Research Article

Congestion Control and Traffic Scheduling for Collaborative Crowdsourcing in SDN Enabled Mobile Wireless Networks

Dawei Shen,¹ Wei Yan,¹ Yuhuai Peng ^{1,2} Yanhua Fu,³ and Qingxu Deng ^{1,2,4}

¹College of Computer Science and Engineering, Northeastern University, Shenyang 110819, China

²Key Laboratory of Vibration and Control of Aero-Propulsion System of Ministry of Education, Northeastern University, Shenyang 110819, China

³College of Jangho Architecture, Northeastern University, Shenyang 110819, China

⁴State Key Laboratory of Rolling and Automation, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Yuhuai Peng; pengyuhuai@mail.neu.edu.cn
and Qingxu Deng; dengqx@mail.neu.edu.cn

Received 8 September 2017; Accepted 3 January 2018; Published 21 February 2018

Academic Editor: Kuan Zhang

Copyright © 2018 Dawei Shen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Currently, a number of crowdsourcing-based mobile applications have been implemented in mobile networks and Internet of Things (IoT), targeted at real-time services and recommendation. The frequent information exchanges and data transmissions in collaborative crowdsourcing are heavily injected into the current communication networks, which poses great challenges for Mobile Wireless Networks (MWN). This paper focuses on the traffic scheduling and load balancing problem in software-defined MWN and designs a hybrid routing forwarding scheme as well as a congestion control algorithm to achieve the feasible solution. The traffic scheduling algorithm first sorts the tasks in an ascending order depending on the amount of tasks and then solves it using a greedy scheme. In the proposed congestion control scheme, the traffic assignment is first transformed into a multiknapsack problem, and then the Artificial Fish Swarm Algorithm (AFSA) is utilized to solve this problem. Numerical results on practical network topology reveal that, compared with the traditional schemes, the proposed congestion control and traffic scheduling schemes can achieve load balancing, reduce the probability of network congestion, and improve the network throughput.

1. Introduction

In recent years, crowdsourcing have received extensive attention from industry and academia, which was originally proposed by American journalist Jeff Howe in 2006. Crowdsourcing means that tasks performed by employees in a company or institution before will be outsourced to the unspecific public networks in a free and voluntary form. Tasks of crowdsourcing are usually undertaken by the individual. But if the people involved need to collaborate to complete the task, there may appear in form of individual production dependent on open source. Many tasks cannot be achieved through a simple algorithm, such as image labeling, commodity evaluation, and entity recognition. These kinds of problems are difficult for machines to handle but can be done with crowdsourcing. In crowdsourcing it publishes tasks directly to the Internet and gathers unknown people on the Internet to solve

problems that are difficult to deal with by traditional computers alone, such as Wikipedia, reCAPTCHA [1], tagged images, and language translations. According to the different forms of public participation in crowdsourcing, it can be divided into collaborative crowdsourcing and crowdsourcing contest. In collaborative crowdsourcing, tasks require collaboration between the masses, but people who perform tasks usually do not have rewards. Crowdsourcing can effectively solve machine-hard tasks by leveraging machine and a large group of people on the web.

Software-Defined Networking (SDN) refers to a new network architecture developed from OpenFlow technology [2]. SDN technology can be programmed by software to control the data forwarding and ultimately achieve the purpose of free data transfer control. SDN technology has outstanding advantages in flow control; therefore, we hope to use SDN technology to solve the problem of congestion control and

traffic scheduling [3] in crowdsourcing-based Mobile Wireless Networks (MWN).

Currently, a number of crowdsourcing-based mobile applications have been applied in mobile networks and Internet of Things (IoT), targeted at real-time services and recommendation, for example, Uber, Elance, Amazon, and Airbnb. These frequent information exchanges and data transmissions are heavily injected into the current communication networks [4], which poses great challenges for congestion control and traffic scheduling problem [5] in Mobile Wireless Networks. To solve the emerging challenges, this paper focuses on the traffic scheduling and load balancing problem in software-defined Mobile Wireless Networks for collaborative crowdsourcing. This paper first presents a network model towards traffic engineering problem and then designs a hybrid routing forwarding scheme as well as a congestion control algorithm to achieve the feasible solution. To validate the performance of the proposals, a lot of simulation experiments are carried out.

The rest of this paper is organized as follows. Related work in recent years is reviewed in Section 2. The network model is then formulated in Section 3. In Section 4, design of congestion control and traffic scheduling scheme are presented in detail. Simulation results and analysis are discussed in Section 5. Finally, conclusions are given in Section 6.

2. Related Work

At present, some researchers have summarized the research work of crowdsourcing from different perspectives.

Yuen et al. in [6] summarized the progress of crowdsourcing from applications, algorithms, performance, and data sets. Kittur et al. in [7] explained the challenges of crowdsourcing in 12 aspects such as synchronous collaboration, real-time response, and dynamic machines. Doan et al. in [8] reviewed the crowdsourcing system applied on the world wide web and summarized the crowdsourcing system according to the problem type and the way of collaboration. Zhao and Zhu in [9] reviewed crowdsourcing research from four perspectives: information, technology, the public, and organization. Kittur et al. in [10] have studied how to decompose complex tasks and how to integrate workers' answers to perform initial tasks and proposed a MapReduce framework to achieve the decomposition of tasks. However, their method is only suitable for specific types of tasks, and the general effect is unsatisfying. Scalability still needs to be solved. References [11–13] focus on the technology of combining machine and human with the join operation in crowdsourcing environment, which first filters the problem through the machine algorithm and then assigns the remaining problems to the workers. The authors of [12] used the transitive relationships of entities to further reduce the number of tasks, thereby saving the cost. Lofi et al. in [14] reduced the cost of the task by preprocessing data sets containing missing data through the "error model" and getting the answers from workers. Sakamoto et al. in [15] studied the ways in which crowdsourcing participants often interact in different task types. Heer et al. in [16] studied how to carry out a survey and found

that the design interface was more suitable for crowdsourcing workers through questionnaires. The authors in [17] proposed a method based on random map generation and messaging task allocation. The limitation of this method is that it can only be used for a specific type of task to the difficulty of the task. However, there are various types of task crowdsourcing platform, and some tasks need special professional knowledge, such as language translation task. Liu et al. in [18] implemented a data analysis system to ensure the quality of the results as the main goal, first through forecasting model number assigned tasks and, then, in the process of task execution, through online quality assessment results to determine whether to terminate the task ahead of time, thus saving cost and time. The authors in [19] proposed a new workers' model in crowdsourcing. Through this model, the workers' quality can be computed accurately and timely. For big data tasks, the number of tasks affects the overall cost of the tasks. The number of tasks can be reduced by effectively designing the task, thus saving the task cost. Marcus et al. in [20] proposed the strategy to transform the problem of each task into multiple subproblems. But when a task contains a large number of subproblems, the price of task needs to improve. Otherwise, it will be easy to cause only a small number of workers selected task. That is to say, even though such an approach reduces the number of tasks, the overall cost of the task is not guaranteed to be reduced. The authors of [21] presented a comprehensive system model of Crowdlet that defines the task, worker arrival, and worker ability models. In [22], the authors designed an approximate task allocation algorithm that is near optimal with polynomial-time complexity and used it as a building block to construct the whole randomized auction mechanism. Compared with deterministic auction mechanisms, the proposed randomized auction mechanism increases the diversity in contributing users for a given sensing job. The authors of [23] presented a new participant recruitment strategy for vehicle-based crowdsourcing. This strategy guarantees that the system can perform well using the currently recruited participants for a period of time in the future. The authors in [24] focused on a more realistic scenario where users arrive one by one online in a random order. The authors in [25] focused on the problem of how to efficiently distribute a crowdsourcing task and recruit participants based on D2D communications. In [26], existing definitions of crowdsourcing were analyzed to extract common elements and to establish the basic characteristics of any crowdsourcing initiative. Based on these existing definitions, an exhaustive and consistent definition for crowdsourcing is presented and contrasted in eleven cases. In [27], the authors defined traffic engineering as a large-scale network project to solve the performance evaluation and network optimization in the network. In [28], traffic engineering has been further explained, and the traffic engineering is a route optimization method to improve the quality of network service by avoiding the link congestion in the network.

3. Network Model

There are a number of possible next hops that may occur after the crowdsourcing task has selected the assignment object in

FIGURE 1: The illustration of the next hop selection in our network model.

the mobile wireless network, and different next hop options affect the load balancing in the network. As shown in Figure 1, if the crowdsourcing task source node in the figure forwards the mission to the destination node through the next hop 1, the maximum link utilization rate in the network is 0.6. If the next hop 2 is chosen to forward the task assignment, then the maximum link utilization in the network is 0.4. Therefore, the SDN controller in the network needs to calculate the next hop periodically to achieve the load balancing in the network.

Taking into account the fact that the routing network already exists in the current mobile wireless network, it requires a lot of manpower and resources to replace all the wireless nodes for the SDN node [29]. Therefore, we consider the SDN node in the mobile wireless network part of the configuration of the scenario. We assume the nodes in the mobile wireless network run the OSPF protocol, so the SDN controller can collect the load information of the links in the network. And the SDN nodes can obtain the link utilization rate of all the links in the network. When the crowdsourcing task leaves the SDN node, it may pass through other SDN nodes on its forwarding path. These nodes can also have multiple next hops, as shown in Figure 2, where the yellow nodes represent SDN nodes, while the white nodes represent non-SDN nodes. In addition, the solid line represents the forwarding path, and the dotted line represents the possible forwarding paths. It is assumed that the current task node 1 is an SDN node, and it selects node 2 as its next hop. There is also another SDN node 4 on the forwarding path, and the next hop of node 4 may be node 5 or node 6. Through the coordination of multiple SDN nodes distributed in the Mobile Wireless Networks, we can have multiple possible forwarding paths to carry the crowdsourcing tasks to achieve load balancing for global networks.

Therefore, we first need to find out all the possible paths that the package task forwards. We use the tree structure to build all possible forwarding paths [30]. First, we construct the source node of the task as the root of the tree. Each node in the tree can be divided into SDN nodes and non-SDN nodes. If it is an SDN node, then it can have multiple child nodes; otherwise, it only has a child node. We assume that when the package task is forwarded, each node in the network will inject an identity packet of the current node. When passing through the SDN node, we check this identity packet and remove the branch path containing the nodes that already exist in the current identity packet, ensuring that the

loopback is not generated when the packet task is forwarded. In Figure 2, for example, the tree structure of all possible forwarding paths can be constructed as shown in Figure 3.

In what we described above, there is only one crowdsourcing task in the wireless sensor network, but, in reality, there can be multiple crowdsourcing task in the network [31].

Formally, the wireless sensor network can be modeled as $G(V, E)$, with the node set $V(1, 2, \dots, v)$ and link set $E(e_{12}, e_{13}, \dots, e_{ij})$.

Assume that there is no interference between nodes and links. Suppose T is the crowdsourcing tasks matrix, and the task set is $N(1_2^1, 2_2^1, \dots, n_d^s)$ (s is the crowdsourcing task source node and d is the crowdsourcing task destination node). And the amount of task is $L(n_d^s)$. Define $C(e)$ as the link capacity. Define link utilization as $U(e)$, which can be formulated as in

$$U(e) = \frac{\sum_e L(n_d^s)}{C(e)}. \quad (1)$$

Define that when a crowdsourcing task passes through a node, all possible forwarding paths are added to set $P_v^{n_d^s}$. There are two scenarios in the mobile wireless network: when a number of crowdsourcing tasks pass through non-SDN nodes, they can only be forwarded in accordance with the OSPF protocol to its next hop. And when multiple crowdsourcing tasks pass through SDN nodes, we have multiple possible forwarding paths. In the mobile wireless network, we can only control SDN nodes. Therefore, when the crowdsourcing task traffic passes through the SDN node, the problem we need to solve is as follows: given $G, N, C(e)$ and P , how we schedule the task N over the path P with the path capacity $C(e)$ to minimize the maximum link utilization U , then achieving load balancing. We describe it as problem $S(N, L(n_d^s), C(e), P, U)$.

Given the definitions above, the problem can be formalized as follows:

$$\text{Minimize } U \quad (2)$$

$$\text{Subject to: } \sum_e L(n_d^s) \leq U \cdot C(e) \quad \forall e \in E \quad (3)$$

$$P^{L(n_d^s)} \geq 0 \quad \forall P \quad (4)$$

$$L(n_d^s) \geq 0 \quad \forall n_d^s \in N. \quad (5)$$

Formula (3) indicates that the size of the task on any link is less than or equal to the maximum link utilization in the

FIGURE 2: Multipath selection illustration.

FIGURE 3: All possible path tree diagram.

network multiplied by the link capacity. Formula (4) indicates that the amount of task on any forwarding path should be nonnegative. Formula (5) indicates that task should be non-negative.

4. Congestion Control and Traffic Scheduling Schemes

4.1. Design of Hybrid Routing and Forwarding Algorithm. In our model, we divided the nodes in the mobile wireless network into two categories: SDN nodes and non-SDN nodes. When the crowdsourcing task traffic passes through the non-SDN node, we use OSPF protocol to perform the next hop routing. When the crowdsourcing task traffic passes through the SDN node, we describe this as problem $S(N, L(n_d^s), C(e), P, U)$. There is a special case of problem S , where $P = 1$ and $U = 1$. The problem $S(N, L(n_d^s), C(e), 1, 1)$ is NP, and we can reduce the well-known 0-1 knapsack problem [32] to this problem. Therefore, $S(N, L(n_d^s), C(e), 1, 1)$ is NP-hard. Thus, the more general problem $S(N, L(n_d^s), C(e), P, U)$ is also NP-hard. This means the computation cannot be completed in a reasonable time for large networks. Therefore, we develop a heuristic algorithm for this problem with polynomial-time complexity.

On the algorithm we make the following assumptions:

- (1) SDN control center can be aware of the relevant information in the network correctly and timely.
- (2) Network topology is stable in a short time, and we do not consider the interference of wireless networks.
- (3) All the nodes are running standard OSPF protocol nodes in the mobile wireless network in addition to SDN nodes.
- (4) Mobile wireless network has only one SDN controller.
- (5) In the process of routing, SDN nodes select only one path to forward when processing a crowdsourcing task flow.
- (6) The task flow is forwarded hop by hop.

In this case, we assume that none of the links in the network will be congested, and there will not be a number of crowdsourcing task traffic on a link exceeding the capacity of the link. Therefore, when the SDN node forwards the crowdsourcing task, we can sort the crowdsourcing tasks according to the task load. Then, according to the greedy algorithm, the crowdsourcing task is distributed to the corresponding link, which makes the value of maximum link utilization in the network minimum.

The hybrid routing and forwarding algorithm is given in Algorithm 1.

Since we define the utilization of the link as the ratio of the link capacity of the data flow on the current link, if the data flow is far greater than our link capacity, our link utilization will be greater than 1. So the network's maximum link utilization is greater than 1, which is contrary to the idea of load balancing in traffic engineering. Therefore, our crowdsourcing task traffic matrix cannot be generated arbitrarily; as

Algorithm for hybrid routing and forwarding

- (1) **Begin**
- (2) **Input:** mobile wireless network topology graph $G(V, E)$, crowdsourcing task flow matrix T ;
- (3) **for** each row in T **do**
- (4) **If** v is non-SDN node **then**
- (5) Assign the task flow to its next hop forwarding link;
- (6) **repeat**
- (7) $\forall v \in V, v++$;
- (8) **until** all non-SDN nodes are traversed;
- (9) **If** v is SDN node **then**
- (10) Sort the task in ascending order according to the load of the task flow;
- (11) Compute all possible forwarding path P ;
- (12) Use the greedy algorithm to assign task to its next hop forwarding link;
- (13) **repeat**
- (14) $\forall v \in V, v++$;
- (15) **until** all SDN nodes are traversed;
- (16) Compute link utilization on all links in the network. Get the maximum link utilization U ;
- (17) Update the crowdsourcing task traffic matrix T to T' ;
- (18) **If** $U' \geq U$ **then**
- (19) $U \leftarrow U'$;
- (20) Return to the third step;
- (21) **Output:** the maximum link utilization U ;
- (22) **End.**

ALGORITHM 1: Hybrid routing and forwarding algorithm.

for the task flow size, according to the method described in literature [33], we generate the formula as follows:

$$d_{ij} = \sigma_i \sum_{\{t|(i,t) \in E\}} c_{(i,t)} \frac{\sum_{\{t|(t,j) \in E\}} c_{(t,j)}}{\sum_{\{(m,n)|(m,n) \in E\}} c_{(m,n)} - \sum_{\{t|(i,t) \in E\}} c_{(i,t)}} \quad (6)$$

$i, j \in V.$

In formula (6), d_{ij} represents the size of the traffic flow from the source node i to the destination node j , σ_i represents a random number in an interval $[0, 1]$, $c(i, t)$ represents the link capacity between the source node i and its neighboring node t , $c(t, j)$ is the link capacity between destination node j and its neighboring node t , and $c(m, n)$ represents the capacity on the link (m, n) . We generate 40 sets of crowdsourcing task flow matrices as simulation data according to formula (6). According to the above conditions, we have simulated the proposed algorithm.

4.2. Design of Congestion Control Algorithm. As mentioned above, we assume that there will be no congestion in the mobile wireless network, but in fact congestion is inevitable in the process of mass crowdsourcing. Therefore the problem $S(N, L(n_d^s), C(e), P, U)$ should be $S(N, L(n_d^s), C(e), P_e, 1)$, because the maximum utilization of the link is 1 and P_e is the first link of the possible path P . In this case, when an SDN node is forwarding the crowdsourcing task, it needs to

select a subset of its task set $N \{1, 2, \dots, n\}$ first. Then, these subtasks will be assigned to the possible forwarding link P_e , with the maximum value of assigned tasks under the limitation of each link. It is a multiknapsack problem. Multiple Knapsack Problem (MKP) refers to the selection of a subset of items in an item collection $N \{1, 2, \dots, n\}$ to be loaded into $M \{1, 2, \dots, m\}$ backpack. The purpose is to maximize the total value of selected items, with the total capacity not exceeding the volume of each backpack. Here we use the AFSA algorithm in [34] to solve this problem. Artificial Fish Swarm Algorithm (AFSA) is a new intelligent optimization algorithm for biomimetic group. Artificial fish can make AFSA better intelligent and suitable for solving large-scale complex optimization problems. We assign the crowdsourcing tasks as many as possible to the link without exceeding the link capacity. According to this heuristic rule, if we want to assign the task i to the link j , there are two possibilities. One is the link capacity $C(j) < L(i)$, and we cannot assign the task to the link. The other one is the link capacity $C(j) \geq L(i)$. Let $C_r(e)$ represent the remaining capacity of the link e . There are two conditions: (1) $C_r(j) \geq L(i)$, if task i is never assigned to any link, then task i is assigned to the link j , and $C_r(j) = C_r(j) - L(i)$; if task i was assigned to link k ($k \neq j$), we firstly execute TakeOut(i, k) (TakeOut(i, k) which means taking the task i out of link k , and then $C_r(k) = C_r(k) + L(i)$). Then we assign the task i to the link j , and the remaining capacity of the link j decreases $L(i)$. (2) $C_r(j) < L(i)$, we


```

Algorithm for Congestion control
(1) Begin
(2) Input: mobile wireless network topology graph  $G(V, E)$ , crowdsourcing task flow matrix  $T$ .
(3) for each row in  $T$  do
(4)   If  $v$  is non-SDN node then
(5)     Assign the task flow to its next hop forwarding link;
(6)   repeat
(7)      $\forall v \in V, v++$ ;
(8)   until all non-SDN nodes are traversed;
(9)   If  $v$  is SDN node then
(10)    Compute all possible forwarding path  $P_e$ ;
(11)    Compute the link capacity  $C_r(P_e)$ ;
(12)    Use the AFSA algorithm to assign task flow to its next hop forwarding link;
(13)  repeat
(14)     $\forall v \in V, v++$ ;
(15)  until all SDN nodes are traversed;
(16) Update the crowdsourcing task traffic matrix  $T$  to  $T'$ ,  $T_m = T_m + 1$ . Return to the third step.
(17) Output: Number of times:  $T_m$ .
(18) End.

```

ALGORITHM 2: Congestion control algorithm.

execute $\text{TakeOut}(p, j)$ (p is any task that is assigned to the link j) until $C_r(j) \geq L(i)$, and then we execute (1). The artificial fish is always kept in a feasible solution and close to the bound boundary. The effective optimization of artificial fish under the guidance of behavior strategy was carried out by artificial fish feeding, rear-ending, and clustering.

Since we can only control SDN nodes in the network, we will take the traffic of non-SDN nodes in the forwarding link first. The remaining capacity of the link is the backpacking capacity of our multibackpack problem. We also need to assume that the crowdsourcing task flow cannot be split. Assume that when the number of tasks on a link exceeds the link capacity of the link, it causes the task to be discarded and needs to be reposted. Define T_m as the number of times that the crowdsourcing task has been forwarded. Finally, we evaluate our congestion control algorithm by calculating the link throughput. We use formula (7) to compute the throughput of the network:

$$\text{Throughput} = \frac{\sum L(n)}{T_m}. \quad (7)$$

The congestion control algorithm is given in Algorithm 2.

5. Number Results and Analysis

We mainly use VS2010 to complete the simulation, which is coded in C/C++. We use the wireless network standard based on IEEE 802.11b [35] to build our mobile wireless network with a maximum bandwidth of 11 Mbps, which means the maximum link capacity can be set to 11 M. Here we use

the method described in [36] to set the link capacity in mobile wireless network. First, divide all the nodes into two categories according to the degree of each node, A class node represents those nodes whose degree is less than 3, and B class represents the set of other degrees of nodes. If a link has two nodes in the B class node set, then the link capacity is 11 M; if there is a node in the link in the A class node, set the link capacity of 6 M.

Simulation of the mobile wireless network topologies are shown in Figures 4 and 5, where yellow nodes represent SDN nodes and white nodes represent non-SDN nodes, and we simulate the experiment by increasing the number of SDN nodes gradually.

For the hybrid routing and forwarding algorithm, we compare the network with no SDN nodes by increasing the amount of SDN nodes in the network, which is the network that we assume all nodes are forwarded according to the OSPF protocol. From (a) to (d) compare the maximum link utilization between our proposed hybrid routing forwarding scheme and the OSPF protocol by increasing the number of SDN nodes. The simulation results are shown in Figures 6 and 7.

Figures 6 and 7 present analyses of the maximum utilization with different SDN nodes deployment in Topology 1 and Topology 2. The simulation results are shown in Figures 6 and 7, the y -axis represents the maximum link utilization, and the x -axis represents the number of crowdsourcing task flow matrixes. We can see intuitively that with the increase in SDN nodes the overall trend of maximum link utilization is decreasing in the mobile wireless network from the simulation results in Figures 6 and 7. However, it can be seen in the

FIGURE 4: Mobile wireless network topology 1.

FIGURE 5: Mobile wireless network topology 2.

figure that when the SDN nodes in the network are relatively few, the maximum link utilization in the network obtained from the hybrid routing and forwarding algorithm is almost the same compared with the OSPF routing algorithm. This is because the SDN controller can only control the SDN nodes to manipulate the traffic in the network. When the SDN nodes in the network are few, the traffic in the whole network becomes uncontrollable. Although traffic through the SDN

nodes can be controlled, the maximum utilization rate of the local link in the network is reduced, and the local network can achieve load balancing, and it is difficult to achieve load balancing for the whole network. In addition, by comparing Topology 1 and Topology 2, the benefits of deploying SDN nodes will become more apparent as the number of nodes in the network increases and the network topology becomes more complex.

FIGURE 6: Comparison of the maximum utilization with different SDN nodes deployment in Topology 1.

For the congestion control algorithm, we gradually increase the number of SDN nodes in the mobile wireless network. We calculate the throughput of the network through formula (7) and then compare it. The simulation results are shown in Figure 8.

Figure 8 shows the analysis of throughput for different SDN nodes deployment in Topology 1 and Topology 2. It can be observed that the throughput performance of Topology 1 and Topology 2 are both better with SDN nodes increasing. From the comparison results in Figure 8, it can be concluded that our congestion control algorithm can effectively improve the network throughput.

6. Conclusion

At present, massive crowdsourcing-based mobile applications have been applied in mobile networks and IoT, targeted at real-time services and recommendation. The frequent information exchanges and data transmissions in collaborative crowdsourcing are continually injected into the current communication networks, which poses great challenges in Mobile Wireless Networks (MWN). This paper focuses on the traffic scheduling and load balancing problem in software-defined MWN and designs a greedy heuristic algorithm as well as a congestion control algorithm to achieve feasible solutions. The proposed traffic scheduling algorithm sorts the

tasks in ascending order according to the amount of tasks and then solves them using the greedy scheme. The packet task is assigned to the corresponding link for forwarding, so that the maximum link utilization in MWN is the least. In the proposed congestion control scheme, the traffic assignment is transformed into a multiknapsack problem, and then the AFSA algorithm is employed to solve the problem. The node selects a subset in its feasible task set and assigns it to the p links, which makes the maximum amount of tasks allocated without exceeding the limited capacity of each link. The simulation results demonstrate that compared with the traditional schemes the proposed congestion control and traffic scheduling methods can achieve load balancing, reduce the probability of network congestion, and improve the network throughput.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61501105), the National Key Research and Development Program of China (2016YFC0801607),

FIGURE 7: Comparison of the maximum utilization with different SDN nodes deployment in Topology 2.

FIGURE 8: Comparison of throughput with different SDN nodes deployment.

the Fundamental Research Funds for the Central Universities (N150404018, N130304001, N150401002, and N161608001), and the Open Research Fund from the State Key Laboratory of Rolling and Automation, Northeastern University, Grant no. 2017RALKFKT002.

References

- [1] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: human-based character recognition via web security measures," *American Association for the Advancement of Science: Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.
- [2] N. McKeown, T. Anderson, H. Balakrishnan et al., "OpenFlow: enabling innovation in campus networks," *ACM Sigcomm Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
- [3] R. Wang, D. Butnariu, and J. Rexford, "Open Flow-based server load balancing gone wild," in *Proceedings of The 11th USENIX Conference on Hot Topics in Management of Internet, Cloud, And Enterprise Networks And Services*, pp. 12–12, USENIX Association, 2011.

- [4] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "LoTAD: long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web Information Systems*, pp. 1–23, 2017.
- [5] Z. Ning, F. Xia, N. Ullah et al., "Vehicular social networks: enabling smart mobility," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 16–55, 2017.
- [6] M. C. Yuen, I. King, and K. S. Leung, "A Survey of crowdsourcing systems," in *Proceedings of the IEEE Third International Conference on Social Computing (SocialCom)*, pp. 766–773, IEEE, Boston, MA, USA, October 2012.
- [7] A. Kittur, J. V. Nickerson, M. Bernstein et al., "The future of crowd work," in *Proceedings of the 2013 Conference on Computer supported cooperative work*, vol. 3-4, pp. 1301–1318, Social Science Electronic Publishing, San Antonio, Texas, USA, February 2013.
- [8] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," *Communications of the ACM*, vol. 54, no. 4, pp. 86–96, 2011.
- [9] Y. Zhao and Q. Zhu, "Evaluation on crowdsourcing research: Current status and future direction," *Information Systems Frontiers*, vol. 16, no. 3, pp. 417–434, 2014.
- [10] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "CrowdForge: Crowdsourcing complex work," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST'11*, pp. 43–52, USA, October 2011.
- [11] J. Wang, T. Kraska, M. J. Franklin et al., "CrowdER: crowdsourcing entity resolution," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1483–1494.
- [12] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng, "Leveraging transitive relations for crowdsourced joins," in *Proceedings of the 2013 ACM SIGMOD Conference on Management of Data, SIGMOD 2013*, pp. 229–240, USA, June 2013.
- [13] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *Proceedings of the 21st Annual Conference on World Wide Web, WWW'12*, pp. 469–478, France, April 2012.
- [14] C. Lofi, K. El Maarry, and W.-T. Balke, "Skyline queries over incomplete data - Error models for focused crowd-sourcing," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8217, pp. 298–312, 2013.
- [15] Y. Sakamoto, Y. Tanaka, L. Yu, and J. V. Nickerson, "The Crowdsourcing design space," in *Proceedings of the International Conference on Foundations of Augmented Cognition: Directing the Future of Adaptive Systems*, vol. 6780 of *Lecture Notes in Computer Science*, pp. 346–355, Springer-Verlag, 2011.
- [16] J. Heer and M. Bostock, "Crowdsourcing graphical perception: Using mechanical Turk to assess visualization design," in *Proceedings of the 28th Annual CHI Conference on Human Factors in Computing Systems, CHI 2010*, pp. 203–212, USA, April 2010.
- [17] D. Karger R, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 1953–1961, Curran Associates Inc., 2011.
- [18] X. Liu, M. Lu, B. C. Ooi et al., "CDAS: A crowdsourcing data analytics system," *Proceedings of the VLDB Endowment*, vol. 5, no. 10, pp. 1040–1051, 2012.
- [19] J. Feng, G. Li, H. Wang et al., "Incremental quality inference in crowdsourcing," in *Proceedings of the International Conference on Database Systems for Advanced Applications, Lecture Notes in Computer Science*, pp. 453–467, Springer International Publishing, 2014.
- [20] A. Marcus, E. Wu, D. Karger, S. Madden, and R. Miller, "Human-powered sorts and joins," *Proceedings of the VLDB Endowment*, vol. 5, no. 1, pp. 13–24, 2011.
- [21] L. Pu, X. Chen, J. Xu, and X. Fu, "Crowdlet: Optimal worker recruitment for self-organized mobile crowdsourcing," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016, USA, April 2016*.
- [22] J. Li, Y. Zhu, Y. Hua, and J. Yu, "Crowdsourcing sensing to smartphones: A randomized auction approach," in *Proceedings of the 23rd IEEE International Symposium on Quality of Service, IWQoS 2015*, pp. 219–224, USA, June 2015.
- [23] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM 2015*, pp. 2542–2550, May 2015.
- [24] D. Zhao, X.-Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *Proceedings of the 33rd IEEE Conference on Computer Communications, IEEE INFOCOM 2014*, pp. 1213–1221, May 2014.
- [25] Y. Han and H. Wu, "Minimum-Cost Crowdsourcing with Coverage Guarantee in Mobile Opportunistic D2D Networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2806–2818, 2017.
- [26] E. Estellés-Arolas and F. González-Ladrón-De-Guevara, "Towards an integrated crowdsourcing definition," *Journal of Information Science*, vol. 38, no. 2, pp. 189–200, 2012.
- [27] Y. Lee and B. Mukherjee, "Traffic engineering in next-generation optical networks," *Communications Surveys & Tutorials IEEE*, vol. 6, no. 3, pp. 16–33.
- [28] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of internet traffic engineering," IETF RFC 3272, Academy of Science, Engineering and Technology, 2002.
- [29] Z. Ning, Q. Song, Y. Yu, Y. Lv, X. Wang, and X. Kong, "Energy-aware cooperative and distributed channel estimation schemes for wireless sensor networks," *International Journal of Communication Systems*, vol. 30, no. 5, Article ID e3074, 2017.
- [30] C.-Y. Chu, K. Xi, M. Luo, and H. J. Chao, "Congestion-aware single link failure recovery in hybrid SDN networks," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM 2015*, pp. 1086–1094, Hong Kong, May 2015.
- [31] Z. Ning, X. Hu, Z. Chen et al., "A cooperative quality-aware service access system for social internet of vehicles," *IEEE Internet of Things Journal*, pp. 1–1.
- [32] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, Calif, USA, 1979.
- [33] K. Li, S. Wang, S. Xu, and X. Wang, "ERMAO: An enhanced intradomain traffic engineering approach in LISP-capable networks," in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference: "Energizing Global Communications", GLOBECOM 2011, USA, December 2011*.
- [34] X. MA and Q. LIU, "Artificial fish swarm algorithm for multiple knapsack problem," *Journal of Computer Applications*, vol. 30, no. 2, pp. 469–471, 2010.

- [35] Z. Ning, F. Xia, N. Ullah, X. Kong, and X. Hu, "Vehicular Social Networks: Enabling Smart Mobility," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 16–55, 2017.
- [36] H. Yang, *IP routing based on topology of network traffic engineering research*, University of electronic science and technology, 2013.