Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Adaptive fractional window increment of TCP in multihop ad hoc networks[*]

Liang-hui DING[†], Wen-jun ZHANG, Xin-bing WANG, Liang QIAN, You-yun XU

(*Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*)

[†]E-mail: lhding@sjtu.edu.cn

Received May 19, 2008; Revision accepted Sept. 16, 2008; Crosschecked Apr. 29, 2009

Abstract: We propose an adaptive fractional window increasing algorithm (AFW) to improve the performance of the fractional window increment (FeW) in (Nahm *et al*., 2005). AFW fully utilizes the bandwidth when the network is idle, and limits the operating window when the network is congested. We evaluate AFW and compare the total throughput of AFW with that of FeW in different scenarios over chain, grid, random topologies and with hybrid traffics. Extensive simulation through ns2 shows that AFW obtains 5% higher throughput than FeW, whose throughput is significantly higher than that of TCP-Newreno, with limited modifications.

Key words: TCP, Multihop, Adaptive fractional window increasing algorithm (AFW)
doi:10.1631/jzus.A0820380 **Document code:** A **CLC number:** TN929.5

INTRODUCTION

A multihop ad hoc network without any infrastructure is composed of independent nodes, which can serve as sources, destinations, and relays. The transmission control protocol (TCP) is a transport protocol widely accepted in wired networks, which is the de facto standard for the reliable data transfer in multihop ad hoc networks. However, the performance of TCP in multihop ad hoc networks is not satisfactory. Lossy wireless channels, inefficient medium access control (MAC) protocols, large overhead of routing protocols, and interactions among different protocol layers are all responsible for the poor performance (Fu *et al*., 2002).

Recently extensive work has been carried out to analyze and improve the performance of TCP in multihop ad hoc networks (Holland and Vaidya, 1999; Chandran *et al.*, 2001; Dyer and Boppana, 2001; Chen *et al*., 2003; Fu *et al*., 2003; Xu *et al.*, 2003; EIRakabawy *et al.*, 2005; Nahm *et al.*, 2005; Zhai *et al.*, 2007; Ding *et al*., 2008). Both path breakage induced by mobility (Holland and Vaidya, 1999; Chandran *et al.*, 2001; Dyer and Boppana, 2001) and fairness of TCP in multihop ad hoc networks (Xu *et al.*, 2003) are considered. Window-based transfer of TCP is changed to be a combination of window- and rate-based transfer in (EIRakabawy *et al.*, 2005; Zhai *et al.*, 2007). Some researchers are focused on limiting the sending rates of TCP with active packet dropping (Fu *et al.*, 2003), the maximum window limit (Chen *et al*., 2003), and small window increasing steps (Nahm *et al.*, 2005; Ding *et al*., 2008).

TCP uses two different phases, i.e., slow start and congestion avoidance, to probe the available bandwidth of dynamic networks. When receiving a new acknowledgement from the sink, the TCP source increases the window size by 1 in the slow start phase and by $1/W$ in the congestion avoidance phase, where W denotes the current window size. This is suitable for wired networks with a large bandwidth delay

[*] Project supported by the National Natural Science Foundation of China (Nos. 60625103, 60702046 and 60832005), the Doctoral Fund of MOE of China (No. 20070248095), the China International Science and Technology Cooperation Program (No. 2008DFA11630), the Shanghai Science and Technology PUJIANG Talents Project (No. 08PJ14067) and Innovation Key Project (No. 08511500400), and the Qualcomm Research Grant

product (BDP). However, the BDP of multihop ad hoc networks is relatively small. The aggressive window increasing policy usually results in overload of the network, and then packet loss, path reestablishment, timeout, retransmission, all of which reduce the throughput of TCP. Nahm *et al*.(2005) identified the problem and proposed a fractional window increment algorithm (FeW), which sets the window increasing step to be smaller than 1 and increases the throughput of TCP significantly. However, the fixed small window increasing step wastes the bandwidth before the congestion window increases up to the BDP, and results in quick timeout events when the window size is equal to or larger than BDP.

Hence, in this paper, we propose an adaptive fractional window increment algorithm (AFW), which sets different window increasing steps to distinct windows. The window increasing steps in AFW are updated when a timeout occurs, which indicates that the network is congested and that the window size is larger than BDP. When the timeout occurs at the congestion window W, the increasing steps of windows smaller than W are increased, while those of windows larger than W are decreased. AFW is more aggressive than FeW by using larger increasing steps when the congestion window is smaller than BDP, while more conservative than FeW by utilizing smaller increasing steps when the congestion window is equal to or larger than BDP. The scheme achieves a tradeoff between large and small window increasing steps and improves the throughput of TCP. Simulation results show that AFW obtains 5% higher throughput than FeW (Nahm *et al*., 2005) with only limited modifications.

The remainder of this paper is organized as follows. The related work and detailed introduction of FeW are given in Section 2. Section 3 presents the motivation, the details of AFW, the parameter tuning of AFW, and the window comparison between AFW and FeW. In Section 4, we evaluate AFW through ns2 over chain, grid, random topologies and with hybrid traffics. The conclusion is given in Section 5.

RELATED WORK AND DESCRIPTION OF FEW

Related work

The work on improving the performance of TCP in multihop ad hoc networks can be divided into three categories: the first aims at removing the impact of mobility, the second focuses on the fairness of TCP in multihop ad hoc networks, while the third mainly considers how to regulate the sending rates of TCP. The work in each category is summarized as follows.

To remove the impact of mobility, the main idea is to distinguish between the packet losses caused by router breakage or transmission errors and those due to the real network congestion. The retransmit timer is fixed after retransmitting once rather than increasing exponentially to avoid unnecessary time waste (Dyer and Boppana, 2001). Feedback at the network layer is utilized to notify the routing breakage (Holland and Vaidya, 1999; Chandran *et al*., 2001). Upon receiving route failure messages, the TCP source snoozes all its variables and stops transmitting anymore to deal with the effects of route breakage. In this paper, we focus on TCP in static multihop ad hoc networks.

TCP in multihop ad hoc networks is unfair due to two major reasons: (1) different round trip time; (2) hidden and exposed terminal problems caused by CSMA/CA (carrier sense multiple access with collision avoidance) based MAC protocols. Random early detection (RED) queue management (Floyd and Jacobson, 1993) is used to improve the fairness in wired networks, in which the routers drop packets actively according to the number of packets in the queue. Xu *et al*.(2003) extended RED to multihop ad hoc networks and proposed the neighborhood RED (NRED). The channel utilization is deployed to express the size of the distributed queue shared by nodes in a two-hop area. When the queue size is larger than a threshold, the related nodes are informed to drop packets.

Another key reason for the poor performance of TCP in multihop ad hoc networks is the mismatch between the TCP transmission rate and the network capacity under specific MAC protocols. Especially, the overload of the network causes more serious waste of bandwidth than that in wired networks because of the large overhead at the routing and MAC layers. Thus, a large amount of work has been done on regulating the sending rates of TCP according to the capacity of the network under specific MAC protocols. The window-based transfer mode of TCP is changed to rate-based mode (EIRakabawy *et al*., 2005; Zhai *et al*., 2007). The inter-packet transmission delay of TCP is set to be four-hop propagation delay (FHD), in which the FHD is estimated from the round trip time (RTT) through the exponential weighted moving average (EWMA) (EIRakabawy *et al*., 2005). The sending rate of the TCP source is set according to the business ratios of all nodes in the path (Zhai *et al*., 2007). The window-based transfer mode is reserved and the transmission rate is controlled with the maximum congestion window or the window increasing step (Chen *et al*., 2003; Fu *et al*., 2003; Nahm *et al*., 2005; Ding *et al*., 2008). The packets are dropped actively at intermediate nodes when the channel business ratio is larger than a threshold (Fu *et al*., 2003), which is similar to RED (Floyd and Jacobson, 1993). The maximum congestion window is set to be the optimal value with the maximum throughput according to the number of hops in (Chen *et al*., 2003). The window increasing step is set to be 0.01 to reduce the probing speed of TCP-Newreno and to increase the throughput (Nahm *et al*., 2005). Larger window increasing thresholds for both slow start and congestion avoidance phases are deployed to reduce the probing speed of TCP-Vegas (Ding *et al*., 2008).

Detail description of FeW

As mentioned above, the wireless ad hoc networks are limited resources and the BDP is usually small. The traditional TCP increases the congestion window by 1 when receiving a new ACK packet, as shown in Eq.(1). The aggressive network probing mechanism usually results in the overload of the network and then the congestion.

$$W^{\text{new}} = W^{\text{current}} + \frac{1}{W^{\text{current}}}, \qquad (1)$$

where W^{new} and W^{current} are the current and newly updated congestion window size, respectively.

FeW proposed in (Nahm *et al*., 2005) changes the window increment from 1 to α, and the window variation becomes

$$W^{\text{new}} = W^{\text{current}} + \frac{\alpha}{W^{\text{current}}}, \qquad (2)$$

where $0<\alpha\leq 1$ controls the speed of probing the network. A smaller value of α leads to a milder probing traffic, while $\alpha=1$ is the same as the legacy TCP. $\alpha=0.01$ is adopted to validate the performance of FeW (Nahm *et al*., 2005). The small window increment reduces the probing speed, and alleviates the congestion in the network, which then increases the throughput of TCP in wireless ad hoc networks.

FeW requires only the modifications at TCP sources and is compatible with other TCP variants, which is a scalable algorithm to improve the performance of TCP in wireless ad hoc networks. However, the small window increasing step of FeW is fixed, which results in bandwidth waste when the network is idle, and quick timeout events when the network is busy.

Hence, in this study, we balance the large and small window increasing steps through the adaptive fractional window increment, AFW, which increases quickly when the network is idle, and slowly when the network is congested.

ADAPTIVE FRACTIONAL WINDOW INCREASE

Motivation

We simulated FeW with dynamic source routing (DSR) (Johnson *et al*., 2004) over a 7-hop chain topology as shown in Fig.1. The dynamics of the congestion window of FeW in this scenario looks like sawtooth, which is shown as the dotted line in Fig.2. The variation from the maximum window size to the minimum value is due to the timeout. The timeout indicates that the network has been congested and a packet has been dropped because the number of retransmissions has been larger than the maximum retry limit. Fig.2 shows that the slope of the window increasing curve from 2 to the maximum value is fixed. However, when the congestion window is smaller than the capacity of the network, the large window increasing step can reduce the waste of the bandwidth. When the congestion window is larger than the capacity of the network, the small window increasing step is able to slow down the probing speed and reduce the number of timeout events. This motivates us to design the adaptive fractional window increasing scheme, which adapts the step according to the state of the network.

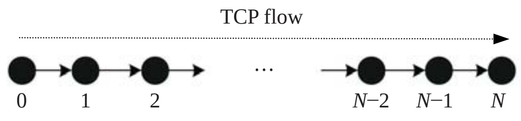

Fig.1 Chain topology with *N*+1 nodes

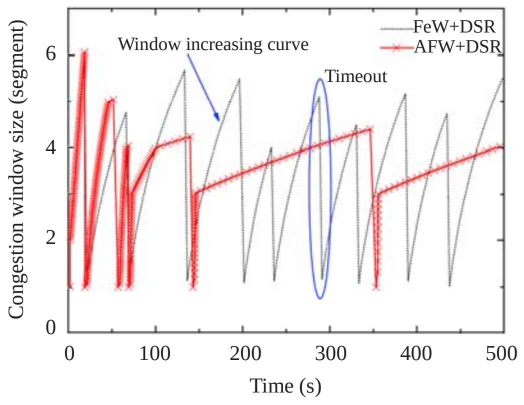

Fig.2 Window variation of both AFW and FeW

Details of AFW

AFW adjusts the congestion window increasing step according to the size of the window when a timeout occurs. Let W_t denote the size of the congestion window when a timeout occurs and S_W the window increasing step of the congestion window W. When the timeout occurs, S_W of the congestion window smaller than W_t is multiplied with α, and S_W of the congestion window larger than W_t is divided by β, where α and β are two parameters set in advance. Since the window increasing step cannot be too small or too large, two parameters S_{max} and S_{min} are deployed to limit the maximum and minimum values of S_W. The details of AFW are illustrated in Fig.3.

```
Initialization: a timeout occurs.
Method: $S_W=S_{max}$;
1.  IF ($W<W_t$)
2.      $S_W=S_W\times\alpha$;
3.      IF ($S_W>S_{max}$)
4.          $S_W=S_{max}$;
5.      END
6.  END
7.  IF ($W>W_t$)
8.      $S_W=S_W/\beta$;
9.      IF ($S_W<S_{min}$)
10.         $S_W=S_{min}$;
11.     END
12. END
```

Fig.3 Adaptive fractional window increment

Parameter tuning

In this subsection, we analyze the influence of parameters α, β, S_{max} and S_{min} on the performance of AFW through observing the throughput of AFW over a 7-hop chain topology for instance.

We firstly considered the influence of the multiplier α and the divisor β on the throughput. We set S_{max}=1, S_{min}=0.001 and simulated AFW over the 7-hop chain topology with different combinations of α and β. The results show that the throughput with the combination α=4 and β=8 is higher than all others. Thus, we chose the combination α=4 and β=8 in the following validation of AFW.

Then we considered the influence of the maximum and the minimum window limits S_{max} and S_{min}. Through analyzing the simulation results with combinations of different S_{max} and S_{min}, we found that the throughputs with S_{min}=0.0001 and S_{max}=0.001 are almost the same, while the throughput decreases as S_{min} increases. Since a too small minimum window limit will reduce the response speed of TCP sources to the network state, S_{min}=0.001 was used in the following simulation. When S_{min}=0.001, the throughputs with different S_{max} are nearly equal to each other. This indicates that the throughput is not very sensitive to S_{max} and the setting of S_{max}=1 in analyzing parameters α and β is reasonable. Considering that the BDP of wireless ad hoc networks is relatively small, we adopted S_{max}=0.1 here.

According to the above analysis, a combination of α=4, β=8, S_{max}=0.1, S_{min}=0.001 was adopted in the following simulations.

Window comparison

We set α=4, β=8, S_{max}=0.1, S_{min}=0.001 according to the above analysis, and ran a TCP flow over the 7-hop chain topology. The window variation of AFW is shown as the solid line with crosses in Fig.2, We can find that AFW is able to adapt to the state of the network, of which the window increasing steps with small congestion windows are relatively large, and the window increasing steps with large congestion windows are relatively small. After the limited modification of AFW, the throughput is about 6% higher than that of FeW.

VALIDATION RESULTS

In this section, we evaluate AFW and compare the total throughput of AFW with that of FeW in different scenarios over chain, grid, random topologies and with hybrid traffics through ns2.

Simulation setup

We implemented AFW into ns2 and evaluated its performance over DSR (Johnson *et al.*, 2004), which is a widely used on-demand routing protocol. IEEE 802.11 (IEEE Std. 802.11 WG, 1999) was set as the MAC layer protocol. Both the basic rate and data rate at the 802.11 MAC layer were set to be 2 Mbps. File transfer protocol (FTP) services with a large enough file size (100 MB) were evaluated over chain, grid, and random topologies and in the scenario with hybrid traffics. The receiving range was 250 m and the carrier sensing range was 500 m. The wireless propagation model was the two-ray ground models. The packet size was 1024 bytes. Other related parameters of 802.11 are listed in Table 1.

Table 1 Parameters for IEEE 802.11 MAC

Parameter	Value
Preamble length (bit)	144*
PLCP length (bit)	48*
RTS length (bit)	160**
CTS/ACK length (bit)	112**
MAC header (bit)	224**
IP header (bit)	160***
SIFS (μs)	10
DIFS (μs)	50
Slot time, σ (μs)	20
Contention window	31
Retry limit, *m*	7

* At 1 Mbps; ** at basic rate; *** at data rate

Chain topology

Chain is a common topology with limited resources and shared by different TCP flows. We first evaluated AFW with 1 and 4 TCP flows from the source node 0 to the sink node *N* over the chain topology shown in Fig.1. The scenarios with 1 and 4 flows represent the network with relatively low and heavy load, respectively. As shown in Fig.4, the throughput of AFW is 6.5% higher than that of FeW in the case of 1 flow (Fig.4a), and 3.7% higher than that of FeW in the case of 4 flows (Fig.4b). The growth comes from the adaptation of AFW. When the network is relatively idle, the window increasing step is large; when the network is relatively busy, the window increasing step is small.

Fig.4a shows that the gap between the throughput of AFW and FeW increases as the number of hops increases. When the number of hops is large, the round trip time is large and the BDP is also larger than that with a smaller number of hops. We know that the gain of AFW mainly comes from the balance of large and small window increasing steps. When the BDP is larger, the impact of the fixed window increasing step of FeW on the TCP throughput is more serious, because the reaction speed of the TCP source to the network state is slower. Hence, the throughput gain of AFW increases as the number of hops increases.

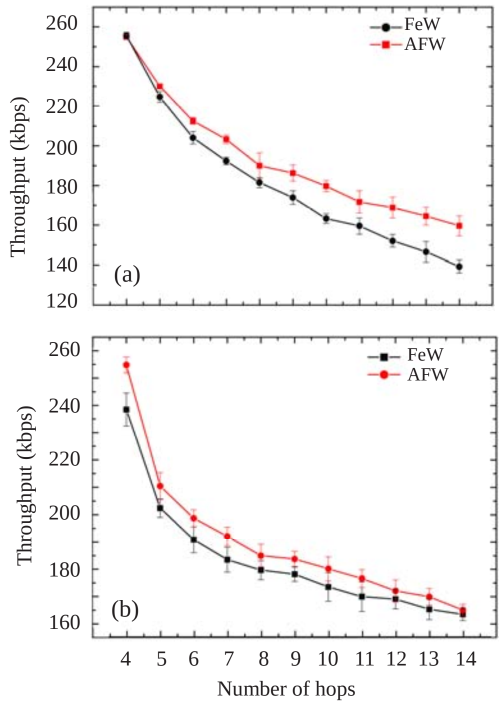

Fig.4 Throughput comparison of AFW and FeW over chain topology with DSR and 95% confidence interval
(a) One flow; (b) Four flows

Fig.4b shows that the percentage growth of the throughput with different numbers of hops is almost identical. This is because the chain with 4 flows is congested, and the congestion window mainly works on *W*=1. When the congestion window is increased up to 2, the congested packets are usually dropped and the timeout event occurs. When the timeout occurs, FeW starts the slow start process with a fixed window increasing step, while AFW regulates the step adaptively to match the state of the network and increases the throughput. Hence, the throughput gain with 4 flows over the chain topology is nearly identical.

Since fairness is an important issue for TCP in wireless ad hoc networks, we explain it here with Jain's fairness index as defined in (Jain *et al.*, 1984):

$$F = \left(\sum_{i=1}^{n} x_i\right)^2 \bigg/ \left(n\sum_{i=1}^{n} x_i^2\right). \tag{3}$$

The fairness comparison between AFW and FeW is shown in Fig.5. Since the window increasing step of AFW is variable, some TCP flows may catch the path with larger steps, while the other flows will run with smaller window increasing steps. Thus, the fairness of AFW is a little worse than that of FeW.

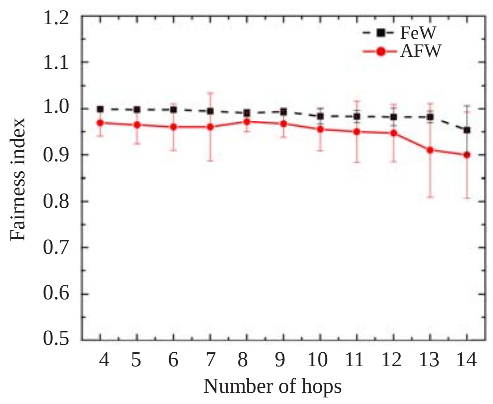

Fig.5 Fairness comparison between AFW and FeW with 4 flows and 95% confidence interval

We also evaluated the performance of AFW over the chain topology with different sources and destinations. The traffic scenario is shown in Fig.6. Flow 1 is from node 0 to node 9, and flows 2 and 3 are from node 2 to node 8 and from node 7 to node 1 respectively. The throughput comparison is shown in Fig.7. For flow 1 with a longer distance between the source and the destination, the throughput of AFW is smaller than that of FeW. For the other two flows with smaller distances, the performance of AFW is better than that of FeW. The total throughput of AFW is about 4% higher than that of FeW.

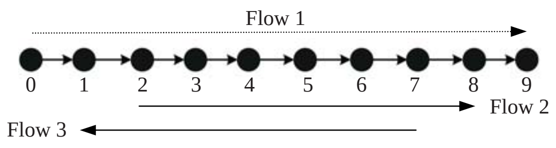

Fig.6 Three different TCP flows over the chain topology

Grid topology

We simulated the AFW algorithm with DSR on 7×7 and 13×13 grid topologies (Fig.8). The distance

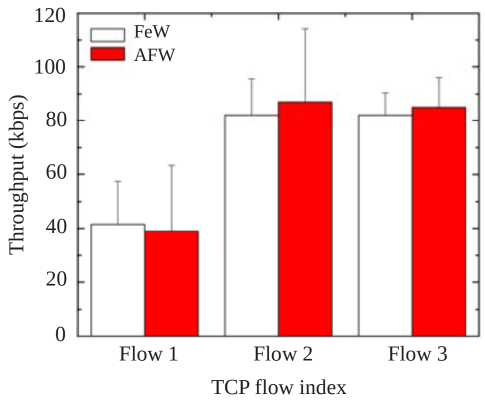

Fig.7 Throughput comparison of AFW and FeW with different sources and destinations

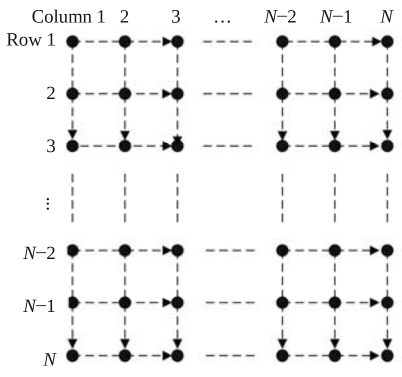

Fig.8 Grid topology with *N*×*N* nodes

between each neighboring node on dashed lines was 200 m. The scenarios with 1, 2, 4, 6, 8 and 14 flows were tested over the 7×7 grid topology, while 1, 2, 4, 6, 12 and 26 flows were evaluated over the 13×13 grid topology. In both topologies, when there was only one flow, the traffic was set on the horizontal middle path from the left edge node to the right edge node. The traffics in all the other scenarios were placed evenly and symmetrically with equal horizontal and vertical numbers of flows from left to right and from top to bottom.

The throughputs of AFW and FeW over grid topologies are shown in Fig.9. In both grid topologies, AFW obtains 5% higher throughput than FeW. We can also find that the percentage growth of the throughput decreases as the number of flows increases. When the number of flows increases, the network is more and more congested. Thus, the increased window is quickly pulled down to 1 in both AFW and FeW, which results in the similar operation and throughput of AFW and FeW.

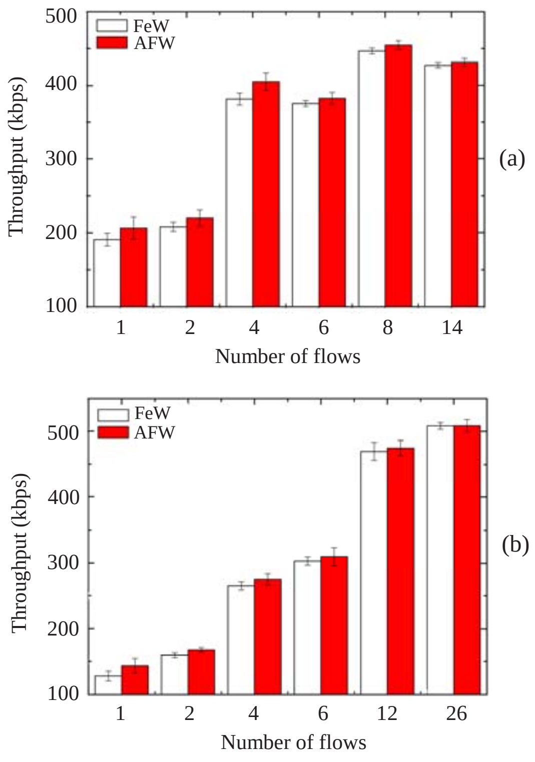

Fig.9 Throughput comparison of AFW and FeW over grid topology with DSR and 95% confidence interval
(a) 7×7 grid; (b) 13×13 grid

Random topology

We distributed 150 nodes randomly on a 2000 m×2000 m square area, and examined 8, 16, 32 and 50 TCP flows with random sources and sinks in the topology. The throughputs of these four scenarios are shown in Fig.10, and AFW still obtains a higher throughput than FeW by about 5%. The gap between AFW and FeW also decreases as the number of flows increases. The reason is the same as that explained in the previous subsection.

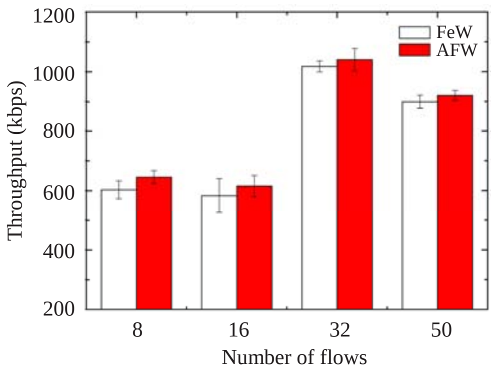

Fig.10 Throughput comparison of AFW and FeW over random topology with DSR and 95% confidence interval

Hybrid traffics

Two kinds of traffics, TCP flow and UDP flow, were placed from the same source to the same destination with different numbers of hops. The throughputs of both AFW and FeW are shown in Fig.11.

When the rate of UDP traffic is 100 kbps, the throughput of AFW is about 4.7% higher than that of FeW. This is because the window increment of AFW can be larger than that of FeW and leads to a higher throughput when the network is relatively idle. However, the throughput is almost the same when the traffic rates of UDP are 200 and 300 kbps, since the network is almost dominated by the UDP traffic.

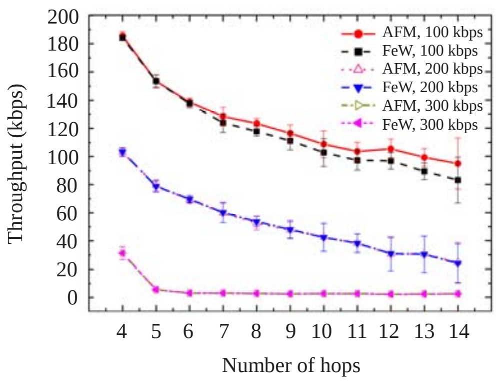

Fig.11 Throughput comparison of AFW and FeW over the chain topology with hybrid traffics and 95% confidence interval

CONCLUSION

In this paper, we propose an adaptive fractional window increment algorithm (AFW) to improve the throughput of TCP in multihop ad hoc networks. AFW adaptively adjusts the window according to the network state and balances the small and large window increasing steps. When the network is idle, it increases the window quickly and fully utilizes the bandwidth; when the network is congested, it increases the window with a smaller step, which keeps the network under a moderate load and avoids frequent packet losses caused by the overload of the network. We analyzed the influence of parameters of AFW on the throughput and validated AFW in a variety of circumstances with the optimal parameters. Simulation results show that the throughput of AFW is about 5% higher than that of FeW, which uses fixed window increasing steps.

References

Chandran, K., Raghunathan, S., Venkatesan, S., Prakash, R., 2001. A Feedback-based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks. *IEEE Pers. Commun.*, **8**(1):34-39. [doi:10.1109/98.904897]

Chen, K., Xue, Y., Nahrstedt, K., 2003. On Setting TCP's Congestion Window Limit in Mobile Ad Hoc Network. Proc. IEEE Int. Conf. on Communications, Anchorage, Alaska, **2**:1080-1084. [doi:10.1109/ICC.2003.1204525]

Ding, L., Wang, X., Xu, Y., Zhang, W., Chen, W., 2008. Vegas-W: An Enhanced TCP-Vegas for Wireless Ad Hoc Networks. Proc. IEEE Int. Conf. on Communications, Beijing, China, p.2383-2387. [doi:10.1109/ICC.2008.453]

Dyer, T.D., Boppana, R.V., 2001. A Comparision of TCP Performance Over Three Routing Protocols for Mobile Ad Hoc Netwoks. Proc. 2nd ACM Int. Symp. on Mobile Ad Hoc Networking and Computing, Long Beach, CA, USA, p.56-66. [doi:10.1145/501422.501425]

EIRakabawy, S.M., Klemm, A., Lindemann, C., 2005. TCP with Adaptive Pacing for Multihop Wireless Networks. Proc. 6th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing, Urbana-Champaign, IL, USA, p.288-299. [doi:10.1145/1062689.1062726]

Floyd, S., Jacobson, V., 1993. Random early detection gateways for congestion avoidance. *IEEE/ACM Tran. Networking*, **1**(4):397-413. [doi:10.1109/90.251892]

Fu, Z., Meng, X., Lu, S., 2002. How Bad TCP Can Perform in Mobile Ad Hoc Networks. IEEE Int. Symp. on Computers and Communications, Taormina, Italy, p.298-303. [doi:10.1109/ISCC.2002.1021693]

Fu, Z., Zerfos, P., Luo, H., Lu, S., Zhang, L., Gerla, M., 2003. The Impact of Multihop Wireless Channel on TCP Throughput and Loss. Proc. INFOCOM, San Francisco, USA, **3**:1744-1753.

Holland, G., Vaidya, N., 1999. Analysis of TCP Performance Over Mobile Ad Hoc Networks. Proc. IEEE/ACM MobiCom, p.219-230. [doi:10.1145/313451.313540]

IEEE Std. 802.11 WG, 1999. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. IEEE Working Group.

Jain, R., Chiu, D., Hawe, W., 1984. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems. DEC Technical Report, DEC-TR-301.

Johnson, D., Maltz, D., Hu, Y., 2004. The Dynamic Source Routing for Mobile Ad Hoc Networks. IETF Internet Draft. Available from http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt

Nahm, K., Helmy, A., Jay Kuo, C.C., 2005. TCP Over Multihop 802.11 Networks: Issues and Performance Enhancement. Proc. 6th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing, Urbana-Champaign, IL, USA, p.277-287. [doi:10.1145/313451.313540]

Xu, K.X., Gerla, M., Qi, L., Shu, Y., 2003. Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighborhood RED. Proc. 9th Annual Int. Conf. on Mobile Computing and Networking, San Diego, California, USA, p.16-28. [doi:10.1145/938985.938988]

Zhai, H.Q., Chen, X., Fang, Y., 2007. Improving transport layer performance in multihop ad hoc networks by exploiting MAC layer information. *IEEE Trans. Wirel. Commun.*, **6**(5):1692-1701. [doi:10.1109/TWC.2007.360371]