2011 IEEE 15th International Symposium on Consumer Electronics

A Proposal of a Traffic Control Method with Consumed Bandwidth Estimation for Real-Time Applications in Wireless Mesh Networks

Sritrusta Sukaridhoto†, Nobuo Funabiki†, Toru Nakanishi†
†Department of Communication Network Engineering, Okayama University
3-1-1 Tsushimanaka, Okayama 700-8530, Japan
Email: dhoto@sec.cne.okayama-u.ac.jp, {funabiki,nakanisi}@cne.okayama-u.ac.jp

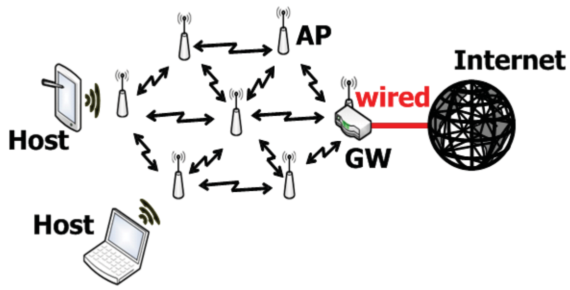


Fig. 1. WIMNET.

***Abstract*—As a flexible and cost effective solution for a scalable Internet-access network, we have studied the *Wireless Internet-access Mesh NETwork (WIMNET)* that is composed of multiple access points (APs). WIMNET utilizes multi-hop communications between APs to provide access to the Internet through a gateway (GW) from associated hosts, which becomes increasingly vulnerable to problems of the available bandwidth degradation and the network latency. As a result, real-time applications such as Voice-over-IP (VoIP) and video conferences are hard in WIMNET. In this paper, we propose a *traffic control method with consumed bandwidth estimation* to afford real-time applications in WIMNET. After the consumed bandwidth by the requesting applications is estimated, the least priority application is repeatedly dropped at GW using the *leaky bucket traffic shaping*, until it does not exceed the available bandwidth in WIMNET. The effectiveness of our proposal is verified through simulations using the *QualNet* simulator.**

I. Introduction

As a flexible and cost effective solution for a scalable Internet-access network, we have studied the *Wireless Internet-access Mesh NETwork (WIMNET)*. WIMNET is a highly promising technology, and may play an important role in next generation wireless mobile networks. As shown in Figure 1, WIMNET consists of multiple access points (APs) as mesh routers with mesh clients, and can be independently implemented or integrated with other mobile communication systems such as cellular systems. In WIMNET, at least one AP acts as a gateway (GW) to the Internet. Any host in the network field covered by WIMNET can connect to the Internet through this GW.

WIMNET is dynamically self-organized and self-configured, with nodes in the network that are automatically establishing and maintaining mesh connectivity among themselves. Besides, WIMNET introduces many advantages such as the low up-front cost, the easy network maintenance, the robustness, and the reliable service coverage. In addition to traditional communication services, WIMNET has a great potential to deliver real-time services such as the Voice over IP (VoIP), the video telephone, and the video conference. This means that WIMNET can be a competitive alternative to the cellular system.

In WIMNET, all the packets to/from client hosts must pass through one of the GWs to access the Internet. If a host is associated with an AP other than GW, the packets must reach GW through multi-hop wireless communications between APs. Since the bandwidth of one wireless link is usually small in wireless networks, the traffic concentration into the limited links around GW increases the communication delay and decreases the performance of WIMNET. Besides, especially in indoor environments, where WIMNET is mainly deployed, the link quality can easily be degraded by obstacles such as walls, doors, and furniture, which can reduce the transmission speed of the link.

In addition, WIMNET faces the *multi-hop dilemma* [1] that can be defined as problems of the bandwidth degradation, the radio interference, and the network latency that are caused by multiple traffic "hops". The *bandwidth degradation* is most severe when the backhaul is shared, as in the single and dual radio approaches. Each time the aggregated traffic "hops" from an AP to another AP, the throughput is almost cut into half. The *radio interference* is a serious issue that affects the performance of any wireless network. It can be defined simply as undesired signals that interfere with the normal operation of other radio communication devices. It easily becomes affected by the radio interference from neighboring devices that operate in the same frequency band. As a packet traverses the network from a node to another node, the *network latency* is naturally introduced. This processing delay becomes the obstacle for real-time applications in WIMNET.

Several challenges exist in effectively deploying real-time applications over WIMNET. Firstly, in WIMNET, because all the APs share and compete for the spectrum, their conflicts can happen among the whole contention domain, which is different from wired networks, cellular networks, and wireless local area networks (WLANs). Under such situations, it becomes very complicated to compute and manage the bandwidth resources. Secondly, real-time applications such as VoIP usually coexist with best-effort applications such as email, FTP, and World-Wide Web. Real-time applications must compete with such data traffics sharing the transmission media in WIMNET.

In this paper, we propose a *traffic control method with consumed bandwidth estimation* for real-time applications in order to provide very small tolerance for the network latency and jitter. Our method consists of two main modules: 1) the *consumed bandwidth estimation module* using the hop count and traffic bandwidths from clients, and 2) the *drop control module* using the leaky bucket traffic shaping. We implemented the proposed method in the *QualNet* simulator [7], and verified the effectiveness through simulations in two types of network topologies for WIMNET.

Section II describes recent related works to this study. Section III presents the design of the traffic control method and modules. Section IV shows simulation results using *Qualnet*. Section V concludes this paper with future works.

II. Related Works

The issue of providing the QoS guarantee in wireless multi-hop networks has been widely investigated in literature, in particular with reference to wireless ad hoc networks. As a survey of QoS approaches for wireless ad hoc networks, readers can be referred to [2]. Given the fully distributed and mobile nature of ad hoc networks, to solve the problems of accurately characterizing the bandwidth and delay characteristic of the links is very challenging.

The authors in [3] review the critical aspect that should be considered when using the IEEE802.16-2004 standard mesh mode as a case-study. In addition to research challenges, they highlight pitfalls and give pointers to realize QoS in wireless mesh networks. In this paper, we follow their objective to use the efficient and adaptive bandwidth management.

The works more related to ours are given in [4]. The authors have the similar goal of providing the QoS guarantee in terms of the bandwidth and delay constraints to final users. However, this paper is based on 802.11 MAC, and on simple interference models based on the notion of the conflict graph. The accuracy of the predicted bandwidth/delay estimation by the proposed framework becomes an issue.

On the other hand, the approach in [5] completely neglects the effect of radio interference, based on TDMA at the MAC layer, since it assumes that enough radio resources are available at the backbone nodes so that an arbitrary number of simultaneous transmissions can occur without mutual interference. Unfortunately, this assumption is hardly met in a practical system, where the amount of available radio resources is severely constrained (e.g., only a few orthogonal channels available in 802.11a/b/g).

III. Traffic Control Method with Consumed Bandwidth Estimation

A. Overview of Proposal

The goal of the proposed traffic control method with the consumed bandwidth estimation is to accept as many real-time flows as possible, and to guarantee no impaction of any already accepted real-time flow by additionally accepting real-time and/or best effort traffics. Both *IntServ* and *DiffServ* are adopted in this traffic control method to satisfy the bandwidth requirement of real-time applications.

Figure 2 illustrates the flowchart of our traffic control method. Our method first classifies the traffics into either real-time applications or best-effort ones. Then, it estimates the *consumed bandwidth* in WIMNET for each real-time flow by multiplying the requested bandwidth ($Bw_{app(i)}$ in the flowchart) with the hop count ($Link_{(i)}$) that indicates the number of links along the path between the host and GW. After that, it calculates the total consumed bandwidth (BWE) by taking the summation among all the real-time flows. If the total consumed bandwidth is larger than the *available bandwidth* in WIMNET, it drops the real-time flow with the largest consumed bandwidth by applying the leaky bucket traffic shaping. This process is repeated until the former one is smaller than the latter one.

Here, we note that through several experiments, we have found that the available bandwidth of a link using IEEE 802.11b protocol for transmitting data is 60% of the designed bandwidth of the link. This means that if each link is assigned $2Mbps$, it actually can send data with $1.4Mbps$.

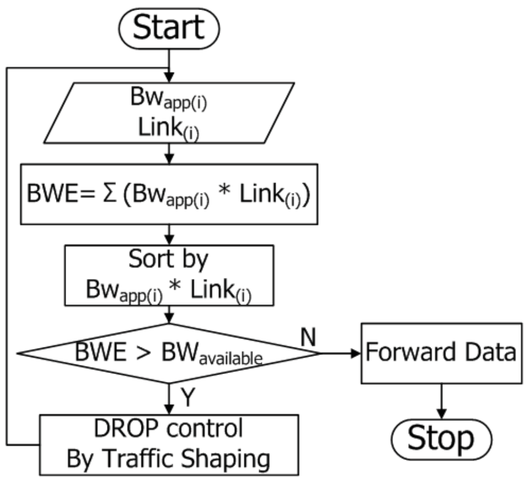


Fig. 2. Flowchart of traffic control method.

B. Consumed Bandwidth Estimation Module

Figure 3 shows an example topology to explain the procedure in the consumed bandwidth estimation. In this example, we assume that IEEE 802.11b protocol is adopted with $2Mbps$ as the bandwidth of any link, and that any client sends data with $256Kbps$ in a real-time application. The client with

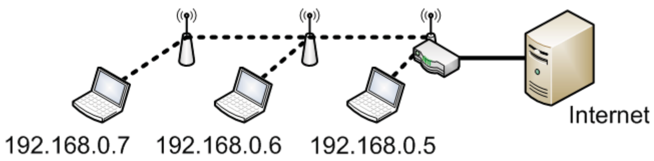


Fig. 3. Example topology for consumed bandwidth estimation.

192.168.0.5 has one hop count because it is connected directly to GW. The client with 192.168.0.6 has two hop counts, and the client with 192.168.0.7 has three hop counts. Then, the consumed bandwidth can be calculated by:

$$\begin{aligned} BWE &= (256 \times 1) + (256 \times 2) + (256 \times 3) \\ &= 1536\text{Kbps}. \end{aligned} \tag{1}$$

C. Drop Control Module

When the estimated bandwidth is larger than the available bandwidth, some of the requested flows are dropped in our method. Usually, both of real-time application flows and best-effort flows exist in WIMNET, which may interfere with each other. Our drop control module drops only real-time application flows until the estimated bandwidth does not exceed the available bandwidth of the WIMNET while reducing impacts to other flows.

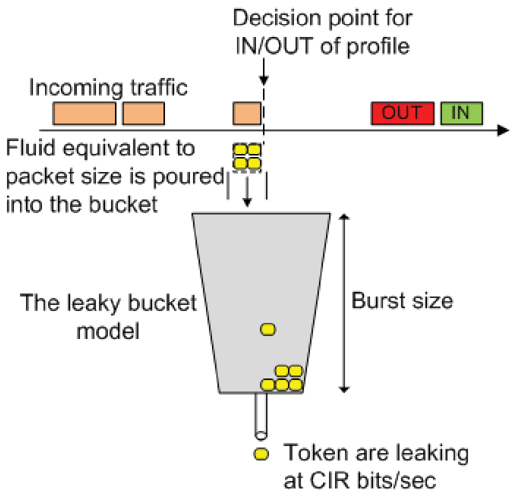


Fig. 4. Leaky bucket operation.

The basic principle of the drop control is to shape the traffic using the drop method in the leaky bucket traffic shaping. The leaky bucket [6] is actually a traffic meter, and is used to measure the amount of information transmitted by a certain data flow. When it is coupled with an algorithmic dropper, it becomes a means of the bandwidth enforcement. The leaky bucket has two parameters: 1) the token rate, and 2) the burst tolerance size or bucket size. The toke rate is often called the *Committed Information Rate (CIR)*. Figure 4 shows the operation of the leaky bucket.

For the implementation of the traffic control method in *QualNet*, we actually use *TRAFFIC-TRACE* that has been implemented there. In *TRAFFIC-TRACE*, we give a large value to the *Bucket Size* parameter for the traffic accepted by our method. To drop the traffic that is rejected by our method, we give a small value to the *Bucket Size* parameter, and set *DROP* rather than *DELAY* to the *Action* parameter. To make the constant rate, we give the transmission rate to the *Token Rate* parameter.

IV. Simulation Results

A. Simulation Environment

We implemented the proposed traffic control method in *Qualnet* for evaluation. Here, we prepared two simulation scenarios, namely, 1) the bandwidth threshold measurement, and 2) the traffic control method analysis. The first scenario is aimed to measure the available bandwidth of a wireless link at IEEE 802.11b protocol for the designed bandwidth, and to evaluate the effect of the hop count between a host and GW. The second scenario is aimed to verify the effectiveness of the traffic control method.

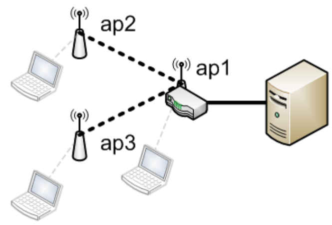


Fig. 5. Tree Topology.

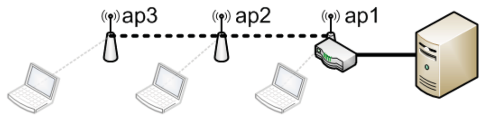


Fig. 6. Line Topology.

In both scenarios, we use two network topologies for the WIMNET backbone, namely, the *tree topology* in Figure 5 and the *line topology* in Figure 6. The distance between neighboring APs is set $250m$, and the transmission rate of any link between APs is $2Mbps$ using IEEE 802.11b protocol. The number of clients is changed from one to three, and the transmission range of any wireless device is fixed to $300m$.

VoIP [8] is adopted as a representative real-time application, where two programs of *TRAFFIC-TRACE* and *CBR* are used as traffics in *QualNet* to simulate VoIP with the leaky-bucket traffic shaping. Here, we note that we modify source codes in *QualNet* to report the consumed bandwidth.

B. Bandwidth Threshold Measurement

The goal of this scenario is to obtain the bandwidth threshold for WIMNET by considering the hop count. From the measurement results in the following three cases, we use 60% of the designed bandwidth for one link for the bandwidth threshold in our traffic control method.

1) Case for Single Client: In the bandwidth threshold measurement, first, only one client host (client) is connected to an AP in the line topology, transmitting $2048Kbps$ real-time application data. The associated AP of the client is changed among AP1, AP2, and AP3. Figure 7 shows the throughput results. Here, we can see that the throughput for AP1 is about $1.4Mbps$. This means that one hop count gives only around 70% of the designed bandwidth as the available bandwidth for data transmissions. Then, the throughput for AP2 is about $0.75Mbps$, which means that two hop counts give only $35-37\%$ of the designed bandwidth, or the *half* of the available bandwidth for the one hop case. The throughput for AP3 is about $0.45Mbps$, which means that three hop counts give only $22-25\%$ of the designed bandwidth, or the *one-third* of the available bandwidth for the one hop case.

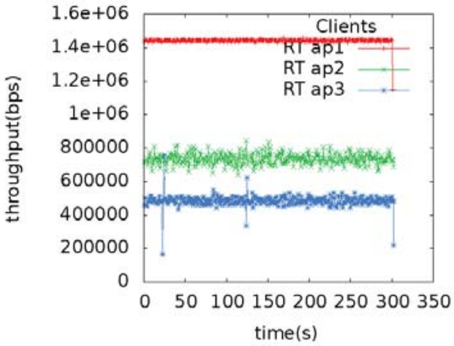


Fig. 7. Throughputs for real-time applications for single client.

2) Case for Two Clients: Then, two clients are associated with two different APs, namely AP2 and AP3, in the both topologies, where each client is transmitting $2048Kbps$ real-time application data. Figures 8 and 9 show the throughput result in each topology respectively. The average accumulated throughput from two clients in the tree topology is about $1.4Mbps$ (=70%), and that in the line topology is about $1.1Mbps$ (=55%). This difference comes from the fact that in the line topology, AP2 needs to forward data from AP3 in addition to data from its associated client. This conflict may cause the degradation of the throughput.

3) Case for Three Clients: Finally, three clients are associated with the three different APs in both topologies, where each client is transmitting $2048Kbps$ real-time application data. Figures 10 and 11 show the throughput result in each topology respectively. The average accumulated throughput from three clients in the tree topology is about $1.4Mbps$ (=70%), and that in the line topology is about $1.3Mbps$ (=65%). In the line topology, the throughput in this case is better than that in the two client case. The reason comes from the fact that the client associated with AP1 can achieve the better throughput than other clients, because it can reach AP1 (GW) by a single hop.

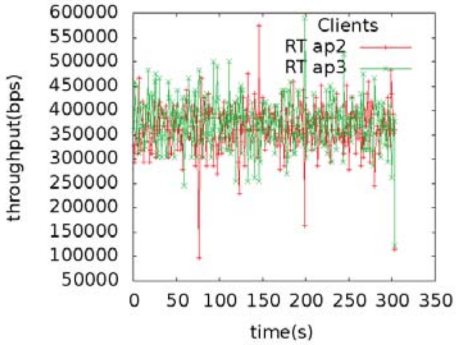


Fig. 8. Throughputs for real-time applications for two clients in tree topology.

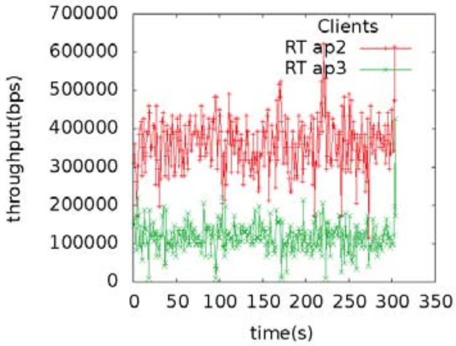


Fig. 9. Throughputs for real-time applications for two clients in line topology.

C. Traffic Control Method Analysis

In the second scenario, both the tree and line topologies are adopted to verify the effectiveness of the proposed traffic control method with the consumed bandwidth estimation.

1) Case for Real-time Application Only: First, we consider the case where only real-time applications exist in WIMNET. In this case, three clients transmit $256Kbps$ real-time application data at the same time. Each client is associated with a different AP, where the location is randomly generated within the circle of the $50m$ radius from the AP. For each topology, both the case with the traffic control and the case without the traffic control are simulated to compare their results. In *QualNet*, *TRAFFIC-TRACE* and *CBR* are actually used as real-time applications.

When all the three clients are sending data at the same time, our traffic control method drops the traffic from the third client associated with AP3, because the estimated total consumed bandwidth is larger than the available bandwidth and the third client is consuming the largest bandwidth among three clients.

Figures 12(a) and 12(b) show that the throughput (consumed bandwidth) of each client is heavily fluctuated when the proposed method is not applied. This fluctuation means that the jitter happens in data transmissions. Besides, the third client associated with AP3 cannot achieve the requested throughput where only about $170Kbps$ can be sent in the line topology. These results indicates the bad quality for real-time

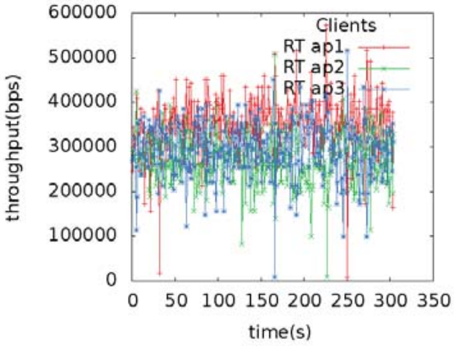


Fig. 10. Throughputs for real-time applications for three clients in tree topology.

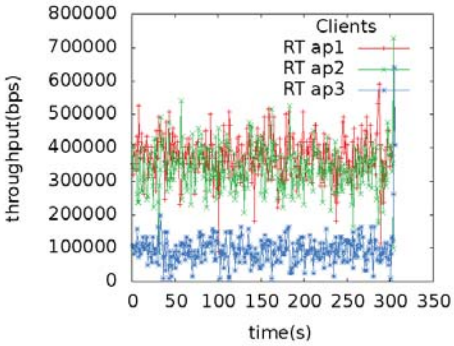


Fig. 11. Throughputs for real-time applications for three clients in line topology.

applications when our proposal is not used.

On the other hand, Figures 12(c) and 12(d) show that the throughput of each client is not fluctuated and is constant at $256Kbps$ by dropping the traffic from the third client by our method. This means that the clients can satisfy the real-time applications in good conditions, although the third client needs to give up the real-time application.

2) Case for Both Real-time and Best-effort Applications: Then, we consider the more practical case where both best-effort applications and real-time applications coexist in WIMNET. In this case, the application of one client is changed to a TCP. As before, both the case with the traffic control and the case without the traffic control are simulated to compare their results. Two clients transmit $256Kbps$ data for real-time applications at the same time, and one client associated with AP2 requests FTP as a TCP application. In *QualNet*, *FTP/GENERIC* is used as this TCP application.

Figures 13(a) and 13(b) show that without the traffic control, throughputs for the real-time applications are far more fluctuated because of the TCP application. Here, the TCP application consumes the large bandwidth. On the other hand, Figures 13(c) and 13(d) show that the throughput of any real-time application becomes constant at $256Kbps$, because the rate of the TCP application is restricted by the traffic control method.

V. Conclusion

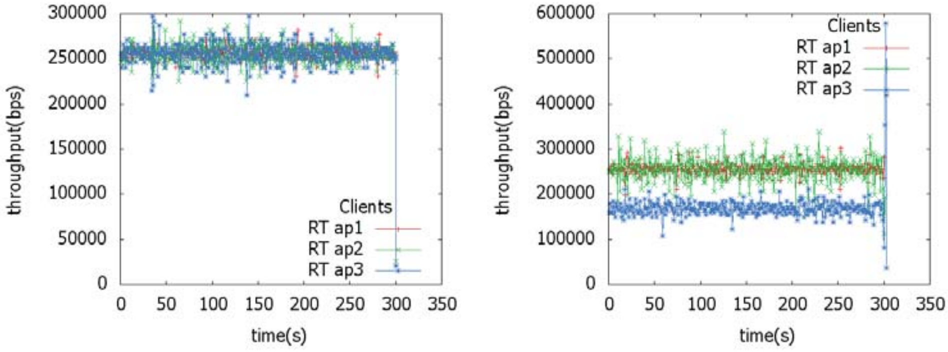
In this paper, we proposed a traffic control method for real-time applications in the wireless Internet-access mesh network (WIMNET), and evaluated the effectiveness through simulations using the *QualNet* simulator. The traffic control method consists of the consumed bandwidth estimation module and the drop control module using the leaky bucket traffic shaping. In future studies, we will improve the accuracy of the consumed bandwidth estimation and extend the drop control module by combining the link sharing algorithm. We will also implement the traffic control method using COTS (Commercial Off-The-Shelf) to realize a practical WIMNET system.

Acknowledgment

This work is partially supported by KAKENHI (22500059).

References

[1] Strix System, ”Solving the wireless mesh multi-hop dilemma”, White Paper, 2005.
[2] T. B. Reddy, I. Karthigeyan, B. Manoj, and C. S. R. Murthy, ”Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions.” Available Online: http://www.sciencedirect.com, April 2004.
[3] P. S. Mogre, M. Hollick, and R. Steinmetz, ”QoS in wireless mesh networks: challenges, pitfalls, and roadmap to its realization”, 17th Int. Work. Network and Operating Sys. Support. Digital Audio & Video, June 2007.
[4] Y. Wei, R. Maosheng, T. Zhao, L. Xiaoming, ”A bandwidth management scheme support for real-time applications in wireless mesh networks”, Proc. ACM SAC, pp.2063-2068, 2008.
[5] X. Chu, ”Provisioning of parameterized quality of service in 802.11e based wireless mesh networks”, Mobile Networks and Applications, vol. 13, pp. 6-18, April 2008.
[6] Leaky Bucket - QoS Guide, http://qos-guide.com/html/leaky_bucket.html.
[7] Scalable Networks Technologies, http://www.scalable-networks.com.
[8] ITU-T Recommendation G.729A, ”Coding of speech at 8kbit/s using conjugate-structure algebraic-code-exited-Linear-prediction (SC-ACELP)”, March 1996.



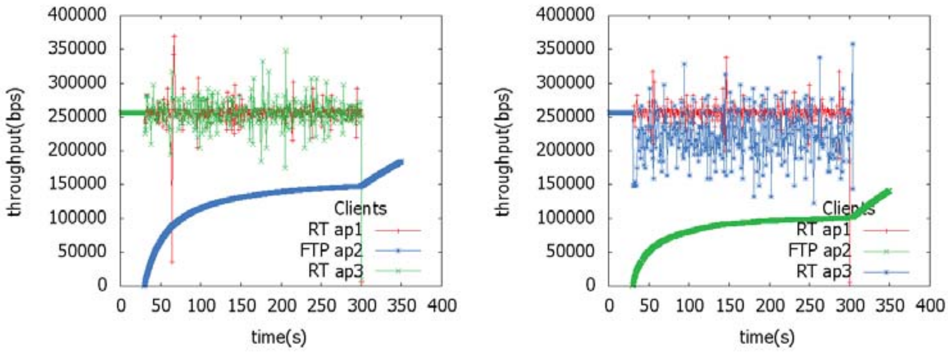
(a) without proposal in tree topology

(b) without proposal in line topology

(c) with proposal in tree topology

(d) with proposal in line topology

Fig. 12. Throughput for real-time application only.



(a) without proposal in tree topology

(b) without proposal in line topology

(c) with proposal in tree topology

(d) with proposal in line topology

Fig. 13. Throughput for both real-time and best-effort applications.