR selection with those of shared MPR selection (in Figure 5), we see that conventional MPR selection has over 10% redundancy in networks containing 100 or more nodes. Figure 6 charts MPR redundancy, which is defined in (13), thereby showing the redundancy more clearly. Based on the results shown in Figure 6, we determine that there is little redundancy in the low density networks of less than 50 nodes and large redundancy in moderately dense networks of over 100 nodes. In other words, when the average number of one-hop neighbors is greater than 5 ($n' = 4.9$ when $n = 100$), there is 10% redundancy in conventional MPR selection.

Figure 7 shows the average number of MPRs per node. Conventional MPR and shared MPR selection have almost the same results. The number of MPRs increases slowly when the number of nodes in the network increases. Even when $n = 300$ ($n' = 14.7$), no more than an average of 4.5 nodes are selected as MPRs by a node.

Figures 8 to 10 show the routing overhead results. These results are averaged per node and per second. Comparing conventional MPR selection to shared MPR selection, we find that, in all three metrics, shared MPR selection is able to reduce routing overhead in networks with 100 or more nodes.

When the number of nodes is greater than or equal to 80, the reduction ratio of shared MPR selection relative to conventional MPR selection is around 9%–12% in the number of TC messages (see Figure 8). More importantly, when the number of nodes is between 140 and 260, the reduction ratio

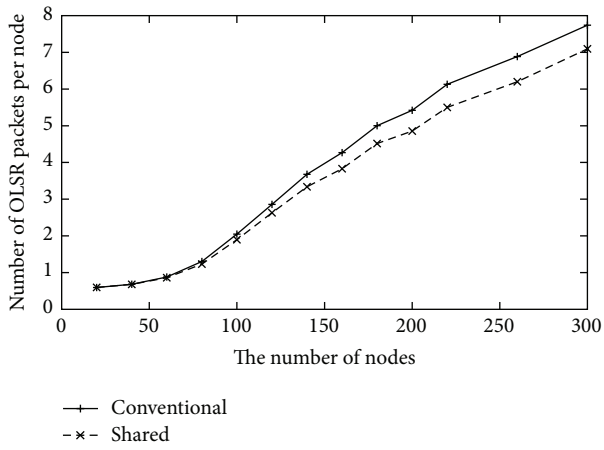


Figure 9: Number of OLSR packets per node per second.

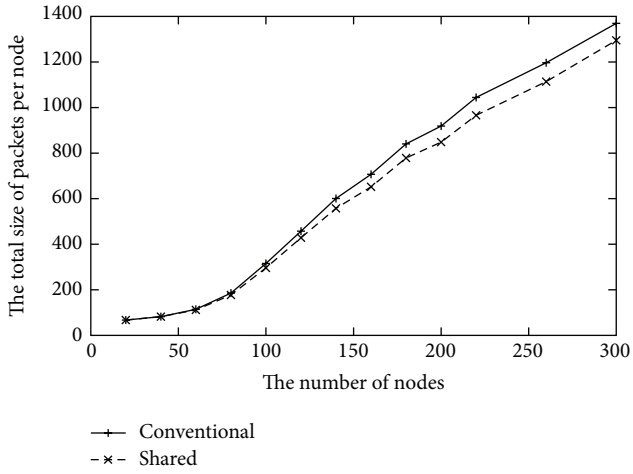


Figure 10: Size of OLSR packets (in bytes) per node per second.

is around 11%-12%. These results are also shown as TC message redundancy (defined in (14)) in Figure 6. From Figure 6, we observe that TC message redundancy is nearly proportional to MPR redundancy in networks of less than 200 nodes; however, we observe a different trend in networks of over 200 nodes. The reason for this different trend in high-density networks cannot be clearly explained, but it is possible that TC message redundancy decreases as networks increase in density. Future studies will examine this trend in greater detail to determine its cause.

Shared MPR selection also reduces the number of OLSR packets, depicted in Figure 9, and the reduction ratio here is even lower than that of the number of TC messages. Most likely, the ability to piggyback several messages into one packet causes the number of packets to decrease. The reduction ratio is around 9%-10% when the number of nodes is between 140 and 260 and around 5%–9% when the number of nodes is between 80 and 120.

Shared MPR selection also decreases the total size of OLSR packets (in bytes), as depicted in Figure 10, and the reduction ratio here is even lower than that of the number of TC messages and the number of OLSR packets. Although the headers of UDP, IP, and MAC will affect these results, the reduction ratio is around 7% when the number of nodes is between 140 and 260 and 4%–6% when the number of nodes is between 80 and 120.

To summarize the simulation results, we see that the MPR ratio increases slowly as the number of nodes in the network also increases, shared MPR selection maintains smaller MPR ratios and less routing overhead than does conventional MPR selection, and conventional MPR selection has room for improvement.

4.4. Comparison with CDS. To compare the MPR ratio with the CDS ratio, we refer to some results reported by Wu et al. [5]. Their evaluation simulates CDS within a $4r \times 4r$ region. We increase the size of the CDS as the number of nodes in the network also increases. For networks of 30, 50, and 80 nodes, the CDS sizes are around 15 (50%), 20 (40%), and 25 (30%), respectively. The CDS ratio shown in parentheses is obtained by dividing the CDS size by the number of nodes in a network. Comparing the density of nodes in their $4r \times 4r$ region with that in our $8r \times 8r$ region, 30, 50, and 80 nodes in the former region have similar densities to 120, 200, and 300 nodes in the latter region.

In high-density networks (e.g., in our simulation, networks of 300 nodes), the MPR ratios of conventional MPR and shared MPR selection are 72% and 64%, respectively, and the CDS ratio of the corresponding network is only 30%. Thus, both MPR ratios in Figure 5 are higher than the CDS ratio.

In moderately dense networks of 200 nodes (see Figure 5), the MPR ratio of conventional MPR selection is 65% (that of shared MPR selection is 57%), and the CDS ratio is 40% in a network of similar density.

When the network is not so dense, the difference between the MPR and CDS ratios is small. In low-density networks (e.g., in our simulation, the network of 120 nodes ($n' = 5.9$)), the MPR ratio of conventional MPR selection is 57% (that of shared MPR selection is 52%), and the CDS ratio is 50% in a network of similar density.

Consequently, if there is a feasible solution for calculating a CDS in a distributed manner and the network is dense, then the CDS will perform better than MPRs; however, the CDS does not guarantee the shortest path between any pair of nodes. Therefore, the next best choice is to find a feasible method for calculating shared MPRs in a distributed manner.

Comparing the MPR ratio with the CDS ratio, we see that the CDS achieves the smallest ratio and that shared MPR selection achieves the next smallest ratio.

5. Discussion

There are several distributed heuristic algorithms for calculating shared MPR sets, such as those put forth by Yamada et al. [7], Maccari and Lo Cigno [8], and ourselves [9]. However, in those studies, the sharing mechanism of MPR is not well analyzed and no bound is shown.

In Section 4, we simulate only a static environment; that is, nodes do not move in the simulation area. This limitation

is the result of the high computational complexity associated with the centralized algorithm described in Section 4.1, which exists because the structure of the problem solved by the centralized algorithm is the same as the local MPR computation, which is an NP-complete problem [1]. Furthermore, the problem size increases; for example, $|\mathcal{N}|$ of (10) is larger than $|N(v)|$ if the graph $G = (V, E)$ is not a complete graph, and $|\mathcal{N}_2|$ of (11) is about $|V|$ times larger than $|N_2(v)|$.

We employ the unit disk graph model for analysis and simulation. This model is valuable for theoretical discussions but does not represent the real environments of wireless communication. For an explanation of the differences between the model and real environments, we refer to the communication gray zones introduced by Lundgren et al. [15]. These zones are defined as areas where data messages cannot be exchanged—although HELLO messages indicate neighbor reachability—for various reasons including bit error rate, variable transmission rate, packet size, and different MAC layer handling between broadcast and unicast packets. To create simulations close to real environment, Chen et al. [16] and Pei and Henderson [17] have redesigned or tuned the IEEE 802.11 WLAN simulation model. More realistic evaluations can be performed with these models than with previous models. Furthermore, when we evaluate urban environments in our future study, obstacles should be modeled; for example, Sommer et al. [18] have proposed an empirical model of IEEE 802.11p path loss, including the attenuation of obstacles. For more realistic simulations, especially with mobile scenarios, these models will be valuable.

If nodes are allowed to move, then node mobility will be another important aspect to be considered in wireless multihop networks. Maccari and Lo Cigno [8] have proposed a stability-driven MPR choice strategy to minimize changes in the MPR selector sets in mobile scenarios. Minimizing these changes implies reducing the routing table calculation in each node. Musolesi and Mascolo [19] have also proposed a mobility model; theirs is based on community such as family members sharing a home or colleagues sharing an office. When mobile nodes are carried by humans, the community structures of humans strongly affect the dynamics of the mobile nodes. Therefore, the model put forth by Musolesi and Mascolo assigns each square area to a community. They have compared their mobility model with the Intel trace [20] and random waypoint mobility model in terms of intercontact times and contact durations. Our future work will include finding a suitable mobility model for evaluation.

6. Conclusion

In this paper, we explored MPR selection in moderately dense wireless multihop networks.

We have analyzed the distance to a two-hop neighbor in moderately dense networks and explained that nodes close to the border of a two-hop disk $D(v, 2r)$, where r is a transmission range, have little probability of being a two-hop neighbor of v. From this observation, we expected that some small number of MPRs could cover all two-hop neighbors of a node, even in moderately dense wireless multihop networks. This expectation was confirmed by the simulation of up to 300 nodes in an $8r \times 8r$ region.

We have also demonstrated the redundancy of conventional MPR sets and described a definition of shared MPR sets that minimizes the MPR ratio. We then provided a heuristic algorithm to select shared MPR sets. This heuristic is not applicable to the OLSR protocol, because it is not a distributed algorithm; however, the heuristic is valuable for showing the redundancy of conventional MPR selection. With this heuristic, we simulated some network topologies and measured the MPR ratio and routing overhead. Simulation results show that the redundancy in the number of TC messages, the number of OLSR packets, and the total size of OLSR packets is up to 12%, 10%, and 7%, respectively. We will continue to explore the feasibility of shared MPR selection as well as CDS schemes.

Conflict of Interests

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

Acknowledgment

This study was partially supported by KAKENHI (23500092 and 26330107).

References

[1] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: an efficient technique for flooding in mobile wireless networks," Research Report RR-3898, INRIA, 2000.

[2] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS '02)*, pp. 3866–3875, Big Island, HI, USA.

[3] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," RFC 3626, IETF, 2003.

[4] T. Clausen, C. Dearlove, P. Jacquet, and U. Herberg, "The optimized link state routing protocol version 2," RFC 7181, IETF, 2014.

[5] J. Wu, W. Lou, and F. Dai, "Extended multipoint relays to determine connected dominating sets in MANETs," *IEEE Transactions on Computers*, vol. 55, no. 3, pp. 334–347, 2006.

[6] T. Kitasuka and S. Tagashira, "Density of multipoint relays in dense wireless multi-hop networks," in *Proceedings of the 2nd International Conference on Networking and Computing (ICNC '11)*, pp. 134–140, Osaka, Japan, November 2011.

[7] K. Yamada, T. Itokawa, T. Kitasuka, and M. Aritsugi, "Redundant TC message senders in OLSR," *IEICE Transactions on Information and Systems*, vol. E93-D, no. 12, pp. 3269–3272, 2010.

[8] L. Maccari and R. Lo Cigno, "How to reduce and stabilize MPR sets in OLSR networks," in *Proceedings of the IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '12)*, pp. 373–380, Barcelona, Spain, October 2012.

[9] T. Kitasuka and S. Tagashira, "Finding more efficient multipoint relay set to reduce topology control traffic of OLSR," in *Proceedings of the IEEE 14th International Symposium on a World*

of Wireless, Mobile and Multimedia Networks (WoWMoM '13), pp. 1–9, IEEE, Madrid, Spain, June 2013.

[10] T. Clausen, C. Dearlove, and J. Dean, "Mobile ad hocnetwork (MANET) neighborhood discovery protocol (NHDP)," RFC 6130, Internet Engineering Task Force (IETF), 2011.

[11] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, "Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models," in *Proceedings of the 2nd IFIP-TC6 Networking Conference*, Pisa, Italy, 2002.

[12] A. Busson, N. Mitton, and E. Fleury, "Analysis of the multipoint relay selection in OLSR and implications," in *Challenges in Ad Hoc Networking*, vol. 197 of *IFIP International Federation for Information Processing*, pp. 387–396, Springer, New York, NY, USA, 2006.

[13] O. Liang, Y. A. Şekercioğlu, and N. Mani, "A survey of multipoint relay based broadcast schemes in wireless ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 30–46, 2006.

[14] F. J. Ros, UM-OLSR, http://masimum.inf.um.es/?Software:UM-OLSR.

[15] H. Lundgren, E. Nordström, and C. Tschudin, "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *Proceedings of the 5th ACM International Workshop on Wireless Mobile Multimedia (WoWMoM '02)*, pp. 49–55, September 2002.

[16] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 modeling and simulation in NS-2," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '07)*, pp. 159–168, October 2007.

[17] G. Pei and T. R. Henderson, "Validation of OFDM error rate model in ns-3," 15 pages, 2010, https://www.nsnam.org/~pei/80211ofdm.pdf.

[18] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A computationally inexpensive empirical model of IEEE 802.11p radio shadowing in urban environments," in *Proceedings of the 8th International Conference on Wireless On-Demand Network Systems and Services (WONS '11)*, pp. 84–90, January 2011.

[19] M. Musolesi and C. Mascolo, "A community based mobility model for ad hoc network research," in *Proceedings of the 2nd International Workshop on Multi-Hop Ad Hoc Networks: From Theory to Reality (REALMAN '06)*, pp. 31–38, Florence, Italy, May 2006.

[20] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket Switched Networks: real-world mobility and its consequences for opportunistic forwarding," Tech. Rep. 617, UCAM-CL-TR-617, University of Cambridge, Cambridge, UK, 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 701418, 11 pages
http://dx.doi.org/10.1155/2015/701418

Research Article

Modeling MAC Protocol Based on Frame Slotted Aloha for Low Energy Critical Infrastructure Sensor Networks

Niamat Ullah,[1] Kifayat Ullah,[2] S. M. Riazul Islam,[3] Pervaiz Khan,[3] Sana Ullah,[4] and Kyung Sup Kwak[3]

[1]*Computer Science Department, Government Postgraduate Jahanzeb College Swat, Khyber Pakhtunkhwa 19130, Pakistan*
[2]*Department of Computer Science, University of Swat, Saidu Sharif, Khyber Pakhtunkhwa 21300, Pakistan*
[3]*Graduate School of Information & Communication Engineering, Inha University, 253 Yonghyun-dong, Nam-gu, Incheon 402-751, Republic of Korea*
[4]*CISTER Research Unit, Polytechnic Institute of Porto, 4200-072 Porto, Portugal*

Correspondence should be addressed to Kyung Sup Kwak; kskwak@inha.ac.kr

Received 23 August 2014; Accepted 3 October 2014

Academic Editor: Liang Zhou

We propose and analyse a medium access control (MAC) protocol for low energy critical infrastructure monitoring (LECIM) networks. As the packet drop probability plays crucial role in LECIM applications, we propose framed slotted aloha based MAC for LECIM using linearly increased contention window size to reduce the packet drop probability. We present a mathematical model of our proposed MAC under both saturated and nonsaturated traffic scenarios. We use probabilistic approach to find the performance metrics such as collision probability, packet drop probability, throughput, and energy consumption. Also, we obtain the probability-generating function of the head-of-line (HoL) delay of packet. The analytical results match with simulations. Our results can be used in the design of a system by providing the optimum system parameters for endpoints satisfying the given quality of service requirements on packet drop probability, energy consumption, and delay.

1. Introduction

Critical infrastructure is a term used by governments to describe assets that are essential for the functioning of a society. Border surveillance, medical alerts for at-risk populations, first-responder tracking, soil monitoring, oil and gas pipeline monitoring, public transport tracking, cargo container monitoring, and railroad condition monitoring are some of the applications and facilities that are associated with LECIM networks. Here, we describe in detail two service applications—railway track condition monitoring and oil and gas pipeline monitoring.

The railway network is the most important system in the transportation infrastructure of a nation. It helps to sustain commerce in almost every sector of the national economy and is used for both pleasure and necessity by almost every citizen. Maintaining this system at a high performance level is vital for public safety, societal well-being, and economic productivity and growth. Railway tracks comprise significant and critical discrete links in the transportation system. Clearly, the job of monitoring the condition of rail infrastructure assets has become increasingly important over the last few years, for reasons of performance optimization to facilitate growth in traffic intensity, cost reduction in maintenance processes, railway patrolling staff role, cheaper measurement and data storage, and analysis capability. The preventive maintenance of railway tracks and structures has long centered in traditional methods for the early detection of potentially catastrophic faults. Such methods include visual inspections, which require a degree of experience to obtain results that are still subjective in interpretation. Further, such tests are invariably time-consuming and tedious to perform.

They are qualitative in nature and can only assess outward appearance. Any internal damage may go unnoticed for a long period of time. With the relentless aging of the railway track infrastructure, an effective railway track monitoring system has become imperative.

Oil and gas installations are assets of high importance and value. To move oil and gas from producing sites to refineries and from there to the markets for distribution, immense networks of pipeline are used. These pipelines are installed through densely populated urban areas, in some cases, over the surface of the earth, whereas in some cases they are located underground. Pipeline faces threats of leaks, damage, and breaks, which can be caused by terror attacks or due to aging equipment, extreme weather, earthquake, and so on. These threats lead to huge revenue losses, bedlam on international oil markets, and environmental pollution problems [1].

An ever increasing number of works are dealing with the protection of the sensed data in critical infrastructure monitoring applications; but accessing the shared medium in such networks has received comparatively little attention. The survey in [2] covers the general ideas about using WSN (wireless sensor network) to ensure the protection of critical infrastructure. In [3], the authors provide an overview of the main challenges and open research issues on critical information infrastructure security. In [4], the authors identify the precise security requirements for distributing the symmetric keys along the one-dimensional WSN used for monitoring an extended piece of linear infrastructure such as a pipe. The authors in [4] also propose lightweight key distribution schemes which could benefit applications like perimeter surveillance and pipeline monitoring. In [5], the authors make two contributions. First, they propose a model to maximize the amount of monitoring-related data that can survive after a portion of the critical infrastructure suffers a disaster. Second, they address the distribution of sensors in a specific application like oil pipeline so that an optimal placement of sensors could be achieved, while satisfying deployment constraints. The IEEE TG4k was formed to facilitate low energy operations for multiyear battery life and a simple and low cost communication environment with reliable data transfer. For this purpose, several proposals and MAC protocols were presented in [1, 6–13]. In [6], authors presented a MAC protocol for downlink communication using wake-up radio. In [1], Ullah et al. proposed multihop MAC for LECIM networks using wake-up radio. In [8], we proposed a MAC protocol based on a framed slotted aloha for LECIM network. In [8], we presented basic collision resolution approach, in which endpoints choose a slot for transmission in a super frame of size M. As the packet drop probability plays an important role in many LECIM applications such as oil/gas pipeline monitoring, railroad condition monitoring, and bridge condition monitoring, we propose a framed slotted aloha based MAC for LECIM using linearly increased contention window size to reduce packet drop probability. We call this mechanism enhanced collision resolution approach. In this approach, the endpoints increase the contention window (CW) linearly after collision instead of retransmitting the packet in the just next super frame.

In this paper, we present the mathematical analysis of our proposed MAC under both saturation and nonsaturation traffic scenarios. For nonsaturation case, we only consider CASE II (packets arrive to and queue in the buffer). We use probabilistic approach to find the performance metrics such as collision probability, packet drop probability, HoL-delay, and energy consumption for both saturation and nonsaturation traffic scenarios. The analytical results are then verified for accuracy by detailed comparison to simulation. Our results can be used to find the optimal number of endpoints while satisfying the QoS (quality of service) on the packet drop probability, energy consumption, and HoL-delay.

The rest of the paper is structured as follows. In Section 2, we discuss our basic collision resolution approach. In Section 3, we discuss the enhanced collision resolution scheme. In Section 4, we describe analytical model under saturated load. Section 5 gives analytical model for nonsaturated traffic scenario. We present analytical and simulation results in Section 6 followed by the conclusions.

2. Basic Collision Resolution Scheme

In the basic approach after packet arrival, the endpoints wait for the beacon. After listening to the beacon, the endpoints then choose a slot randomly using uniform distribution on [1, M] and then transmit in the slot. After collision, the endpoints choose a slot in the same manner in the just next super frame of size M. The packet gets dropped when the maximum retransmission limit R is reached (see Figure 1).

3. Enhanced Collision Resolution Approach

Unlike the basic approach, in the enhanced approach, the endpoints increase their contention window in linear order after collision. At the first transmission attempt, endpoints choose a random number in the interval $[1, M]$ using uniform distribution, where M is the size of the super frame representing the minimum CW. The size of the CW depends on the number of failed transmission attempts. After each collision, endpoints increase their CW linearly up to a maximum value of mM, where m is the maximum back-off stage. Once the CW reaches mM, the endpoints retain this value until it is reset back to M (see Figure 2). The endpoints reset the CW either after successful transmission or after packet drop. Endpoints drop the packet after R unsuccessful retransmission attempts. The CW can be calculated as follows:

$$\begin{aligned} \mathrm{CW}_i &= i * M \quad \text{if } 1 \le i < m \\ &= mM \quad \text{if } m \le i \le R. \end{aligned} \tag{1}$$

4. Performance Analysis under Saturated Load

In saturation condition, endpoints' queue never empties and, just after the completion of a transmission, endpoints take another packet from the buffer and start the transmission process. In the case of collision, the collided endpoints handle the collision as discussed above. The flowchart in

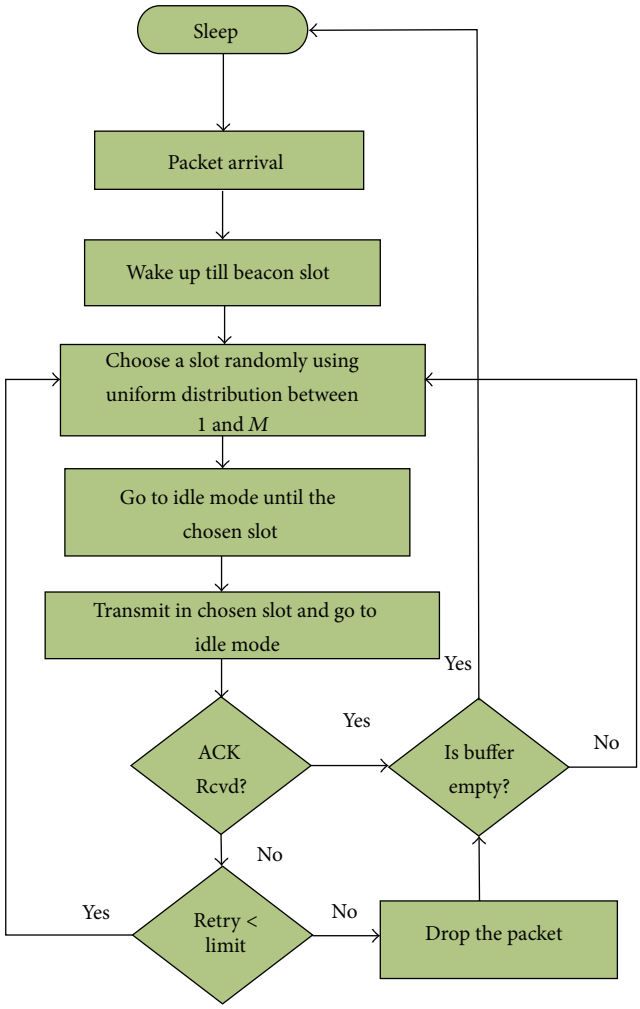


FIGURE 1: Flowchart describing the basic collision resolution scheme.

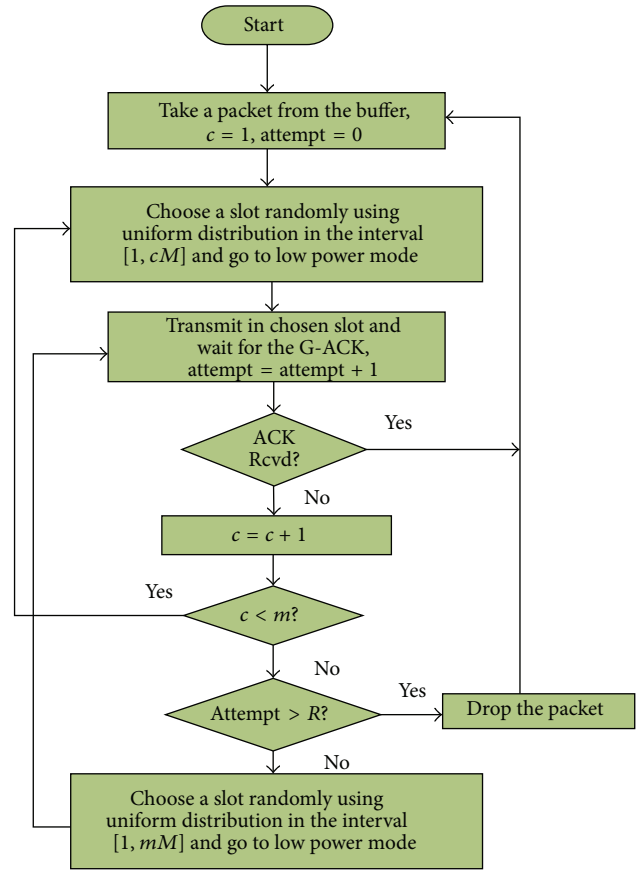


FIGURE 2: Flowchart depicting the transmission in saturated mode using enhanced collision resolution scheme.

Figure 2 describes the proposed collision resolution scheme under saturated traffic condition.

A transmission of the tagged endpoint is said to be successful if no other endpoint will select the slot chosen by the tagged endpoint. The probability of success P_S and the probability of collision P_C can be expressed as

$$P_S = \left(1 - \frac{1}{M}\right)^{N-1}$$
$$P_C = 1 - P_S. \tag{2}$$

4.1. Packet Drop Probability. In our proposed approach, the tagged endpoint drops the packet and takes another packet from the buffer if the packet reaches specified retransmission limit. The packet drop probability P_{drop} of the tagged endpoint can be written as

$$P_{\text{drop}} = \left\{1 - \left(1 - \frac{1}{M}\right)^{N-1}\right\}\left\{1 - \left(1 - \frac{1}{2M}\right)^{N-1}\right\} \cdots \left\{1 - \left(1 - \frac{1}{(m-1)M}\right)^{N-1}\right\} \cdot \left(\left\{1 - \left(1 - \frac{1}{mM}\right)^{N-1}\right\}\right)^{R-m+1}. \tag{3}$$

The first factor in (3) shows the failure probability of the tagged endpoint on the first attempt in a super frame of size M. The second factor describes the failure probability of the tagged packet in a super frame of size $2M$. Similarly, the last factor accounts for the fact that the tagged endpoint did not succeed in a super frame of size mM and will attempt to transmit in the super frame for $R - m + 1$ times.

4.2. HoL-Delay. The HoL-delay is defined as the time interval from the time when the packet is at the head of the queue ready to be transmitted until an acknowledgement for the packet is received or the packet is dropped.

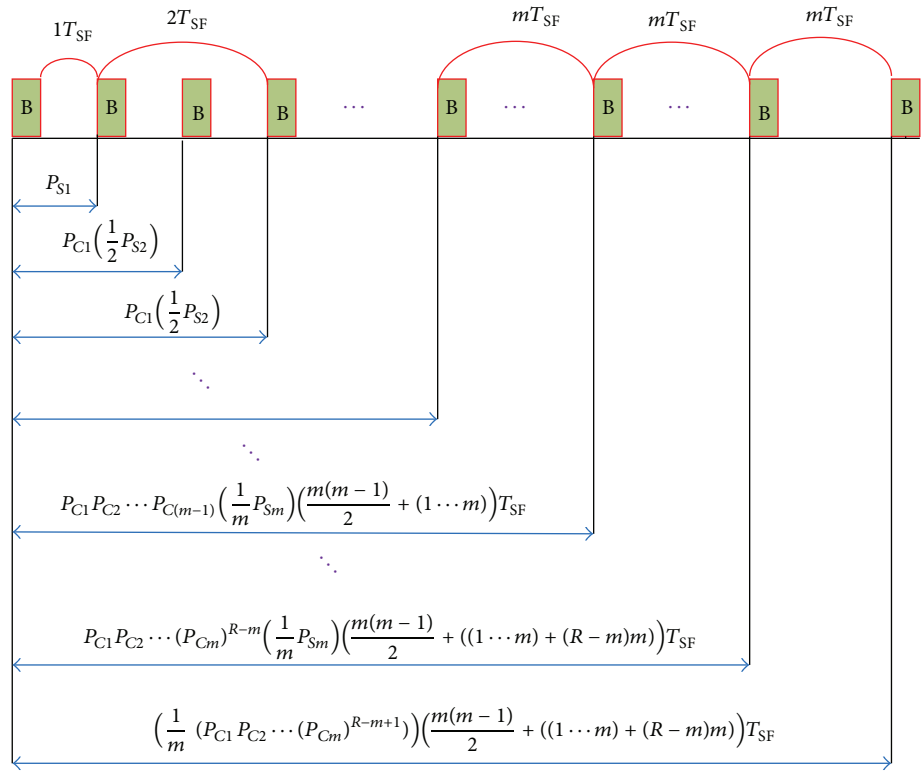


Figure 3: Head-of-line delay of the tagged packet.

Using Figure 3, the average HoL-delay of packet, $E[D]$, can be derived as follows.

Let

$$P_{S1}=\left(1-\frac{1}{M}\right)^{N-1},$$

$$P_{S2}=\left(1-\frac{1}{2M}\right)^{N-1}\cdots P_{Sm}=\left(1-\frac{1}{mM}\right)^{N-1}$$

$$P_{c1}=1-\left(1-\frac{1}{M}\right)^{N-1},\ldots,P_{cm}=1-\left(1-\frac{1}{mM}\right)^{N-1}$$

$$T_c=P_{C1}P_{C2}\cdots P_{C(m-1)}\left(\frac{1}{m}P_{Sm}\right),\qquad T_m=\frac{m\,(m-1)}{2},$$

$$T_n=((3R-R)\,\delta)\,P_{RX}+(R*\delta)\,P_{TX}. \tag{4}$$

The $E[D]$ can be expressed as

$$\begin{aligned}E[D]=&\,P_{S1}T_{SF}+P_{C1}\left(\frac{1}{2}P_{S2}\right)(2T_{SF})+P_{C1}\left(\frac{1}{2}P_{S2}\right)(3T_{SF})\\&+P_{C1}P_{C2}\left(\frac{1}{3}P_{S3}\right)(4T_{SF})\\&+P_{C1}P_{C2}\left(\frac{1}{3}P_{S3}\right)(5T_{SF})+\cdots+T_c\,(T_m+1)\,T_{SF}\\&+\cdots+T_c\,(T_m+m)\,T_{SF}\\&+T_c\,(P_{Cm})^{R-m}\,(T_m+(1+(R-m)\,m))\,T_{SF}\\&+T_c\,(P_{Cm})^{R-m}\,(T_m+(m+(R-m)\,m))\,T_{SF}\\&+T_c\left((P_{Cm})^{R-m+1}\right)(T_m+(1+(R-m)\,m))\,T_{SF}\\&+\cdots+T_c\left((P_{Cm})^{R-m+1}\right)\\&\cdot(T_m+(m+(R-m)\,m))\,T_{SF}.\end{aligned} \tag{5}$$

4.3. Energy Consumption. Energy consumption is one of the most important performance metrics in LECIM. Using symbols P_I, P_{RX}, P_{TX}, and P_{SLEEP} for power consumption in idle, receive, transmit, and sleep modes, respectively, while δ for slot length and T_{SF} for the super frame length, we can derive the formula for energy consumption as follows:

$$\begin{aligned}E_{AVG}=&\,P_{S1}\{(T_{SF}-3\delta)*P_I+(2\delta)*P_{RX}+\delta*P_{TX}\}\\&+P_{C1}\left(\frac{1}{2}P_{S2}\right)\{(2T_{SF}-6\delta)*P_I+(4\delta)*P_{RX}\\&\qquad+2\delta*P_{TX}\}+P_{C1}\left(\frac{1}{2}P_{S2}\right)\\&\cdot\{(3T_{SF}-6\delta)*P_I+(4\delta)*P_{RX}+2\delta*P_{TX}\}\\&+\cdots+T_c\{((T_m+1)\,T_{SF}-3m*\delta)\,P_I+T_n\}\end{aligned}$$

$$
\begin{aligned}
&+\cdots+T_c\left\{\left(\left(T_m+m\right)T_{\mathrm{SF}}-3m*\delta\right)*P_{\mathrm{I}}+T_n\right\}\\
&+\cdots+T_c\left(P_{Cm}\right)^{R-m}\\
&\cdot\left\{\left(\left(T_m+\left(1+\left(R-m\right)m\right)\right)T_{\mathrm{SF}}\right.\right.\\
&\qquad\left.\left.-3R*\delta\right)P_{\mathrm{I}}+T_n\right\}+\cdots+T_c\left(P_{Cm}\right)^{R-m}\\
&\cdot\left\{\left(\left(T_m+\left(m+\left(R-m\right)m\right)\right)T_{\mathrm{SF}}\right.\right.\\
&\qquad\left.\left.-3R*\delta\right)P_{\mathrm{I}}+T_n\right\}+T_c\left(P_{Cm}\right)^{R-m+1}\\
&\cdot\left\{\left(\left(T_m+\left(1+\left(R-m\right)m\right)\right)T_{\mathrm{SF}}-3R*\delta\right)P_{\mathrm{I}}+T_n\right\}\\
&+T_c\left(P_{Cm}\right)^{R-m+1}\\
&\cdot\left\{\left(\left(T_m+\left(m+\left(R-m\right)m\right)\right)T_{\mathrm{SF}}-3R*\delta\right)P_{\mathrm{I}}+T_n\right\}.
\end{aligned}
\tag{6}
$$

We know that duration of beacon packet, acknowledgement (ACK) packet, and data packet is equal to one slot time, and therefore in (6) we use δ instead of using the individual packet names explicitly.

5. Performance Analysis under Nonsaturated Load

Nonsaturation mode means that endpoints sometimes have no packets to transmit. Nonsaturated mode can be classified into two cases. Case I refers to the situation where a new packet is not generated when the previous packet is in service. Case II is that packets arrive to (e.g., according to a Poisson process) and queue in a buffer at the endpoints even during the service of the preceding packet. We consider here Case II of nonsaturated mode, because in LECIM the chances of packets arrival during the processing of the existing packets are high. For analysis, we assume the tagged endpoint as $M/G/1$ queue with exceptional first service D_0 and ordinary service time D which can be represented in the busy period by the formula $D = D_0 - Y$ (see Figure 4).

In Figure 4, we see the packet that arrives to an empty queue has to wait till the beacon (this waiting time is denoted by Y) and after listening to the beacon the transmission process will start, while the subsequent packets (i.e., those that arrive during the service of the first packet) start their transmission process just after the transmission of the preceding packet. They need not to wait for Y.

Thus, $E[D]$ can be written as

$$
E\left[D\right]=E\left[D_0\right]-E\left[Y\right].
\tag{7}
$$

The HoL-delay, D_0, is made up of waiting time until the first beacon followed by possibly zero or more colliding transmission attempts until success or packet drop.

The expression for D_0 can be written as

$$
D_0=Y+\begin{cases}
T_{\mathrm{SF}} & \text{with prob } P_{\mathrm{S1}}\\
\begin{cases}2T_{\mathrm{SF}}\\ 3T_{\mathrm{SF}}\end{cases} & \text{with prob } P_{\mathrm{C1}}*\left(\frac{1}{2}P_{\mathrm{S2}}\right)\\
\begin{cases}4T_{\mathrm{SF}}\\ 5T_{\mathrm{SF}}\\ 6T_{\mathrm{SF}}\end{cases} & \text{with prob } P_{\mathrm{C1}}P_{\mathrm{C2}}*\left(\frac{1}{3}P_{\mathrm{S3}}\right)\\
\vdots & \\
\begin{cases}\left(T_m+1\right)T_{\mathrm{SF}}\\ \left(T_m+2\right)T_{\mathrm{SF}}\\ \vdots\\ \left(T_m+m\right)T_{\mathrm{SF}}\end{cases} & \text{prob } P_{\mathrm{C1}}P_{\mathrm{C2}}\cdots P_{\mathrm{C}(m-1)}*\left(\frac{1}{m}P_{\mathrm{S}m}\right)\\
\vdots & \\
 & \text{prob } P_{\mathrm{C1}}\cdots {P_{cm}}^{R-m}*\left(\frac{1}{m}P_{\mathrm{S}m}\right)\\
 & \text{prob } \frac{1}{m}P_{\mathrm{C1}}\cdots {P_{cm}}^{R-m+1},
\end{cases}
\tag{8}
$$

where Y is uniformly distributed over $[1, M]$.

The $E[D_0]$ can be expressed as

$$
\begin{aligned}
E\left[D_0\right]=&\frac{T_{\mathrm{SF}}}{2}+P_{\mathrm{S1}}T_{\mathrm{SF}}+P_{\mathrm{C1}}\left(\frac{1}{2}P_{\mathrm{S2}}\right)\left(2T_{\mathrm{SF}}\right)\\
&+P_{\mathrm{C1}}\left(\frac{1}{2}P_{\mathrm{S2}}\right)\left(3T_{\mathrm{SF}}\right)+P_{\mathrm{C1}}P_{\mathrm{C2}}\left(\frac{1}{3}P_{\mathrm{S3}}\right)\left(4T_{\mathrm{SF}}\right)\\
&+P_{\mathrm{C1}}P_{\mathrm{C2}}\left(\frac{1}{3}P_{\mathrm{S3}}\right)\left(5T_{\mathrm{SF}}\right)\\
&+\cdots+T_c\left(T_c+1\right)T_{\mathrm{SF}}+\cdots+T_c\left(T_m+m\right)T_{\mathrm{SF}}\\
&+T_c\left(P_{Cm}\right)^{R-m}\left(T_m+\left(1+\left(R-m\right)m\right)\right)T_{\mathrm{SF}}\\
&+\cdots+T_c\left(P_{Cm}\right)^{R-m}\left(T_m+\left(m+\left(R-m\right)m\right)\right)T_{\mathrm{SF}}\\
&+\frac{1}{m}\left(P_{\mathrm{C1}}P_{\mathrm{C2}}\cdots\left(P_{Cm}\right)^{R-m+1}\right)\\
&\cdot\left(T_c+\left(1+\left(R-m\right)m\right)\right)T_{\mathrm{SF}}\\
&+\cdots+\frac{1}{m}\left(P_{\mathrm{C1}}P_{\mathrm{C2}}\cdots\left(P_{Cm}\right)^{R-m+1}\right)\\
&\cdot\left(T_c+\left(m+\left(R-m\right)m\right)\right)T_{\mathrm{SF}}.
\end{aligned}
\tag{9}
$$

It is known [14, Equation (5.153)] that mean busy period (BP) for $M/G/1$ queue with exceptional service time is BP $= E[D_0]/(1-\lambda E[D])$. Thus, by renewal theory, ρ can be written as

$$
\rho=\frac{\mathrm{BP}}{1/\lambda+\mathrm{BP}}.
\tag{10}
$$

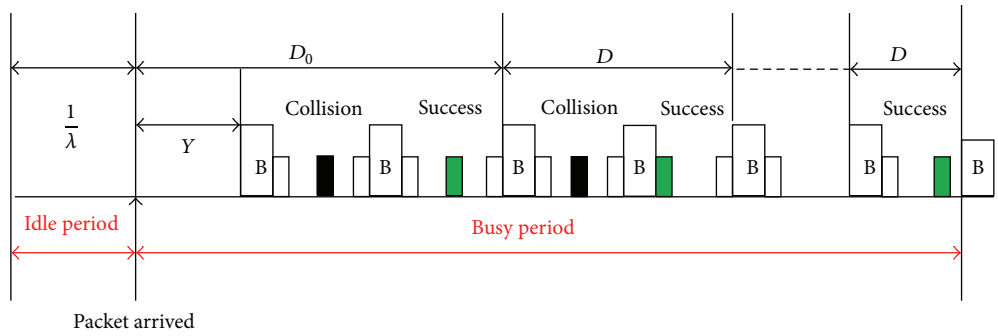


FIGURE 4: Waiting time using exception first service time.

Using the law of total probability, the probability of success P_{S1} at the first transmission attempt can be calculated as (see Figure 5)

$$\begin{aligned}
P_{S1} &= \sum_{k=0}^{N-1} \left(\left(1 - \frac{1}{M}\right)^k P(X = k) \right) \\
&= \sum_{k=0}^{N-1} \left(1 - \frac{1}{M}\right)^k \binom{N-1}{k} \rho^k (1-\rho)^{N-1-k}, \\
&\qquad \text{since } X \sim B(N-1, \rho) \\
&= \binom{N-1}{0} \left(1 - \frac{1}{M}\right)^0 \rho^0 (1-\rho)^{(N-1)-0} \\
&\quad + \cdots + \binom{N-1}{N-1} \left(1 - \frac{1}{M}\right)^{(N-1)} \\
&\quad \cdot \rho^{(N-1)} (1-\rho)^{(N-1)-(N-1)} \\
&= \binom{N-1}{0} \left(\left(1 - \frac{1}{M}\right) \rho \right)^0 (1-\rho)^{(N-1)-0} \\
&\quad + \cdots + \binom{N-1}{N-1} \left(\left(1 - \frac{1}{M}\right) \rho \right)^{N-1} (1-\rho)^0 .
\end{aligned} \tag{11}$$

After simplification, we get

$$P_{S1} = \left[\left(1 - \frac{1}{M}\right) \rho + (1-\rho) \right]^{N-1}. \tag{12}$$

Thus, P_{C1} can be written as $P_{C1} = 1 - P_{S1}$.

Similarly, we can derive the formulas for $P_{S2}, \ldots, P_{Sm}$ as

$$\begin{aligned}
P_{S2} &= \left[\left(1 - \frac{1}{2M}\right) \rho + (1-\rho) \right]^{N-1} \cdots \\
P_{Sm} &= \left[\left(1 - \frac{1}{mM}\right) \rho + (1-\rho) \right]^{N-1}.
\end{aligned} \tag{13}$$

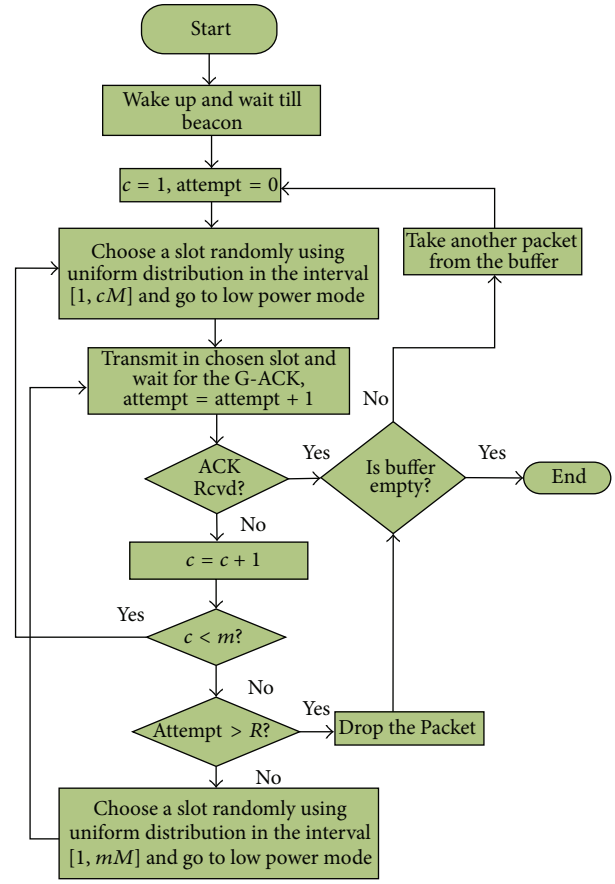


FIGURE 5: Flowchart depicting the transmission in nonsaturated case.

The success probability P_S, collision probability P_C, and packet drop probability P_{drop} of the tagged endpoint in this case can be represented by (14), (15), and (16), respectively:

$$P_S = P_{S1} + P_{C1}P_{S2} + \cdots + P_{C1}P_{C2} \cdots (P_{Cm})^{R-m} P_{Sm} \tag{14}$$

$$P_C = 1 - P_S \tag{15}$$

$$P_{\text{drop}} = P_{C1}P_{C2} \cdots P_{C(m-1)} (P_{Cm})^{R-m+1}. \tag{16}$$

Solving (7), (9), and (10) simultaneously, we can obtain the value of $E[D]$ and can then find ρ, P_{drop} along with other performance metrics.

5.1. PGF of D_0 and D. The tagged packet is transmitted successfully with probability $1 - P_{\text{drop}}$ or is dropped with probability P_{drop}. Let D_1 be the HoL-delay of packet being transmitted successfully and let D_2 be the HoL-delay of packet being dropped after the maximum retransmission limit.

The PGF $E[z^{D_0}]$ of HoL-delay D_0 can be expressed as

$$D_0[z] = E\left[z^{D_0}\right] = E\left[z^{D_1}\right] + E\left[z^{D_2}\right] \tag{17}$$

The $E[z^{D_1}]$ and $E[z^{D_2}]$ can be calculated as

$$\begin{aligned}
E\left(z^{D_1}\right) = {} & \left[\left(\frac{1}{M} * \frac{z\left(1-z^{M-1}\right)}{1-z}\right)\right. \\
& * \left\{P_{S1}z^{M} + P_{C1}\left\{\left(\frac{1}{2}P_{S2}\right)\left(z^{2M}\right)\right.\right. \\
& \qquad \left.\left. + \left(\frac{1}{2}P_{S2}\right)\left(z^{3M}\right)\right\}\right. \\
& + P_{C1}P_{C2}\left\{\left(\frac{1}{3}P_{S3}\right)\left(z^{4M}\right) + \left(\frac{1}{3}P_{S3}\right)\left(z^{5M}\right)\right. \\
& \qquad \left. + \left(\frac{1}{3}P_{S3}\right)\left(z^{6M}\right)\right\} \\
& + \cdots + T_c\left(P_{Cm}\right)^{R-m} \\
& \cdot \left\{\left(z^{(T_m+(1+(R-m)m))M}\right)\right. \\
& \qquad \left.\left. + \cdots + \left(z^{(T_m+(m+(R-m)m))M}\right)\right\}\right\}\Bigg] \\
E\left(z^{D_2}\right) = {} & \left[E\left[z^{Y}\right]\left(P_{C1}\cdots\left(P_{Cm}\right)^{R-m+1}\right)\right. \\
& \left. \cdot\, z^{(1+\cdots(T_m+(1+(R-m)m)))M}\right],
\end{aligned} \tag{18}$$

where $E[z^Y] = ((1/M) * (z(1 - z^{M-1})/(1 - z)))$.

As $D = D_0 - Y$, therefore, by excluding $E[z^Y] = ((1/M) * (z(1 - z^{M-1})/(1 - z)))$ from (18), we can find $D[z] = E[z^D]$.

The mean HoL-delay of packet $E[L]$ can be written as

$$E[L] = (1-\rho)D_0'[1] + \rho D'[1]. \tag{19}$$

5.2. Energy Consumption. We use (20) to find the energy consumption of the tagged endpoint using our enhanced collision resolution scheme:

$$E_{\text{AVG}} = \frac{1 \cdot E_{D_0} + \left(1/(1-\lambda E[D_0]) - 1\right)E_D + E_{\text{SLEEP}}}{E[D_0]/(1-\lambda \cdot E[D]) + 1/\lambda}. \tag{20}$$

Here $E[D]$, $E[D_0]$, E_D, E_{SLEEP}, and E_{D_0} represent mean ordinary service time, mean exceptional first service time, energy consumption during ordinary service time, energy consumption during sleep time, and energy consumption during exceptional first service time, respectively. Their values can be computed as

$$\begin{aligned}
E_D &= E_{D_0} - P_{\text{RX}} * \frac{T_{\text{SF}}}{2}, \\
E[D] &= \mathrm{D}'[1], \qquad E[D_0] = D_0'[1], \\
E_{\text{SLEEP}} &= \frac{1}{\lambda} * P_{\text{SLEEP}}.
\end{aligned} \tag{21}$$

To derive the formula for E_{D_0}, let us assume that

$$\begin{aligned}
E_{D_0} = {} & P_{\text{RX}} * \frac{T_{\text{SF}}}{2} \\
& + P_{S1}\left\{\left(T_{\text{SF}} - 3\delta\right) * P_{\text{I}} + (2\delta) * P_{\text{RX}} + \delta * P_{\text{TX}}\right\} \\
& + P_{C1}\left(\frac{1}{2}P_{S2}\right)\left\{\left(2T_{\text{SF}} - 6\delta\right) * P_{\text{I}}\right. \\
& \qquad \left. + (4\delta) * P_{\text{RX}} + 2\delta * P_{\text{TX}}\right\} \\
& + P_{C1}\left(\frac{1}{2}P_{S2}\right)\left\{\left(3T_{\text{SF}} - 6\delta\right) * P_{\text{I}}\right. \\
& \qquad \left. + (4\delta) * P_{\text{RX}} + 2\delta * P_{\text{TX}}\right\} \\
& + \cdots + T_c\left\{\left(\left(T_m + 1\right)T_{\text{SF}} - 3m * \delta\right)P_{\text{I}}\right. \\
& \qquad \left. + \left((3m - m)\delta\right)P_{\text{RX}} + (m * \delta)P_{\text{TX}}\right\} \\
& + T_c\left(P_{Cm}\right)^{R-m}\left\{\left(\left(T_m + (m + (R-m)m)\right)T_{\text{SF}}\right.\right. \\
& \qquad \left.\left. - 3R * \delta\right) * P_{\text{I}} + T_n\right\} \\
& + T_c\left(P_{Cm}\right)^{R-m+1} \\
& \cdot \left\{\left(\left(T_m + (1 + (R-m)m)\right)T_{\text{SF}} - 3R * \delta\right)P_{\text{I}} + T_n\right\} \\
& + \cdots + T_c\left(P_{Cm}\right)^{R-m+1} \\
& \cdot \left\{\left(\left(T_m + (m + (R-m)m)\right)T_{\text{SF}} - 3R * \delta\right)P_{\text{I}} + T_n\right\}.
\end{aligned} \tag{22}$$

6. Results and Discussions

In this section, we present analytical results of both the saturated and nonsaturated traffic scenarios to evaluate the performance metrics. We set the length of a super frame to be 64, 128, and 256 slots long. The data rate is 40 Kbps. We assume a fixed size packet of length 80 bytes. The power consumption parameters are [15]

$$\begin{aligned}
P_{\text{I}} &= 0.00005673\ \text{mJ/ms}, \\
P_{\text{SLEEP}} &= 0.00000016\ \text{mJ/ms}, \\
P_{\text{RX}} &= 0.0113472\ \text{mJ/ms}, \\
P_{\text{TX}} &= 0.0100224\ \text{mJ/ms}.
\end{aligned} \tag{23}$$

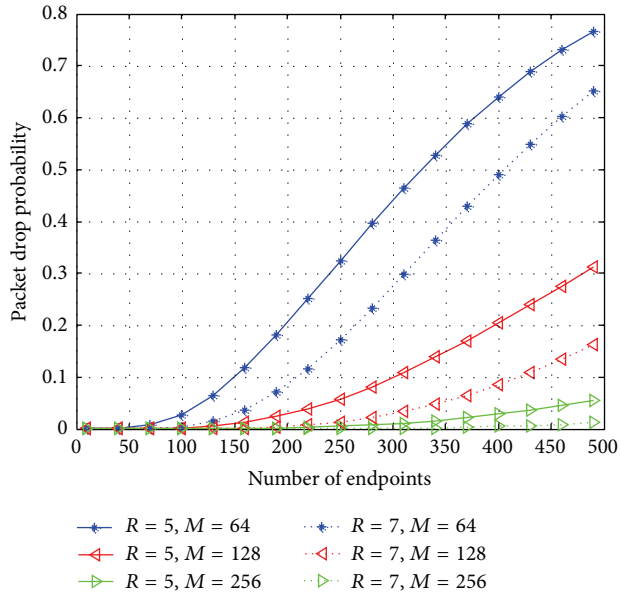


Figure 6: Packet drop probability.

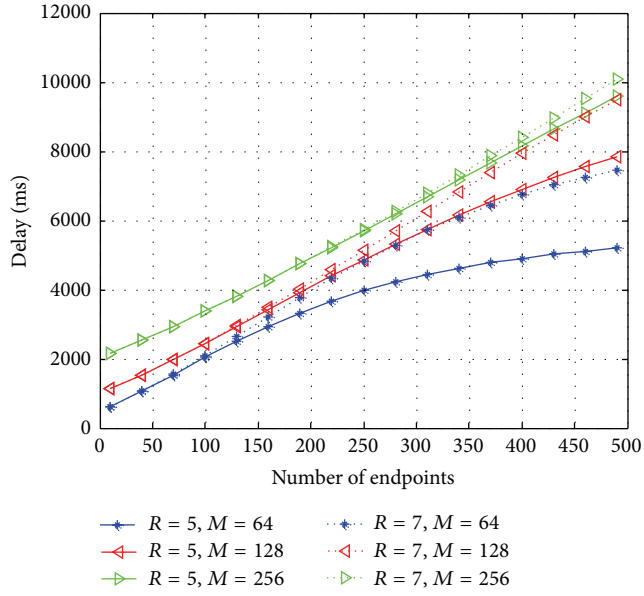


Figure 7: Head-of-line delay of packet.

Figure 6 shows the packet drop probability in saturated case for different values of R and M. In the figure, we see rapid increase in the packet drop probability as the number of endpoints increases for small values of R and M (e.g., $R = 5$ and $M = 64$). It is due to the fact that as the number of endpoints increases, the collision probability for small M and R values also increases quickly.

In the case of large super frame size and R values, the packet drop probability is quite low (e.g., for $R = 7$ and $M = 128$, the packet drop probability is less than 18% even for large value of N; i.e., $N = 480$).

Figure 7 depicts the HoL-delay of packet. Unlike packet drop probability, the HoL-delay increases with the increase of R and M values. As the number of endpoints increases, collision among the packets also increases which causes the endpoints to increase the length of the contention window. The large contention window size causes more HoL-delay of packet and small packet drop probability. For example, we see in Figures 6 and 7, for small values of R and M, the delay is low but the packet drop probability is high, while for large values of the same parameters the delay is high and the packet drop probability is low.

Figure 8 depicts the average energy consumption (mJ). We see in the figure that, due to the high ratio of collisions in a super frame with small values of M than large values of M, more transmission attempts are needed, which results in more energy consumption. We see less amount of energy consumption for the longer super frame sizes (e.g., $M = 256$) at the cost of some delay. In order to keep the delay as well as energy consumption in balance, we propose to use the value of $R = 7$ and $M = 128$. Optimal values of parameters can be chosen depending on the needs of each application. For example, given the number of endpoints equal to 400, $P_{\text{drop}} \leq 20\%$, and energy consumption (mJ) $\leq 3 \times 10^{-4}$, the optimal size of NAP is 128.

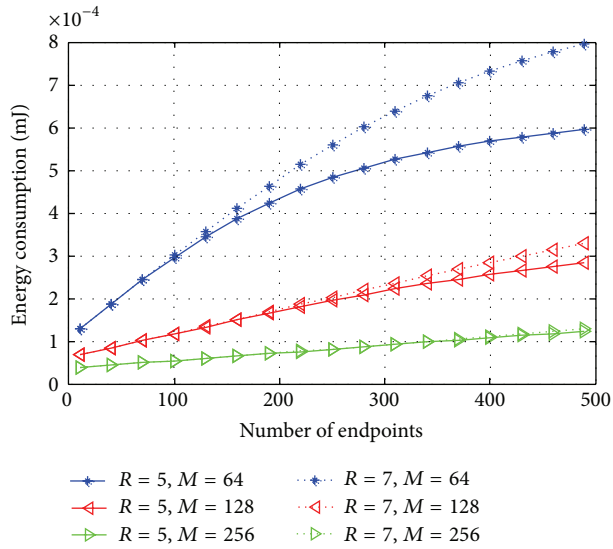


Figure 8: Average energy consumption.

Figures 9 and 10 depict the performance comparison in terms of packet drop probability and HoL-delay of the enhanced collision resolution scheme against the basic scheme under saturated traffic conditions. According to our intuition, we see the reduction in the packet drop probability in the enhanced approach as compared to the basic approach. As in the enhanced approach, the endpoints increase their contention window after collision; therefore, the delay increases and the packet drop probability decreases.

Figure 11 shows the average energy consumption against the number of endpoints in the basic as well as in the enhanced methods for different values M and a fixed value of R which is equal to 7. In the figure, we see that, for the small number of endpoints, the energy consumption

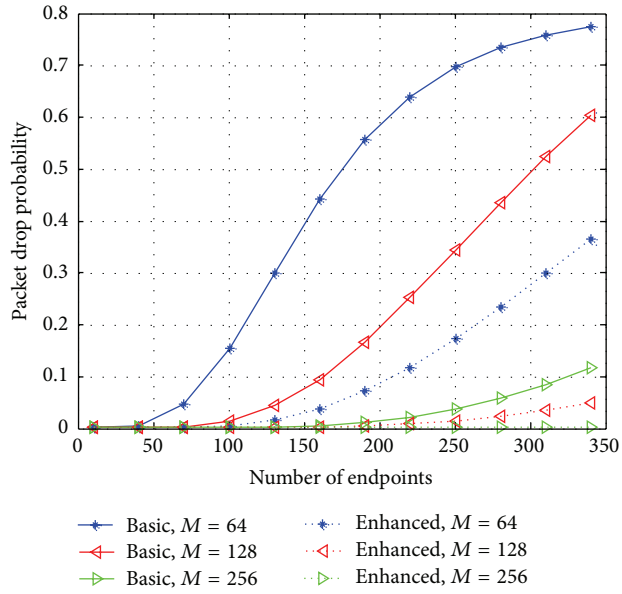


FIGURE 9: Packet drop probability comparison between the basic and enhanced approaches.

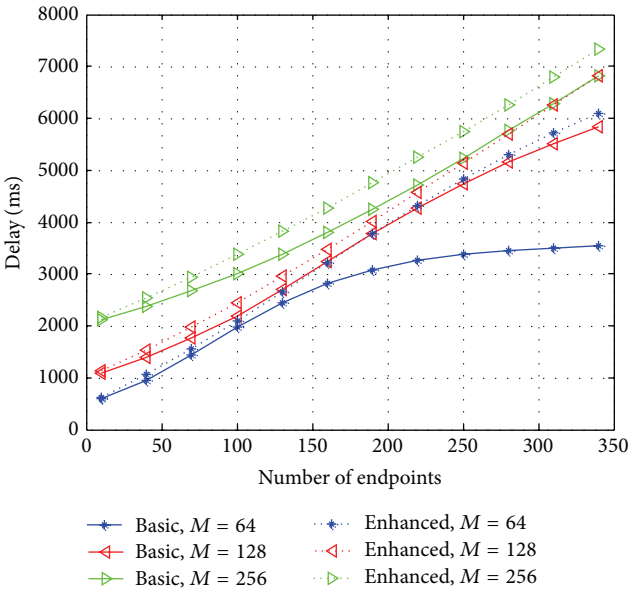


FIGURE 10: Delay comparison of the basic and enhanced approach.

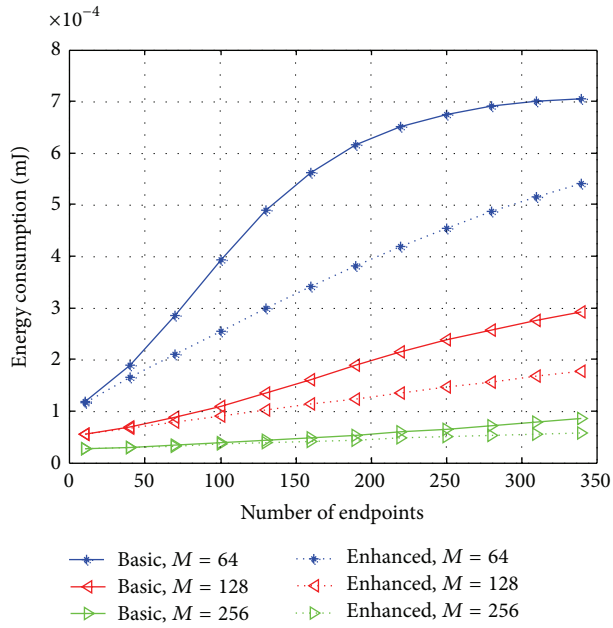


FIGURE 11: Energy consumption comparison under saturated traffic conditions between the basic and enhanced approaches.

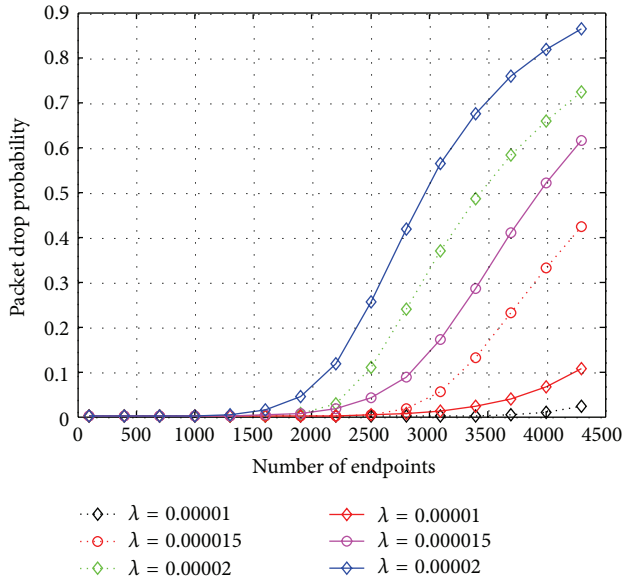


FIGURE 12: Packet drop probability under nonsaturated case. Dotted curves represent enhanced approach; solid curves represent basic approach.

of both methods is almost same and when the number of endpoints increases, the enhanced method dominates the basic method in terms of energy efficiency. The decrease in energy consumption of the enhanced method is due to the reduction of packet collision probability.

Figures 12 and 13 show the comparison of packet drop probability and HoL-delay of packet for different arrival rates under nonsaturated traffic condition. We see that, for small number of endpoints and low arrival rates, the packet drop probability and HoL-delay in both methods is very close to each other. For large number of endpoints, the enhanced approach performs well in terms of packet drop probability as compared to the basic approach. Although the delay in the enhanced approach is more than the basic approach, this small increase in delay has fewer effects on the overall performance of the protocol.

Figure 14 displays the comparison of both methods in terms of average energy consumption for different values of λ. The gap between the energy consumption curves for the same arrival rates of both the methods increases slowly as the number of endpoints increases. In the case of low

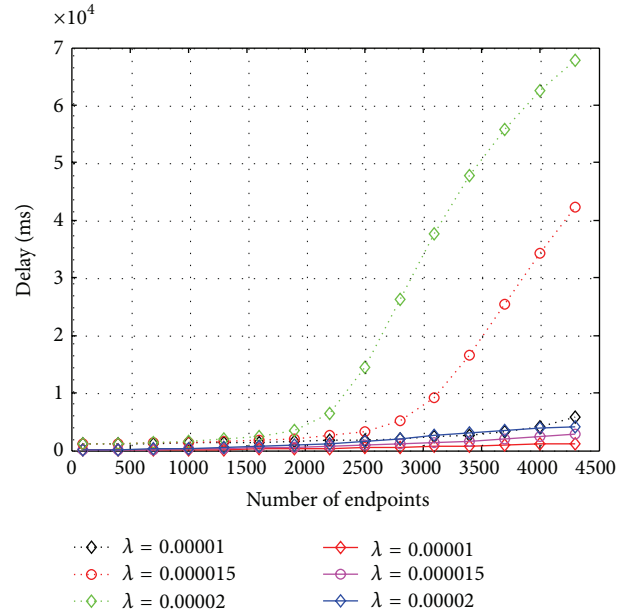


Figure 13: HoL-delay under nonsaturated case. Dotted curves represent enhanced approach; solid curves represent basic approach.

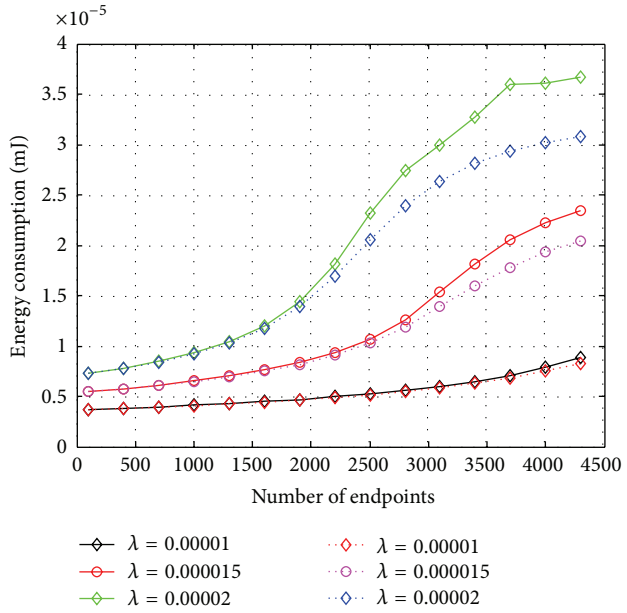


Figure 14: Comparison of energy consumption between the basic and enhanced collision resolution schemes. Dotted curves represent enhanced approach; solid curves represent basic approach.

arrival rates, due to fewer collisions, the gap between the curves is very small (e.g., for $\lambda = 0.00001$). We see large gap between the curves for $\lambda = 0.00002$ when the number of endpoints increases. The more energy consumption in the basic approach happens due to the large number of collisions and retransmission attempts compared to the enhanced approach. The energy consumption converges to a fixed value as the number of endpoints tends to infinity.

7. Conclusion

In this paper, we presented our proposed enhanced collision resolution scheme for uplink communication in LECIM networks. We investigated packet drop probability, HoL-delay, and average energy consumption under both saturated (endpoints always have data to transmit to the coordinator) and nonsaturated (endpoints sometimes have no data to send) traffic conditions. We considered Case II (packets arrive to and queue in the buffer) of nonsaturation traffic scenario.

We compared our enhanced collision resolution approach with the basic approach and found that the new approach performs better than the basic approach in terms of packet drop probability and energy consumption. Optimal values of parameters can be chosen depending on the needs of each application. For example, given the number of endpoints equal to 400, $P_{\text{drop}} \leq 20\%$, and energy consumption (mJ) $\leq 3 \times 10^{-4}$, the optimal size of NAP is 128.

Conflict of Interests

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

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MEST) (no. 2010-0018116).

References

[1] N. Ullah, M. S. Chowdhury, P. Khan, and K. S. Kwak, "Multi-hop medium access control protocol for low energy critical infrastructure monitoring networks using wake-up radio," *International Journal of Communication Systems*, 2012.

[2] J. Lopez, J. A. Montenegro, and R. Roman, "Service-oriented security architecture for CII based on sensor networks," in *Proceedings of the 2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU '06)*, pp. 1–6, Lyon, France, June 2006.

[3] R. Roman, C. Alcaraz, and J. Lopez, "The role of wireless sensor networks in the area of critical information infrastructure protection," *Information Security Technical Report*, vol. 12, no. 1, pp. 24–31, 2007.

[4] M. Albano, S. Chessa, and R. di Pietro, "A model with applications for data survivability in critical infrastructures," *International Journal of Information Assurance and Security*, vol. 4, pp. 629–639, 2009.

[5] K. M. Martin and M. B. Paterson, "Ultra-lightweight key pre-distribution in wireless sensor networks for monitoring linear infrastructure," in *Proceedings of the 3rd IFIP WG 11.2 International Workshop on Information Security Theory and Practice (WISTP '09)*, vol. 5746 of *Lecture Notes in Computer Science*, pp. 143–152, 2009.

[6] N. Ullah, M. S. Chowdhury, M. Al Ameen, and K. S. Kwak, "Energy efficient MAC protocol for low-energy critical infrastructure monitoring networks using wakeup radio," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 504946, 15 pages, 2012.

[7] H. Liu, G. Yao, and P. Xu, "Energy efficient MAC protocol for low energy critical infrastructure networks," IEEE 802.15-11-0619-01-004 k, IEEE Standards Association, 2011, https://mentor.ieee.org/802.15/documents.

[8] M. S. Chowdhury, N. Ullah, M. A. Ameen, and K. S. Kwak, "Framed slotted aloha based MAC protocol for low energy critical infrastructure monitoring networks," *International Journal of Communication Systems*, 2012.

[9] https://mentor.ieee.org/802.15/dcn/10/15-10-0053-00-wng0-low-energy-critical-infrastructure-monitoring.pptx.

[10] https://mentor.ieee.org/802.15/dcn/11/15-11-0147-02-004k-lecim-call-for-proposals.docx.

[11] J. Schwoerer and N. Dejean, "Elster & France Telecom proposal, doc.: IEEE 802. 15-11-0479-01-004 k," 2011, https://mentor.ieee.org/802.15/documents.

[12] Y. Yang, "Low energy MAC proposal for TG4k_WSNIRI, doc.: IEEE 802. 15-11-0596-00-004 k," 2011, https://mentor.ieee.org/802.15/documents.

[13] S. S. Joo, J. A. Jun, and C. S. Pyo, "MAC proposal for low-energy wide area monitoring, doc.: IEEE 802. 15-11-0597-01-004 k," 2011, https://mentor.ieee.org/802.15/documents.

[14] L. Kleinrock, *Queuing Systems, Vol. I: Theory*, Wiley-Inter science, New York, NY, USA, 1975.

[15] T. O. Kim, J. S. Park, H. J. Chong, K. J. Kim, and B. D. Choi, "Performance analysis of IEEE 802.15.4 non-beacon mode with the unslotted CSMA/CA," *IEEE Communications Letters*, vol. 12, no. 4, pp. 238–240, 2008.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 631092, 13 pages
http://dx.doi.org/10.1155/2015/631092

Research Article

An Adaptive Routing Protocol Based on QoS and Vehicular Density in Urban VANETs

Yongmei Sun, Shuyun Luo, Qijin Dai, and Yuefeng Ji

State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Yongmei Sun; ymsun@bupt.edu.cn

Received 8 December 2014; Accepted 3 March 2015

Academic Editor: Xiaohong Jiang

Multihop data delivery between vehicles is an important technique to support the implementation of vehicular ad hoc networks (VANETs). However, many inherent characteristics of VANETs (e.g., dynamic network topology) bring great challenges to the data delivery. In particular, dynamic topology and intermittent connectivity make it difficult to design an efficient and stable geographic routing protocol for different applications of VANETs. To solve this problem, the paper proposes an adaptive routing protocol based on QoS and vehicular density (ARP-QD) in urban VANETs environments. The basic idea is to find the best path for end-to-end data delivery, which can satisfy diverse QoS requirements by considering hop count and link duration simultaneously. To reduce the network overhead furthermore, ARP-QD adopts an adaptive neighbor discovery algorithm to obtain neighbors' information based on local vehicular density. In addition, a recovery strategy with carry-and-forward is utilized when the routing path is disrupted. Numerical simulations show that the proposed ARP-QD has higher delivery ratio than two prominent routing protocols in VANETs, without giving large compromise on delivery delay. The adaptivity of ARP-QD is also analyzed.

1. Introduction

With the development of wireless technologies and dedicated short-range communication technologies, vehicular ad hoc networks (VANETs) have been paid increasing attention [1]. In vehicular settings, the availability of navigation system, global positioning system (GPS), and other sensors that can perceive the vehicle speed, location, and other useful information makes it possible to exploit many applications, such as intelligent transportation system (ITS) applications and infotainment applications [2, 3]. ITS applications include cooperative traffic monitoring, traffic control, blind crossing, collision prevention, nearby information services, and real-time detour route computation [4], which have attracted attention from many car manufacturers, research institutes, and national transportation departments. Vehicle communications [5, 6] are the basic foundation of the above applications of VANETs.

Unfortunately, the traditional wireless technologies cannot be applied for VANETs directly, since they have some inherent characteristics, such as dynamic radio environments and frequent topology changes, which cause the network disconnection from time to time. Due to high speeds of vehicular movements, link duration between two vehicles is hard to keep stable for a period of time. As communication relays or information broadcasters, the equipment of road-side-units (RSUs) can help improve the vehicle communications. However, the RSUs usually have high costs. Therefore, the dynamic network topology is the most critical issue in VANETs. In particular, it brings significant challenges for designing an efficient and stable geographic routing protocol.

The existing routing protocols lack the friendly adaptation to diverse QoS requirements of different applications. The objectives of the current routing protocols focus on either the fastest path with the minimum hop count or the most stable path with the longest link duration or connectivity but neglect the adaptive balance of the routing protocol with consideration of path efficiency and path stability. In this paper, we propose an adaptive routing protocol based on QoS and vehicular density (ARP-QD) over urban VANETs. It balances the path efficiency and path stability by an optimal forwarding algorithm and an adaptive neighbor discovery

algorithm with friendly adaptation to different QoS requirements and urban VANETs environments. The main intellectual contributions of this paper are summarized as follows.

(1) For describing the dynamic link quality in VANETs, we define two new metrics, named product of connectivity and distance (CDP) and segment selection weight (SSW), by considering the hop count and link duration simultaneously.

(2) We present an optimal forwarding algorithm based on CDP and SSW, which can obtain a qualified path to satisfy the diverse QoS requirements of different applications by balancing the path efficiency and path stability. As an essential part, a quick recovery strategy with carry-and-forward is also provided when the routing path is disrupted.

(3) To reduce the network overhead and improve resource usage, we propose an adaptive neighbor discovery algorithm to obtain the neighbors' information based on local vehicular density.

(4) The extensive simulation results show that the proposed ARP-QD has higher delivery ratio than two prominent routing protocols in VANETs, without giving large compromise on the delivery delay. The adaptivity of ARP-QD is also analyzed.

The remainder of the paper is organized as follows. Section 2 briefly reviews related routing mechanisms proposed in VANETs and details the motivation of this paper. In Section 3, we design two metrics combining hop count and link duration for forwarding optimization. The adaptive neighbor discovery algorithm is also presented as well as the recovery strategy to improve the robustness of ARP-QD. Numerical simulations and the results are analyzed in Section 4. We conclude the paper and list some possible future works in Section 5.

2. Related Works

Generally, path efficiency and path stability are two important criterions in designing routing protocol for VANETs. To achieve high efficiency, the shortest (generally fastest) path with minimum hop count is usually selected as the best path. To pursue high stability, the path with the longest duration is considered as the best candidate. However, most of existing researches focus on either efficiency or stability. We review related works in both directions as follows.

Path Efficiency. One objective of a routing protocol in VANETs is to find an efficient (or a fast) path with the shortest number of hops for data delivery [5, 7–10]. Greedy Perimeter Stateless Routing (GPSR) algorithm uses the positions of routers and a packet's destination to make packet forwarding decisions [7]. It chooses the nearest node to the destination as the next hop within communication range, which will increase the link loss because of high mobility and radio obstacles. Like GPSR, Geographic Source Routing (GSR) [8] is also a position based routing protocol. The weakness of GSR is not flexible to the sparse network, since GSR works on the foundation of end-to-end connectivity. Another similar method of GPSR is Greedy Perimeter Coordinator Routing (GPCR) [9], which assigns the routing decision to the nodes located at the street intersections and uses the greedy forwarding strategy to route the packet path between the street intersections. However, GPCR does not take the link connectivity into consideration to select the best path. An improved Greedy Traffic Aware Routing Protocol (GyTAR) has been presented in [5], which is based on the geographical intersection information to find robust and optimal routes within urban environments. In [10], a two-stage routing algorithm has been presented to find out the practically fastest route to a destination at a given departure time in terms of taxi drivers' intelligence learned from a large number of historical taxi trajectories.

In short, most of the above researches regard the shortest path, but fail to concern the diverse QoS requirements of different applications. Some applications require more stable path for high delivery ratio, while the link connectivity between the current and farthest neighbor node is always most vulnerable, which may cause shorter link duration than other links. Hence, the above protocols are not suitable for applications which require high delivery ratio.

Path Stability. One of the simple but efficient methods to improve the path stability is to find the next hop with the longest link duration (or the most stable connectivity) [11–15]. A Receive On Most Stable Group-Path (ROMSGP) scheme [11] has been designed to choose the most stable path with the longest link expiration time. However, ROMSGP only broadcasts specific and well-defined packets, which will result in the loss of other packets. The goal of [12, 13] is to find the routing path with the least probability of network disconnection and avoid carry-and-forward delay. However, the links with good connectivity usually have short distance, which makes the selected paths include more hops and therefore brings longer delivery delay. A stable VANETs routing protocol [14] has been proposed to provide fast and reliable message delivery based on the real-time road vehicular density. However, the real-time update of density information incurs a large number of communication overheads, which results in its performance deterioration with the augment of network scale. An intersection-based geographical routing protocol has been proposed in [15], which aims to find the path with high connectivity probability and other QoS constraints.

In a word, all aforementioned researches mainly focus on the link connectivity and make less use of the geographical distance information among vehicles, such that the selected paths may have unnecessary loops, which causes longer delivery delay. Thus, the above protocols are not suitable for the applications which require low delivery delay.

Some researches, like [16, 17], take the link state and hop count into account. In [16], the authors have presented an Optimized Link State Routing (OLSR) algorithm to provide optimal routes. However, the link state is only used to obtain the neighbors' information and OLSR provides the path with minimum hop count as the best path. Moreover, OLSR is a topology-based routing algorithm, which consumes a large amount of topology control messages. To improve GPSR,

[17] uses the vehicle speed and position to find relatively stable links, which is based on the forecast of the speed fluctuations. However, the above works failed to adaptively trade off the path efficiency and path stability for diverse QoS requirements in different scenarios and could not achieve the purpose of friendly communications.

To the best of our knowledge, there is no prior work that has thoroughly researched the adaptive routing protocol which can balance the path efficiency and stability based on diverse QoS requirements of different applications. In this paper, based on the information of intersection location, vehicle speed, and position, we take the hop count and link duration into consideration and propose a novel optimal forwarding algorithm to trade off the path efficiency and stability with friendly adaptation to different QoS requirements of applications. Furthermore, we present an adaptive neighbor discovery algorithm, which exploits different ways to acquire the neighbors' information according to the local vehicular density. Based on the above two main algorithms, we build the adaptive routing protocol based on QoS and vehicular density (ARP-QD), which has higher delivery ratio and reasonable delivery delay.

3. The Proposed Adaptive Routing Protocol (ARP-QD)

In this section, we first introduce the system model used for urban VANETs. Then we present the optimal forwarding algorithm which adaptively balances the path efficiency and stability based on QoS requirements, as well as the adaptive neighbor discovery algorithm based on the real-time vehicular density. To improve the robustness of ARP-QD, the recovery strategy with carry-and-forward is adopted when the routing path is disrupted. Finally, an example is given to illustrate how the proposed ARP-QD works.

3.1. System Model. As shown in Figure 1, we consider a VANET road environment with intersections and segments within two intersections, which is a typical scenario in urban areas. The circle with the intersection ID inside denotes the intersection. $\vec{v}$ and $\vec{p}$ indicate the moving directions of the vehicle and the packet, respectively. The yellow arrow means the moving direction of vehicles on that road segment. The purple arrow with a right angle denotes the candidate path of the packet from the source node S to the destination D. Vehicles move through the segments in the same or opposite direction, while, when moving into the intersection, they will find their neighbors moving in various directions.

Since the RSUs are costly, the paper focuses on the routing protocol for vehicle-to-vehicle (V2V) communications without RSUs. We assume that all vehicles are equipped with onboard navigation system and wireless communication capability as described in [18]. Each vehicle has a digital street map of the area using the onboard navigation system to determine the positions of its neighboring intersections. Meanwhile, it can acquire a landscape of the road environment, including the vehicular velocity and density on each road. The above information can be obtained through the commercialized applications [19]. Furthermore, through the periodic

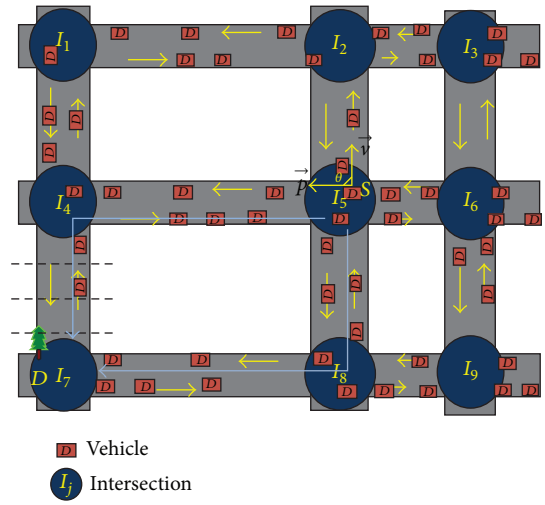


Figure 1: System illustration (S: the source node; D: the destination).

information exchange, each vehicle knows its neighbors' information including the positions and velocities, which is maintained in its neighbor table. For easy illustration, we assume that all vehicles have the same transmission range. In addition, the location service can make the source node have the knowledge of destination position in real time. The above assumptions are the same as the previous works [4, 20, 21].

3.2. Optimal Forwarding Algorithm. As mentioned above, the real road environment contains two parts: intersections and segments within two intersections. Many vehicles, which are regarded as mobile nodes, move along the road as shown in Figure 1. We aim to find the best path hop by hop from the source node S which creates the packets to the destination D. D can be the nearest Internet gateway or data collection center. Thus, we assume the destinations are always located in the intersections. The proposed ARP-QD is a geographic routing protocol including optimal forwarding decision, adaptive neighbor discovery, and robust route recovery. It selects the whole path hop by hop from S to D, and each sender decides its next hop locally. It is easy to observe that a node traveling in the segment or intersection should use different tactics to calculate the metric to choose the next hop. For a node in the segment, it only chooses its next hop in the parallel directions, while for a node in the intersection it should first choose the next segment and then decide the next hop within the selected segment. Therefore, we define a new metric, that is, product of connectivity and distance (CDP), in two cases, respectively.

3.2.1. Metric Design in the Segment Case. Two seemingly contradictory, yet related, objectives of routing performance exist: improving the path efficiency with less hop count and improving the path stability with longer link duration. In general, the longer the link distance is, the smaller the hop count is. In contrast, the shorter the link distance is, the more stable the link is. We aim to design a novel metric for selecting the best next hop on the road segment, which can balance the requirements of the path efficiency and stability. We first

consider the one lane case and later show that the case of multiple lanes has the same result. In the one-lane case, we just consider that all vehicles drive in the same direction, and the result in the opposite direction can be easily induced in the same way. To formally design the metric, that is, CDP, the notations used in the following analysis are described in Notations.

We regard a neighbor node n with $\mathrm{PL}_n < \mathrm{PL}_s$ as a candidate neighbor. Note that the path length means the distance along the selected roads.

First, we discuss the case of one lane. As depicted in Figure 2(a), we can obtain

$$R_n + L_n = R. \tag{1}$$

Since $\mathrm{PL}_n < \mathrm{PL}_s$,

$$R_n = \mathrm{PL}_s - \mathrm{PL}_n > 0. \tag{2}$$

It can be observed that the neighbor node n, which is closest to the destination, has the largest R_n.

We define T_n in (3) to denote the link connection duration time between candidate neighbor node n and the sender s:

$$T_n = \begin{cases} \dfrac{L_n}{(v_n - v_s)}, & v_n > v_s, \\ K, & v_n = v_s, \\ \dfrac{(2R - L_n)}{(v_s - v_n)}, & v_n < v_s, \end{cases} \tag{3}$$

where K is a default constant set by the VANETs system. In order to improve the path efficiency and stability, we prefer to choose the neighbor node with the largest product of T_n and R_n as the next hop. Hence, the basic CDP of a neighbor node n is defined as

$$\mathrm{CDP}_n^b = R_n T_n = \begin{cases} \dfrac{R_n L_n}{(v_n - v_s)}, & v_n > v_s, \\ R_n K, & v_n = v_s, \\ \dfrac{R_n (2R - L_n)}{(v_s - v_n)}, & v_n < v_s. \end{cases} \tag{4}$$

As we can see, the CDP value depends on the relative speed and distance between the sender s and candidate neighbor node n. Indeed, for a given lane with some nodes, the CDP function combines the factors of the distance from n to s and the link connection duration. Since larger R_n means less hop count and larger T_n means longer link duration, larger CDP is preferred. The node with the largest CDP among the candidates is selected to be the next hop. Figure 2(a) shows an example of vehicles driving on one lane. In this scenario, once the sender s obtains the information of neighbors' positions and velocities, it computes the CDP value of each neighboring vehicle. Considering its path length to the destination and the link duration with s, neighboring vehicle 1 (i.e., node 1) is assumed to get the maximum value of CDP. It is then chosen as the next hop. Note that if there are multiple neighbor nodes with the same largest CDP, s will randomly pick up one as the next hop.

Then, we will discuss the case of multiple lanes as depicted in Figure 2(b). The relation is changed as shown in the following:

$$Q_n + L_n = \sqrt{R^2 - (kl)^2}. \tag{5}$$

Here it is assumed that a sender s drives in lane 2 and the candidate neighbor node n drives in lane $k+2$, where k indicates the number of interval lanes. Although transmission range R is more than 100 m, l is usually less than 3 m. We can get $Q_n \approx R_n$ and $\sqrt{R^2 - (kl)^2} \approx R$. Consequently, (5) can be simplified to

$$R_n + L_n \approx R. \tag{6}$$

No matter where the vehicles drive in one lane or multiple lanes, their basic CDP can be calculated by (4).

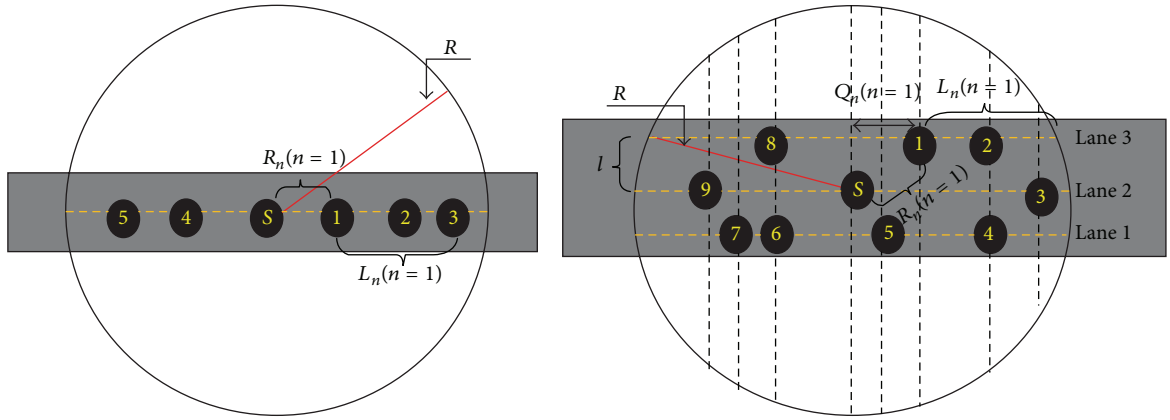
Next, we modify the definition of CDP to satisfy diverse QoS requirements of different applications. In this paper, two prominent QoS requirements, that is, delivery delay and delivery ratio, are considered. For real-time applications such as video on demand, which require high priority on delivery delay, they need to find the efficient path with minimum hop count, while, for other applications such as file transmissions, which require the reliable transmission with high delivery ratio, they need to find the stable path with longest link duration. We use adaptive factors α and β to represent diverse QoS requirements of different applications, where α implies the priority weight of hop count, while β means the importance of link duration under the condition of $\alpha + \beta = 1$. To friendly adapt to diverse QoS requirements of different applications, α and β will be set according to the application requirements on delivery delay and delivery ratio. It is easily obtained that the larger α makes higher priority on delivery delay, which requires finding a path with smaller hop count. Therefore, the advanced CDP is defined as

$$\mathrm{CDP}_n^a = R_n^\alpha T_n^\beta = \begin{cases} R_n^\alpha \cdot \left(\dfrac{L_n}{(v_n - v_s)} \right)^\beta, & v_n > v_s, \\ R_n^\alpha K^\beta, & v_n = v_s, \\ R_n^\alpha \cdot \left(\dfrac{(2R - L_n)}{(v_s - v_n)} \right)^\beta, & v_n < v_s. \end{cases} \tag{7}$$

From (1) and (7), we can obtain an optimal value of CDP_n^a among different neighbors.

ARP-QD will select the one with the maximum CDP_n^a among all candidate neighbor nodes as the best next hop. In conclusion, using the metric CDP_n^a defined in (7), ARP-QD can friendly adapt to diverse QoS requirements when packets are delivered on the segment areas.

3.2.2. Metric Design in the Intersection Case. This part discusses how to design a new metric expression for the best next hop selection in the intersection by taking hop count and link duration into consideration. There are two stages for the sender in the intersection to choose the best next hop. First, the sender needs to choose which segment the packet will be



(a) One lane (b) Multiple lanes

FIGURE 2: Road segment illustration.

delivered. Then, based on the selected segment, the sender computes the best next hop located in that segment.

Segment Selection. Obviously, the segment selection for a sender in the intersection is to find the best next intersection. Candidate intersections are defined as the adjacent intersections whose path lengths are shorter than the current intersection. The mobile vehicles moving along the roads are formalized to form a mobile ad hoc vehicular network. To find an efficient routing path, we prefer to choose the connected one. The reason is that the disconnection brings in the vehicle carrying the packet until it connects to another vehicle, but the vehicle's moving speed is significantly slower than that of wireless communications. Thus, we aim to find the next intersection which is connected to the current intersection through these mobile nodes. We define a binary parameter, named U_j, to indicate the connectivity of intersection j. $U_j = 1$ means that the intersection j is connected with the current intersection. Otherwise, $U_j = 0$. The formal expression can be illustrated as follows:

$$U_j = \begin{cases} 1, & j \text{ can be connected,} \\ 0, & j \text{ cannot be connected.} \end{cases} \tag{8}$$

With the precondition of intersection connectivity, we aim to combine the hop count and link duration time into the metric design. On the one hand, we want to choose the path with the shortest path length, which means minimum hop count. On the other hand, in order to choose the next hop with long link duration, we tend to choose the neighbors in the same moving direction as the sender. Hence we prefer to select the segment with smaller θ, which is the angle between candidate segment and movement direction of the current sender. Based on the above analysis, we define a metric, named segment selection weight (SSW), to select the best next intersection. The SSW of the intersection j is

$$\mathrm{SSW}_j = \alpha \frac{\mathrm{PL}_{sj}}{\mathrm{PL}} + \beta \frac{(1-\cos\theta)}{2} + \left[1 - U_j\right], \tag{9}$$

where PL_{sj} indicates the path length of packet delivery from the sender s to the destination through the intersection j. PL is the summed length of paths through all candidate intersections, formally shown as $\mathrm{PL} = \sum \mathrm{PL}_{sj}$, where j represents the ID of candidate intersections. PL_{sj} is divided by PL for normalization. For the sender in a given intersection, PL is fixed and the path with smaller PL_{sj} is preferred. In order to satisfy diverse QoS requirements of different applications, we also use the adaptive factors α and β to represent the weight of hop count and link duration, respectively, in (9). ARP-QD will select the one with the minimum SSW among all candidate intersections as the best next intersection.

Next Hop Selection. Once the next segment is selected, the direction of packet delivery is determined. In the following we give the process to select the next hop among the selected segments, which can be classified into two cases.

(1) $\theta = 0$: in this case, the sender's moving direction is the same as the next hop's. Hence, we can use the same method to select the best next hop as that used in the segment case.

(2) $\theta \neq 0$: in this case, $R_n + L_n \neq R$. We need to obtain new CDP equations. As shown in Figure 3, we assume that both the sender s and the candidate neighbor node n are moving in constant speed, which are noted as v_s and v_n, respectively. Using the cosine law, we can obtain the equation as follows:

$$R^2 = \left(R_n + v_n T_n\right)^2 + \left(v_s T_n\right)^2 - 2 v_s T_n \left(R_n + v_n T_n\right) \cos\theta. \tag{10}$$

From (10), we can compute T_n as follows:

$$T_n = \frac{R_n\left(v_s \cos\theta - v_n\right)}{v_n^2 + v_s^2 - 2 v_n v_s \cos\theta} + \frac{\sqrt{R^2\left(v_n^2 + v_s^2 - 2 v_n v_s \cos\theta\right) - R_n^2 v_s^2 \sin^2\theta}}{v_n^2 + v_s^2 - 2 v_n v_s \cos\theta}, \tag{11}$$

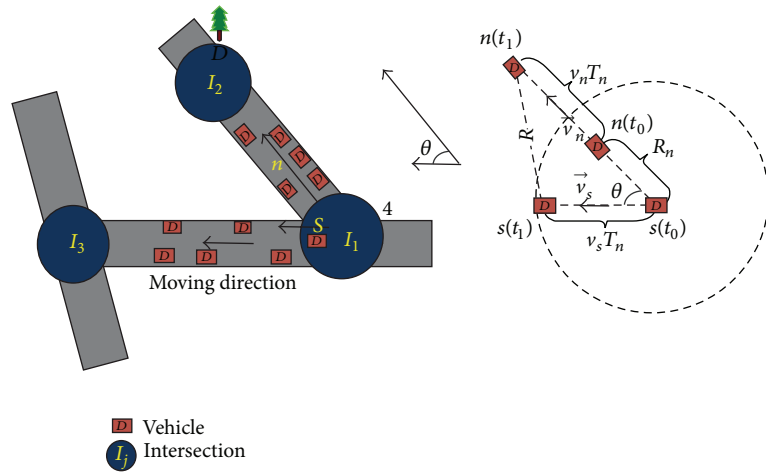


Figure 3: Intersections illustration.

where R_n is the Euclidean distance from the sender s to its candidate neighbor n and θ is the angle between candidate segment and movement direction of the current sender. Hence, the basic CDP is defined as

$$\mathrm{CDP}_n^b = R_n \cdot \left(\frac{R_n (v_s \cos\theta - v_n)}{v_n^2 + v_s^2 - 2v_n v_s \cos\theta} + \frac{\sqrt{R^2 (v_n^2 + v_s^2 - 2v_n v_s \cos\theta) - R_n^2 v_s^2 \sin^2\theta}}{v_n^2 + v_s^2 - 2v_n v_s \cos\theta} \right). \tag{12}$$

Accordingly, the advanced CDP is obtained as

$$\mathrm{CDP}_n^a = R_n^\alpha \cdot \left(\frac{R_n (v_s \cos\theta - v_n)}{v_n^2 + v_s^2 - 2v_n v_s \cos\theta} + \frac{\sqrt{R^2 (v_n^2 + v_s^2 - 2v_n v_s \cos\theta) - R_n^2 v_s^2 \sin^2\theta}}{v_n^2 + v_s^2 - 2v_n v_s \cos\theta} \right)^\beta. \tag{13}$$

The sender s chooses its candidate neighbor with the maximum CDP_n^a in (13) as the best next hop, which is located in the selected segment.

3.2.3. Optimal Forwarding Algorithm. In this part, we present a novel optimal forwarding algorithm, as described in Algorithm 1, to choose the best next hop for multihop packet delivery. The best next hop is selected from the sender's neighbor list, which is obtained by neighbor discovery algorithm (described in Section 3.3). Note that neighbor list contains the information of neighbors' IDs and CDP_n^a values, while neighbor table is composed of neighbors' IDs, velocities, and positions. Each CDP_n^a value in the neighbor list is computed by (7) or (13) using the information in the neighbor table. As mentioned above, there are two cases to analyze the next hop selection. On the one hand, when the sender s is moving along a road segment, it will choose the candidate neighbor with the maximum CDP_n^a value, from its neighbor list, as the best next hop. On the other hand, when s approaches an intersection, it needs to firstly find the best next intersection with the minimum SSW and then choose the best next hop located in the selected segment.

To find the best next intersection (or segment), s needs to get the information of which intersection is connected with the current intersection. Hence, s broadcasts a beacon packet, which contains a connectivity probe request (CP_REQ) and its own information as shown in Figure 4. CP_REQ includes the current intersection ID, source and destination of the data, request time, and expired time. It is used to probe the connectivity of each candidate intersection, which is indicated by U_j. If a candidate intersection j is connected to the current intersection by mobile nodes (i.e., vehicles) moving between the current intersection and candidate intersection j, the sender s will receive a responding packet from its neighbor node before the expired time; then $U_j = 1$; otherwise $U_j = 0$. The responding packet contains a connectivity probe reply (CP_REP) and neighbor's information as shown in Figure 4. CP_REP includes the candidate intersection ID, source and destination of the data, reply time, and expired time. Neighbor's information includes its ID, velocity, and position used for calculation of SSW and CDP_n^a. The beacon will be dropped if the expired time is over. Based on the received responding packets, s calculates values of SSW according to (9) for all candidate intersections and then picks out the candidate intersection with the minimum SSW as the next intersection. Thus, the next delivery segment is selected accordingly. Finally, to find the best next hop, s chooses the candidate neighbor with the maximum CDP_n^a value as the best next hop from its neighbor list.

Input: The information of sender s and destination D
Output: The next hop of the delivered packet
(1) **if** s approaches the intersection **then**
(2) Broadcast a beacon packet with CP_REQ to each candidate intersection and active the Time 1 (expired time).
(3) **repeat**
(4) Receive responding packets with CP_REP and neighbors' information
(5) **until** Timer 1 expires
(6) **if** s receives a responding packet with CP_REP and neighbor's information from intersection j **then**
(7) $U_j = 1$.
(8) **else**
(9) $U_j = 0$
(10) Compute SSW of each candidate intersection.
(11) Select the next intersection with the minimum SSW.
(12) Select the next hop with the maximum CDP_n^a based on (7) or (13).
(13) **else**
(14) Choose the next hop with the maximum CDP_n^a according to (7).

Algorithm 1: The optimal forwarding algorithm.

CP_REQ					Node information		
src	dst	Intersection ID	REQ time	Expired time	ID	Velocity	Position

(a) Beacon packet

CP_REP					Neighbor information		
src	dst	Intersection ID	REP time	Expired time	ID	Velocity	Position

(b) Responding packet

Figure 4: Packet format.

3.3. Adaptive Neighbor Discovery Algorithm. The neighbor list of each node is updated at fixed intervals to keep neighbors' information in real time, which is the precondition of the optimal forwarding algorithm. Vehicular density has a tremendous impact on the network performance, and high density incurs serious congestions during the update process of neighbors' information. In other words, heavy periodic beacons for neighbor discovery will decrease the average throughput of network, which causes negative influence on the end-to-end data delivery. In this section, we aim to design an adaptive neighbor discovery algorithm based on the vehicular density to obtain the neighbor list.

The proposed neighbor discovery algorithm can adaptively reduce the communication overhead according to the local vehicular density, which is defined as the number of nodes in transmission range of node i, denoted as d_l. We set a density threshold d_{th} to evaluate the local vehicular density d_l. The basic principle of the adaptive neighbor discovery algorithm is to choose a centralized way to discover neighbors and update neighbor list when d_l is lower than d_{th}, while using a distributed fashion on the opposite. The detailed process of neighbor discovery is illustrated in Algorithm 2.

In the centralized way, node i first broadcasts a start beacon to request all neighbors' information. Next each neighbor answers to the beacon with the information of its own position and velocity. Based on the neighbors' information of positions and velocities, node i can compute CDP_n^a value of each neighbor n by (7) or (13). Thus the neighbor table and neighbor list of node i are updated. The optimal forwarding algorithm will select the best next hop from this neighbor list, as mentioned in Section 3.2. Since the destination of all neighbors' answers is node i, they adopt distributed coordination function in IEEE 802.11 to avoid the transmission collision. Request to send (RTS) and clear to send (CTS) control frames are used to reserve channel bandwidth and to minimize the amount of wasted bandwidth when collision occurs [22]. Since d_l is lower than d_{th}, such a centralized way for neighbor discovery will not result in heavy communication overheads.

In the distributed fashion, we propose a receiver-based approach for neighbor discovery. Node i broadcasts a start beacon that informs its neighbors about its position and velocity. Each receiver computes its own CDP_n^a value by (7) or (13). In order to reduce the communication overhead, it can only answer to the beacon after a waiting time based on its CDP_n^a value by a uniform rule as defined in the following:

$$T_n = \frac{T^*}{\mathrm{CDP}_n^a}, \tag{14}$$

where T^*, set by the VANETs system, is a time parameter to control the relation between CDP_n^a value and waiting time of receiver n. n means the node which can receive the start beacon from node i, which is node i's neighbor. The waiting time of neighbor n is inverse correlation with the value of CDP_n^a calculated by (7) or (13). It is easily observed that the neighbor with the maximum CDP_n^a has the smallest waiting time; therefore it will answer to node i at the first time. Once node i hears this answer, it will broadcast a stop message to all neighbors to terminate the current neighbor discovery. Thus the neighbor list of node i will have only one node. If node i has not received any answers before the expired time, its neighbor list will be empty at current time. The optimal forwarding algorithm will select the best next hop from this

Input: The local vehicular density d_l of node i, the vehicular density threshold d_{th}
Output: The neighbor list of node i
(1) **if** $d_l < d_{\text{th}}$ **then**
(2) Use the centralized way to obtain the neighbor list of node i based on CDP_n^a.
(3) **else**
(4) Use the distributed way to obtain the neighbor list of node i based on CDP_n^a.

Algorithm 2: The adaptive neighbor discovery algorithm.

neighbor list, as mentioned in Section 3.2. Since d_l is higher than d_{th}, such a distributed way for neighbor discovery will significantly reduce the communication overheads.

This adaptive neighbor discovery algorithm requires each node to previously know the local vehicular density, which is easily to be obtained by the current commercial applications [19], as mentioned before. Intuitively, this adaptive approach will increase the average data delivery ratio by reducing the communication overheads during the neighbor discovery in dense networks, while decreasing the delay by reducing the waiting time in sparse networks.

Remark. The adaptive neighbor discovery algorithm still works when the update of neighbor list is triggered by the forwarding event.

3.4. Routing Path Recovery Strategy. In the dynamic wireless environment, it is inevitable that the routing path fails or breaks. Once a selected link breaks, a local recovery procedure takes place. To improve the robustness of ARP-QD, the adopted recovery strategy is based on the idea of carry-and-forward [23]. The sender which detects the broken link will explore the one-hop neighbors to find a backup link. If the sender has no one-hop neighbor, it will carry the packet until another node moves into its transmission range to transfer the packet. Furthermore, such a carry-and-forward strategy guarantees loop-free routing and avoids endlessly forwarding loop by marking the previous hops.

3.5. A Walk-Through Example. The whole ARP-QD contains the two main novel algorithms proposed above: optimal forwarding algorithm and adaptive neighbor discovery algorithm. In order to improve the robustness of ARP-QD, the carry-and-forward strategy for routing path recovery is also complemented. We use the following example, depicted in Figure 5, to illustrate how ARP-QD works. According to the QoS requirement of certain application, the adaptation factors α and β are set for computation of CDP_n^a and SSW_j. With the help of onboard GPS, navigation system, and digital map, the source node S can obtain the position of destination. The dotted parallel lines denote the transmission range of the source node S. We assume all nodes have the same transmission range.

(1) The source node is in the segment area, and the local vehicular density around S is smaller than the certain density threshold d_{th}. Thus, S exploits the centralized way to discover neighbors and compute the CDP_n^a value of each candidate neighbor. After collecting the CDP_n^a information, S chooses the one with maximum CDP_n^a as the best next hop, which is s_1 in this case. For the same way, s_2 is selected to be the best next hop of s_1.

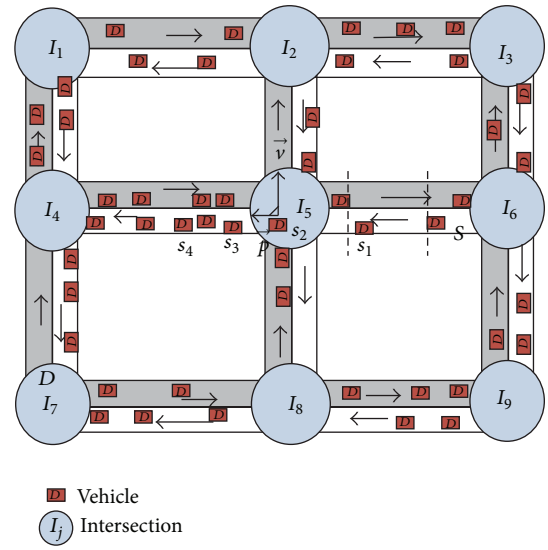


Figure 5: A walk-through example.

(2) The sender s_2 approaches the intersection I_5. First, s_2 needs to choose the best next intersection with the minimum value of SSW. s_2 broadcasts a beacon with CP_REQ to request the information of connectivity from the current intersection to all candidate intersections. It aims to find the intersection with the shortest path length PL_{sj}, the least direction change angle θ, and the connectivity with the current intersection. In this example, when s_2 traveling to I_2 arrives at I_5, it has two candidate intersections, that is, I_4 and I_8. Note that $\vec{v}$ and $\vec{p}$ are the moving directions of sender s_2 and the delivered packet, respectively. s_2 computes SSW_4 and SSW_8 according to (9). It is easy to get that $U_4 = 1$ and $U_8 = 0$ because there are no vehicles between intersections I_5 and I_8. Hence, s_2 chooses I_4 with the minimum SSW as its next intersection. Next, s_2 selects its best next hop. Assume that the local vehicular density of s_2 is smaller than the density threshold d_{th}. s_2 uses the centralized way to compute the CDP_n^a values of candidate neighbors located in the selected segment. Assuming that s_3 has the maximum CDP_n^a, it is selected to be the next hop.

(3) The local vehicular density of s_3 is higher than the density threshold d_{th}; therefore s_3 adapts to the distributed fashion to discover neighbors. s_3 broadcasts a start beacon to inform its neighbors about its position and velocity. Each neighbor which received the beacon will compute its unique waiting time for sending answer to s_3 based on (14). In this case, we assume s_4 is the one which first replies and then s_4 is selected to be the best next hop of s_3.

(4) If the link fails when s_4 is sending packets, the recovery mode of routing path is active. s_4 will notice its neighbors and find a backup link from its current neighbors. If s_4 has no neighbor to deliver packets, it will carry them until some appropriate nodes move into its transmission range. The following process of packet forwarding is the same as the above illustrated until the packet is delivered to its destination D.

4. Performance Evaluation

To evaluate the performance of the proposed ARP-QD, we simulate the protocol on a variety of data transmission rates and network densities. To compare the performance of ARP-QD with the previous works in VANETs routing, we also simulate basic GPSR [7], which aims to find a path with minimum hop count, and ROMSGP [11] which can guarantee a high level of stable communication to some extent. Note that most of geographic VANET routing protocols are based on GPSR with little differences in essence. ROMSGP is a classical stable VANET routing protocol for comparison.

4.1. Simulation Environment. We simulate ARP-QD in the vehicle traffic model using the standard NS2 simulator [24], which offers full simulation of the IEEE 802.11 physical and MAC layers. In our simulation, network size is set to be 50, 100, 150, 200, and 250 nodes with 802.11 WaveLAN radios. The assumptions are that all vehicles have the same transmission range of 250 m and all packets have the same size of 512 bytes. We simulate 20 constant bit rate (CBR) traffic flows to destinations, and sources and destinations are picked up randomly. The transmission rate of each CBR flow is set to be 0.5, 1.0, 1.5, 2.0, 2.5, and 3.0 packets per second (p/s). Each simulation lasts for 1000 seconds. Table 1 summarizes the key parameters in the simulation.

4.2. Mobility Model. The mobility model has a great impact on the studied protocol behavior in the simulation and the corresponding results [25]. For evaluating protocol performance accurately in such a complex and dynamic vehicular environment, we use VanetMobiSim [25] to initially place nodes uniformly at random and generate the random movement of the nodes within a $10 \times 10\,\text{km}^2$ rectangular region with a maximum speed of 30 m/s. Figure 5 shows the simulation scenario, including 9 intersections and 12 road segments. We assume that a road segment composes two lanes without traffic signals. When a node approaches the intersection, it will randomly choose a road segment to turn its direction without pause.

Table 1: Simulation parameters.

Parameter	Value
Number of lanes	2
Number of nodes	50, 100, 150, 200, 250
Velocity	10–30 m/s
Simulation duration	1000 s
Simulation area	$10 \times 10\,\text{km}^2$
Channel capacity	2 Mbps
Wireless communication range	250 m
Mac protocol	802.11 DCF
Beacon interval	0.5 s
Data packet size	512 bytes
CBR rate	0.5, 1.0, 1.5, 2.0, 2.5, 3.0 p/s
Routing protocol	ARP-QD, GPSR, ROMSGP

4.3. Simulation Results. We focus mainly on the performance of delivery ratio and delivery delay in the simulation. (1) Delivery ratio is measured as the ratio of the number of successfully delivered data packets to the total number of transmitted data packets. The packet will be dropped when it fails to be delivered, without retransmission rule. (2) Delivery delay is measured as the average time elapsed from sending the packet by the source node to receiving it by the destination. Without loss of generality, we first fix the adaptive weigh factors (α, β) at $(0.5, 0.5)$ to evaluate the impact of transmission rate and network density. Next, we fix the transmission rate at 1.5 p/s and the number of nodes at 150 to observe the impact of adaptive weight factors α and β.

4.3.1. Delivery Ratio. The number of nodes is set to 150 when we study the impact of transmission rate, while the transmission rate is fixed at 1.5 p/s when we focus on the impact of network density. Figures 6 and 7 show the delivery ratio with respect to varied transmission rate and the number of nodes, respectively. The two figures show that the proposed ARP-QD has higher delivery ratio compared with that of GPSR and ROMSGP in all cases. The reason is that ARP-QD considers the whole path based on the SSW metric, while GPSR works on the vehicle-by-vehicle forwarding and ROMSGP makes the vehicles with the same moving direction into groups, which only considers the local segment, rather than the whole path. Another reason is that the adaptive neighbor discovery algorithm reduces the communication overheads. Furthermore, from Figure 6 we can see that the delivery ratio of ARP-QD does not change much as the transmission rate is increased, while that of GPSR and ROMSGP deteriorates. This comes from the fact that the routing paths found by ARP-QD are more tolerant to the high network load due to the adaptive neighbor discovery algorithm. The main reason is that the adaptive neighbor discovery algorithm largely reduces the beacon cost to require neighbors' information, which reserves more bandwidth for data delivery. Thus, the network load is still tolerable when the transmission rate rises up to 3.0 p/s. From Figure 7 we can observe that the delivery ratio increases with the rise of the number of nodes but decreases when the number of nodes goes up to 200.

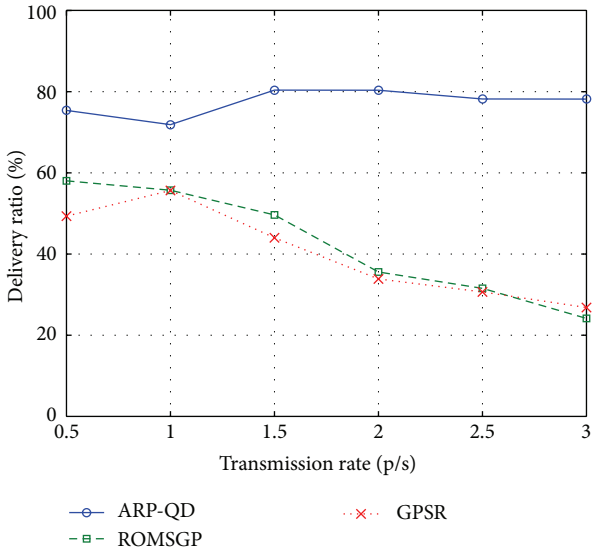


Figure 6: Delivery ratio versus transmission rate.

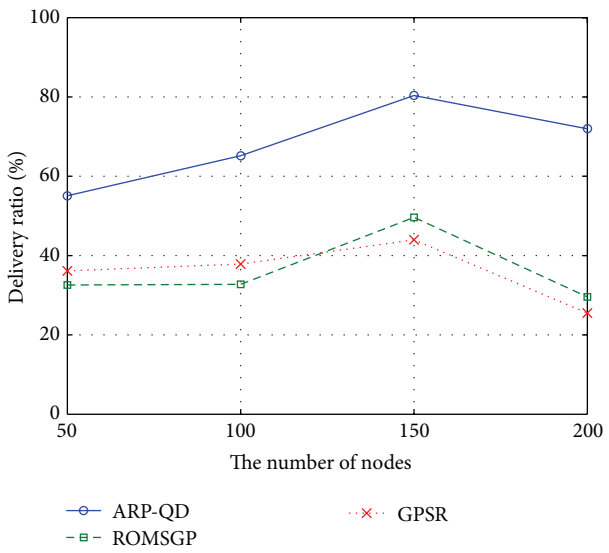


Figure 7: Delivery ratio versus the number of nodes.

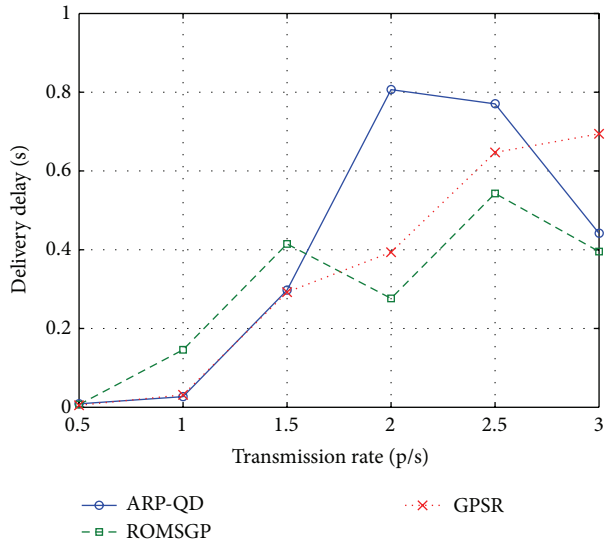


Figure 8: Delivery delay versus transmission rate.

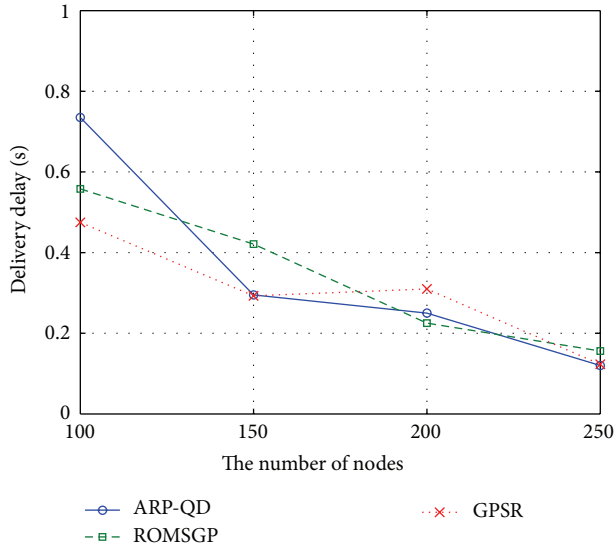


Figure 9: Delivery delay versus the number of nodes.

The reason is that before the number of nodes reaches 150 or other values less than 200, the increased network density becomes higher than the density threshold and the enhanced connectivity and the reduced communication overheads during the neighbor discovery procedure improve the delivery ratio. With the continuous increase of node density, the overheads increase for updating all nodes' neighbor list. Thus the performance of delivery ratio diminishes.

4.3.2. Delivery Delay. As shown in Figure 8, the delay of ARP-QD is the same as that of GPSR but is lower than that of ROMSGP at lower transmission rate. That is because the collisions are rare to happen when the transmission rate is lower and ROMSGP tends to choose the path with more hops for stability. However, when the transmission rate increases, the performance of ARP-QD deteriorates in terms of delivery delay. That is because high transmission rate makes the sender fail to find a backup neighbor quickly; when the link breaks, the time of carry-and-forward procedure prolongs the delivery delay. In brief, ARP-QD is not suitable for the applications with high QoS requirement on delivery delay when the network load is higher. Figure 9 shows that the delay of all protocols decreases along with the increase of the number of nodes. The reason is that packets can be delivered quickly with less caching time when the network density is high. Moreover, ARP-QD only has little difference on the delivery delay compared with the other two protocols when the number of nodes increases, which means ARP-QD does not give high compromise on the delivery delay.

4.3.3. The Impact of Adaptive Factor α. In order to evaluate the impact of weight factors α and β for different QoS

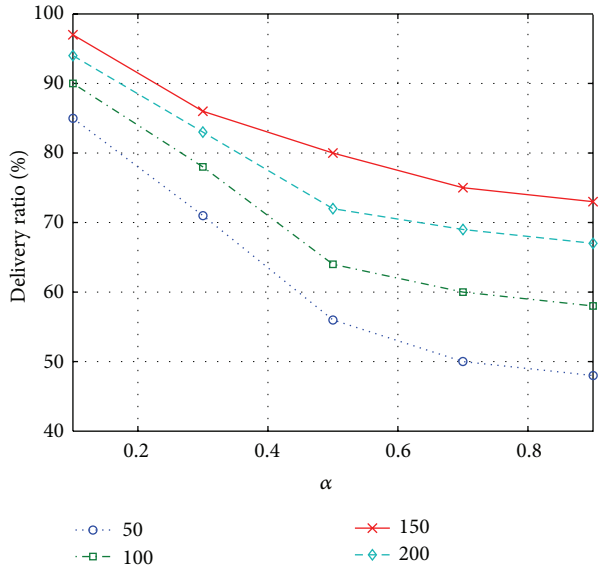


Figure 10: Delivery ratio versus α.

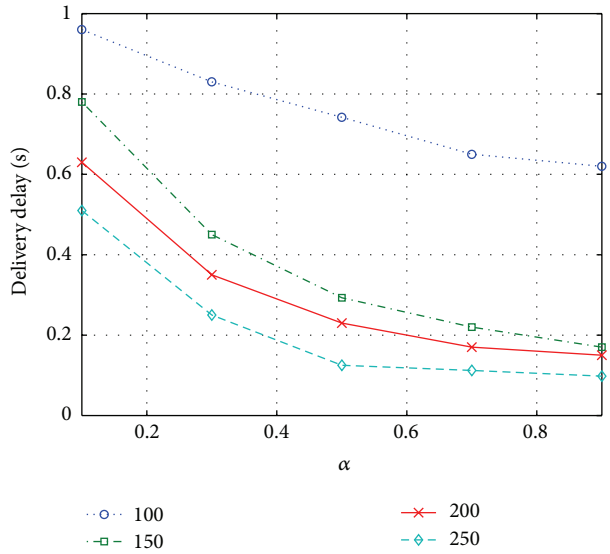


Figure 11: Delivery delay versus α.

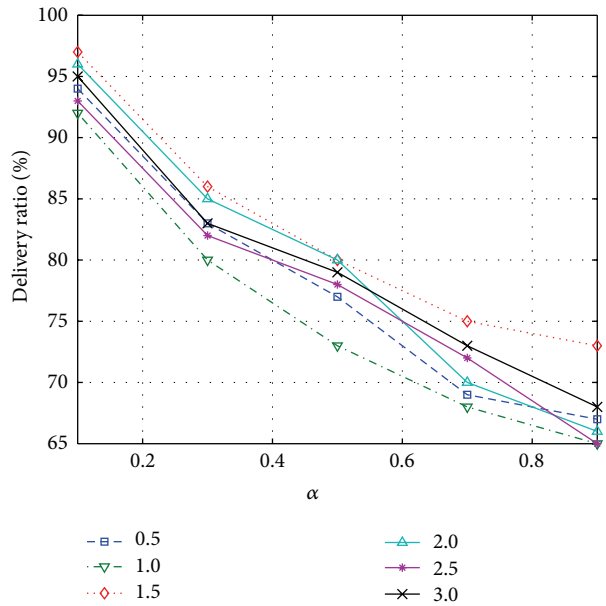


Figure 12: Delivery ratio versus α.

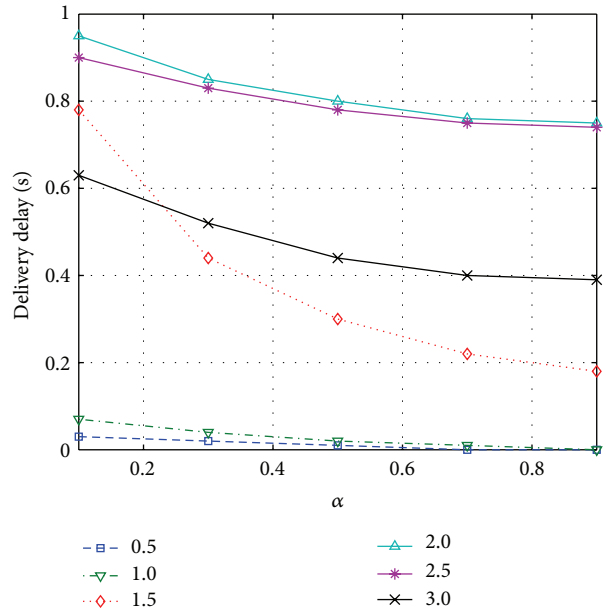


Figure 13: Delivery delay versus α.

requirements, we obtain the simulation results of delivery ratio and delivery delay when α increases from 0.1 to 0.9 with the interval of 0.2. In Figures 10 and 11 where the transmission rate is set to 1.5 p/s, the four curves represent different number of nodes. In Figures 12 and 13 where the number of nodes is set to 150, the six curves represent different transmission rate.

From Figures 10, 11, 12, and 13, we can draw the conclusion that the delivery ratio declines, while the delivery delay goes down along with the increase of α. That is because the link efficiency has larger weight and the link stability has smaller weight accordingly when the factor α is increased. The link has more probability to break down along with the rise of α; thus the delivery ratio turns worse. At the same time, the number of hops for each path is decreased with the higher requirements on link efficiency; thus the delivery delay is improved. In addition, from Figure 12, we can observe that the delivery ratio of ARP-QD does not vary much when the transmission rate changes, which is the same as that observed in Figure 6. Figure 13 shows that the delivery delay of ARP-QD varies much when the network density changes as analyzed in Figure 8. These results show that the weight factor α can adaptively satisfy the diverse QoS requirements of different applications.

5. Conclusion

In this paper, the proposed adaptive routing protocol based on QoS and vehicular density (ARP-QD) is capable of

finding a fast and reliable path for end-to-end data delivery within urban VANETs environments according to diverse QoS requirements of different applications. To reduce the communication overheads furthermore, ARP-QD adopts the adaptive fashion to obtain the neighbors' information based on local vehicular density and recovers quickly when the routes are disrupted. Numerical simulations showed that ARP-QD has a higher delivery ratio than GPSR and ROMSGP, without giving large compromise on the delivery delay. In the future, we shall take the real data trace into consideraion to validate ARP-QD protocol and combine the link correlations to estimate link quality.

Notations

n: The neighbor node
s: The current sender
S: The source node
D: The destination
R: The vehicular transmission range
l: The distance of two adjacent lanes
R_n: The Euclidean distance between the neighbor node n and the sender s
L_n: The distance that the neighbor node n moves out the transmission range of the sender s
Q_n: The projection of R_n in the road direction
PL_n: The path length between the neighbor node n and the destination D
PL_s: The path length between the sender s and the destination D
v_n: Velocity of the neighbor node n
v_s: Velocity of the current sender s
T_n: The link duration time between the neighbor node n and the sender s.

Conflict of Interests

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

Acknowledgment

This research was supported by Major Program of National Natural Science Foundation of China (no. 61190114).

References

[1] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, 2008.

[2] 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.

[3] 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.

[4] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: a survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.

[5] M. Jerbi, S.-M. Senouci, T. Rasheed, and Y. Ghamri-Doudane, "Towards efficient geographic routing in urban vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 5048–5059, 2009.

[6] D. Lee, Y.-H. Kim, and H. Lee, "Route prediction based vehicular mobility management scheme for VANET," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 679780, 9 pages, 2014.

[7] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, ACM, August 2000.

[8] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 156–161, IEEE, Columbus, Ohio, USA, June 2003.

[9] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, pp. 69–72, 2005.

[10] J. Yuan, Y. Zheng, C. Zhang et al., "T-drive: driving directions based on taxi trajectories," in *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10)*, pp. 99–108, ACM, San Jose, Calif, USA, November 2010.

[11] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, "A stable routing protocol to support its services in vanet networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3337–3347, 2007.

[12] C. Rezende, R. W. Pazzi, and A. Boukerche, "Enhancing path stability towards the provision of multimedia support in vehicular ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 1–5, IEEE, 2010.

[13] Q. Yang, A. Lim, and P. Agrawal, "Connectivity aware routing in vehicular networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08)*, pp. 2218–2223, IEEE, April 2008.

[14] H. Yu, S. Ahn, and J. Yoo, "A stable routing protocol for vehicles in urban environments," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 759261, 9 pages, 2013.

[15] H. Saleet, R. Langar, K. Naik, R. Boutaba, A. Nayak, and N. Goel, "Intersection-based geographical routing protocol for VANETs: a proposal and analysis," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 9, pp. 4560–4574, 2011.

[16] T. Clausen, P. Jacquet, C. Adjih et al., *Optimized Link State Routing Protocol (OLSR)*, RFC, 2003.

[17] H. Huang and S. Zhang, "A routing algorithm based on dynamic forecast of vehicle speed and position in vanet," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 390795, 9 pages, 2013.

[18] H. Saleet, O. Basir, R. Langar, and R. Boutaba, "Region-based location-service-management protocol for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 917–931, 2010.

[19] Digital map from mapmechanics, http://www.allmapdata.com/.

[20] Z. Mo, H. Zhu, K. Makki, and N. Pissinou, "MURU: a multi-hop routing protocol for urban vehicular ad hoc networks," in *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '06)*, pp. 1–8, July 2006.

[21] R. N. Murty, G. Mainland, I. Rose et al., "Citysense: an urban-scale wireless sensor network and testbed," in *Proceedings of the IEEE Conference on Technologies for Homeland Security*, pp. 583–588, IEEE, 2008.

[22] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, 1997.

[23] J. A. Davis, A. H. Fagg, and B. N. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned Ad-Hoc networks," in *Proceedings of the 5th International Symposium on Wearable Computers (ISWC '01)*, pp. 141–148, IEEE, October 2001.

[24] The Network Simulator, "The vint project. The ucb/lbnijvint network simulator-ns (version 2)," http://mash.cs.berkeley.edu/ns.

[25] J. Härri, F. Filali, C. Bonnet, and M. Fiore, "Vanetmobisim: generating realistic mobility patterns for vanets," in *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET '06)*, pp. 96–97, ACM, September 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 239267, 10 pages
http://dx.doi.org/10.1155/2015/239267

Research Article

A Novel Interest Detection-Based Video Dissemination Algorithm under Flash Crowd in Mobile Ad Hoc Networks

Shijie Jia,[1] Shengli Jiang,[1] Yuanchen Li,[1] Xihu Zhi,[1] and Mu Wang[2]

[1]*Academy of Information Technology, Luoyang Normal University, Luoyang 471022, China*
[2]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*

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

Received 5 December 2014; Accepted 5 March 2015

Academic Editor: Liang Zhou

The peer-to-peer-based video resource dissemination is important for handling extreme conditions such as flash crowds which severely break the balance between supply and demand of video content and bring negative effects for quality of service (QoS). In this paper, we propose a novel interest detection-based video dissemination algorithm under flash crowd in mobile ad hoc networks (IDVD). IDVD classifies the user behavior of requesting video content in terms of the popularities and playback time of video and considers the demand of popular video as the cause of flash crowd. IDVD makes use of the prediction of necessary bandwidth and period time of intensive request to be aware of period time and scale of flash crowd. IDVD employs a resource dissemination algorithm which can fast discover interested nodes and achieve the on-demand dissemination of video resources in neighboring geographical area, in order to handle imbalance between supply and demand. IDVD uses the epidemic model to describe the state transition of nodes and define the convergence condition of resource dissemination algorithm. Extensive tests show how IDVD achieves much better performance results in comparison with other state of the art solutions.

1. Introduction

The mobile ad hoc networks (MANETs) rely on packets hopping between nodes which act as hosts and routers to achieve communication of nodes, so they are well known for flexible architectures and rapid deployment [1, 2]. The multimedia streaming services are more popular applications in Internet and mobile Internet [3–10]. The deployment of video services in MANETs supports ubiquitous access for the mobile users, enhancing user experience [11, 12]. P2P technologies provide the solution for large-scale video sharing in wireless networks such as MANETs and wireless sensor networks (WSNs), so that the mobile users conveniently fetch desired video content from the peers [13–17]. The video with wonderful content always attracts mass peers to seek and download the resources from other peers or the media server. However, the relatively limited network bandwidth and capacities of mobile nodes cannot afford traffic requirement caused by intensive resource request, reducing quality of service (QoS). Therefore, a solution based on green communication requirement, which can understand behavior of pursuing popular video content, fast discover the nodes of cooperatively caching resources, and efficiently spread video content should be considered for video streaming system in MANETs.

The flash crowds lead to serious imbalance between supply and demand for video streaming resources and bring negative influence in the scalability and QoS of video streaming systems. For instance, the blowout of video resource request results in the long delay of response and network congestion. The mass researchers recently focus on the study of flash crowd in order to improve QoS of video streaming system and reduce the cost of system deployment [18–20]. A fluid-based model for P2P live streaming systems was proposed in [18], which describes the relationship between the system capacity, peer startup latency, and system recovery time with and without admission control. A mathematical model proposed in [19] captures the relationship between time and scale in P2P live streaming systems and designs a principle

of scale control. A model proposed in [20] predicts the scalability of system with increasing number of nodes and provides enough upload bandwidth for sudden resource request according to the estimation of server bandwidth, so as to ensure high QoS. The aforementioned solutions focus on modeling the living streaming systems to address the influence of flash crowd and are unsuitable for the mobile environment with limited network resources and high mobility of mobile nodes. The high-efficiency video resource sharing in wireless networks also is important solution for handling flash crowd. The management of video resources regulates distribution of resources in terms of dynamic user demand, which fast perceives and responds to the variation of video resource demand. The video resource dissemination strategies can fast spread requested resources in overlay networks, which reduce recovery time of balance between supply and demand. Therefore, an efficient solution based on dynamic balance between supply and demand, which supports fast video resource dissemination and efficiently prevents the degraded QoS should be considered for P2P-based video streaming systems over wireless mobile networks.

In this paper, we propose *a novel interest detection-based video dissemination algorithm under flash crowd in mobile ad hoc networks (IDVD)*. By the analysis of resource request behavior of users and the estimation of the popularities of video resources, IDVD constructs a novel "H"-model to explain that the intensive request of popular video content is the main cause of the flash crowd in multimedia streaming system. IDVD makes use of the historical information (e.g., bandwidth and period time) of large-scale request for popular video content to predict the period time and scale of flash crowd. IDVD designs a novel resource dissemination algorithm which can fast spread resources to meet the demand of upload bandwidth. In order to reduce the cost of resource dissemination, the resource carriers only search and spread the interested mobile nodes in neighboring geographical area and dynamically regulate range of dissemination in terms of predicted necessary bandwidth and capacities of carriers. IDVD uses the epidemic model to describe the dissemination process of resources and state transition of nodes and define convergence condition of spreading process. Simulation results show how IDVD achieves much better performance results in comparison with other state of the art solutions.

2. Related Work

The video system models under flash crowd have attracted increasing research interests from various researchers. For instance, a fluid-based model for P2P live streaming systems was proposed in [18], which studies the relationship between capacity and recovery time of system and peer startup latency with and without admission control for flash crowds. In the systems without admission control, this paper finds that there is an independent relationship between the capacity and initial state of system while power law decreases with the departure rate of stable peers. The paper also shows that the admission control can help the system relieve the large flash crowds in the systems with admission control and proposed the flash crowd handling strategies in order to satisfy the peer startup performance under various circumstances. The mathematical framework in [19] researches the inherent relationship between time and scale of P2P live streaming system during a flash crowd. The population control procedure improves the system scale by trading peer startup delays. This paper also analyzed the effects of partial knowledge of peers and the competition of limited upload bandwidth resources between peers. Moreover, an analytical model in [20] for flash crowds is based on the evolution of the utilization of available bandwidth at peer side in order to investigate impact of the utilization of available bandwidth. The model can predict the system scalability with increasing number of nodes and provide necessary bandwidth for sudden request.

Recently, some researchers focus on the resource management strategies in order to optimize resource distribution and make the balance between supply and demand. Kozat et al. proposed a hybrid P2P video-on-demand architecture, which improves transmission efficiency of popular videos [21]. In this architecture, each system member caches a video chunk and makes use of surplus upload bandwidth to serve other nodes. The server schedules the video resources in the system to respond to the request of nodes and provide reliable streaming service. In order to balance the load between server and system members, the architecture considered the caching problem as a utility optimization problem based on supply and demand and used the multiple caching mechanisms to optimize the performance of system. PECAN in [22] proposed a peer cache adaptation strategy, in which each peer dynamic regulates the local storage capacity to improve the scalability of system. PECAN employs a cache replacement algorithm to improve the resource distribution in terms of the popularities of video chunk and show that the storage capacities of peers are corresponding to the request rate of resources. PECAN designed a distributed reputation and monitoring system to discover selfish peers.

Moreover, some file (video) resource dissemination algorithms recently are proposed. For instance, Mokhtarian and Hefeeda show that the problem of allocating the seed server capacity is NP-complete and proposed a seeding capacity allocation algorithm to address the optimal allocation problem [23]. The paper proposed an analytical model to predict the performance of P2P-based video system by using an allocation algorithm, which estimates the long-term network throughput according to video quality and total served bitrate. Venkatramanan and Kumar analyzed the evolution process of the interest in the content under the linear threshold model and made use of an epidemic spread model to control the content copying process [24]. This paper modeled the coevolution process of popularity and delivery of video content according to homogeneous influence linear threshold model. This paper used fluid limit ordinary differential equations to provide the selection of parameters for the control of content suppliers and address optimization problems for content delivery. Altman et al. proposed an extensional epidemic model to characterize file sharing behavior in P2P networks including free-riding peers [25]. This paper modeled P2P network dynamics by a Markov chain, where the state of P2P system evolves from branching process to a supercritical P2P swarm with increasing network

size. The paper shows that there are the phase transitions; the small change of parameters causes a large change in the network behavior for two models of epidemic and branching.

3. IDVD Detailed Design

3.1. Media Server. The video resources are stored at the media server in order to provide original video data for all mobile nodes in MANETs; namely, a video file set is defined as $S_V = (f_1, f_2, \ldots, f_n)$. When the server receives a request message, it assigns a candidate supplier which caches the requested file for the requester and adds the information of requester into local node set $S_N = \{(n_1, f_a, t_1), (n_2, f_b, t_2), \ldots, (n_m, f_h, t_m)\}$ where f denotes the requested video file and t is the timestamp of joining system of nodes. The nodes also contact other system members to obtain desired resources, which is detailed next. A system member which leaves system sends a quit message containing the information of played files and corresponding playback start-stop timestamp (local system time of node) to the server. The server considers this information as playback logs and analyzes these logs to be aware of the condition of request and playback of resources such as the file popularity and the mean playback time of video files. The popularities of items in S_V follow a Zipf distribution [26, 27] and the popularity value of each file f_i can be obtained according to

$$\text{pop}_i = \frac{\gamma_i}{\sum_{c=1}^{n} \gamma_c}, \tag{1}$$

where γ_i is the access frequency of f_i and $\sum_{c=1}^{n} \gamma_c$ is the total access frequency of all files. The mean playback time ratio of video file f_i is defined as

$$\bar{l}_i = \frac{\sum_{c=1}^{N_u} l_c}{N_u \times L_i}, \tag{2}$$

where l_c is playback time of cth user, L_i is the length of video f_i, and N_u is the total number of users which have watched f_i. We use the mean playback time ratio of f_i as weight value of popularities, so the weighted popularities of f_i can be defined as $\text{pop}_i^w = \bar{l}_i \times \text{pop}_i$. The files whose weighted popularities are larger/less than $(1/n) \sum_{c=1}^{n} \text{pop}_c^w$ (average value of weighted popularities of n files) are popular/unpopular. In terms of the fetched content, the user playback behavior can be classified as (1) the users which are watching popular files request popular files; (2) the users which are watching unpopular files request popular files; (3) the users which are watching unpopular files request unpopular files; (4) the users which are watching popular files request unpopular files. As Figure 1 shows, the popular/unpopular files and channels between them form an "H"-model". The first and second kind behaviors generate the traffic to the popular files, namely, the channel C_p; the third and fourth kind behaviors form the traffic of channel C_u. The popular content always can attract more users, namely, the traffic value in C_p is far greater than that in C_u. The less active system members in C_u do not require mass upload bandwidth, but this leads to the shortage of available resources. For instance, if there are less nodes which cached these files and have the asynchronous playback point with the requesters in overlay networks, this leads to fragile logical link between suppliers and requesters due to dynamic resource demand. The requesters need to fetch desired resources from the server. If there are no requested resources in overlay networks, the server needs to provide the initial streaming data for the requesters. Therefore, the shortage of available resources causes too much intervention of server for scheduling resources, which increases the load of server. We make use of distributed hash table (DHT) to group these system members in C_u [28] and optimize the resource distribution according to cooperative cache strategies [29, 30], which can improve the efficiency of sharing resources and reduce the load of server.

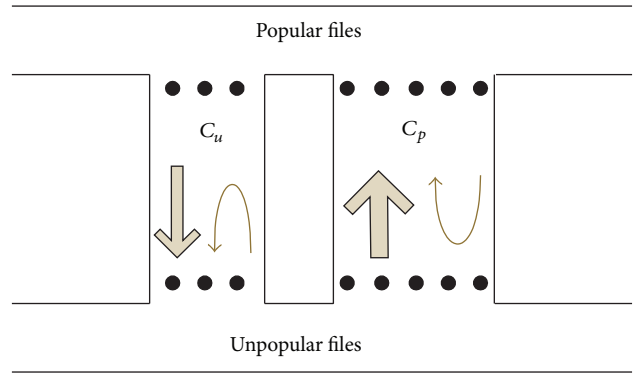


Figure 1: "H" model.

The demand from the large number of nodes for popular video is one of the main causes of flash crowd in C_p. The transient large-scale request breaks the balance between supply and demand for upload bandwidth, which results in high serving delay and the overload of server. The server needs to be aware of the bandwidth requirement level and period time of intensive request in advance. The huge traffic in C_p usually is the main reason for the system bottleneck, so we introduce the traffic prediction scheme for any popular video f_i in C_p. The length of f_i is defined as a period time. We use an average value of traffic during each period time to define the channel traffic set; namely, $S_{f_i} = (\bar{b}_{t_1}, \bar{b}_{t_2}, \ldots, \bar{b}_{t_m})$ where any average traffic value can be obtained by

$$\bar{b}_{t_c} = \frac{\sum_{j=1}^{u} b_j}{t_c}, \tag{3}$$

where u is the number of request bandwidth values in C_p during t_c. S_{f_i} is considered as the original series $S_{f_i}^{(O)}$ and is processed to an accumulated series $S_{f_i}^{(A)} = (b_{t_1}^{(A)}, b_{t_2}^{(A)}, \ldots, b_{t_m}^{(A)})$, according to

$$b_{t_k}^{(A)} = \left\{ \sum_{c=1}^{k} \bar{b}_{t_c} \mid k = 1, 2, \ldots, m \right\}. \tag{4}$$

Because the random traffic values are difficult to form a stable predictable variation trend, we make use of the Grey Forecast Model GM($b^{(A)}$) [31, 32]; the first-order differential equation of GM($b^{(A)}$) is defined as

$$\frac{db^{(1)}}{dt} + rb^{(1)} = h, \tag{5}$$

where t is the time series variable and b is the series variable of traffic accumulation with increasing time interval. r and h denote the grey level of development and control, respectively. We use the ordinary least square method to solve the values of r and h according to

$$\widehat{U} = \begin{bmatrix} \hat{r} \\ \hat{h} \end{bmatrix} = \left(D^T D\right)^{-1} D^T y, \quad D = \begin{bmatrix} \frac{1}{2}\left[b_{t_2}^{(A)} + b_{t_1}^{(A)}\right] 1 \\ \cdots \\ \frac{1}{2}\left[b_{t_m}^{(A)} + b_{t_{m-1}}^{(A)}\right] 1 \end{bmatrix}, \tag{6}$$

where $\hat{r}$ and $\hat{h}$ are the solutions of r and h, respectively. D^T is the transposition matrix of D and $y = (\overline{b}_{t_2}, \overline{b}_{t_3}, \ldots, \overline{b}_{t_m})^T$. Equation (7) denotes the solution of (5) by using $\hat{r}$ and $\hat{h}$:

$$\hat{b}(k+1) = \left(\overline{b}_{t_1} - \frac{\hat{h}}{\hat{r}}\right) e^{-\hat{r}k} + \frac{\hat{h}}{\hat{r}}. \tag{7}$$

$\hat{b}(k+1)$, $k \le m$, is the fitting value; $\hat{b}(k+1)$, $k > m$, is the forecast value; namely, we compare $\hat{b}(k+1)$, $k > m$, with the upper limit of server load to make a decision whether to spread f_i in networks. We use a posteriori variance ratio $R^{(P)}$ and an occurrence probability $P^{(P)}$ to ensure the confidence level of prediction values. $R^{(P)}$ and $P^{(P)}$ are defined as

$$R^{(P)} = \frac{F}{\sigma}, \qquad P^{(P)} = P\left\{\left|C(k) - \overline{C}\right| < 0.6745\sigma\right\}, \tag{8}$$

where $C(k)$ is the residual value; namely, $C(k) = \overline{b}_{t_k} - \hat{b}(k)$, $k = 2, 3, \ldots, m$. $\overline{C}$ and F are the residual mean value and variance, respectively, according to

$$\overline{C} = \frac{\sum_{c=2}^{m} C(c)}{m-1}, \qquad F = \sqrt{\frac{\sum_{c=1}^{m}\left(C(c) - \overline{C}\right)^2}{m-1}}. \tag{9}$$

σ and b_{mean} are the variance and mean value of items in $S_{f_i}^{(O)}$ according to

$$\sigma = \sqrt{\frac{\sum_{c=1}^{m}\left(\overline{b}_{t_c} - b_{\text{mean}}\right)^2}{m}}, \quad b_{\text{mean}} = \frac{\sum_{c=1}^{m} \overline{b}_{t_c}}{m}. \tag{10}$$

$R^{(P)}$ and $P^{(P)}$ can reflect the confidence level of prediction values of channel traffic, so we make use of two threshold values $T^{(R)}$ and $T^{(P)}$ in terms of the Grey Forecast Model to measure the confidence level. If $R^{(P)} \ge T^{(R)}$ and $P^{(P)} \ge T^{(P)}$, the prediction value is credible. When the prediction value $\widehat{b}(u)$, $u > m-1$, is credible and is larger than the summation of upload bandwidth known by the server (total bandwidth of the server and items in S_N), the server sends a message containing $\widehat{b}(u)$ to the nodes which store f_i in S_N and are considered as the carriers of f_i. These carriers return a statistical information about available upload bandwidth to the server by collecting the upload bandwidth from the nodes which are downloading f_i from the carriers. If $\widehat{b}(u)$ still is larger than the summation of upload bandwidth, the server requires these nodes to fast disseminate f_i in order to cope with intensive request.

3.2. Resource Disseminate Model. In order to reduce the load of server, the server only sends a message containing necessary upload bandwidth $\widehat{b}(u)$ and a carrier list to each carrier. The carriers are responsible for controlling the dissemination process by the message exchange. We use a token-based message exchange strategy to achieve the synchronization of information between carriers. Each item in the carrier list has a random number and successively sends the message to other carriers according to the value of number. For instance, the number of items in the list is k. The $k-1$ carriers send collected information (e.g., the number of discovered nodes which are interested in the resources) to kth carrier n_s which has the token. After n_s handles these received messages, it returns a message containing the calculation results to other carriers. Meanwhile, n_s turns the token over to $k-1$th carrier. After the carrier with the smallest number return message to other carriers, it returns the token over to n_s. The token-based exchange strategy can balance the load between carriers and does not cause high message overhead.

When these carriers receive the request of spreading f_i from the server, they start to discover interested nodes (INs) and require INs caching f_i. The INs include two types of nodes: the interested mobile nodes (IMNs) and the interested system members (IMs) which are playing other videos. The carriers also are considered as the inquirers due to searching the INs. In order to reduce the cost of spreading f_i, we employ a guidance-based dissemination strategy to implement geographic region-based file diffusion.

Each inquirer n_j makes use of cross-layer method to add the information of viewing file (current playback state) to one-hop multicast message at the MAC layer. If there are the system members in the one-hop neighbor nodes of n_j, these members return the information of current played video file. Moreover, if the one-hop neighbor nodes are interested in f_i, they also add an interest mark into the return messages. When n_j has exchanged messages with the one-hop nodes, it records the information of played content of one-hop nodes and stores the information of INs. We set a variable period time UT for the above neighbor node discovery process according to our previous work in [33]. The nodes

dynamically change their own UT in terms of the variation level of mobility of one-hop nodes.

n_j needs to select an IMs n_p from an encountered node list L_j during recent p update period as a cooperative inquirer. The list is defined as $L_j = \{(n_1, t_1, f_a), (n_2, t_2, f_b), \ldots, (n_u, t_u, f_h)\}$ where t is the encountered timestamp and f denotes the viewing file in the process of encounter. n_p has the nearest encountered timestamp and does not cache f_i. n_j requires that n_p continues to search the INs from its one-hop nodes and select next cooperative inquirer from L_p. n_j sends a message containing the condition of convergence of iterative search to n_p (for instance, the number of iteration is defined as q times). After n_p has inquired for its one-hop nodes, n_p delivers the collected information of the INs to the selected next cooperative inquirer. After the qth cooperative inquirer has inquired, it directly returns the collected information of INs to n_j. Because the mobile nodes may be inquired by multiple inquirers or cooperative inquirers, we define the following rule in order to ensure the accuracy of statistical information in the process of inquiry. (1) If a node has sent the interest mark to an inquirer or a cooperative inquirer, it only return the message containing uninterested mark after the reception of inquiry message of other inquirers or cooperative inquirers; (2) a node cannot become the cooperative inquirer of two carriers in the same process of inquiry; (3) the carriers cannot become the cooperative inquirers of other carriers.

After n_j receives and stores the collected information of INs from the qth cooperative inquirer, it keeps the contact with discovered INs and forwards the messages to the carrier n_t which has the token. When n_t receives the information of discovered INs from all carriers, it estimates needed number of INs. We make use of the epidemic model (SIR) to calculate needed number of INs. All parameters of the epidemic model are listed in the Notations. The necessary and sufficient condition of implementing SIR model is defined as $N_t > N_I$, $N_I = \widehat{b}(u)/b_p$ where N_I denotes needed number of nodes which store f_i and b_p is the transmission rate of f_i. If $N_t < N_I$, n_t requires that the carriers continue to search more INs. Once the carriers find the IMs, they transmit the data of f_i to the IMs; namely, the IMs can be immediately infected. However, the IMNs are considered as the potential infected nodes so that the carriers do not require them to cache f_i at once. The dynamic number of carriers leads to the change of available upload bandwidth. For instance, the carriers which have watched the whole video content remove f_i from local buffer. When the inquirers do not discover more IMs to meet the demand of bandwidth due to the limited detection range, they require that the IMNs cache f_i. The IN discovery rate DR_j of any inquirer n_j is defined as

$$\mathrm{DR}_j = \frac{N_{\mathrm{IN}_j}}{N_t}, \tag{11}$$

where N_{IN_j} denotes the number of INs discovered by n_j. λ is defined as $\lambda = N_{\mathrm{IM}}/N_t$ where N_{IM} is the number of discovered IMs. The members which have watched f_i are usually uninterested to view f_i again; namely, their state becomes uninterested. Therefore, the value of μ is set to 1. We use a differential equation to denote the SIR model as follows:

$$\begin{aligned} \frac{dI}{dt} &= \lambda SI - \mu I, \\ \frac{dS}{dt} &= -\lambda SI, \\ I(t_0) = I_0, &\qquad S(t_0) = S_0. \end{aligned} \tag{12}$$

We obtain the solution of the above differential equation; namely, $I = S_0 + I_0 - S + (\mu/\lambda)\ln(S/S_0)$. Further, we can obtain the solution of S according to known value of S_0, I_0, I, μ, and λ, namely, $\widehat{S} = W(-\lambda e^{\lambda I - \lambda S_0 - \lambda I_0 - \ln S_0})/ - \lambda$ where W is the Lambert W Function and $I = N_I/N_t$. $\widehat{S}$ is the needed number of INs in order to ensure the required scale of nodes cached f_i based on current spreading rate. n_t reassigns the needed number of INs for the carriers according to the collected number of inquired nodes of carriers during an inquiry period and requires that the carriers continue to find new INs. The more the number of inquired nodes of a carrier is, the higher the probability of discovering INs is. Therefore, the carriers which have more inquired nodes should be assigned more number of IN discovery. The above process is considered as an IN discovery period. n_t turns the token to next carrier after a discovery period. If the total number of discovered INs is equal to or greater than $\widehat{S}$, the carrier which has the token requires that the carriers keep the state of equalisation during predicted period time t_u. After the system went through t_u, the carriers disconnect the contact with all INs.

4. Testing and Test Results Analysis

We investigate the performance of the proposed IDVD in comparison with HILT-SI model [24]. We chose a 100-second long video clip f_i which is considered as copied content. IDVD was modeled and implemented in NS-2, as described in the previous sections.

4.1. Testing Topology and Scenarios. Table 1 lists some NS-2 simulation parameters of the MANET for the two solutions. We define initial random speed and target location of movement for all mobile nodes. After the mobile nodes arrive at the target location, they continue to move according to newly assigned speed and target location of movement. All mobile nodes follow the above iteration of moving behavior during the whole simulation time. The default distance between server and nodes is set to 6 hops in order to ensure the consistency of cost of accessing to the server for all mobile nodes. The variation of default distance can influence the cost of fetching video content from the server. For instance, the increase/decrease of default distance brings high/low transmission delay of video data and packet loss rate. Initially, there are 200 system members where 20 members play f_i and 80 members are uninterested in f_i. 50 members which are viewing another video want to watch f_i and 50 members request f_i per 0.5 s from 80 s to 105 s. Moreover,

Table 1: Simulation parameter setting for MANET.

Parameters	Values
Area	$1000 \times 1000\ m^2$
Channel	Channel/WirelessChannel
Network interface	Phy/WirelessPhyExt
MAC interface	Mac/802_11
Number of mobile nodes	400
Mobile speed range of nodes	[1, 30] m/s
Simulation time	200 s
Signal range of mobile nodes	200 m
Default distance between server and nodes	6 hops
Transmission protocol	UDP
Wireless routing protocol	DSR
Interface queue	CMUPriQueue
Bandwidth of server	20 Mb/s
Bandwidth of mobile nodes	10 Mb/s
Transmission rate of video data	128 kb/s
Travel direction of mobile nodes	random
Pause time of mobile nodes	0 s
p	2
q	3

50 mobile nodes are interested in f_i. The INs cache and play f_i after they are discovered (influenced). The members which have watched f_i quit the system and remove f_i from local buffer. The uninterested nodes do not cache f_i during the whole simulation time. In HILT-SI, each IN is independently assigned a random infected threshold θ, $0 < \theta < 1$ and the values of Γ and α are set to 0.9 and 0.3, respectively.

4.2. Performance Evaluation. The performance of IDVD is compared with that of HILT-SI in terms of capacity of IN discovery and content spreading, message overhead, average data transmission delay, packet loss rate (PLR), and throughput, respectively.

Capacity of IN Discovery and Content Spreading. The number of discovered INs (the interested members and mobile nodes) and carriers (the nodes carry f_i) denotes the capacities of content dissemination for two solutions.

As Figure 2 shows, HILT-SI's blue curve has slow rise from $t = 100$ s to $t = 200$ s after fast increase from $t = 0$ s to $t = 80$ s, after IDVD's red curve also has a slow increase from $t = 80$ s to $t = 200$ s after fast rise from $t = 0$ s to $t = 60$ s. The increment of HILT-SI curve is larger than that of IDVD and HILT-SI has a longer increase period time than that of IDVD; namely, HILT-SI nearly searches all interested nodes. Figure 3 presents the variation of number of carriers in the video system with increasing simulation time. The blue curve corresponding to HILT-SI's results experiences a fast decrease from $t = 120$ s to $t = 200$ s after it suddenly reaches the peak value 143 at $t = 100$ s. IDVD's red curve also has similar trend; namely, it has a slow rise, suddenly arrives at the peak value

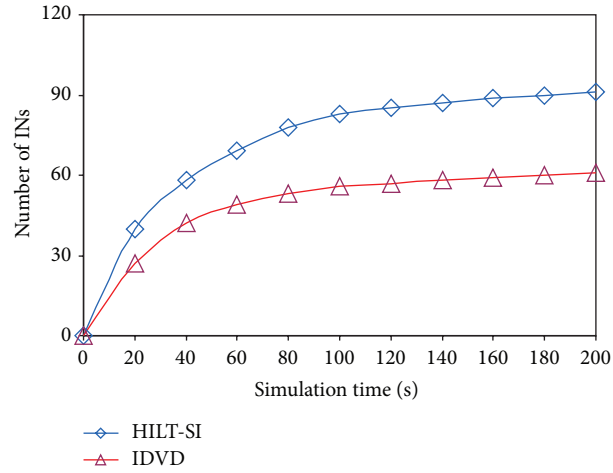


Figure 2: The number of discovered INs against simulation time.

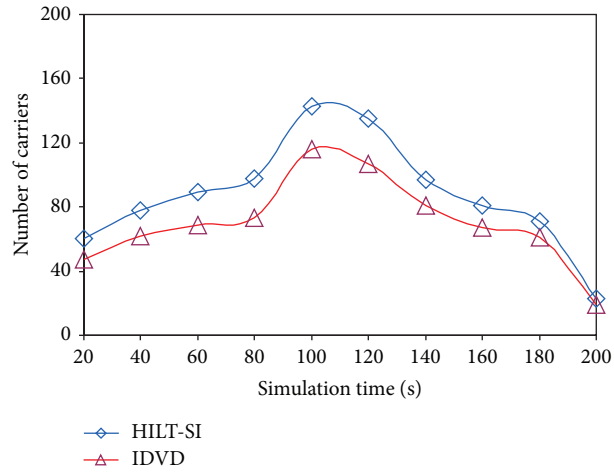


Figure 3: The number of carriers against simulation time.

116 at $t = 100$ s, and fast falls from $t = 120$ s to $t = 200$ s. HILT-SI's results are both higher values and larger fluctuation than those of IDVD during whole simulation time.

In HILT-SI, the carriers continually influence the nodes connected with them according to the given threshold. When the nodes do not become interested nodes, the influence values of carriers increase by the accumulation so that the state of influenced nodes finally becomes interested and these new interested nodes help the carriers influence other nodes by making use of its own influence value. Because the server periodically broadcasts the state of all nodes in the whole network, all INs and carriers try to influence other potential INs. The efficiency of IN discovery HILT-SI is higher than that of IDVD. When the potential INs become new INs, they cache and play f_i; namely, these new INs immediately become new carriers after they fetch f_i. In HILT-SI, the number of carriers has a slow increase from $t = 20$ s to $t = 80$ s. 50 members suddenly join the system and request f_i from $t = 80$ s to $t = 105$ s, so the number of carriers fast reaches the peak value. With increasing simulation time, the initial carriers

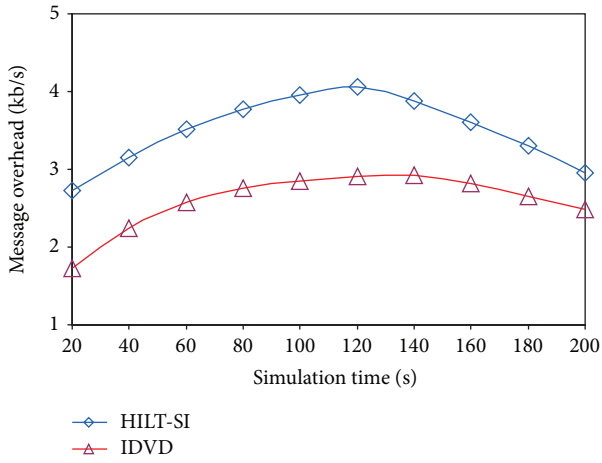


Figure 4: Message cost against simulation time.

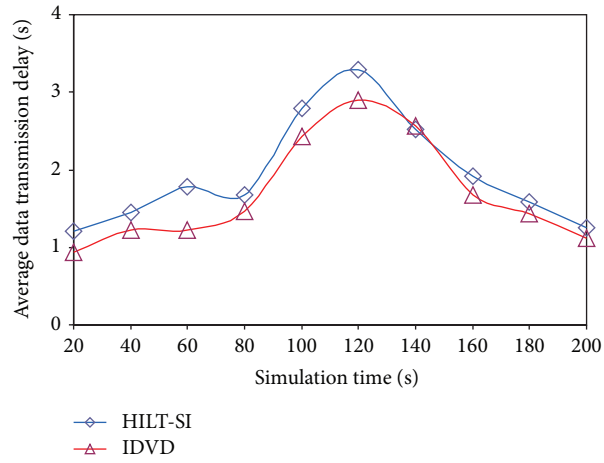


Figure 5: Average data transmission delay against simulation time.

have watched whole video content and quit the system, so the number of carriers fast decreases from $t = 120$ s to $t = 200$ s. HILT-SI does not include the mechanism of controlling the range of IN discovery and spreading resources, so it can fast disseminate the video content to the whole network. In IDVD, the carriers make use of MAC multicast to invite the mobile nodes joining INs and inquire recent encountered system members. IDVD uses p and q to control detection range; namely, the carriers search INs in their neighbor geographic area. IDVD's capacity of IN discovery is limited by local search strategy, but IDVD can regulate the speed of discovery by changing the values of p and q and dynamically assign the number of INs discovered in terms of the capacities of carriers. Moreover, IDVD predicts the needed bandwidth in the future to control the process of discovery, so that the number of INs slowly increases in terms of the needed bandwidth from $t = 80$ s to $t = 200$ s. The variation of number of carriers is based on the number of INs, so the increment and decrement of carriers are less than those of HILT-SI.

Message Overhead. The average bandwidth which is used by the sent messages for the IN discovery is considered as the message overhead.

As Figure 4 shows, the message cost values of two systems have similar changing trend with increasing simulation time. The curve corresponding to HILT-SI's results fast decreases from $t = 120$ s to $t = 200$ s after fast increases from $t = 20$ s to $t = 100$ s, where the peak value is 4.05 kb/s at $t = 120$ s. The curve corresponding to IDVD has a slow rise trend from $t = 20$ s to $t = 100$ s, reaches the peak value 2.92 kb/s at $t = 140$ s, and slightly decreases from $t = 160$ s to $t = 200$ s. The CSPV results are roughly 30% lower than those of HILT-SI, with increasing simulation time.

In HILT-SI, the server periodically broadcasts the state of all nodes in network. When the INs and carriers receive the state information of nodes, they update the state of nodes and continue to influence other potential INs. In order to obtain the capacity of fast IN discovery, the server needs to frequently interact with the INs and carriers. HILT-SI needs to consume the large number of network bandwidth to maintain the process of fast discovery. In IDVD, the carriers only inquire the small number of system members and detect the mobile nodes in one-hop range. Moreover, the token-based message exchange strategy also reduces the message exchange between carriers. Therefore, IDVD's message cost can maintain lower level than that of HILT-SI. By regulating the values of p and q to change the range of IN discovery, IDVD can control the range of resource dissemination and adapt dynamic network environment.

Average Data Transmission Delay. We calculate the transmission delay of received video data at application layer during each time slice according to

$$\overline{d} = \frac{\sum_{c=1}^{k} d_c}{t_s}, \tag{13}$$

where t_s is the time slice, k is the number of all received data during a time slice, d_c denotes the delay of received cth data, and $\sum_{c=1}^{k} d_c$ is the sum of delay of all received data during a time slice. In terms of the settings of simulation time and the defined strategies of requesting resources, the value of t_s is set to 20 s.

As Figure 5 shows, HILT-SI's blue curve experiences a slight fluctuation from $t = 20$ s to $t = 80$ s and fast decreases from $t = 140$ s to $t = 200$ s after suddenly reaching the peak value 3.3 s at $t = 120$ s. The red curve corresponding to IDVD's results fast reaches the peak value 2.89 s at $t = 120$ s after having a slow increase from $t = 20$ s to $t = 80$ s and it falls from $t = 140$ s to $t = 200$ s. IDVD's delay is roughly 20% better than the values associated with HILT-SI.

In HILT-SI, the carriers and INs fetch the information of nodes from the broadcast messages. They make use of logical connection with the INs to push the video content. HILT-SI does not consider the geographical location relationship between carriers and INs, so that the average data transmission delay maintains higher level than that of IDVD. Moreover, when the large number of request suddenly arrives, the nodes do not assume huge network traffic so

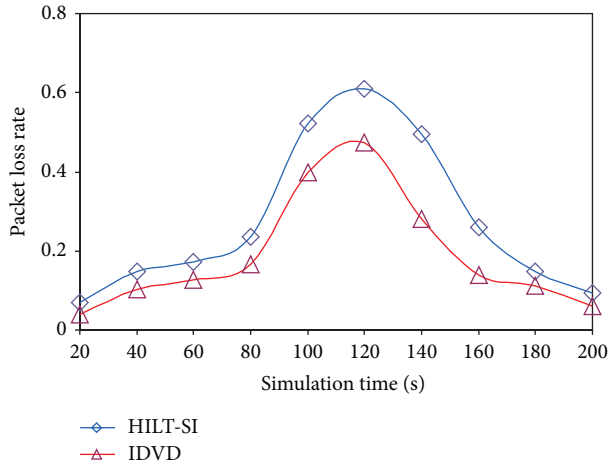


Figure 6: PLR against simulation time.

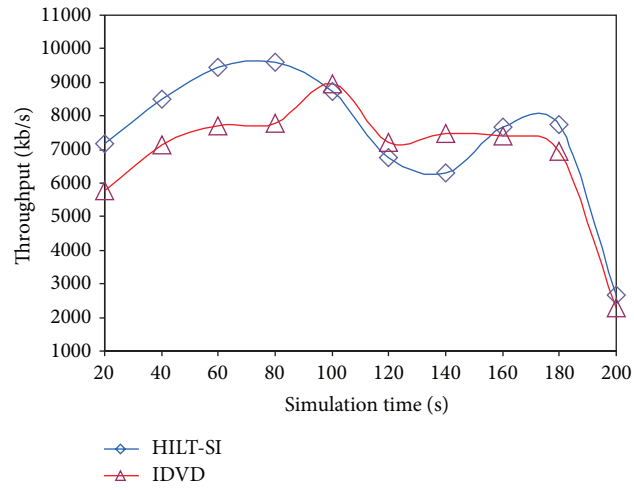


Figure 7: Throughput against simulation time.

as to result in the network congestion from $t = 100$ s to $t = 140$ s. Therefore, HILT-SI's delay is higher than that of IDVD. In IDVD, the members and mobile nodes are aware of resource information by receiving push messages of inquirers and cooperative inquirers. They fetch the video content from neighbor carriers. The local resource dissemination and the small number of video streaming relative to HILT-SI (the number of INs in IDVD is less than that of HILT-SI) do not consume more bandwidth of other relay nodes. Therefore, IDVD's peak value is less than that of HILT-SI and the time of duration of network congestion is shorter than that of HILT-SI.

Packet Loss Rate (PLR). The ratio between the number of packets lost in the process of video data transmission and the total number of packets of video data sent is defined as PLR.

As Figure 6 shows, the curves corresponding to HILT-SI and IDVD show a fall after rise with increasing simulation time. The results of HILT-SI and IDVD maintain low levels from $t = 20$ s to $t = 80$ s and fast increase from $t = 100$ s to $t = 120$ s and reach the peak values, respectively. The PLR values of HILT-SI and IDVD fast decrease from $t = 140$ s to $t = 200$ s. IDVD's PLR values are roughly 15% lower than those of HILT-SI.

The small number of system members fetching the video content only consumes less bandwidth, so the PLR values of HILT-SI and IDVD show slow increase from $t = 20$ s to $t = 80$ s. With sudden arrival of mass resource request, the high requirement of network bandwidth introduces the network congestion, so that HILT-SI and IDVD have high PLR from $t = 100$ s to $t = 120$ s. When the carriers constantly quit the system, the decreasing network traffic alleviates the congestion level. The PLR values of HILT-SI and IDVD fast decrease from $t = 140$ s to $t = 200$ s. In HILT-SI, the video data transmission relies on the logical link between carriers and INs; namely, the geographical distance of the communicating parties cannot be considered. The long-distance delivery of video data consumes the large number of bandwidth of relay nodes. Moreover, the more number of INs requires much network bandwidth. In IDVD, the carriers disseminate the message containing the information of video resources in neighbor geographical area. When the INs receive the resource information, they can download the video content from their neighbor carriers. The data only is forwarded by less relay nodes. Moreover, the small number of INs fetching video content cannot consume more network bandwidth. The congestion level of IDVD is lower than that of HILT-SI, so the PLR values of IDVD are less than those of HILT-SI.

Average Throughput. The total number of packets received in the overlay during a certain time period divided by the length of this time period is defined as the average throughput.

As Figure 7 shows, the average throughput curve of HILT-SI experiences severe fluctuation, which fast increases from $t = 20$ s to $t = 80$ s, immediately decreases from $t = 80$ s to $t = 140$ s, and finally has a fall after rise from $t = 160$ s to $t = 200$ s. The curve corresponding to IDVD results fast increases from $t = 20$ s to $t = 100$ s, reaches the peak value at $t = 100$ s, and slowly decreases from $t = 120$ s to $t = 200$ s.

The more number of INs introduces the transmission of much video streaming data in network. The throughput of HILT-SI fast increases from $t = 20$ s to $t = 80$ s. When the intensively arrival of mass resource request leads to the network congestion, the throughput of HILT-SI fast decreases due to the increase in PLR. With the increase in the number of carriers leaving the system, the decreasing congestion level enables the throughput rise. When the large number of carriers quits the system, the throughput fast falls. The more number of streaming and long-distance delivery result in high congestion level, so the throughput of HILT-SI severely jitters. In IDVD, the small number of data transmission requirement and the neighbor distance between nodes only introduce low-level congestion. When the congestion occurs at $t = 100$ s, the throughput of IDVD reaches the peak value. The congestion influence of IDVD is lighter than that of HILT-SI.

5. Conclusion

In this paper, we propose a novel interest detection-based video dissemination algorithm under flash crowd in mobile ad hoc networks (IDVD). IDVD prevents the degradation of QoS and network congestion caused by large-scale sudden request for popular video content. IDVD constructs an "H" model to build the categories of user request according to the popularities of video content and predict the amount demanded of upload bandwidth and period time of sudden request. The proposed resource dissemination algorithm formulates the area coverage of interested node discovery and resource dissemination and defines the convergence condition of spreading resources according to the epidemic model. The results show how IDVD obtains better performance than HILT-SI.

Notations

The Symbols Used in the Epidemic Model

N_t: The total number of inquired nodes
N_c: The number of nodes which store f_i
N_{IN}: The total number of INs (IMNs and IMs)
N_{um}: The number of uninterested members
I: The ratio of N_c and N_t
S: The ratio of N_{IN} and N_t
R: The ratio of N_{um} and N_t
I_0: The initial value of I
S_0: The initial value of S
λ: The spreading rate
μ: The recovery rate.

Conflict of 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 no. 61372112; the Beijing Natural Science Foundation (4142037); the key project of science and technology of Henan province (142102210473); the major projects of science and technology research of education department of Henan province (14B520016).

References

[1] D. C. Karia and V. V. Godbole, "New approach for routing in mobile ad-hoc networks based on ant colony optimisation with global positioning system," *IET Networks*, vol. 2, no. 3, pp. 171–180, 2013.

[2] L. Zhou, R. Q. Hu, Y. Qian, and H.-H. Chen, "Energy-spectrum efficiency tradeoff for video streaming over mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 5, pp. 981–991, 2013.

[3] Y. Zhou, T. Z. J. Fu, and D. M. Chiu, "On replication algorithm in P2P VoD," *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 233–243, 2013.

[4] 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*, p. 1, 2015.

[5] O. B. Maia, H. C. Yehia, and L. de Errico, "A concise review of the quality of experience assessment for video streaming," *Computer Communications*, vol. 57, pp. 1–12, 2015.

[6] Y. Xiao and M. V. D. Schaar, "Optimal foresighted multi-user wireless video," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 1, pp. 89–101, 2015.

[7] 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.

[8] Y. Chang and M. Kim, "Binocular suppression-based stereoscopic video coding by joint rate control with KKT conditions for a hybrid video codec system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 1, pp. 99–111, 2015.

[9] L. Zhou, Z. Yang, H. Wang, and M. Guizani, "Impact of execution time on adaptive wireless video scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 760–772, 2014.

[10] D. Bethanabhotla, G. Caire, and M. Neely, "Adaptive video streaming for wireless networks with multiple users and helpers," *IEEE Transactions on Communications*, vol. 63, no. 1, pp. 99–111, 2015.

[11] 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.

[12] 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.

[13] G. Zhang, W. Liu, X. Hei, and W. Cheng, "Unreeling Xunlei Kankan: understanding hybrid CDN-P2P video-on-demand streaming," *IEEE Transactions on Multimedia*, vol. 17, no. 2, pp. 229–242, 2015.

[14] 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.

[15] D. Wu, H. Liu, Y. Bi, and H. Zhu, "Evolutionary game theoretic modeling and repetition of media distributed shared in P2P-based VANET," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 718639, 14 pages, 2014.

[16] W.-T. Lo, Y. S. Chang, R.-K. Sheu, C.-T. Yang, T.-Y. Juang, and Y.-S. Wu, "Implementation and evaluation of large-scale video surveillance system based on P2P architecture and cloud computing," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 375871, 11 pages, 2014.

[17] H. Xu, H.-C. Huang, R. Wang, and J. Dong, "Peer selection strategy using mobile agent and trust in peer-to-peer streaming media system," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 791560, 15 pages, 2013.

[18] Y. Chen, B. Zhang, C. Chen, and D. M. Chiu, "Performance modeling and evaluation of peer-to-peer live streaming systems under flash crowds," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1106–1120, 2014.

[19] F. Liu, B. Li, L. Zhong, H. Jin, and X. Liao, "Flash crowd in P2P live streaming systems: fundamental characteristics and design implications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1227–1239, 2012.

[20] C. Carbunaru, Y. M. Teo, B. Leong, and T. Ho, "Modeling flash crowd performance in peer-to-peer file distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2617–2626, 2014.

[21] 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.

[22] J. Kim and S. Bahk, "PECAN: peer cache adaptation for peer-to-peer video-on-demand streaming," *Journal of Communications and Networks*, vol. 14, no. 3, pp. 286–295, 2012.

[23] K. Mokhtarian and M. Hefeeda, "Capacity management of seed servers in peer-to-peer streaming systems with scalable video streams," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 181–194, 2013.

[24] S. Venkatramanan and A. Kumar, "Co-evolution of content popularity and delivery in mobile P2P networks," in *Proceedings of the IEEE INFOCOM*, 2012.

[25] E. Altman, P. Nain, A. Shwartz, and Y. Xu, "Predicting the impact of measures against P2P networks: transient behavior and phase transition," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 935–949, 2013.

[26] L. Guo, S. Chen, and X. Zhang, "Design and evaluation of a scalable and reliable P2P assisted proxy for on-demand streaming media delivery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 5, pp. 669–682, 2006.

[27] J. Choi, A. S. Reaz, and B. Mukherjee, "A survey of user behavior in VoD service and bandwidth-saving multicast streaming schemes," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 1, pp. 156–169, 2012.

[28] W.-P. K. Yiu, X. Jin, and S.-H. G. Chan, "VMesh: distributed segment storage for peer-to-peer interactive video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1717–1731, 2007.

[29] 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.

[30] L. Tu and C.-M. Huang, "Collaborative content fetching using MAC layer multicast in wireless mobile networks," *IEEE Transactions on Broadcasting*, vol. 57, no. 3, pp. 695–706, 2011.

[31] J. Deng, "Grey linear programming," in *Proceedings of the International Conference Information Processing Management Uncertainty Knowledge-Based System*, 1986.

[32] J. L. Deng, "Introduction to grey system theory," *The Journal of Grey System*, vol. 1, no. 1, pp. 1–24, 1989.

[33] 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.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 238386, 13 pages
http://dx.doi.org/10.1155/2015/238386

Research Article

A Novel Energy-Efficient Reception Method Based on Random Network Coding in Cooperative Wireless Sensor Networks

Yulun Cheng[1,2] and Longxiang Yang[1,2]

[1] *Jiangsu Key Lab of Wireless Communications, Nanjing University of Posts & Telecommunications, Nanjing 210003, China*
[2] *Key Lab of Broadband Wireless Communication & Sensor Network Technology, Nanjing University of Posts & Telecommunications, Ministry of Education, Nanjing 210003, China*

Correspondence should be addressed to Longxiang Yang; allennupt@qq.com

Received 15 January 2015; Revised 25 March 2015; Accepted 26 March 2015

Academic Editor: Gabriel-Miro Muntean

This paper presents an opportunistic reception (OR) algorithm for energy-efficient transmission in cooperative wireless sensor networks (WSNs), where the characteristics of random linear network coding and the energy consumption property of WSNs are jointly considered. In OR, the sensor nodes in intermediate cluster generate the independent coding vector through simple forwarding or decoding-recoding manners opportunistically, so that the number of received packets can be reduced. To evaluate the algorithm, we derive the average number of received packets, the transmitting nodes, and decoding failure probability under specific assumption. The obtained theoretical and simulation results verify the effectiveness offered by the proposed approach.

1. Introduction

Due to the wide range of applications, wireless sensor networks (WSNs) have attracted numerous research interests in recent years, such as target tracking [1], military surveillance [2], and environment monitoring [3]. In most cases, the sensor nodes in WSNs are equipped with low-complexity hardware and single antenna, which restrain their communication quality in wireless fading channels. Therefore, transmission reliability becomes one of the key issues in the applications of WSNs.

Cooperative communication (CC) [4] is one approach to improve transmission reliability of single-antenna devices. A transmitting node that uses CC can share its packet with neighboring nodes, and then these nodes can transmit the packet to the intended receiver cooperatively, thereby creating a virtual multiple-input-multiple-output (MIMO) system. The intended receiver can obtain diversity gains by combining the received signals, which brings a signal-to-noise ratio (SNR) advantage over the single-input-single-output (SISO) case. Several works have focused on the applications of CC in WSNs (cooperative WSNs); for example, Li et al. [5] proposed a space time block code- (STBC-) based cooperative scheme to improve the bit error rate (BER), while Cui et al. [6] employed CC to reduce the transmission energy of the sensor nodes. All these cooperative schemes require strict synchronization between the nodes; however, due to the inexpensive hardware and limited resources, strict synchronization is difficult to be realized in WSNs, which deteriorates the performance of CC [7].

On the other hand, by means of combining the incoming packets at intermediate nodes in the networks, network coding (NC) [8] brings a breakthrough to the transmission efficiency. As a class of NC, Li et al. demonstrated that the network throughput can achieve the max-flow-min-cut bound through linear network coding (LNC) [9]. Furthermore, Koetter and Médard [10] presented an algebraic framework to construct the NC coefficient for LNC. On the basis of that, random LNC (RLNC) [11–13] was proposed to reduce the complexity and make LNC to be deployed as the distributed manner. RLNC enables the intermediate nodes to select combination coefficient randomly from a Galois field of size q, while the incoming packets are also treated and combined as a vector over this field. In this way, RLNC provides a simple,

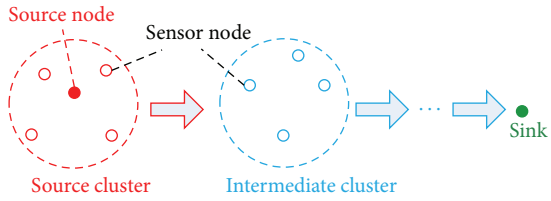


Figure 1: System model of cooperative WSNs.

yet effective, approach to improve the latency [14] and delay-tolerance [15] in wireless networks.

The distributed deployment and simplicity of RLNC loose the synchronization requirement, which makes it very suitable to be deployed in cooperative WSNs. Recent work [16] has employed RLNC to increase the redundancy of the original packets, so that the transmission reliability of the cooperative WSNs can be improved. In [16], the intermediate node tries to receive and combine all the incoming packets to construct its linear independent coding vector (CV). However, since the sensor nodes are usually deployed in hostile circumstances and powered by limited batteries, energy consumption [17, 18] is another important issue in cooperative WSNs. For popular sensor transceiver today, the receiving circuit energy consumption of a packet is even larger than the transmitting one, because decoding is a rather complex operation requiring a lot of computing power, which has been a big threat to the lifetime of the networks [19]. Therefore, it indicates that, for cooperative WSNs, not only transmission reliability [16] but also energy consumption characteristic should be taken into consideration in RLNC scheme design.

In this paper, we propose an opportunistic reception (OR) algorithm to reduce the received packets at the intermediate nodes. Different from the former work [16], by considering some characteristics of RLNC, our OR algorithm enables the intermediate sensor nodes to generate independent CV through simple forwarding or decode-and-select manners opportunistically. To examine its performance, the average number of received packets, the transmitting nodes, and decoding failure probability are derived under specific assumption. In this way, our study provides a trade-off between energy efficiency and transmission reliability for cooperative WSNs.

The remainder of this paper is organized as follows. Section 2 describes the system model. The proposed OR algorithm is presented in Section 3, and the corresponding performances are theoretically analyzed in Section 4. Simulation results are presented and compared for performance evaluation in Section 5. Finally, Section 6 concludes the paper.

2. System Model

We consider multihop clustered WSNs, in which a source node in the source cluster sends data to a sink with the aid of several intermediate clusters, as depicted in Figure 1. Each cluster is composed of n sensor nodes, and the operation of the system is broken into rounds. It is assumed that the source node has data to send in each round.

The transmission round consists of four phases, as shown in Figure 2. Phase I is the intrasource cluster broadcasting, as shown in Figure 2(a); the source node splits the original data into m packets as $\mathbf{P} = (P_1, \ldots, P_k, \ldots, P_m)^T$ and then broadcasts it to the other nodes in the source cluster, named as r_{Si}, $i = 1, \ldots, n$. It is assumed that the intracluster communication is error-free, so r_{Si} randomly selects a $1 \times m$ CV $\mathbf{V}_{Si} = (V_{Si}^{(1)}, \ldots, V_{Si}^{(k)}, \ldots, V_{Si}^{(m)})$ and combines all the incoming packets to generate its data as

$$P_{Si} = \mathbf{V}_{Si} \cdot \mathbf{P}. \tag{1}$$

Note that P_k, $V_{Si}^{(k)}$, and P_{Si} are the elements over Galois field of size q and $V_{Si}^{(k)}$ is randomly selected with probability $1/q$.

Phase II is the source-intermediate cluster transmission, as depicted in Figure 2(b). In this phase, each r_{Si} transmits its outgoing packet to the intermediate cluster in the next hop. Note that the outgoing packet encapsulates both CV $\mathbf{V}_{Si}$ and data P_{Si}. Each sensor node in the intermediate cluster, named as r_{Ii}, tries to receive all the packets [16] from the source cluster. After that, it recodes all the successfully received data as

$$P_{Ii} = \mathbf{A}_{Ii} \cdot \left[\ldots, P_{Sj}, \ldots, P_{Sk}\right]^T. \tag{2}$$

$\mathbf{A}_{Ii} = (\ldots, A_{Ii}^{(j)}, \ldots, A_{Ii}^{(k)})$ is also a randomly selected L-length vector over Galois field, where L is the number of successfully received packets at r_{Ii}. With (1), we rewrite the coded data P_{Ii} as the product of a CV and $\mathbf{P}$:

$$P_{Ii} = \mathbf{V}_{Ii} \cdot \mathbf{P}, \tag{3}$$

where $\mathbf{V}_{Ii} = \mathbf{A}_{Ii} \cdot [\ldots, \mathbf{V}_{Sj}^T, \ldots, \mathbf{V}_{Sk}^T]^T$. After that, each r_{Ii} reencapsulates $\mathbf{V}_{Ii}$ and P_{Ii} into its outgoing packet, and the recoding and reencapsulation procedure is shown in Figure 3.

Phase III is the inter-intermediate clusters communications, as shown in Figure 2(c). In this phase, if r_{Ii} fails to receive all the incoming packets in Phase II, it keeps silent. Otherwise, r_{Ii} sends its recapsulated packet to the next cluster. The remaining intermediate clusters perform the operation in the same manner as the one in Figure 2(b), so the coded packets are delivered one by one, until the last cluster near the sink.

Phase IV is the decoding phase, in which the sink tries to receive all the packets from the last cluster nearby, as shown in Figure 2(d). We assume that

$$\mathbf{P}_D = \left(P_{I'1}, \ldots, P_{I'k}, \ldots, P_{I'w}\right)^T \tag{4}$$

is the w received coded data, and

$$\mathbf{V}_D = \left[\mathbf{V}_{I'1}^T \ldots, \mathbf{V}_{I'k}^T, \ldots, \mathbf{V}_{I'w}^T\right]^T \tag{5}$$

is the w received CVs, in which $P_{I'k}$ and $\mathbf{V}_{I'k}$ are extracted from the packet of $r_{I'k}$. Thus, (4) can be rewritten as

$$\mathbf{P}_D = \mathbf{V}_D \cdot \mathbf{P}, \tag{6}$$

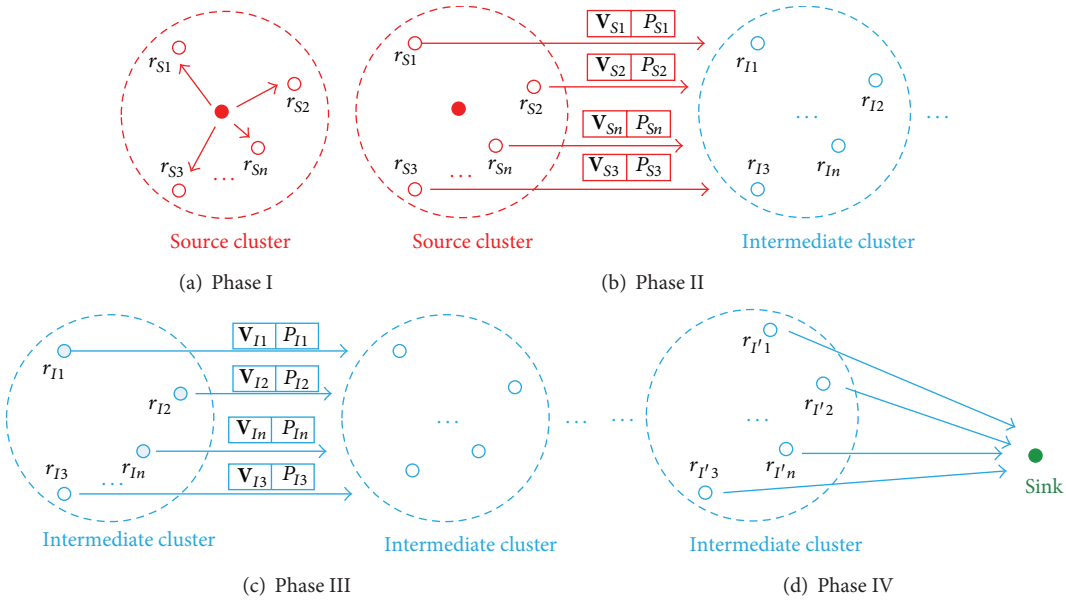


Figure 2: Transmission procedure of cooperative WSNs.

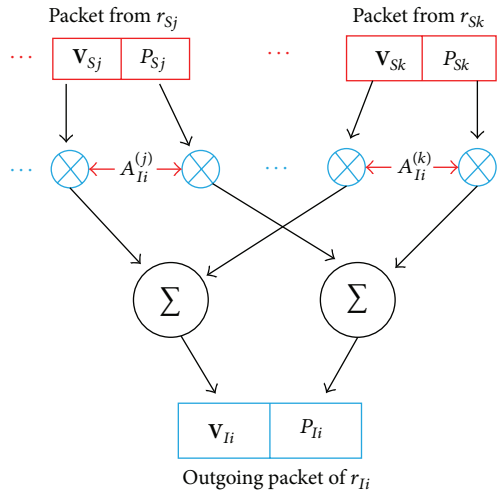


Figure 3: Recoding and recapsulation procedure.

and if no less than m out of the CVs in $\mathbf{V}_D$ are linearly independent, the original data $\mathbf{P}$ can be recovered through Gaussian elimination [11]. Otherwise, the decoding failure occurs at sink.

3. Opportunistic Reception Algorithm

To facilitate the analysis, we simplify the above model to a two-hop one, which is the basic building block of more complex multihop networks. Firstly, according to the principle of RLNC [11], it should be confirmed that the only purpose of combining all the incoming packets at the sensor nodes [16] is nothing else but generating linearly independent CV. From this perspective, the linear combination in (3) can be replaced by the following two manners. (1) Forwarding: since $\mathbf{V}_{Si}$ and $\mathbf{V}_{Sj}$ are randomly selected by r_{Si} and r_{Sj}, $\forall i, j \in \{1, \ldots, n\}$, $i \neq j$, it is naturally guaranteed that they are linearly independent of each other at high probability; in other words, they can be directly employed as the CV of the sensor nodes in intermediate cluster. For example, r_{I1} and r_{I2} can simply forward the packets from r_{S1} and r_{S2}, respectively, without receiving and combining other packets anymore. (2) Decoding-recoding: if r_{Ii} successfully receives m packets, the received data can be written as

$$\mathbf{P}_I = \mathbf{V}_S \cdot \mathbf{P}, \tag{7}$$

in which

$$\begin{aligned} \mathbf{P}_I &= \left(P_{I1}, \ldots, P_{Ik}, \ldots, P_{Im}\right)^T, \\ \mathbf{V}_S &= \left[\mathbf{V}_{S1}^T, \ldots, \mathbf{V}_{Sk}^T, \ldots, \mathbf{V}_{Sm}^T\right]^T. \end{aligned} \tag{8}$$

r_{Ii} can decode $\mathbf{P}$ by solving (7), and, then, it locally recodes $\mathbf{P}$ as

$$P_{Ii} = \mathbf{A}_{Ii} \cdot \mathbf{P}. \tag{9}$$

In this way, r_{Ii} successfully generates linearly independent CV while further reception is also avoided.

By considering the above two characteristics, we propose a reception algorithm for the nodes in intermediate cluster to reduce the received packets. Let L_j be the number of successfully received packets at r_{Ij}. Φ_U is the index set of the nodes which fail to generate independent CV, while Φ_{S1} and Φ_{S2} are those which generate independent CV due to forwarding and decoding-recoding, respectively. F_j is the flag to identify whether r_{Ij} is selected from Φ_{S1}, through the value *true* or *false*. Thus, the pseudocode description of the proposed algorithm is presented in Algorithm 1.

We propose an example to illustrate the effectiveness of OR algorithm. As depicted in Figure 4, an OR process is

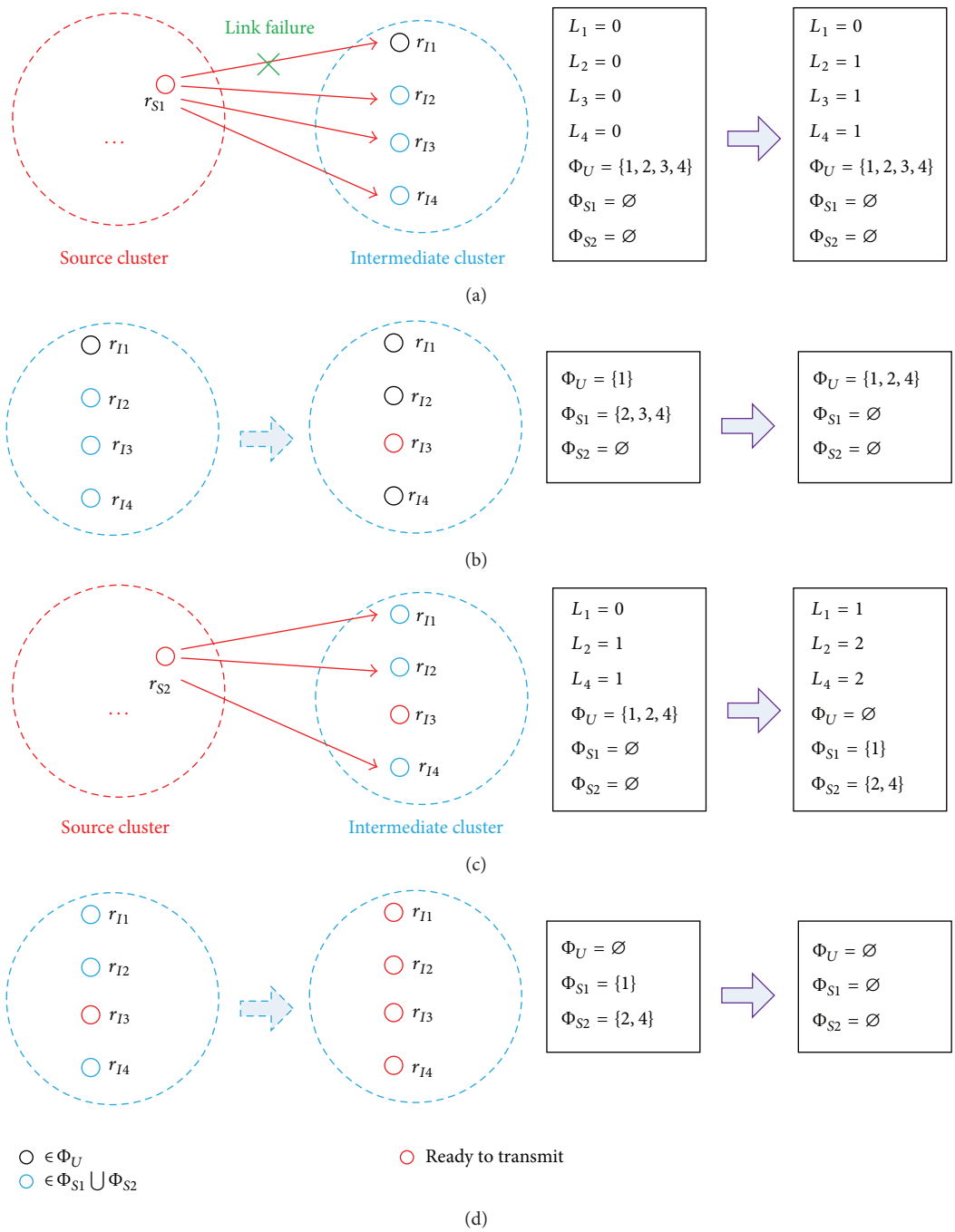


Figure 4: An example of OR process.

presented when $m = 2$, $n = 4$. In Figure 4(a), r_{S1} successfully broadcasts its packet to all the nodes in intermediate cluster, except for r_{I1}, which experiences link failure. According to the reception state, L_i is updated, $i = 1, 2, 3, 4$, while indexes 2, 3, and 4 are removed from Φ_U to Φ_{S1}. Figure 4(b) depicts the node selection in Φ_{S1}; that is, one of the nodes in Φ_{S1} is selected to generate its CV by the manner of forwarding. Here, we assume that index 3 is selected, so it is deleted while the others are removed from Φ_{S1} to Φ_U. Besides, r_{I3} is ready to transmit and will not receive packets anymore. In Figure 4(c), r_{S2} broadcasts and all the nodes in intermediate cluster try to receive, except for r_{I3}. Similarly, L_i is updated and index i is classified into Φ_{S1} or Φ_{S2} according to the rules of OR. In Figure 4(d), since index 1 is the only element of Φ_{S1}, r_{I1} is selected to forward the packet of r_{S2}. Meanwhile, because $\Phi_{S2} = \{2, 4\}$, r_{I2} and r_{I4} will generate their CVs by the decoding-recoding. Thus, all the nodes are ready to transmit, without further reception from r_{S3} and r_{S4}. In this example,

Initialization: $\Phi_U = \{1, \ldots, n\}$, $\Phi_{S1} = \Phi_{S2} = \varnothing$. $L_j = 0$, $F_j = true$, $\forall j \in \Phi_U$.
(1) **for** $i = 1$ to n,
(2) r_{Si} transmits its packet.
(3) **for** $\forall j \in \Phi_U$,
(4) r_{Ij} receives and updates L_j.
(5) **if** $L_j == 1$ && $F_j = true$,
(6) j is removed from Φ_U to Φ_{S1}.
(7) **elseif** $L_j == m$ && $F_j = false$,
(8) j is removed from Φ_U to Φ_{S2}.
(9) **end if**
(10) **end for**
(11) Select an element w from Φ_{S1} with probability $1/|\Phi_{S1}|$,
(12) r_{Iw} forwards its received packet in Phase III,
(13) $\Phi_{S1} = \Phi_{S1} - \{w\}$.
(14) **for** $\forall j \in \Phi_{S1}$,
(15) $F_j = false$, j is removed from Φ_{S1} to Φ_U.
(16) **end for**
(17) **for** $\forall j \in \Phi_{S2}$,
(18) $\Phi_{S2} = \Phi_{S2} - \{j\}$,
(19) r_{Ij} decodes and recodes **P** by (7) and (9),
(20) r_{Ij} transmits the packet in Phase III.
(21) **end for**
(22) **end for**

Algorithm 1: OR algorithm.

$1 + 2 \times 3 = 7$ packets are received in total. Comparing with $4 \times 4 = 16$ packets by the manner of combination [16], the merit of OR is revealed.

4. Performance Analysis

To evaluate the performance of OR, we derive the mathematical expectation of the received packets, the transmitting nodes, and the decoding failure probability when $m = 2$, $n = 4$. (In fact, it is difficult to exhaust all the cases with large values of m and n. However, the low parameters do not influence the evaluation result of this paper. On the other hand, it has been proved in [20] that $m = 2$ is optimal for NC and $n = 4$ can also be used to describe the cluster with many nodes, in which 4 are alive while the others are asleep.) For simplicity, it is assumed that all incoming links at any sensor node exhibit the same value of failure probability P.

4.1. Mathematical Expectations of the Received Packets and Transmitting Nodes. In OR algorithm, r_{Ij} is able to generate independent CV only in the following 2 cases: (1) index $j \in \Phi_{S1}$ as well as j is selected from Φ_{S1}; (2) index $j \in \Phi_{S2}$. Let V and VV denote these two events, respectively, and we denote d as the number of the sensor nodes that generate independent CV. In this way, CV generation can be presented by a d-size vector which consists of V and VV. For example, (V, VV) represents the event that in total 2 sensor nodes generate independent CV through cases (1) and (2), respectively. Thus, all cases of CV generations can be described by such vectors, as shown in Table 1.

In Table 1, when $d = 1$, a possible sample of (V) can be written as (V_1, O, O, O). These 4 elements represent the reception situation of the 4 sensor nodes in intermediate cluster after Phase II, where V_1 denotes that a node successfully receives the packet from r_{S1}, while O represents that a node fails to receive all the incoming packets. To derive the probability of each case, we introduce the following lemma.

Table 1: All cases of CV generations.

d	Vector description
1	(V)
2	(V, V), (V, VV)
3	(V, V, V), (V, V, VV), (V, VV, VV)
4	(V, V, V, V), (V, V, V, VV), (V, V, VV, VV), (V, VV, VV, VV)

Lemma 1. *In the proposed OR algorithm, if the CV generation is represented by a vector consisting of only one element V, then the sensor nodes that fail to generate independent CV can successfully receive at most $m - 1$ packets.*

Proof. If r_{Ii} successfully receives m packets, the index i will be included in Φ_{S2} and r_{Ii} will generate CV by (7) and (9), so the corresponding vector description will contain the element VV, which contradict the hypothesis. □

With Lemma 1, it can be deduced that, in all samples of (V), the sensor nodes that fail to generate independent CV can receive at most 1 ($m = 2$) packet. Thus, all possible samples of (V) can be listed in Table 2.

Let P_i denote the probability of the event that $\{d = i\}$, $i = 0, 1, 2, 3, 4$. In Table 2, the event $\{d = 1\}$ can be divided

Table 2: All possible realizations of (V).

Case	Possible sample
1	(V_1, O, O, O), (V_2, O, O, O), (V_3, O, O, O), (V_4, O, O, O)
2	(V_1, V_1, O, O), (V_2, V_2, O, O), (V_3, V_3, O, O), (V_4, V_4, O, O)
3	(V_1, V_1, V_1, O), (V_2, V_2, V_2, O), (V_3, V_3, V_3, O), (V_4, V_4, V_4, O)
4	(V_1, V_1, V_1, V_1), (V_2, V_2, V_2, V_2), (V_3, V_3, V_3, V_3), (V_4, V_4, V_4, V_4)

into 4 cases; for each case, the probability is denoted as P_{1i}, so P_1 can be expressed as

$$P_1 = \Pr(d = 1) = \sum_{i=1}^{4} P_{1i}. \tag{10}$$

Each case is further divided, and P_{1i} can be written as

$$P_{1i} = \sum_{j} P_{1i}^{(j)}, \tag{11}$$

where $P_{1i}^{(j)}$ denotes the probability of the jth sample of case i in Table 2. Let N denote the total number of the received packets by intermediate cluster per transmission round, and its mathematical expectation can be written as

$$E(N) = \sum_{k=0}^{4} \sum_{i} \sum_{j} P_{ki}^{(j)} N_{ki}^{(j)}, \tag{12}$$

where $N_{ki}^{(j)}$ is the number of received packets in the corresponding sample. $E(N)$ can be used to evaluate the average energy consumption of the network by the reception per round.

We take $P_{12}^{(1)}$ as an example to show the derivation. As listed in Table 2, it can be derived as

$$\begin{aligned} P_{12}^{(1)} &= \Pr\left(\text{sample} = \left(V_1, V_1, O, O\right)\right) \\ &= \binom{4}{2} P^8 \binom{2}{1} (1-P) \left(\frac{1}{2}\right) (1-P) P^3, \end{aligned} \tag{13}$$

where $\binom{4}{2} P^8$ stands for the probability that 2 out of 4 nodes in intermediate cluster fail to receive all the incoming packets. In this sample, the remaining 2 nodes both successfully receive the packet from r_{S1}, and one of them is selected with probability 1/2 to forward this packet, so the factor $\binom{2}{1}$ in (13) stands for the above event. $(1 - P)P^3$ accounts for the probability that a node successfully receives the packet from r_{S1}, but it is selected to neither forward nor successfully receive the rest of incoming packets. Meanwhile, we can calculate $N_{12}^{(1)} = 1 + 4 + 4 + 4 = 13$ for this sample, since the node that is selected to simply forward receives only one packet, while the others receive all 4 incoming ones.

In Table 1, for the vector descriptions that only contain the element V, the possible samples and probabilities can be derived in similar way. For the ones that contain both V and VV, we take (V, VV) as an example. We denote V_i and V_iV_j as the samples of V and VV, respectively. Thus, the following lemma can be proved.

Table 3: All possible realizations of (V, VV).

Case	Possible sample
1	(V_1, V_1V_2, V_i, V_i), $\forall V_i \in \Phi_{31}$
2	(V_1, V_1V_3, V_i, V_i), $\forall V_i \in \Phi_{31}$
3	(V_1, V_1V_4, V_i, V_i), $\forall V_i \in \Phi_{31}$
4	(V_2, V_2V_3, V_i, V_i), $\forall V_i \in \Phi_{32}$
5	(V_2, V_2V_4, V_i, V_i), $\forall V_i \in \Phi_{32}$
6	(V_3, V_3V_4, V_i, V_i), $\forall V_i \in \Phi_{33}$

Lemma 2. *In OR algorithm, all the realizations of (V, VV) can be written as the form of* (V_i, V_iV_j), $\forall i, j \in \{1, 2, 3, 4\}$, $i \neq j$.

Proof. If a sample of (V, VV) can be expressed as (V_m, V_iV_j), $m \neq i$, then according to OR it can be concluded that, besides V_m and V_iV_j, the sample must contain the element V_i. This fact indicates that the realization is a case of (V, V, VV), which contradicts the assumption. □

With Lemma 2, all the samples of (V, VV) can be expressed as in Table 3, where $\Phi_{31} = \{O, V_1\}$, $\Phi_{32} = \{O, V_2\}$, and $\Phi_{33} = \{O, V_3\}$. The calculation is similar to the one of (V); for example, let $P_{21}^{(2)}$ denote the probability of (V_1, V_1V_2, V_1, O), and it can be written as

$$\begin{aligned} P_{21}^{(2)} &= \Pr\left(\text{sample} = \left(V_1, V_1V_2, V_1, O\right)\right) \\ &= \left(\frac{1}{3}\right) \binom{4}{1} \binom{3}{1} \binom{2}{1} (1-P)^4 P^7, \end{aligned} \tag{14}$$

while $N_{21}^{(2)} = 1 + 2 + 4 + 4 = 11$. In this way, all the possible samples and probabilities of (V, VV) can be derived, so as the ones that contain both V and VV in Table 1. By substituting these results into (12), $E(N)$ can be obtained. The derivations of the other possible samples and probabilities listed in Table 1 are presented in the appendix.

Similarly, the mathematical expectation of transmitting nodes $E(d)$ can be obtained as

$$E(d) = \sum_{i=0}^{4} d_i P_i, \tag{15}$$

where P_i is derived in the Appendix. $E(d)$ can be used to evaluate the average energy consumption of the network by the transmission per round.

4.2. Decoding Failure Probability. As defined in Section 2, the decoding failure probability can be expressed as

$$P_f = \Pr(s < m) = \sum_{i=0}^{n} \Pr(s < m \mid d = i) P_i, \tag{16}$$

where s denotes the number of independent CVs at sink. In [21], it has been proved that two randomly selected CVs are correlated with each other at very small probability (5.63×10^{-8}) when $q = 256$. In fact, small values of q increase the correlation of randomly selected CVs, which will dominate the decoding performance. On the other hand, too large value is also unnecessary, which increases the complexity. Hence, we assume that s equals the number of successfully received packets by sink. In our studied case, $m = 2$, $n = 4$, and it is assumed that all incoming links at sink exhibit the same failure probability P_{rd}. Thus, $\Pr(s < m \mid d = i)$ can be written as

$$\begin{aligned}
&\Pr(s < 2 \mid d = i) = 1, \quad i = 0, 1,\\
&\Pr(s < 2 \mid d = 2) = 1 - (1 - P_{rd})^2,\\
&\Pr(s < 2 \mid d = 3) = P_{rd}^3 + \binom{3}{1}(1 - P_{rd})P_{rd}^2, \qquad (17)\\
&\Pr(s < 2 \mid d = 4) = P_{rd}^4 + \binom{4}{1}(1 - P_{rd})P_{rd}^3.
\end{aligned}$$

By substituting (17) and P_i into (16), the decoding failure probability can be obtained.

5. Simulation Results

We compare OR with NC based cooperative communication (NCCC) in [16], in terms of $E(N)$, $E(d)$, lifetime, and P_f. The simulation model follows the simplified two-hop one in Section 3 with $m = 2$, $n = 4$, and $q = 256$. We define the lifetime as the time that the first node in intermediate cluster dies, which is widely used in the literatures. The energy model in [19] is employed, which has transmitting circuit energy of 45 nJ/packet and receiving circuit energy of 135 nJ/packet, and all the nodes have the initial energy of 500 mJ.

Figure 5 presents the mathematical expectation of received packet $E(N)$ conditioned to the link failure probability P. For OR, the theoretical values are calculated by (12). It shows that our theoretical analysis perfectly matches the simulation results, and OR always yields less received packets comparing with NCCC. In the figure, the theoretical and simulation results also confirm that, for OR, the maximum and minimum values of $E(N)$ are 16 and 7 when $P = 1$ and 0, respectively. These points show the performance of OR when the link is in extreme state, for the sample (O, O, O, O) corresponds to $N = 16$ while $(V_1, V_1V_2, V_1V_2, V_1V_2)$ corresponds to $N = 7$. Due to less reception, OR consumes less receiving circuit energy than NCCC.

Figure 6 shows the mathematical expectation of transmitting nodes $E(d)$ versus P. For OR, the theoretical values are calculated by (15). It shows that OR is close to NCCC when P is near 0, which indicates that these two algorithms supply similar redundancy for original data when the link is in good state. In brief, OR always enables less nodes to transmit under the same P. Combining this result with the one in Figure 5, we can conclude that OR consumes less energy than NCCC.

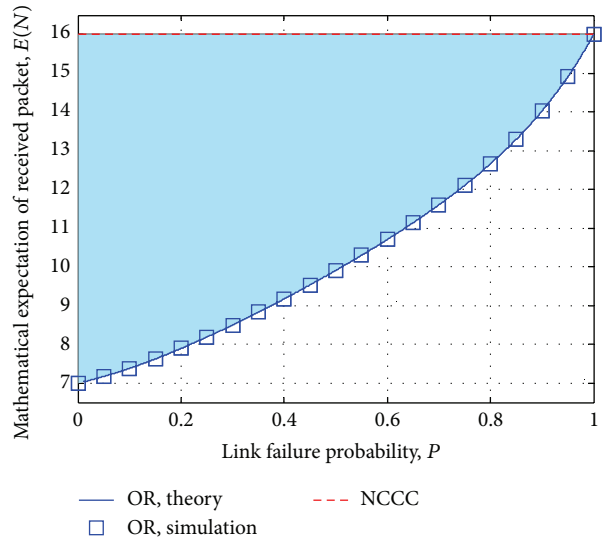


Figure 5: Mathematical expectation of received packet $E(N)$ versus P.

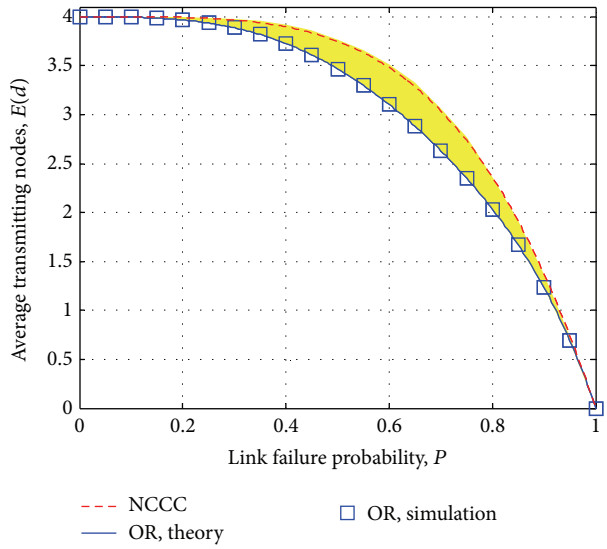


Figure 6: Mathematical expectation of transmitting nodes $E(d)$ versus P.

Figure 7 presents the lifetime comparisons, in which the minimal residual energy of the nodes is simulated after each round. The result shows that OR can support more transmission rounds than NCCC when the energy is exhausted. Besides, it also shows that when $P = 0.01$, OR performs better than $P = 0.3$. This result corresponds well with the one in Figure 5, because smaller value of P will yield less received packets, which dominate the energy consumption in the employed model.

Figure 8 depicts the gradient comparisons of the minimal residual energy presented in Figure 7. This metric essentially reflects the energy consumption per round. We observe that, for NCCC, the gradient is fixed to −585 nJ. This is due to the fact that each node will always receive 4 packets and transmit

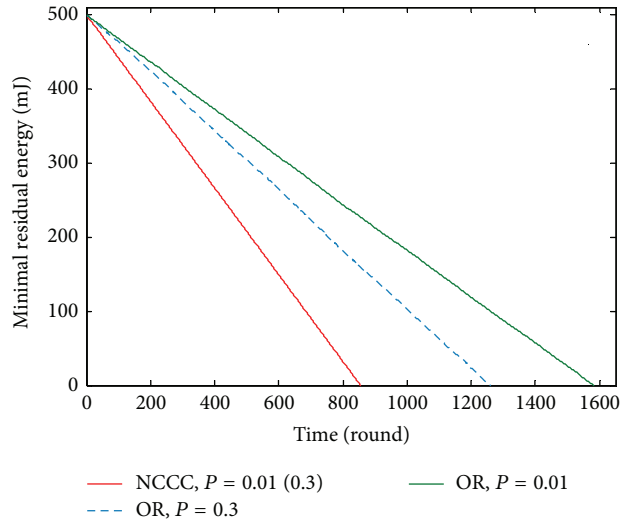


Figure 7: Lifetime comparisons.

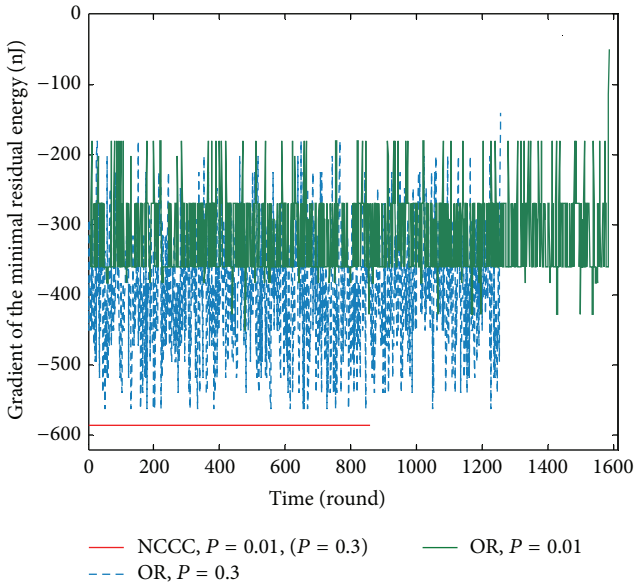


Figure 8: Gradient comparisons.

1 per round. Comparing with NCCC, the result shows that OR always yields less gradient, and it fluctuates with time. This is caused by the dynamic manner selection, through which the nodes opportunistically generate their CVs. Moreover, it also shows that superior link quality yields smaller gradient for OR.

Figure 9 depicts the decoding failure probability P_f versus P. It indicates that OR performs close to NCCC when P is near 0, which corresponds well with the result in Figure 6. Again, our theoretical analysis matches the simulations very well. On the other hand, the gap between the two compared algorithms grows as P_{rd} increases, which shows that OR is less efficient under poor relay-sink links. However, different from our simplified model, the sink generally locates in the sink cluster in practice and hence the other sensor nodes in

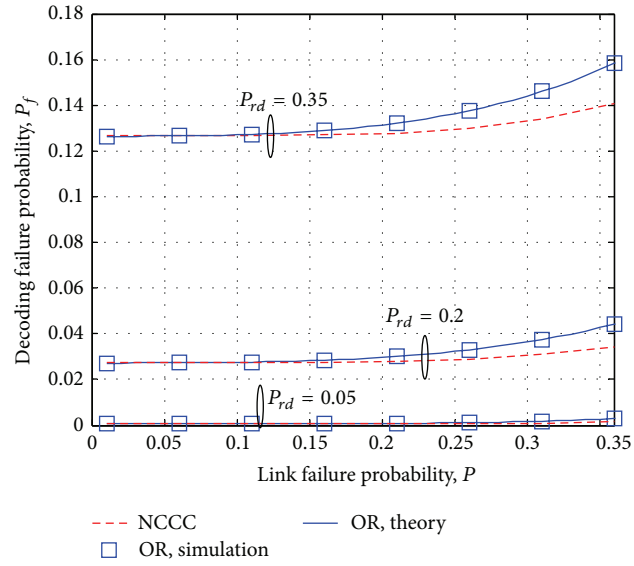


Figure 9: Decoding failure probability P_f versus P.

the sink cluster also help to receive, which yields superior reception performance even when P_{rd} is in poor state.

6. Conclusion

In this paper, we proposed an energy-efficient OR algorithm to reduce the energy consumption of the sensor nodes in cooperative WSNs. The theoretical and simulation results show that OR outperforms NCCC in energy consumption, while it also obtains similar decoding failure probability to that of NCCC when the incoming links are in good state. The proposed algorithm provides a trade-off between energy efficiency and transmission reliability.

Appendix

The Derivation of P_i, $i = 0, 1, 2, 3, 4$

According to the definition in Section 4, P_0 can be expressed as

$$P_0 = P^{16}. \tag{A.1}$$

All possible realizations of (V) are listed in Table 2, and the corresponding probabilities can be written as

$$\begin{aligned} P_{11} &= \binom{4}{3} P^{12}(1-P)\left(1+P+P^2+P^3\right), \\ P_{12} &= \frac{\binom{4}{2}\binom{2}{1} P^{11}(1-P)^2\left(1+P+P^2+P^3\right)}{2}, \\ P_{13} &= \frac{\binom{4}{1}\binom{3}{1} P^{10}(1-P)^3\left(1+P+2P^2\right)}{3}, \\ P_{14} &= \frac{\binom{4}{1} P^9(1-P)^4\left(1+P+P^2+P^3\right)}{4}. \end{aligned} \tag{A.2}$$

Substituting (A.2) into (10), P_1 can be obtained.

All possible realizations of (V, V) can be listed as in Table 4, in which $\Phi_{21} = \{V_1, V_2\}$, $\Phi_{22} = \{V_1, V_3\}$, $\Phi_{23} = \{V_1, V_4\}$, $\Phi_{24} = \{V_2, V_3\}$, $\Phi_{25} = \{V_2, V_4\}$, and $\Phi_{26} = \{V_3, V_4\}$, and the corresponding probabilities can be written as

$$
\begin{aligned}
P_{21} &= \binom{4}{2}\binom{2}{1}P^9(1-P)^2 \\
&\quad + \binom{4}{1}\binom{3}{1}\binom{2}{1}P^8(1-P)^3 \\
&\quad + \frac{7\binom{4}{1}\binom{3}{1}P^7(1-P)^4}{6}, \\
P_{22} &= \binom{4}{1}\binom{3}{1}P^{10}(1-P)^2 \\
&\quad + \binom{4}{1}\binom{3}{1}\binom{2}{1}P^9(1-P)^3 \\
&\quad + \frac{7\binom{4}{1}\binom{3}{1}P^8(1-P)^4}{6}, \\
P_{23} &= \binom{4}{1}\binom{3}{1}P^{11}(1-P)^2 \\
&\quad + \binom{4}{1}\binom{3}{1}\binom{2}{1}P^{10}(1-P)^3 \\
&\quad + \frac{7\binom{4}{1}\binom{3}{1}P^9(1-P)^4}{6}, \\
P_{24} &= \binom{4}{1}\binom{3}{1}P^{11}(1-P)^2 \\
&\quad + \binom{4}{1}\binom{3}{1}\binom{2}{1}P^{10}(1-P)^3 \\
&\quad + \frac{7\binom{4}{1}\binom{3}{1}P^9(1-P)^4}{6}, \\
P_{25} &= \binom{4}{1}\binom{3}{1}P^{12}(1-P)^2 \\
&\quad + \binom{4}{1}\binom{3}{1}\binom{2}{1}P^{11}(1-P)^3 \\
&\quad + \frac{7\binom{4}{1}\binom{3}{1}P^{10}(1-P)^4}{6}, \\
P_{26} &= \binom{4}{1}\binom{3}{1}P^{13}(1-P)^2 \\
&\quad + \binom{4}{1}\binom{3}{1}\binom{2}{1}P^{12}(1-P)^3 \\
&\quad + \frac{7\binom{4}{1}\binom{3}{1}P^{11}(1-P)^4}{6}.
\end{aligned}
\tag{A.3}
$$

Meanwhile, all the possible realizations of (V, VV) are listed in Table 3, and the corresponding probabilities can be expressed as

$$
\begin{aligned}
P_{27} &= \frac{\binom{4}{1}\binom{3}{1}P^8(1-P)^3}{2} \\
&\quad + \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^7(1-P)^4}{3} \\
&\quad + \frac{\binom{4}{1}\binom{3}{1}P^6(1-P)^5}{4}, \\
P_{28} &= \frac{\binom{4}{1}\binom{3}{1}P^9(1-P)^3}{2} \\
&\quad + \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^8(1-P)^4}{3} \\
&\quad + \frac{\binom{4}{1}\binom{3}{1}P^7(1-P)^5}{4}, \\
P_{29} &= \frac{\binom{4}{1}\binom{3}{1}P^{10}(1-P)^3}{2} \\
&\quad + \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^9(1-P)^4}{3} \\
&\quad + \frac{\binom{4}{1}\binom{3}{1}P^8(1-P)^5}{4}, \\
P_{210} &= \frac{\binom{4}{1}\binom{3}{1}P^{10}(1-P)^3}{2} \\
&\quad + \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^9(1-P)^4}{3} \\
&\quad + \frac{\binom{4}{1}\binom{3}{1}P^8(1-P)^5}{4}, \\
P_{211} &= \frac{\binom{4}{1}\binom{3}{1}P^{11}(1-P)^3}{2} \\
&\quad + \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^{10}(1-P)^4}{3} \\
&\quad + \frac{\binom{4}{1}\binom{3}{1}P^9(1-P)^5}{4},
\end{aligned}
$$

Table 4: All possible realizations of (V, V).

Case	Possible sample
1	$(V_1, V_2, O, O); (V_1, V_2, V_i, O), \forall V_i \in \Phi_{21}$; $(V_1, V_2, V_i, V_j), \forall V_i, V_j \in \Phi_{21}$;
2	$(V_1, V_3, O, O); (V_1, V_3, V_i, O), \forall V_i \in \Phi_{22}$; $(V_1, V_3, V_i, V_j), \forall V_i, V_j \in \Phi_{22}$;
3	$(V_1, V_4, O, O); (V_1, V_4, V_i, O), \forall V_i \in \Phi_{23}$; $(V_1, V_4, V_i, V_j), \forall V_i, V_j \in \Phi_{23}$;
4	$(V_2, V_3, O, O); (V_2, V_3, V_i, O), \forall V_i \in \Phi_{24}$; $(V_2, V_3, V_i, V_j), \forall V_i, V_j \in \Phi_{24}$;
5	$(V_2, V_4, O, O); (V_2, V_4, V_i, O), \forall V_i \in \Phi_{25}$; $(V_2, V_4, V_i, V_j), \forall V_i, V_j \in \Phi_{25}$;
6	$(V_3, V_4, O, O); (V_3, V_4, V_i, O), \forall V_i \in \Phi_{26}$; $(V_3, V_4, V_i, V_j), \forall V_i, V_j \in \Phi_{26}$;

Table 5: All possible realizations of (V, V, V).

Case	Possible sample
1	$(V_1, V_2, V_3, V_i), \forall V_i \in \{O, V_1, V_2, V_3\}$
2	$(V_1, V_2, V_4, V_i), \forall V_i \in \{O, V_1, V_2, V_4\}$
3	$(V_2, V_3, V_4, V_i), \forall V_i \in \{O, V_2, V_3, V_4\}$
4	$(V_1, V_3, V_4, V_i), \forall V_i \in \{O, V_1, V_3, V_4\}$

$$P_{212} = \frac{\binom{4}{1}\binom{3}{1}P^{12}(1-P)^3}{2} + \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^{11}(1-P)^4}{3} + \frac{\binom{4}{1}\binom{3}{1}P^{10}(1-P)^5}{4}. \tag{A.4}$$

According to the results in Table 1, P_2 can be written as

$$P_2 = \sum_{i=1}^{12} P_{2i}. \tag{A.5}$$

Substituting (A.3) and (A.4) into (A.5), P_2 can be calculated.

All possible realizations of (V, V, V) can be listed as in Table 5, and the corresponding probabilities can be written as

$$\begin{aligned} P_{31} &= \binom{4}{1}\binom{3}{1}\binom{2}{1}P^6(1-P)^3\left(P + \frac{3(1-P)}{2}\right), \\ P_{32} &= \binom{4}{1}\binom{3}{1}\binom{2}{1}P^7(1-P)^3\left(P + \frac{3(1-P)}{2}\right), \\ P_{33} &= \binom{4}{1}\binom{3}{1}\binom{2}{1}P^9(1-P)^3\left(P + \frac{3(1-P)}{2}\right), \\ P_{34} &= \binom{4}{1}\binom{3}{1}\binom{2}{1}P^8(1-P)^3\left(P + \frac{3(1-P)}{2}\right). \end{aligned} \tag{A.6}$$

On the other hand, all possible realizations of (V, V, VV) can be listed as in Table 6, in which $\Phi_{41} = \{O, V_1, V_2\}$, $\Phi_{42} = \{V_1V_2, V_1V_3, V_1V_4, V_2V_3, V_2V_4\}$, $\Phi_{43} = \{O, V_1, V_3\}$, $\Phi_{44} = \{V_1V_2, V_1V_3, V_1V_4, V_3V_4\}$, $\Phi_{45} = \{O, V_1, V_4\}$, $\Phi_{46} = \{V_1V_2, V_1V_3, V_1V_4\}$, $\Phi_{47} = \{O, V_2, V_3\}$, $\Phi_{48} = \{V_2V_3, V_2V_4, V_3V_4\}$, $\Phi_{49} = \{O, V_2, V_4\}$, $\Phi_{410} = \{V_2V_3, V_2V_4\}$, and $\Phi_{411} = \{O, V_3, V_4\}$. The corresponding probabilities can be expressed as

Table 6: All possible realizations of (V, V, VV).

Case	Possible sample
5	$(V_1, V_2, V_i, V_j), \forall V_i \in \Phi_{41}, \forall V_j \in \Phi_{42}$
6	$(V_1, V_3, V_i, V_j), \forall V_i \in \Phi_{43}, \forall V_j \in \Phi_{44}$
7	$(V_1, V_4, V_i, V_j), \forall V_i \in \Phi_{45}, \forall V_j \in \Phi_{46}$
8	$(V_2, V_3, V_i, V_j), \forall V_i \in \Phi_{47}, \forall V_j \in \Phi_{48}$
9	$(V_2, V_4, V_i, V_j), \forall V_i \in \Phi_{49}, \forall V_j \in \Phi_{410}$
10	$(V_3, V_4, V_3V_4, V_i), \forall V_i \in \Phi_{411}$

$$\begin{aligned} P_{35} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^5(1-P)^4\left(1+2P+2P^2\right)}{2} + \frac{7\binom{4}{1}\binom{3}{1}\binom{2}{1}P^4(1-P)^5\left(1+2P+2P^2\right)}{12}, \\ P_{36} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^6(1-P)^4\left(1+P+2P^2\right)}{2} + \frac{7\binom{4}{1}\binom{3}{1}\binom{2}{1}P^5(1-P)^5\left(1+P+2P^2\right)}{12}, \\ P_{37} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^7(1-P)^4\left(1+P+P^2\right)}{2} + \frac{7\binom{4}{1}\binom{3}{1}\binom{2}{1}P^6(1-P)^5\left(1+P+P^2\right)}{12}, \\ P_{38} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^8(1-P)^4(1+2P)}{2} + \frac{7\binom{4}{1}\binom{3}{1}\binom{2}{1}P^7(1-P)^5(1+2P)}{12}, \\ P_{39} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^9(1-P)^4(1+P)}{2} + \frac{7\binom{4}{1}\binom{3}{1}\binom{2}{1}P^8(1-P)^5(1+P)}{12}, \\ P_{310} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^{11}(1-P)^4}{2} + \frac{7\binom{4}{1}\binom{3}{1}\binom{2}{1}P^{10}(1-P)^5}{12}. \end{aligned} \tag{A.7}$$

Table 7: All possible realizations of (V, VV, VV).

Case	Possible sample
11	$(V_1, V_i,\ V_jV_k,\ V_jV_k), \forall V_i \in \{O, V_1\}$, $\forall V_jV_k \in \{V_1V_2, V_1V_3, V_1V_4\}$
12	$(V_1, V_i,\ V_1V_2, V_1V_3),\ \forall V_i \in \{O, V_1\}$
13	$(V_1, V_i,\ V_1V_2, V_1V_4),\ \forall V_i \in \{O, V_1\}$
14	$(V_1, V_i,\ V_1V_3, V_1V_4), \forall\ V_i \in \{O, V_1\}$
15	$(V_2, V_i,\ V_jV_k,\ V_jV_k),\ \forall V_i \in \{O, V_2\}, \forall V_jV_k \in \{V_2V_3, V_2V_4\}$
16	$(V_2, V_i,\ V_2V_3, V_2V_4),\ \forall V_i \in \{O, V_2\}$
17	$(V_3, V_i,\ V_3V_4, V_3V_4),\ \forall V_i \in \{O, V_3\}$

All possible realizations of (V, VV, VV) can be listed as in Table 7, and the corresponding probabilities can be written as

$$
\begin{aligned}
P_{311} &= \frac{\binom{4}{1}\binom{3}{1}P^3(1-P)^5\left(1+P^2+P^4\right)(3+P)}{12},\\
P_{312} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^4(1-P)^5(3+P)}{12},\\
P_{313} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^5(1-P)^5(3+P)}{12},\\
P_{314} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^6(1-P)^5(3+P)}{12},\\
P_{315} &= \frac{\binom{4}{1}\binom{3}{1}P^6(1-P)^5\left(1+P^2\right)(3+P)}{12},\\
P_{316} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^7(1-P)^5(3+P)}{12},\\
P_{317} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^9(1-P)^5(3+P)}{12}.
\end{aligned}
\tag{A.8}
$$

From the results of Table 1, it can be observed that the event $d = 3$ can be described by (V, V, V), (V, V, VV), and (V, VV, VV), so P_3 can be written as

$$P_3 = \sum_{i=1}^{17} P_{3i}. \tag{A.9}$$

Substituting (A.6), (A.7), and (A.8) into (A.9), P_3 can be obtained.

For (V, V, V, V), the only possible realization is (V_1, V_2, V_3, V_4), so the probability can be written as

$$P_{41} = \binom{4}{1}\binom{3}{1}\binom{2}{1}P^6(1-P)^4. \tag{A.10}$$

All possible realizations of (V, V, V, VV) are listed in Table 8, and the corresponding probabilities can be expressed as

$$
\begin{aligned}
P_{42} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^3(1-P)^5\left(1+2P+3P^2\right)}{2},\\
P_{43} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^4(1-P)^5\left(1+2P+2P^2\right)}{2},\\
P_{44} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^5(1-P)^5\left(1+P+2P^2\right)}{2},\\
P_{45} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^7(1-P)^5(1+2P)}{2}.
\end{aligned}
\tag{A.11}
$$

For (V, V, VV, VV), all possible realizations are listed in Table 9, in which $\Phi_{51} = \{V_1V_2, V_1V_3, V_1V_4, V_2V_3, V_2V_4\}$, $\Phi_{52} = \{V_1V_2, V_1V_3, V_1V_4, V_3V_4\}$, $\Phi_{53} = \{V_1V_2, V_1V_3, V_1V_4\}$, and $\Phi_{54} = \{V_2V_3, V_2V_4, V_3V_4\}$, and the corresponding probabilities can be expressed as

$$
\begin{aligned}
P_{46} &= \frac{\binom{4}{1}\binom{3}{1}P(1-P)^6\left(1+2P^2+2P^4\right)}{3},\\
P_{47} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^2(1-P)^6\left(7+10P+14P^2+3P^3\right)}{12},\\
P_{48} &= \frac{\binom{4}{1}\binom{3}{1}P^2(1-P)^6\left(1+P^2+2P^4\right)}{3},\\
P_{49} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^3(1-P)^6\left(4+7P+7P^2+3P^3\right)}{12},\\
P_{410} &= \frac{\binom{4}{1}\binom{3}{1}P^3(1-P)^6\left(1+2P^2\right)}{3},\\
P_{411} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^4(1-P)^6\left(1+P+P^2\right)}{3},\\
P_{412} &= \frac{\binom{4}{1}\binom{3}{1}P^7(1-P)^6}{3},\\
P_{413} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^6(1-P)^6(7+3P)}{12},\\
P_{414} &= \frac{\binom{4}{1}\binom{3}{1}P^6(1-P)^6\left(1+P^2\right)}{3},\\
P_{415} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^7(1-P)^6\left(1+2P+2P^2\right)}{3},\\
P_{416} &= \frac{\binom{4}{1}\binom{3}{1}P^9(1-P)^6\left(1+P+2P^2\right)}{3}.
\end{aligned}
\tag{A.12}
$$

Table 8: All possible realizations of (V, V, V, VV).

Case	Possible sample
2	$(V_1, V_2, V_3, V_iV_j),\ \forall V_iV_j \in \{V_1V_2, V_1V_3, V_1V_4, V_2V_3, V_2V_4, V_3V_4\}$
3	$(V_1, V_2, V_4, V_iV_j),\ \forall V_iV_j \in \{V_1V_2, V_1V_3, V_1V_4, V_2V_3, V_2V_4\}$
4	$(V_1, V_3, V_4, V_iV_j),\ \forall V_iV_j \in \{V_1V_2, V_1V_3, V_1V_4, V_3V_4\}$
5	$(V_2, V_3, V_4, V_iV_j),\ \forall V_iV_j \in \{V_2V_3, V_2V_4, V_3V_4\}$

According to Lemma 2, all realizations of (V, VV, VV, VV) can be listed as in Table 10, and the corresponding probabilities can be written as

$$
\begin{aligned}
P_{417} &= \frac{\binom{4}{1}(1-P)^7\left(1+P^3+P^6\right)}{4},\\
P_{418} &= \frac{\binom{4}{1}\binom{3}{1}P^2(1-P)^7\left(1+P^2\right)}{4},\\
P_{419} &= \frac{\binom{4}{1}\binom{3}{1}P(1-P)^7\left(1+P^4\right)}{4},\\
P_{420} &= \frac{\binom{4}{1}\binom{3}{1}P^2(1-P)^7\left(1+P^2\right)}{4},\\
P_{421} &= \frac{\binom{4}{1}\binom{3}{1}\binom{2}{1}P^3(1-P)^7}{4},\\
P_{422} &= \frac{\binom{4}{1}P^4(1-P)^7\left(1+P^3\right)}{4},\\
P_{423} &= \frac{\binom{4}{1}\binom{3}{1}P^5(1-P)^7}{4},\\
P_{424} &= \frac{\binom{4}{1}\binom{3}{1}P^6(1-P)^7}{4},\\
P_{425} &= \frac{\binom{4}{1}P^8(1-P)^7}{4}.
\end{aligned}
\tag{A.13}
$$

From the results of Table 1, it can be observed that the event $d = 4$ can be described by (V, V, V, V), (V, V, V, VV), (V, V, VV, VV), and (V, VV, VV, VV), so P_4 can be written as

$$
P_4 = \sum_{i=1}^{25} P_{4i}. \tag{A.14}
$$

Substituting (A.10), (A.11), (A.12), and (A.13) into (A.14), P_4 can be calculated.

Conflict of Interests

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

Table 9: All possible realizations of (V, V, VV, VV).

Case	Possible sample
6	$(V_1, V_2,\ V_iV_j,\ V_iV_j), \forall V_iV_j \in \Phi_{51}$
7	$(V_1, V_2,\ V_iV_j,\ V_mV_k), \forall V_iV_j, V_mV_k \in \Phi_{51}, V_iV_j \neq V_mV_k$
8	$(V_1, V_3,\ V_iV_j,\ V_iV_j), \forall V_iV_j \in \Phi_{52}$
9	$(V_1, V_3,\ V_iV_j,\ V_mV_k), \forall V_iV_j, V_mV_k \in \Phi_{52}, V_iV_j \neq V_mV_k$
10	$(V_1, V_4,\ V_iV_j,\ V_iV_j), \forall V_iV_j \in \Phi_{53}$
11	$(V_1, V_4,\ V_iV_j,\ V_mV_k), \forall V_iV_j, V_mV_k \in \Phi_{53}, V_iV_j \neq V_mV_k$
12	$(V_2, V_3,\ V_iV_j,\ V_iV_j), \forall V_iV_j \in \Phi_{54}$
13	$(V_2, V_3,\ V_iV_j,\ V_mV_k), \forall V_iV_j, V_mV_k \in \Phi_{54}, V_iV_j \neq V_mV_k$
14	$(V_2, V_4,\ V_iV_j,\ V_iV_j), \forall V_iV_j \in \{V_2V_3, V_2V_4\}$
15	$(V_2, V_4,\ V_iV_j,\ V_mV_k), \forall V_iV_j, V_mV_k \in \{V_2V_3, V_2V_4\}, V_iV_j \neq V_mV_k$
16	$(V_3, V_4,\ V_3V_4,\ V_3V_4)$

Table 10: All possible realizations of (V, VV, VV, VV).

Case	Possible sample
17	$(V_1, V_iV_j,\ V_iV_j,\ V_iV_j), \forall V_iV_j \in \{V_1V_2, V_1V_3, V_1V_4\}$
18	$(V_1, V_1V_2,\ V_iV_j,\ V_iV_j), \forall V_iV_j \in \{V_1V_3, V_1V_4\}$
19	$(V_1, V_1V_3,\ V_iV_j,\ V_iV_j), \forall V_iV_j \in \{V_1V_2, V_1V_4\}$
20	$(V_1, V_1V_4,\ V_iV_j,\ V_iV_j), \forall V_iV_j \in \{V_1V_2, V_1V_3\}$
21	$(V_1, V_1V_2,\ V_1V_3, V_1V_4)$
22	$(V_2, V_iV_j,\ V_iV_j,\ V_iV_j), \forall V_iV_j \in \{V_2V_3, V_2V_4\}$
23	$(V_2, V_2V_3,\ V_2V_4, V_2V_4)$
24	$(V_2, V_2V_3,\ V_2V_3, V_2V_4)$
25	$(V_3, V_3V_4,\ V_3V_4, V_3V_4)$

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program) (2013CB329104), the National Natural Science Foundations of China (61372124, 61171093, and 61427801), the Key Projects of Natural Science Foundations of Jiangsu University (11KJA510001), Jiangsu 973 Projects (BK2011027), and NUPTSF (NY214138).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[3] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks," *Computer*, vol. 37, no. 8, pp. 50–56, 2004.

[4] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[5] X. Li, M. Chen, and W. Liu, "Application of STBC-Encoded cooperative transmissions in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 134–137, 2005.

[6] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1089–1098, 2004.

[7] T.-D. Nguyen, O. Berder, and O. Sentieys, "Impact of transmission synchronization error and cooperative reception techniques on the performance of cooperative MIMO systems," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 4601–4605, May 2008.

[8] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[9] S. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[10] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[11] T. Ho, M. Medard, R. Koetter et al., "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[12] D. E. Lucani, M. Stojanovic, and M. Médard, "Random linear network coding for time division duplexing: when to stop talking and start listening," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 1800–1808, Rio de Janeiro, Brazil, April 2009.

[13] D. E. Lucani, M. Medard, and M. Stojanovic, "Random linear network coding for time-division duplexing: field size considerations," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, pp. 1–6, December 2009.

[14] M. Xiao, J. Kliewer, and M. Skoglund, "Design of network codes for multiple-user multiple-relay wireless networks," *IEEE Transactions on Communications*, vol. 60, no. 12, pp. 3755–3766, 2012.

[15] M. Nistor, D. E. Lucani, T. T. V. Vinhoza, R. A. Costa, and J. Barros, "On the delay distribution of random linear network coding," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 5, pp. 1084–1093, 2011.

[16] X. Liu, X. Gong, and Y. Zheng, "Reliable cooperative communications based on random network coding in multi-hop relay WSNs," *IEEE Sensors Journal*, vol. 14, no. 8, pp. 2514–2523, 2014.

[17] M. Z. Siam, M. Krunz, and O. Younis, "Energy-efficient clustering/routing for cooperative MIMO operation in sensor networks," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 621–629, April 2009.

[18] Q. Gao, Y. Zuo, J. Zhang, and X.-H. Peng, "Improving energy efficiency in a wireless sensor network by combining cooperative MIMO with data aggregation," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, pp. 3956–3965, 2010.

[19] J. W. Jung and M. A. Weitnauer, "On using cooperative routing for lifetime optimization of multi-hop wireless sensor networks: analysis and guidelines," *IEEE Transactions on Communications*, vol. 61, no. 8, pp. 3413–3423, 2013.

[20] Z. Ding, T. Ratnarajah, and K. K. Leung, "On the study of network coded AF transmission protocol for wireless multiple access channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4568–4574, 2008.

[21] O. Trullols-Cruces, J. M. Barcelo-Ordinas, and M. Fiore, "Exact decoding probability under random linear network coding," *IEEE Communications Letters*, vol. 15, no. 1, pp. 67–69, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 943475, 17 pages
http://dx.doi.org/10.1155/2015/943475

Research Article

Security Trade-Off and Energy Efficiency Analysis in Wireless Sensor Networks

Damian Rusinek,[1] Bogdan Ksiezopolski,[1,2] and Adam Wierzbicki[2]

[1]*Maria Curie-Sklodowska University, 20-031 Lublin, Poland*
[2]*Polish-Japanese Academy of Information Technology, 02-008 Warsaw, Poland*

Correspondence should be addressed to Bogdan Ksiezopolski; bogdan.ksiezopolski@acm.org

Received 8 December 2014; Accepted 4 February 2015

Academic Editor: Gabriel-Miro Muntean

With a rapid progress of numerous applications in wireless sensor networks (WSNs), performance evaluation and analysis techniques face new challenges in energy efficiency area in WSN applications. One of the key issues is to perform the security trade-off and energy efficiency analysis. In this paper, the energy analysis module for the QoP-ML (quality of protection modeling language) is proposed by means of which one can analyze the influence of various security levels on the energy consumption of a protocol. Moreover, an advanced communication module is proposed as an extension of the QoP-ML language, which enhances the abilities to analyze complex wireless sensor networks. The case study of WSN deployed on the Jindo Bridge in South Korea was carried out and the lifetime of protocols with various security levels was simulated. The results show that the introduction of various security levels can entail large differences in performance and energy consumption, and hence result in different lifetime. Therefore, the designers of WSN protocols should search for balance between the required lifetime and security level. The introduced QoP-ML extension, along with the AQoPA (automated quality of protection analysis) tool, has been developed to meet the above requirements.

1. Introduction

In today's world we witness a rapid growth of information and communication techniques for wireless sensor networks (WSNs). This progress has created a need for their analysis and performance evaluation. One of the most investigated problems of WSN applications is energy efficiency [1, 2]. In addition, the search for trade-offs between energy effectiveness and security assurance needs to be taken into consideration. Designing secure protocols which satisfy the required performance is an important issue to be solved. The traditional approach assumes that the best way is to apply the strongest possible security mechanisms, which make the system as secure as possible. Unfortunately, such reasoning leads to the overestimation of security measures, which causes an unreasonable increase in system load [3, 4]. Determination of the required quality of protection (QoP) and adjustment of some security measures to meet these concerns (QoP modeling) can be a solution to the above problems.

In the literature, many energy-efficient solutions have been proposed due to the scare battery resources of the sensors, which limits the network lifetime. Many of them concentrate on the MAC and PHY layers (standards [5–7]) and on routing and messaging protocols [8, 9]. However, there exist also application-specific solutions like data reduction (aggregation, compression) and new technologies used for harvesting energy [10].

Energy-efficient solutions are always measured and compared with their predecessors. Measurements can be done either by experiments or simulations. As the first solution is in many cases quite hard to perform the simulation is used instead. There exists many evaluation techniques, such as data or bits flow analysis, the state transition modeling based on Markov chain, and Petri net or model-driven architecture analysis. One can use tools like [11], which is a real-time

network emulator, or evaluation platforms like [12]. However, in [13] authors point out that most of classical energy models are generally oversimplified and focus only on RF transceivers ignoring other components, what may result in imprecise evaluation especially when taking into account the cases with heavy workloads on processors and sensors. They propose an event-driven queuing Petri net (QPN) [13] model to simulate the energy consumption behaviors of sensor. The QPN model allows us to evaluate the energy consumption of sensor, transceiver, and processor units including their state transitions.

Besides the energy effectiveness, security is another requirement present among the other requirements in the most of WSN applications. In [10] authors present interdependences between energy-efficient mechanisms and application requirements. Despite the fact the security is listed as one of requirements the interdependence between security and energy effectiveness is not analyzed.

Many modeling languages and tools to analyze the security of cryptographic protocols have been developed. However, the proposed approaches do not consider the topic of trade-off between security and energy efficiency. There exist tools like Scyther [23], Avispa [24], and Proverif [25] which perform a formal, automatic verification of protocol by proving the correctness of specified security requirements or by finding the flaw in the protocol. These tools, however, do not evaluate the performance. Other tools, like UMLSec [26], which deal with the security level of analysed systems are used for software development and fail to include the analysis of communication steps and their impact on system performance and security level.

To the best of the authors' knowledge, the QoP-ML (quality of protection modeling language) is the only modeling language which allows us to balance security against performance and accomplishes a multilevel analysis of the protocol, extending the possibility of describing the state of a cryptographic set of actions. Every single operation defined by the QoP-ML is characterised by security metrics which evaluate the impact of this operation on overall system security [27]. The QoP-ML was used to simulate cryptographic protocols designed for a wireless sensor network. The correctness of this analysis was positively verified by experiments [28].

The relevant type of operation is a communication process which must be included in the performance analysis of a system. The original communication model of the QoP-ML has a few limitations caused by the use of channels representing the link between each pair of hosts. The first limitation is the impossibility to determine the receiver of the message when many hosts use the same channel. In such a case, the message will be delivered to the first host in the queue of hosts waiting for the message on the channel. The inability to define the sender of the message in order to send back the response is another known limitation.

The main contributions of this paper are summarized as follows.

(i) We propose an extension of the QoP-ML which allows us to accomplish a complex network analysis as part of protocol performance analysis. Furthermore, we introduce an advanced communication module, which during the analysis takes into account the following elements: network topology, routing, and packet filtering. This new module removes all the above-listed limitations of the QoP-ML.

(ii) We propose an energy efficiency module by means of which one can analyze the influence of given operations on energy consumption and system lifetime.

(iii) The two modules introduced in this paper are implemented in the Automatic Quality of Protection Analysis Tool (AQoPA). The AQoPA performs automatic evaluation and optimization of complex system models created in QoP-ML.

(iv) We present a case study of energy efficiency analysis and security trade-offs for a complex wireless sensor network. Using this example, we want to present a method to find a trade-off between security and energy efficiency. The case study is based on an existing WSN deployed on the Jindo Bridge in South Korea [29].

The remaining part of this paper is organized as follows. Section 2 contains the comparison of the QoP-ML to other solutions used to analyze the security of protocols and their influence on performance. Section 3 describes briefly the elements of the QoP-ML language. In Section 4, a new communication model and its features and structures are described. In Section 5, the energy analysis module is explained, and in Section 6, we present a case study which uses the new functionality of the introduced communication model. Last section, Section 7, concludes the paper.

2. Related Work

All services provided by information systems of any nature (e.g., WSN, cloud, etc.) should be guaranteed by the provider and formalized in contracts. This is achieved by the SLA (service-level agreement) which defines a process of continuous monitoring and maintaining the quality of service (QoS) on the agreed level. In particular, SLA specifies the conditions (QoS parameters) under which service is delivered [30]. Conditions can be very different depending on the type of service. For example, in case of call center, a condition can specify average time it takes for a call to be answered by the service desk. On the other hand, data storage companies can specify availability as one of conditions which is the ratio of the total time a system is capable of being used during a given interval of time to the length of the interval.

The ideal system ensures the quality of service on the highest level. However, it involves high costs and when the expense of mechanisms to provide QoS is justified the provider negotiates QoS parameters (conditions) with clients. The result of negotiations is the agreed level of quality of service to be guaranteed by the provider.

The quality of service term can have various meanings [31]. Usually it is referred to the overall performance of computer network. In RFC 2386 [32], QoS has been defined as

a set of service requirements to be fulfilled when transmitting a stream of packets from source to destination. However, in the literature the requirements of QoS are defined from two perspectives: mentioned network QoS and application specific QoS [31, 33]. In the application communities, QoS generally refers to the quality as perceived by the user/application.

Assuming such broad interpretation of QoS term, one can find a subset of conditions that refer directly to the security. These can be confidentiality, authentication, integrity, availability, and many other conditions. Most of them can be associated with network QoS but some can be also application specific. For example, availability condition in network QoS can be understood as successful transmission of data from source to destination (with additional time requirements), while an application can add its derived requirements like coverage [34] when application requires whole monitored area to be covered.

Extraction of the subset of conditions that refer directly to the security gives possibility to measure the quality of protection (QoP) in analyzed system. In such case, QoP is understood as the part of QoS. Some of the conditions can overlap; for example, performance requirements (e.g., transmission time, protocol execution time, energy efficiency, and lifetime) which are strictly connected with QoS have great impact on the availability requirement which belongs to QoP requirements.

Introduction of QoP term allows us to concentrate on security requirements and extends the SLA negotiations of requirements by adding new variable (QoP derived from QoS) to previous two: QoS (as performance) and costs.

In the literature, the security trade-off is based on the quality of protection (QoP) models. These models were created for different purposes and have different features and limitations. The related research in this area is presented below.

Lindskog attempts to extend security layers in a few quality of service (QoS) architectures [17]. Unfortunately, the descriptions of the methods are limited to the confidentiality of data and based on different configurations of the cryptographic modules. Ong et al. in [19] present the QoP mechanisms which define security levels depending on security parameters. These parameters are as follows: key length, block length, and the contents of an encrypted block of data. Schneck and Schwan [21] propose an adaptable protocol concentrating on authentication. By means of this protocol, one can change the version of the authentication protocol, which finally changes the parameters of the asymmetric and symmetric ciphers. Sun and Kumar [22] create the QoP models based on vulnerability analysis which is represented by attack trees. The leaves of the trees are described by means of the special metrics of security. These metrics are used for describing individual characteristics of the attack. In the article [15], Ksiezopolski and Kotulski introduce mechanisms for adaptable security which can be used for all security services. In this model, the quality of protection depends on the risk level of the analysed processes. Luo et al. [18] provide the quality of protection analysis for the IP multimedia systems (IMS). This approach presents the IMS performance evaluation using the queuing networks and stochastic Petri nets. In the paper [14], Agarwal and Wang present the performance impact of security protocols in wireless LANs with IP mobility and introduce the QoP model to quantify the benefits of security policies and demonstrate the relationship between the QoS and the QoP. LeMay et al. [16] create an adversary-driven, state-based system security evaluation, a method which evaluates quantitatively the strength of system security. In the paper [20], Petriu et al. present the performance analysis of security aspects in the UML models. This approach takes as an input the UML model of the system designed by the UMLsec extension [26] of the UML modeling language. This UML model is annotated with the standard UML profile for schedulability, performance, and time and then analysed for performance. In the article [35], Ksiezopolski introduces the quality of protection modeling language which provides the modeling language for making abstraction of cryptographic protocols with emphasis on the details concerning the quality of protection. Table 1 demonstrates the approach presented in this paper as compared to the existing methodologies. These approaches can be characterised by the following main attributes.

Table 1: The characterisation of the QoP models.

	QA	*E*	*Con*	*EE*	*H*	*Com*	*PE*
Agarwal and Wang [14]	√	—	—	—	√	√	√
Ksiezopolski and Kotulski [15]	√	√	—	—	√	√	—
LeMay et al. [16]	—	√	√	—	—	—	—
Lindskog [17]	√	—	√	—	—	—	√
Luo et al. [18]	√	—	—	—	√	√	√
Ong et al. [19]	√	—	—	—	—	—	—
Petriu et al. [20]	—	√	√	—	—	√	√
Schneck and Schwan [21]	√	—	√	—	—	—	√
Sun and Kumar [22]	√	—	—	—	—	—	—
QoP-ML	√	√	√	√	√	√	√

(i) *Quantitative assessment (QA)* refers to the quantitative assessment of the estimated quality of protection of the system.

(ii) *Executability (E)* specifies the possibility of the implementation of an automated tool able to perform the QoP evaluation.

(iii) *Consistency (Con)* is the ability to model the system maintaining its states and communication steps consistency.

(iv) *Performance evaluation (PE)* gives the possibility of performance evaluation of the analysed system.

(v) *Energy evaluation (EE)* gives the possibility of energy efficiency evaluation of the analysed system.

(vi) *Holistic (H)* approach gives the possibility of the evaluation of all security attributes.

(vii) *Completeness (Com)* is the possibility of the representation of all security mechanisms. This attribute is provided for all models.

One can notice that only QoP-ML can be used for finding a trade-off between security (*QA*) and performance (*PE*)

including energy efficiency evaluation (*EE*) of the system which is modeled in a formal way with communication steps consistency (*Con*). By means of QoP-ML, one can evaluate all security attributes (*H*) and abstract all security mechanisms which protect the system (*C*). Additionally, the QoP-ML approach is supported by the tool (*E*) required for the analysis of complex systems.

3. QoP-ML

In the paper [35], Ksiezopolski introduces the quality of protection modeling language, which provides the modeling language for making abstraction of cryptographic protocols with emphasis on the details concerning the quality of protection. The intended use of the QoP-ML is to represent a series of steps described as a cryptographic protocol. The QoP-ML has introduced a multilevel protocol analysis which extends the possibility of describing the state of a cryptographic protocol.

3.1. General View. Structures used in the QoP-ML represent a high level of abstraction which allows us to focus on the quality of protection analysis. The QoP-ML consists of processes, functions, message channels, variables, and QoP metrics. Processes are global objects grouped into the main process, which represents a single computer (host). A process specifies behaviour, functions represent a single operation or a group of operations, and channels define the environment in which a process is executed.

The QoP metrics define the influence of functions and channels on the quality of protection. In the paper [35], the syntax, semantics, and algorithms of the QoP-ML are presented.

3.2. Data Types. In the QoP-ML, an infinite set of variables is used for describing communication channels, processes, and functions. Variables are used to store information about the system or a specific process. The QoP-ML is an abstract modeling language, so there are no special data types, sizes, or value ranges. Variables do not have to be declared before they are used. They are automatically declared when they are used for the first time.

The scope of variables declared inside a high hierarchy process (`host`) is global for all processes defined inside a `host`.

3.3. Functions. System behaviour is changed by functions which modify the states of variables and pass objects by communication channels. When defining a function, one has to set the arguments of this function which describe two types of factors. Functional parameters written in round brackets are necessary for the execution of a function while additional parameters written in square brackets influence the system quality of protection. The names of arguments are unrestricted.

3.4. Equation Rules. Equation rules play an important role in the quality of protection protocol analysis. Equation rules for a specific protocol consist of a set of equations asserting the equality of function calls. For instance, the decryption of the encrypted data with the same key is equal to the encrypted data.

3.5. Process Types. Elements describing system behaviour (functions, message passing) are grouped into processes which constitute the main objects in the QoP-ML. In a real system, processes are executed and maintained by a single computer. In the QoP-ML, sets of processes are grouped into a higher hierarchy process named `host`.

All variables used in a high hierarchy process (`host`) have a global scope for all processes grouped inside this structure. Normally, variables used inside a `host` process cannot be applied for another high hierarchy process. This operation is possible only when a variable is sent by a communication channel.

3.6. Message Passing. Communication between processes is modeled by means of channels which are used to pass messages between hosts and processes in the FIFO (first-in first-out) order. Before a message is sent, a channel must be declared because its declaration contains a buffer size and other channel's characteristics. When channels are declared with a nonzero buffer size, communication is asynchronous, whereas a buffer size equal to zero stands for synchronous communication. In synchronous communication, the sender transmits data through a synchronous channel only if the receiver listens to this channel. When the size of the buffer channel equals at least 1, a message can be sent through this channel even if no one is listening on this channel. This message will be transmitted to the receiver when the listening process in this channel is executed.

3.7. Security Metrics. System behavior, which is formally described by a cryptographic protocol, can be modeled by the proposed QoP-ML. One of the main aims of this language is to abstract the quality of protection of a particular version of the analysed cryptographic protocol. In the QoP-ML, the influence of system protection is represented by means of functions. While declaring a function, the quality of protection parameters is defined and the details about this function are described. These factors do not influence the flow of a protocol, but they are crucial for the quality of protection analysis. During such an analysis, functions' QoP parameters are combined with the next structure of the QoP-ML, that is, security metrics. In this structure, one can abstract functions' time performance, their influence on the security attributes required for a cryptographic protocol, or other factors important during the QoP analysis.

4. Advanced Network Analysis Module

The introduction of new network analysis module eliminates the weaknesses of the original one (from QoP-ML). Briefly mentioning, the first weakness is the impossibility to determine the receiver of the message when many hosts use the same channel while the second one is the inability to define the sender of the message in order to send back the response.

```
(1)  communication {
(2)
(3)   medium[cable] {
(4)    default_q = 0.1;
(5)    default_time = 1 ms;
(6)
(7)    topology {
(8)     Gateway -> Sensor[0];
(9)    }
(10)  }
(11)
(12)  medium[air_channel] {
(13)   default_q = 1;
(14)   default_time = 18 ms;
(15)
(16)   topology {
(17)    Sink <-> Gateway : t = wsn_time[ms];
(18)
(19)    Sensor -> * : time = 17 ms;
(20)
(21)    Sensor[0] -> Gateway, time = 5 ms;
(22)    Sensor[0] <- Gateway : q = 2.5, time = 5 ms;
(23)    Sensor[1] <- Sensor[2] : q = 3.5, time = 5 ms;
(24)
(25)    Sensor[2:5] -> Sensor[3];
(26)    Sensor[2:] -> Sensor[4] : time = 15 ms;
(27)    Sensor[:2] -> Sensor[5] : q = 3.5, time = 15 ms;
(28)
(29)    Sensor -> Sensor[i + 1] : q = 2;
(30)    Sensor[0:5] <- Sensor[i − 2];
(31)    Sensor[4:] <- Sensor[i − 3];
(32)   }
(33)  }
(34) }
```

Listing 1: An example of a topology definition connected with *channel_name* tag.

Removal of the limitations enumerated above requires the creation of new mechanisms and structures in the QoP-ML model. In this section, we describe three new mechanisms: topology, routing, and packet filtering. In addition, we introduce a methodology which provides time analysis of communication steps in a network. Depending on the selected path in a network, the time of delivering a message from the sender to the receiver can vary. The model allows to determine the characteristics of a channel and calculate the time of transmission.

The syntax of all structures introduced in this paper is presented in Supplementary Material available online at http://dx.doi.org/10.1155/2015/943475 using the BNF (Backus-Naur form) [36] standard.

4.1. Topology. A topology is defined by a graph where vertices are hosts and edges are connections between them. All existing connections must be defined and have a weight representing the quality of connection (the lower the weight, the better the quality). A special type of connection is a link between a host and a medium used for broadcasting messages. This connection does not have the quality parameter.

A topology is defined in the *topology* structure (from line 16 to line 32 in Listing 1) which is a part of the *communication* structure (see lines from 1 to 34 in Listing 1). The aim of the *communication* structure is to describe the communication characteristics of mediums (channels). It includes the definition of topology and default topology parameters for all mediums. The *communication* structure can be located in two places. First, it can be one of the main structures (like *hosts*, *functions*, etc.) and affect the whole model and all versions. Secondly, the structure can be placed in the *version* structure after the *run* section. In such a case, it affects only the selected version. If the element of the *communication* structure (e.g., a topology) for a given medium is defined in the *version* structure, it overrides the main *communication* structure (i.e., the topology is determined on the basis of the version *communication* structure only).

4.1.1. Connections Definition. A *topology* consists of rules which define connections between hosts or between a host

and a medium (used for broadcast). A rule has two sets of hosts (left and right), a direction, and, optionally, after the colon, the connection-specific values of parameters. There are three types of direction:

(1) A $\rightarrow$ B, the connection is created from host A to B;

(2) A $\leftarrow$ B, the connection is created from host B to A;

(3) A $\leftrightarrow$ B, the connection is created in both ways.

There are three possible ways of declaring the left set of hosts, all of which are presented in Listing 1.

(1) The first way (without indices) includes all hosts with a given name. In Listing 1, they are the rules in lines number 17 and 29. These rules can be used in the main *communication* structure since the structure does not specify the index of the host.

(2) The second way (with one index in square brackets) selects only one host, the one with a given index. In Listing 1, they are the rules in lines number 21, 22, and 23.

(3) The third way (with indices and a colon in square brackets) selects the range of hosts with indices larger than or equal to the first index and lower than or equal to the second index. If the first index is not specified, zero is used, and if the second index is not specified, the number of all hosts with a given name is chosen. The examples are the rules in lines 25, 26, 27, 30, and 31 in Listing 1.

Besides the three methods described above, one can use two additional ways to declare the right set of hosts.

(1) The hosts can be specified with a special *i* index and its modified (increased or decreased) value. In such a case, the hosts with indices shifted by a given value (modification of *i*) in relation to all hosts from the left set are selected. The example rules are in lines number 29, 30, and 31 in Listing 1. The first rule (line number 29) defines the links between all *Sensor*s and their next neighbours (forming a line) while the second one (line number 30) defines the links between *Sensor*s with indices 0, 1, 2, 3, 4, and 5 and their second predecessor. The last rule (line number 31) creates the link between *Sensor*s with an index larger than or equal to 4 and their third predecessors. When a host does not have a selected neighbour, the link is not created. This type of rule can be used only in the *version* structure when indices are used on the left side.

(2) The hosts can be replaced with a star sign (∗) which represents a medium. In this case, the quality parameter is not defined and the direction can only be right (from the left hosts to the medium). This type of rule is used to define the parameters for broadcasting a message. The example rule is in line number 19 in Listing 1.

4.1.2. Quality of Connections. Each connection in a topology can be parameterized. Parameters are used to perform the analysis of communication steps. Each parameter can have a default value. To define a default value, one has to precede its name with *default_* and place it in the *medium* structure (lines number 13 and 14 in Listing 1). When a parameter is not defined for a particular connection in a topology, the default value is used.

There is one required parameter *q* (e.g., numbers 22 and 23 in Listing 1) which represents the quality (weight) of a connection between hosts (the lower the value, the better the quality). The quality parameter is used by the routing algorithm to find the best route between two hosts in multihop communication. It is the resultant value of the environmental factors (e.g., distance, barriers, etc.). This parameter can either be defined statically or estimated dynamically by a defined algorithm. We do not consider the algorithm determining the quality because the QoP-ML is the modeling language not only for WSNs, but also for other systems and protocols. Therefore, algorithms may be entirely different.

4.1.3. Transmission Time. Another important factor in the communication analysis is the time analysis which introduces the *time* parameter. The proposed parameter represents the time of data transmission between hosts or between a host and a medium (used for broadcast). An example of a definition of a default transmission time is in line number 14 in Listing 1, while a definition of time for a specific connection can be found, for example, in lines number 21, 22, or 23. Its value can be specified as

(i) a constant or random number from a specified range in seconds or milliseconds (e.g., line number 19 in Listing 1);

(ii) a value depending on the size of data: mspb, mspB, kbps, and mbps (a constant or random value from a specified range per bit or byte);

(iii) a constant or random value from a specified range in seconds or milliseconds per each block of data (e.g., 100 ms per each 16 bytes);

(iv) the result of an algorithm (e.g., line number 17 in Listing 1) in seconds or milliseconds (algorithms are discussed further in this section).

Depending on the number of receivers, the communication time can vary. The main rules are presented below.

(i) When a message is sent to one receiver, the time of communication is equal to the result of the *time* parameter. The time of the sender and the receiver is increased with the result time.

(ii) When a message is sent to zero receivers (no one is waiting for a message), the time of communication is equal to the result of the *time* parameter between a host and a medium (broadcast time). Only the time of the sender is increased.

(iii) When a message is sent to many receivers, the time of communication can be different for all hosts. The time

```
(1)  algorithms {
(2)     alg wsn_time(msg) {
(3)        sending = 18;
(4)        size_factor = 0.12;
(5)        full_time = 0.0;
(6)        msg_size = size(msg);
(7)        while (msg_size > 0) {
(8)           current_size = 110;
(9)           if (msg_size < 110) {
(10)             current_size = msg_size;
(11)          }
(12)          full_time += sending + current_size * 0.12;
(13)          msg_size = msg_size − 110;
(14)       }
(15)       return full_time;
(16)    }
(17) }
```

Listing 2: An example of an algorithm for the communication time.

of sending is equal to the result of the *time* parameter between the sender and the medium (broadcast time). The sender's time is increased with this value. The time of receiving for each receiver is equal to the maximum value of the time of sending and the result of the *time* parameter between the sender and the given receiver. As the times of communication between the sender and different receivers can vary, the times of receiving can differ as well.

The easiest way to determine the transmission time in a medium is to take its bandwidth. However, this measure is inaccurate in many cases. In order to define the transmission times more precisely, we introduced the algorithms structure, which provides the possibility of adding nonlinear values of metrics.

An example of an algorithm is presented in Listing 2. It calculates the transmission time between two TelosB motes [28]. The time of transmission is equal to constant 18 ms plus 0.12 ms per each byte. The while loop is used to handle messages with payload larger than 110 bytes, which is the maximal payload size in ZigBee assuming that header has 17 bytes size (the maximal size of packet is 127 bytes) [37]. When the maximal size is exceeded, payload is divided into many packets with a 110 bytes payload size.

An algorithm is defined like a function but started with the word *Alg*. Each algorithm has one parameter which is a message being sent in the case of a communication step or a function call expression in the case of an operation in process.

The body of an algorithm includes arithmetic operations, constructions known from the C language: if, while, and two predefined function calls:

(i) *quality*, which can be used only in the algorithm for calculating a communication time step and which returns the quality of the link between the sender and the receiver (parameter q),

(ii) *size*, which takes one argument and returns its size.

The function *size* is called with the algorithm parameter as the argument in order to obtain the size of the called function, the sent message, or its indexed element.

An example usage of an algorithm as the value of the communication parameter is presented in Listing 1 (line 17). In order to calculate the time of transmission of a message between the *Sensor* and the gateway hosts, the *wsn_time* algorithm is used and the return value is determined in milliseconds.

4.2. Packet Filtering. Packet filtering is a feature which allows us to determine which packets should be delivered to a selected host. While the receiver specifies what kind of packets would like to receive, the sender determines the type of the transmitted packet. Such an approach allows many hosts to communicate on the same channel.

The process of filtering packets is presented in Algorithm 1. It contains *FilteredRequest* function which accepts a message and channel and returns the requests that can accept the message.

The *FilteredRequests* function uses *PassFilters* function which is presented in Algorithm 2. The *PassFilters* accepts two parameters: message being sent and filters taken from *in* instruction (described later in this section). Function returns boolean *True* when message contains values (representing headers) acceptable by filters.

4.2.1. Channels. The structure of channels is presented in Listing 3. The value in square brackets at the end of the channel definition is a tag which determines channel characteristics, the medium name. This tag is used to link the channel with the medium. Many channels can be assigned to the same medium. Then each channel is treated independently but has the same characteristics (topology, topology parameters, etc.).

An example is presented in Listing 3. There is the *channels* structure which contains one channel named *channel_WSN*.

```
(1) procedure FilteredRequests(message, channel) ▷ Procedure returns list of requests that can accept the message.
(2)     filteredRequests ← empty list
(3)     sender ← get_sender(message)            ▷ Pull out sender from message.
(4)     for request in get_waiting_requests(channel) do    ▷ Get all requests that wait on channel
(5)         receiver ← get_receiver(request)    ▷ Pull out receiver from request.
(6)         if link between sender and receiver does not exist in router topology then
(7)             Continue to the next loop
(8)         end if
(9)         if message cannot be accepted by request then          ▷ Modules can flag messages that cannot be assigned to
        selected requests, eg. when message is sent before the request is created while the channel is synchronous.
(10)            Continue to the next loop
(11)        end if
(12)        requestFilters ← get_filters(request)     ▷ Retrieve filters from in instruction linked with request.
(13)        if not PassFilters(message, filters) then
(14)            Continue to the next loop
(15)        end if
(16)        Add request to filteredRequests list.
(17)    end for
(18)    return filteredRequests
(19) end procedure
```

Algorithm 1: Algorithm of filtering requests.

```
(1)  procedure PassFilters(message, filters)          ▷ Procedure returns True if message passes the filters from request.
(2)      if filters is empty then
(3)          return True
(4)      else
(5)          expression ← get_expression(message)     ▷ Pull out expression sent in message.
(6)          if expression is not tuple or size of expression is smaller than filters size then ▷ For packet filtering expression must
         be a tuple because its elements are compared with filters.
(7)              return False
(8)          else
(9)              for filter in filters do
(10)                 if filter is star then              ▷ Filter accepts everything.
(11)                     Continue to the next filter
(12)                 else
(13)                     tupleElement ← get_next_element_from_expression(expression)
(14)                     if tupleElement is not equal to filter then
(15)                         return False
(16)                     end if
(17)                 end if
(18)             end for
(19)         end if
(20)         return True
(21)     end if
(22) end procedure
```

Algorithm 2: Algorithm of checking if message is sent for the request by comparing its elements with request's filters.

```
(1) channels { channel channel_WSN (*)[air_channel]; }
```

Listing 3: Definition of one channel called **channel_WSN** with *air_channel* characteristics.

```
(1) in(channel_name: var_name: |*, id(), init_cmd()|);
```

Listing 4: An example of the extended **in** instruction.

```
(1) MSG = (id(), id(Sensor), init_cmd(), data());
(2) out(channel_name: MSG);
```

Listing 5: An example of the **out** instruction sending a message with a header.

It has an unlimited buffer of messages (the star sign) and is connected with a medium from the communication structure called *air_channel*.

4.2.2. Input and Output Messages. Packet filtering introduces a new (optional) part of the *in* instruction in the QoP-ML (input messages). An example is presented in Listing 4.

This instruction waits for a message from channel *channel_name* and saves it in the *var_name* variable. The new part starts with the second colon. The values between "|" signs specify the first three values of the incoming message. In the case when a message has different values, the instruction *in* will continue to wait until the message with the three specified values is delivered. These filtered values can be understood as the header values.

The typical use of this feature is to reject packets which are not addressed to the host (or process). Such an approach requires the introduction of new predefined functions: *id* (used in the example in Listing 4) and *pid*, which can be executed with one optional parameter. If the parameter is specified, they return the identification number of host (*id*) or process (*pid*) with the same name as the passed argument. Otherwise, they return the identification of host or process in which the function is executed.

The designer can use four types of elements as the filtering value in the *in* instruction:

(i) a simple function call, the *init_cmd*() in Listing 4;

(ii) functions *id* and *pid* described above;

(iii) a variable name when its value should be used to filter the packet;

(iv) sign $*$ (star) which states any value is accepted.

In Listing 4, the host waits for the message that has any value in the first element, its identification in the second, and the *init_cmd* function call as the third. The host can wait for many messages from many other hosts. In order to recognise these messages, the third parameter has been used as the message type. In Listing 4, the host waits for a message that is in some way understood as *initial command* (*init_cmd()*). However, the number of parameters is not fixed and the designer can use a different number of filtering parameters compared to the 3 used in the above example.

From the perspective of the sending host, packet filtering needs to include the filtered values in the message. In Listing 5, the examplary message *MSG* is created and sent through the channel *channel_name*. It is a 4-tuple which contains the sender's identification, the receiver's identification, the message type (*initial command*), and the data. This type of message can be accepted by the instruction in Listing 4.

The syntax and semantics of the *out* instruction are unchanged in comparison to those defined in the QoP-ML. The *out* instruction still accepts any variable. However, when the packet filtering feature is used, the values of variables must be tuples because the *in* instruction needs to access their indexed elements.

The introduction of the *id* and *pid* predefined functions provides the possibility to send back a message to the sender when hosts are replicated.

The processes of sending a message and waiting for the message on channel are presented in Algorithms 3 and 4, respectively.

Functions *SendMessage* and *WaitForMessage* from Algorithms 3 and 4 use *BindMessagesWithRequests* function which is responsible for binding all waiting requests with messages being sent through the given channel at the given moment. Its algorithm is presented in Algorithm 5.

4.3. Routing. Routing is an integral part of all networks. It can be defined as static, when all connections are defined in advance and cannot change, or dynamic, when the path from host A to B can be modified in time. The presented communication model uses a topology to find the shortest path between a pair of hosts using the Dijkstra algorithm [38]. The edges are compared using the connection qualities defined in the topology. The routing feature solves the problem of multihop communication in the QoP-ML. The sender can check which host is the next hop in the path between the sender and the receiver. It is obtained with the use of a new, predefined function, namely, *routing_next*, which takes three parameters: the first one is the topology name, the second one is the identification of the receiver, and the third (optional) one is the identification of the sender (it is the identification of the host which calls the function by default). The function returns the identification of the sender's next hop host.

An example use of the *routing_next* function is presented in Listing 6. In the first line, the host obtains the identifier of the host which is its neighbour in the path leading to

```
(1) procedure SendMessage(sender, channel, message)   ▷ Procedure sends message from sender through channel.
(2)   if sender can use channel then
(3)     filteredRequests ← FilteredRequests(message, channel)
(4)     for request in filteredRequests do
(5)       receiver ← get_receiver(request)   ▷ Pull out receiver from request.
(6)       buffer ← get_buffer(channel, receiver)   ▷ Retrieve receiver's buffer from channel.
(7)       Add message to buffer.
(8)     end for
(9)     BindMessagesWithRequests(channel)
(10)  end if
(11) end procedure
```

Algorithm 3: Algorithm for **out** instruction.

```
(1) procedure WaitForMessage(channel, request)   ▷ Procedure processes in instruction and adds request to channel's
    requests list.
(2)   if request exists in channel's requests list then
(3)     if request is not in waiting state then
(4)       request's state ← WAITING
(5)     end if
(6)   else
(7)     Add request to channel's requests list
(8)   end if
(9)   BindMessagesWithRequests(channel)
(10) end procedure
```

Algorithm 4: Algorithm for **in** instruction.

the *Sensor* host (the second argument). The first argument (*air_channel*) is used to select the medium. In this case, the topology from the *air_channel* medium is used. In the second line, the third (optional) argument is added. It tells the function from which host it should start the path (when the argument is not given, the algorithm starts from the host which calls the function). In this case, it would start from the first neighbour (obtained in the first line), and therefore the result of the function call would return the second neighbour of the host which calls the function. In the third line, the 5-tuple message is created. The first three values are understood as header: sender, received, and message types. The last two are payload: the first contains some data and the second contains the identifier of the second neighbour which can be used, for example, to manually define the next hop in the path (e.g., the protocol requires that the first neighbour must send *data* through the second neighbour included in the message).

5. Energy Analysis and Lifetime Prediction Module

One of the main contributions of this paper is to add the energy analysis and lifetime prediction module to the QoP-ML and its implementation as an extension to the AQoPA.

5.1. Energy Analysis. The aim of the energy analysis module is to evaluate the energy consumption of the modeled system. To determine these values, the time analysis module must be included in the performance analysis process because it tracks the times of operations and communication steps. Energy consumption is calculated as the sum of the energy consumed by simple operations which use only the CPU (security operations, other arithmetic operations, etc.) and communication operations (listening, receiving, and sending) which use the radio. The energy consumption of one CPU or communication operation is calculated as follows:

$$E_{\text{op}} = T * I * V, \tag{1}$$

where E_{op} is the energy consumption of CPU or communication operation, op is the index of operation, T is the time of the operation, I is the electric current of the operation, and V is the voltage of the host.

The time is retrieved from the time analysis module and the voltage is defined for each host as constant. The remaining factor, the current, can be defined for each operation independently or for a group of operations. Its value is specified in metrics with the *current* header. In the case of communication steps, the current is defined in the *medium* structure.

Finally, the energy module analysis evaluates the energy consumption for each host as follows:

$$E_H = E_{H_{\text{CPU}}} + E_{H_{\text{COMM}}}, \tag{2}$$

where E_H is the energy consumption of the host, $E_{H_{\text{CPU}}}$ is the sum of energy consumption of all CPU operations and operations with a separately specified electric current, and $E_{H_{\text{COMM}}}$

```
(1)  procedure BindMessagesWithRequests(channel)        ▷ Procedure binds messages from the buffers with
     matching requests.
(2)    for request in get_waiting_requests(channel) do
(3)      receiver ← get_receiver(request)    ▷ Pull out receiver from request.
(4)      buffer ← get_buffer(channel, receiver)    ▷ Retrieve receiver's buffer from channel.
(5)      if request is waiting for message and message has not been assigned yet and buffer is not empty then
(6)        filters ← get_filters(request)          ▷ Retrieve filters from in instruction linked with request.
(7)        for message in buffer do
(8)          if PassFilters(message, filters) then
(9)            Assign message to request
(10)           Remove message from buffer
(11)           Break                                ▷ Leave for loop.
(12)         end if
(13)       end for
(14)     end if
(15)     if request is ready to fulfill then    ▷ Request was waiting for message and obtained it.
(16)       Set variable from request in receiver with value from message
(17)       Move receiver to the next instruction
(18)       if channel is synchronous then
(19)         Remove request from buffer    ▷ Delete the request-a new one will be created when the instruction is
     executed again.
(20)       else
(21)         Set request's status ← NOT  WAITING    ▷ Request has been fulfilled but still can accept messages to the buffer.
(22)       end if
(23)     end if
(24)   end for
(25)   if channel is synchronous then
(26)     Clean all buffers.
(27)   end if
(28) end procedure
```

Algorithm 5: Algorithm of binding sent messages with matching receivers.

```
(1) FIRST_NEXT_ID = routing_next(air_channel, id(Sensor));
(2) SECOND_NEXT_ID = routing_next(air_channel, id(Sensor),
          FIRST_NEXT_ID);
(3) MSG = (id(), FIRST_NEXT_ID, init_cmd(), data(),
         SECOND_NEXT_ID);
(4) out(channel_name: MSG);
```

Listing 6: Obtaining the addresses of the next two hops and sending the message.

is the sum of the energy consumption of all communication operations (sending, receiving, and listening).

The energy analysis module introduces three parameters: *sending_current*, *receiving_current*, and *listening_current*. All of them describe the electric current in three different states. The *listening_current* defines the electric current when a host is waiting on the channel for a message. The electric current in the transmission state has been divided in two: the *sending_current* and the *receiving_current* because hosts can send and receive data with different electric currents (e.g., the sending current in the sensors can vary depending on signal strength).

The value of the current can be specified as a constant in milliamps or as the result of an algorithm in milliamps. In Listing 7, the *wsn_sending_current* algorithm (line 10) is used to calculate the electricity current of the message sending process. The value is determined in milliamps (the unit is defined in square brackets). The *wsn_sending_current* algorithm must be placed in the *algorithms* structure and return the value of the current. An example of the algorithm is presented in Listing 2: it returns the time.

5.2. Lifetime Prediction. In the proposed module we measure the energy efficiency of a secured network by means of its lifetime (in days). The longer the lifetime of a network, the more energy efficient a protocol. We introduce two types of lifetime: the nodal lifetime and the network lifetime.

```
(1) communication {
(2)  medium[wsn] {
(3)   default_q = 1;
(4)   default_t = 20 ms;
(5)   default_sending_current = 14.8 mA;
(6)   default_receiving_current = 22.4 mA;
(7)   default_listening_current = 1.8 mA;
(8)
(9)   topology {
(10)    Sensor <-> Gateway : sending_current = wsn_sending_current[mA];
(11)  }
(12) }
(13) }
```

Listing 7: An algorithm used to calculate the value of a metric.

The *nodal lifetime* $\mathrm{nl}(G, v)$ of node v in the network represented by graph G is defined as follows:

$$\mathrm{nl}(G, v) = \frac{E_r(v)}{E_{\mathrm{CPU}}(v) + E_{\mathrm{COMM}}(v)}, \tag{3}$$

where $E_r(v)$ indicates the residual energy of node v. $E_{\mathrm{CPU}}(v)$ and $E_{\mathrm{COMM}}(v)$ are the sums of energy of all CPU and the communication operations, respectively, of node v. They are defined in (4) and (5).

The sum of all CPU operations is defined as follows:

$$E_{\mathrm{CPU}}(v) = \sum_{i \in \mathbb{CPU}} E_i(v), \tag{4}$$

where $\mathbb{CPU}$ is the set of indexes of all CPU operations and operations with a separately specified electric current.

The sum of all communication operations is defined as follows:

$$E_{\mathrm{COMM}}(v) = \sum_{i \in \mathbb{COMM}} E_i(v), \tag{5}$$

where $\mathbb{COMM}$ is the set of indexes of all communication operations (sending, receiving, and listening).

The *network lifetime* $\mathrm{NL}(G)$ is defined as the minimum of nodes' lifetimes because we assume that each node must be operative in order to keep network working correctly. Usually the *Sink* is the bottle neck of the network. The network lifetime is defined as follows:

$$\mathrm{NL}(G) = \min_{v \in G} \mathrm{nl}(G, v). \tag{6}$$

The trade-off between the security and energy efficiency is achieved by selecting the most energy efficient version of a protocol which provides security at the required level in a given unit of time.

6. Case Study

In this section, the authors present a case study which uses the mechanisms described in previous sections and introduces network analysis into the process of balancing security against performance and energy consumption. In this case study, we have created the QoP-ML model of a wireless sensor network deployed on the new Jindo Bridge, a cable-stayed bridge in South Korea with a 344 m main span and two 70 m side spans [29]. In total, 70 sensor nodes and two base stations have been deployed to monitor the bridge using an autonomous SHM (structural health monitoring) application with excessive wind and vibration triggering the system to initiate monitoring. The central components of the WSN deployment are TelosB motes and the security metrics for communication and cryptographic primitives (symmetric and asymmetric encryption) were taken from previous experiments [28].

Figure 1 presents the locations of all nodes. The whole network consists of two independent single-hop subnetworks, one per each pylon. Both subnetworks have their own gateway node placed on the corresponding pylon on the neighbouring bridge.

The SHM software installed on the sensors includes four services.

(i) *SnoozeAlarm* is a strategy that allows the network to sleep most of the time and wake up periodically to measure data.

(ii) *ThresholdSentry* wakes up the network in the case of an important event. The sentry nodes wake up at predefined times and measure a short period of acceleration or wind data. When the measured data exceeds a predefined threshold, the sentry node sends an alarm to the gateway node, which subsequently wakes the entire network for a synchronized data measurement.

(iii) *Watchdog Timer* is used to reset the nodes to ensure network reliability in the case of a node hanging due to an unexpected error.

(iv) *RemoteSensing* is a remote data measurement application and data collection to the gateway node and the base station.

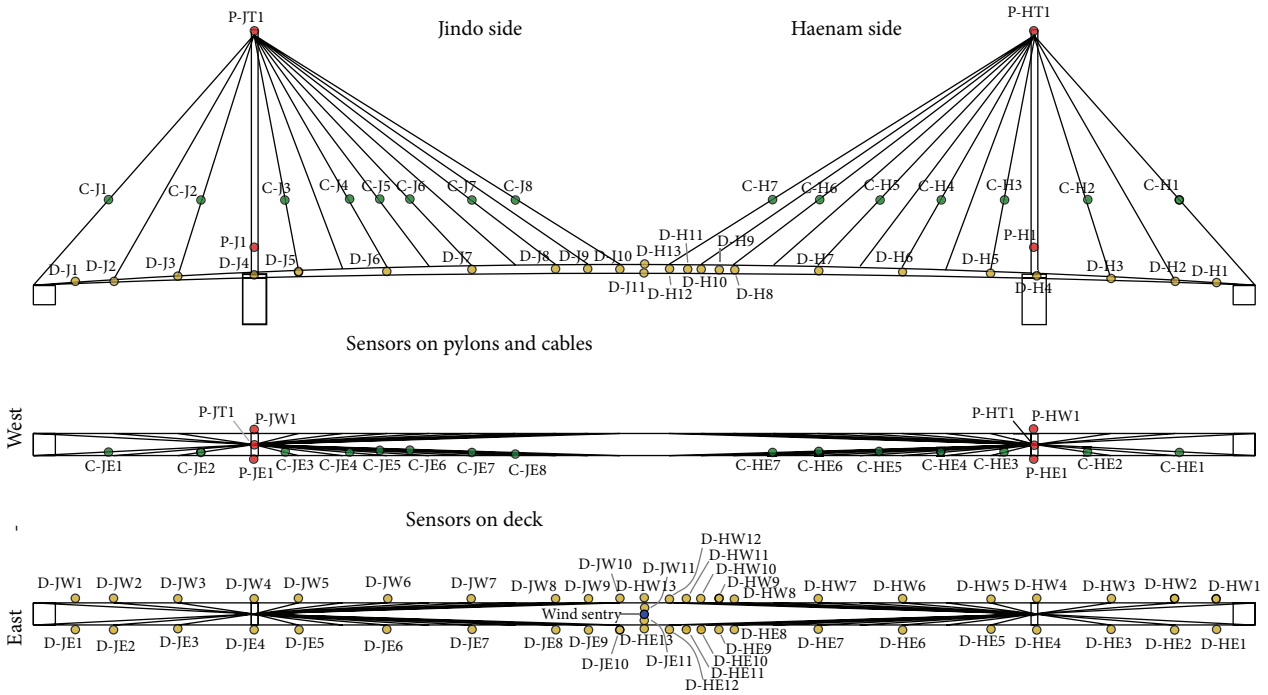


Figure 1: Locations of the sensors on the Jindo bridge [29].

Table 2: The parameters of the *RemoteSensing* application.

Parameter	Value
Time synchronization wait time	30 sec
Sensing start delay	50 sec
Node sensing start delay	1.5 sec
Sampling frequency	10 Hz
Channels sampled	3
Number of data points per channel	5000

Since the *RemoteSensing* application consumes the most time and energy, it became the subject of our case study. The details of this service are presented in Table 2. This application periodically collects the acceleration data (in three dimensions) from the sensors deployed on the whole bridge. The QoP-ML model representing the *RemoteSensing* application is available in the AQoPA's library [39].

The flow of the *RemoteSensing* application is as follows.

Step 1. Network time synchronization is held during the *Time synchronization wait time* period (Table 2).

Step 2. Send measurement parameters from the gateway to the leaf nodes.

Step 3. Each channel in the data sampling phase is sampled for the number of data points given in the parameter and the given frequency.

Step 4. Transfer data back to the gateway node and saving data on the base station.

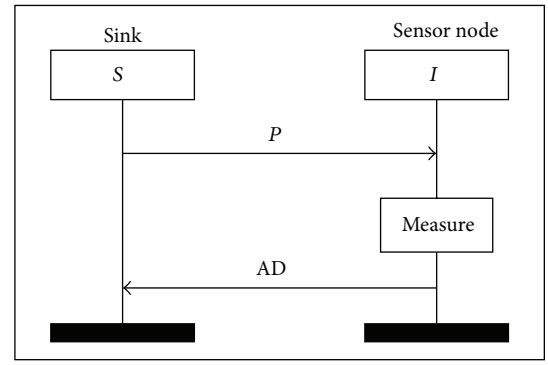


Figure 2: The flow of *LOW* security level protocol. The original protocol which ensures neither confidentiality nor authentication.

6.1. Cryptographic Protocols. The deployed network [29] is unsecured as it does not ensure any security attributes. In the case study, we intend to evaluate the influence of security attributes on the performance of the network. We introduce three protocols which guarantee three different levels of security: *LOW* (Figure 2), *MID* (Figure 3), and *HIGH* (Figure 4).

In the *LOW* security level protocol, the *Sink* starts with message *P* containing the measurement parameters. Upon its reception, the *Sensor* starts measurement and sends back the acceleration data (AD), the result of the measurement. In this level no security attributes are guaranteed.

The *MID* security level protocol introduces confidentiality of accelerated data. After measurement, the *Sensor* encrypts the data with a predeployed network key (NK).

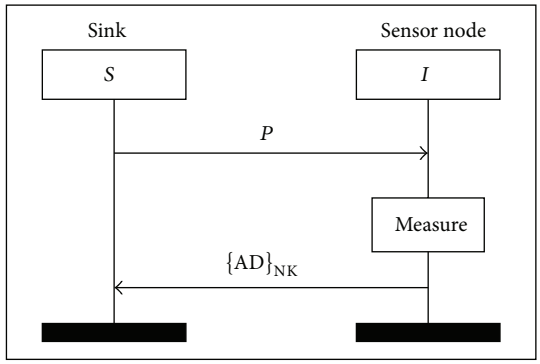


Figure 3: The flow of the *MID* security level protocol. The protocol encrypts the samples data with the AES-CTR-256 cipher and thus ensures confidentiality.

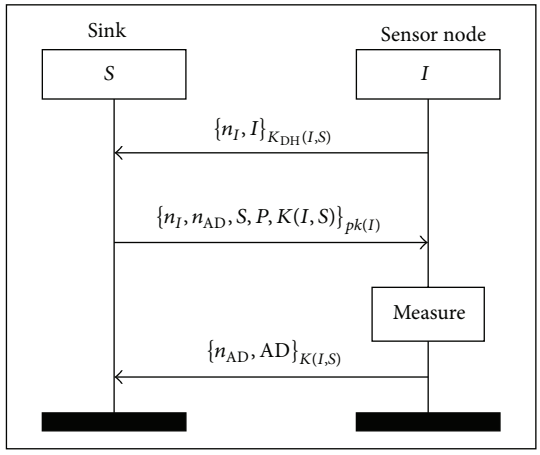


Figure 4: The flow of the *HIGH* security level protocol. Authentication is added by the use of the ECC with 160 bit key length.

In this protocol, the AES algorithm is chosen for the encryption in the CTR mode and with 256 bits of the key.

In the *MID* security level protocol, sensor nodes are not authenticated and a malicious node can deceive sensor nodes by impersonating the *Sink* and sending fake parameters P. This is avoided by the introduction of the sensors and parameters authentication achieved with the modified version of the *DJS* protocol from [28].

The *HIGH* security level protocol is started by sensor nodes. They generate key $K_{\mathrm{DH}}(I, S)$ using the Diffie-Hellman method (ECC 160 bits of key) without communication with their private key and the *Sink*'s public key as it is predeployed on all sensor nodes. The generated key is used to encrypt the request which is sent to the *Sink*. The request contains nonce n_I and the *Sensor*'s id (I). Upon receiving the request, the *Sink* decrypts it with the key generated using the same method as the sensors used and creates a response which contains the received nonce n_I, a new nonce n_{AD}, the *Sink*'s id (S), the new session key used to encrypt data $K(I, S)$, and parameters P. The response is encrypted with the *Sensor*'s public key and sent back to the sensor. When the *Sensor* receives the parameters, it checks the nonce and starts the measurement process. When the process is finished, the acceleration data AD and nonce n_{AD} are encrypted with session key $K(I, S)$ and sent to the *Sink*. In this protocol, the AES algorithm is chosen for the encryption in the CTR mode and with 256 bits of the key.

Nonces n_I and n_{AD} are used to keep the messages fresh. Session key $K(I, S)$ which is generated by the *Sink* for each sensing session independently solves the problem of the distribution of the network key NK appearing in the *MID* protocol.

The security mechanisms and metrics of cryptographic primitives for TelosB motes are described in [28]. The metrics for the electricity current are taken from [40].

The summary of the analysed cryptographic protocols which ensure security on different levels is presented in Table 3.

Table 3: Security levels evaluated in the case study.

Security level	Confidentiality	Authentication
LOW	—	—
MID	AES-CTR-256	—
HIGH	AES-CTR-256	ECC-160

Table 4: The number of *RemoteSensing* events (sensing sessions) in a day for all scenarios.

	S.1	S.2	S.3	S.4	S.5	S.6	S.7	S.8	S.9	S.10
LOW level	4	0	0	24	0	0	4	20	20	8
MID level	0	4	0	0	24	0	0	0	4	8
HIGH level	0	0	4	0	0	24	20	4	0	8

6.2. Scenarios. The operation of the original version of the evaluated protocol consists of 4 sensing events per day. In this section, we introduce a situation in which the retrieved data needs to be more accurate. It takes place when the data retrieved from sensor nodes overcomes a threshold value. We defined ten scenarios in which the acceleration data is retrieved every hour for the subsequent 24 hours. Differences between scenarios are caused by various numbers of sensing events for each security level (LOW, MID, and HIGH) which are presented in Table 4. The introduction of the proposed scenarios largely increases the energy consumption of the network. Therefore, we want to evaluate several implementations with different levels of security and check their influence on energy consumption.

The first three scenarios refer to the original version of the protocol where 4 sensing events are conducted. The difference comes from different security levels. The other scenarios refer to the situation with 24 sensing events.

As a result of the analysis, we predict the maximal energy consumption of the node and the lifetime of the network represented as the battery level remaining after given months of operation. In our case study, we assume that each node has two AA batteries with 1200 mAh capacity and take the maximal energy consumption of nodes as the energy consumption of the network for lifetime prediction.

Table 5: The energy consumption and lifetime prediction for all scenarios.

Scenario	Energy consumption (J)	Lifetime prediction (days)
S.1	43.34	299
S.2	68.99	187
S.3	75.20	172
S.4	260.06	49
S.5	413.92	31
S.6	451.21	28
S.7	378.90	34
S.8	262.01	49
S.9	261.90	49
S.10	294.16	44

7. Results

Table 5 contains the energy consumption and lifetime prediction results for the presented scenarios (Table 4).

The *Energy consumption* column contains the maximal amount of energy consumed by one sensor during the execution of one scenario.

The *Lifetime prediction* column contains the number of days passed before the battery of any sensor is depleted (according to the assumptions from Section 6.2).

The results from Table 5 show that various numbers of remote sensing events can have significant influence on the lifetime of wireless sensor networks. The lifetime of the first three scenarios is about 6 times longer because the number of sensing events is equally increased.

However, the lifetime of scenario number 4 (24 unsecured sensing events) is almost twice as long as the lifetime of the most secured scenario (number 6). The last scenario with the same number of sensing events for all three security levels seems to be a good compromise. The results show that the designers of WSN protocols should search for balance between the acceptable energy consumption and security level.

Obtained results suggest that in some situations ensuring security at the expense of energy consumption is inevitable. However, before implementing designed solutions, there is a need to carefully examine considered environment and choose the option which fulfills given requirements best (in terms of, for instance, time or energy consumption).

The proposed approach can automatically answer the question what is the difference in performance between the created scenarios. Through this analysis you can make a trade-off between the means of information protection and the required performance. In addition, this analysis allows us to create scenarios to cope with a situation that will require greater efficiency or security. Such events may include a sudden and significant change of environmental factors, for example, sudden weather change that implies stronger requirements for efficiency. On the other hand, the detection of unexpected communication can be treated as an attack and the stronger security is applied. To summarize, the system can switch the operation mode when such cases appear (adaptable security [15]).

8. Conclusions

In this paper, the authors present the advanced communication module as an extension of the QoP-ML. Described module allows us to perform complex network examination as part of protocol performance analysis. It is utilized to include the time and energy analysis of communication steps.

Another contribution of this paper is adding the energy analysis and lifetime prediction modules to the QoP-ML. The modification of the QoP-ML is also implemented as an extension to AQoPA tool, utilized for automatic use in performance analysis.

The authors use the proposed communication model to perform an analysis of an existing wireless sensor network deployed on the Jindo Bridge in South Korea. The aim of such an analysis is to predict the lifetime of the existing network with additional security attributes. The authors introduce new scenarios in which the operation of the actual sensor network is modified and the number of sensing events is increased in order to collect more precise acceleration data. In the case study, ten scenarios with miscellaneous security levels are analysed.

The results allow us to draw conclusions about the influence of security attributes on time and energy consumption of wireless sensor networks. In the presented case study it is shown that the introduction of security attributes can have significant influence on network lifetime. Therefore, the designers of WSN protocols should search for balance between the required lifetime and security level. The QoP-ML along with the AQoPA tool have been created to accomplish this task.

Conflict of Interests

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

Acknowledgments

The research is partially supported by the grant "Reconcile: Robust Online Credibility Evaluation of Web Content" from Switzerland through the Swiss Contribution to the enlarged European Union. This work is supported by Polish National Science Centre Grant 2012/05/B/ST6/03364.

References

[1] P. K. Sahoo, "Efficient security mechanisms for mhealth applications using wireless body sensor networks," *Sensors*, vol. 12, no. 9, pp. 12606–12633, 2012.

[2] L. X. Hung, N. T. Canh, S. Lee, Y.-K. Lee, and H. Lee, "An energy-efficient secure routing and key management scheme for mobile sinks in wireless sensor networks using deployment knowledge," *Sensors*, vol. 8, no. 12, pp. 7753–7782, 2008.

[3] B. Ksiezopolski, Z. Kotulski, and P. Szalachowski, "Adaptive approach to network security," in *Computer Networks*, vol. 39 of *Communications in Computer and Information Science*, pp. 233–241, Springer, Wisla, Poland, 2009.

[4] P. Szalachowski, B. Ksiezopolski, and Z. Kotulski, "On authentication method impact upon data sampling delay in wireless sensor network," in *Computer Networks*, vol. 79 of *Communications in Computer and Information Science*, pp. 280–289, Springer, Ustron, Poland, 2010.

[5] IEEE 802.15.4 Standard, 2015, http://www.ieee802.org/15/pub/TG4.html.

[6] Zigbee Alliance, 2015, http://www.zigbee.org/.

[7] IEEE 802.15 wpan task group (tg6) body area networks, 2015, http://www.ieee802.org/15/pub/TG6.html.

[8] P. O. Kamgueu, E. Nataf, T. Djotio, and O. Festor, "Energy-based metric for the routing protocol in low-power and lossy network," in *Proceedings of the 2nd International Conference on Sensor Networks (SENSORNETS 2013)*, pp. 145–148, Barcelona, Spain, February 2013.

[9] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—a publish/subscribe protocol for wireless sensor networks," in *Proceedings of the 3rd IEEE/Create-Net International Conference on Communication System Software and Middleware (COMSWARE '08)*, pp. 791–798, January 2008.

[10] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: a top-down survey," *Computer Networks*, vol. 67, pp. 104–122, 2014.

[11] The ns-3 network simulator, 2008, http://www.nsnam.org/.

[12] D. Blouin and E. Senn, "CAT: an extensible systemlevel power consumption analysis toolbox for model-driven design," in *Proceedings of the 8th IEEE International NEWCAS Conference (NEWCAS '10)*, pp. 33–36, 2010.

[13] J. Li, H. Y. Zhou, D.-C. Zuo, K. M. Hou, H. P. Xie, and P. Zhou, "Energy consumption evaluation for wireless sensor network nodes based on queuing Petri net," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 262848, 11 pages, 2014.

[14] A. K. Agarwal and W. Wang, "On the impact of quality of protection in wireless local area networks with IP mobility," *Mobile Networks and Applications*, vol. 12, no. 1, pp. 93–110, 2007.

[15] B. Ksiezopolski and Z. Kotulski, "Adaptable security mechanism for dynamic environments," *Computers & Security*, vol. 26, no. 3, pp. 246–255, 2007.

[16] E. LeMay, W. Unkenholz, D. Parks, C. Muehrcke, K. Keefe, and W. H. Sanders, "Adversary-driven state-based system security evaluation," in *Proceedings of the 6th International Workshop on Security Measurements and Metrics (MetriSec '10)*, pp. 5:1–5:9, ACM, September 2010.

[17] S. Lindskog, *Modeling and tuning security from a quality of service perspective [Ph.D. thesis]*, Chalmers University of Technology, Gothenburg, Sweden, 2005.

[18] A. Luo, C. Lin, K. Wang, L. Lei, and C. Liu, "Quality of protection analysis and performance modeling in IP multimedia subsystem," *Computer Communications*, vol. 32, no. 11, pp. 1336–1345, 2009.

[19] C. S. Ong, K. Nahrstedt, and W. Yuan, "Quality of protection for mobile multimedia applications," in *Proceedings of the International Conference on Multimedia and Expo (ICME '03)*, vol. 2, pp. II-137–II-140, Baltimore, Md, USA, July 2003.

[20] D. C. Petriu, C. M. Woodside, D. B. Petriu et al., "Performance analysis of security aspects in UML models," in *Proceedings of the 6th International Workshop on Software and Performance (WOPS '07)*, pp. 91–102, ACM, New York, NY, USA, February 2007.

[21] P. A. Schneck and K. Schwan, "Authenticast: an adaptive protocol for high-performance, secure network applications," Tech. Rep., Georgia Institute of Technology, 1997.

[22] Y. Sun and A. Kumar, "Quality-of-protection (qop): a quantitative methodology to grade security services," in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pp. 394–399, IEEE Computer Society, Beijing, China, June 2008.

[23] C. J. F. Cremers, "The scyther tool: verification, falsification, and analysis of security protocols," in *Aarti Gupta and Sharad Malik*, vol. 5123 of *Lecture Notes in Computer Science*, pp. 414–418, Springer, Princeton, NJ, USA, 2008.

[24] A. Armando, D. Basin, Y. Boichut et al., "The AVISPA tool for the automated validation of internet security protocols and applications," in *Computer Aided Verification: 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, July 6–10, 2005. Proceedings*, K. Etessami and S. K. Rajamani, Eds., vol. 3576 of *Lecture Notes in Computer Science*, pp. 281–285, Springer, Berlin, Germany, 2005.

[25] B. Blanchet, "Automatic proof of strong secrecy for security protocols," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 86–100, Berkeley, Calif, USA, May 2004.

[26] J. Jürjens, *Secure Systems Development with UML*, Springer, Berlin, Germany, 2005.

[27] B. Ksiezopolski, T. Zurek, and M. Mokkas, "Quality of protection evaluation of security mechanisms," *The Scientific World Journal*, vol. 2014, Article ID 725279, 18 pages, 2014.

[28] I. Mansour, D. Rusinek, G. Chalhoub, P. Lafourcade, and B. Ksiezopolski, "Multihop node authentication mechanisms for wireless sensor networks," in *Ad-Hoc, Mobile, and Wireless Networks*, S. Guo, J. Lloret, P. Manzoni, and S. Ruehrup, Eds., vol. 8487 of *Lecture Notes in Computer Science*, pp. 402–418, Springer International Publishing, Benidorm, Spain, 2014.

[29] S. Jang, H. Jo, S. Cho et al., "Structural health monitoring of a cable-stayed bridge using smart sensor technology: deployment and evaluation," *Smart Structures and Systems*, vol. 6, no. 5-6, pp. 439–459, 2010.

[30] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheimy, "Security SLAs for federated Cloud services," in *Proceedings of the 6th International Conference on Availability, Reliability and Security (ARES '11)*, pp. 202–209, Vienna, Austria, August 2011.

[31] D. Chen and P. K. Varshney, "QoS support in wireless sensor networks: a survey," in *Proceedings of the International Conference on Wireless Networks (ICWN '04)*, vol. 13244, pp. 227–233, Las Vegas, Nev, USA, 2004.

[32] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, *A Framework for QoS-Based Routing in the Internet*, RFC, 1998.

[33] B. Bhuyan, H. K. D. Sarma, N. Sarma, A. Kar, and R. Mall, "Quality of service (QoS) provisions in wireless sensor networks and related challenges," *Wireless Sensor Network*, vol. 2, no. 11, pp. 861–868, 2010.

[34] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 3, pp. 1380–1387, Anchorage, Alaska, USA, April 2001.

[35] B. Ksiezopolski, "QoP-ML: quality of protection modelling language for cryptographic protocols," *Computers & Security*, vol. 31, no. 4, pp. 569–596, 2012.

[36] J. W. Backus, "The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GRAMM

conference," in *Proceedings of the International Conference on Information Processing (ICIP '59)*, UNESCO, Paris, France, June 1959.

[37] Zigbee Alliance, "Zigbee specification," Tech. Rep., 2008.

[38] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[39] Aqopa, 2014, http://qopml.org/aqopa/.

[40] A. Prayati, C. Antonopoulos, T. Stoyanova, C. Koulamas, and G. Papadopoulos, "A modeling approach on the TelosB WSN platform power consumption," *Journal of Systems and Software*, vol. 83, no. 8, pp. 1355–1363, 2010.