Towards Simulation Study of FAST TCP Compared to XCP in Satellite Networks

Yandong Sun, Zhenzhou Ji, Hui Wang

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

Summary
More and more attention has been paid to the Internet access via satellite networks. But theory and simulations have proved that TCP suffers severe degradation in high bandwidth-delay product networks. Some improved mechanisms have been proposed, and FAST TCP and XCP are two promising representatives. FAST TCP uses a window-based policy and XCP uses a feedback-based policy. The performance of FAST TCP and XCP is thoroughly evaluated in this paper under the context of GEO satellite networks. The results show that FAST TCP and XCP maintain good fairness, low queue length and identical friendliness to TCP. In steady state, the throughput of FAST TCP is larger than that of XCP, but the convergence time is longer than that of XCP. Through comparison, the pros and cons of two protocols are illustrated and can be the guideline to design new congestion control mechanisms in satellite networks.

Key words:
Satellite Networks, FAST TCP, XCP, High Bandwidth-Delay, Congestion Control

1. Introduction

TCP is the de facto transmission protocol in the Internet, whereas, TCP suffers difficulties in high bandwidth-delay product networks. The main reason is that TCP adopts AIMD policy which increases the window too slowly but decreases the window too drastically when packet loss occurs. Moreover, in wireless networks, TCP suffers throughput degradation because TCP misinterprets the packet loss due to Bit Error Rate (BER) as congestion loss. In recent years, many improved congestion control mechanisms have been proposed featured high bandwidth-delay [1~5]. FAST TCP [6] and eXplicit Control Protocol (XCP) [3] are two promising representatives. FAST TCP is similar to TCP Vegas [7] which uses queuing delay variation to predict congestion in the network. FAST TCP performs well, but delay due to non-congestion, such as rerouting, may make it inefficient. FAST TCP is also a window-based mechanism as TCP. XCP [3] is a router-assisted solution, which uses the accurate feedback from the routers to adjust the congestion window.

More and more Internet applications based on the satellite networks have emerged. Satellite networks, especially Geostationary Earth Orbit (GEO) features not only long propagation delay, but high BER and asymmetric bandwidth. Although FAST TCP and XCP perform well in some high bandwidth-delay product networks, the performance of them in satellite networks is unveiled. So, in the paper, the performance of two protocols is evaluated and compared in terms of fairness, throughput, queue length and friendliness to TCP under the context of GEO satellite networks. The results show the pros and cons of the window-based and feedback-based protocols and provide reference for the design of congestion control solutions in satellite networks.

Manuscript received January 5, 2012
Manuscript revised January 20, 2012

2. Related Work

2.1 FAST TCP [6]

TCP suffers the following difficulties in high bandwidth-delay product networks [6]:
1) At the packet level, additive increase by one packet per Round Trip Time (RTT) is too slow, and multiplicative decrease per loss event is too drastic.
2) At the packet level, the binary congestion signal (packet loss) makes TCP oscillate.
3) At the flow level, maintaining large average congestion windows requires an extremely small equilibrium loss probability.
4) At the flow level, the accurate estimation of packet loss probability and a stable design of the flow dynamics are required to reduce the oscillations due to the dynamics.

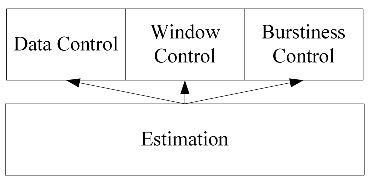


Fig. 1 Architecture of FAST TCP.

To overcome the difficulties of TCP in high bandwidth-delay product networks, FAST TCP which using queuing delay as the congestion measure was proposed. The architecture of FAST TCP consists of four components, as depicted in Fig. 1. These four components are functionally independent so that they can be designed separately and upgraded asynchronously.

(i). Estimation

This component provides input information to other components for decision making. Two kinds of feedback information are estimated per packet sent. When a positive acknowledgement (ACK) is received, the RTT of the packet is estimated and the queuing delay and the minimum RTT are updated. When a negative ACK (three duplicated ACKs or timeout) is received, the packet loss signal is sent to other components. The estimation component can provide both a multi-bit queuing delay estimation and a one-bit loss-or-no-loss signal for each data packet.

(ii). Window Control

Window control component determines the window size based on the congestion signal, queuing delay and packet loss. FAST TCP updates the congestion window periodically according to the Eq.(1):

$$w \leftarrow \min\{2w, (1-\gamma)w + \gamma(\frac{baseRTT}{RTT}w + \alpha(w, qdelay))\} \quad (1)$$

where $\gamma \in (0,1]$, *baseRTT* is the minimum RTT observed, $qdelay$ is the average queuing delay. In FAST TCP, $\alpha(w, qdelay)$ is a constant which represents the number of packets each flow attempts to maintain in the buffers at equilibrium.

In [8], a Generalized FAST TCP algorithm was proposed, which can acquire proportional fairness at equilibrium and buffer requirements grow only as the number of flows increase. In [9], an adaptive FAST TCP algorithm was proposed, which adjusts α according to the difference between current estimated queuing delay and the target queuing delay. So, the queuing delay can be limited to a specific value, and the sensitivity of setting α is eliminated. In [10], a new congestion control algorithm for TCP based on integrated feedback (IF-TCP) was proposed. In IF-TCP, packet loss probability and queuing delay both serve as a new integrated congestion feedback. IF-TCP can achieve approximately proportional fairness and good stability whether there is packet loss or not.

2.2 XCP [3]

XCP is an explicit congestion control mechanism, and the sender adjusts the congestion window size based on the feedback from the routers. The explicit control manner allows a flow to acquire the available bandwidth quickly. Some research results show that XCP is greatly stable, effective and scalable [3]. In XCP, a congestion header (Fig. 2) is attached to each packet sent. The field *H_throughput* is the sender's current congestion window, and *H_rtt* is the sender's current Round Trip Time (RTT) estimate. These are filled in by the sender and never modified in transit. The field *H_feedback* carries the feedback and is initialized by the sender. Routers along the path modify this field to inform the window adjustment to senders [11].

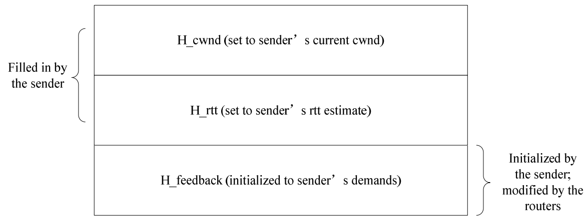


Fig. 2 Congestion header.

XCP protocol functions as follows [12]. Refer to [3] for details of XCP.

(i). The XCP sender

On a packet departure, the sender fills the congestion header with its current congestion window, current estimated RTT and desired window increase, respectively. Whenever an ACK arrives, the sender updates the window size according to the feedback from the routers as below:

$$cwnd = \max(cwnd + H\_feedback, s) \quad (2)$$

where s is the packet size. The sender responds to the losses in a similar manner to TCP.

(ii). The XCP router

The XCP router has two controllers to compute the feedback, efficiency controller (EC) and fairness controller (FC). The target of EC is to maximize link utilization while minimizing drop rate and persistent queues. According to the Multiplicative Increase and Multiplicative Decrease (MIMD) principle, the EC computes the desired increase/decrease of aggregate traffic each control interval (i.e. average RTT of all flows traversing the link). The aggregate feedback δ is computed as:

$$\delta = \alpha \cdot \tau \cdot S - \beta Q \quad (3)$$

where α and β are control parameters, and the values of them are set 0.4 and 0.226, respectively. The term τ is the average RTT, and S is the spare bandwidth denoted by the difference between the input traffic rate and link capacity. Q is the persistent queue size.

According to the Additive Increase and Multiplicative Decrease (AIMD) principle, the FC apportions the feedback to the flows sharing the same link. The per-packet feedback is distributed as follows:

If δ >0, distribute δ equally to all XCP flows.

If $\delta<0$, distribute δ to a flow proportional to its current throughput.
When δ is about zero, the concept of bandwidth shuffling is introduced to ensure local fairness. The shuffled traffic is computed as follows:

$$\xi = \max(0, \gamma \cdot \chi - |\delta|) \quad (4)$$

where γ is a control parameter and set to 0.1. The term χ is the input traffic in the control interval.
(iii).The XCP receiver
An XCP receiver is similar to a TCP receiver except that the XCP receiver copies the congestion header from a packet to its ACK when it acknowledges the packet.
In recent years, some mechanisms have been proposed to improve the performance of XCP in particular environment. The P-XCP [11], solves low throughput under high BER conditions and output link underutilization in presence of rate-limited connections. In P-XCP, all packet losses are attributed to BER and the dropped packets are just retransmitted. Wireless eXplicit Congestion control Protocol (WXCP) [13], based on XCP, tackles the problem of TCP in multi-hop wireless networks. XCP-i [14], addresses the problem that XCP behaves worse than TCP in presence of non-XCP routers. A smart window adjustment for XCP [15], is proposed to shorten the response time for packet loss under congestion in high bandwidth-delay product networks. However, the packet loss due to BER isn't taken into account. The iXCP [16], an improved XCP mechanism, overcomes the underutilization of XCP in multi-bottleneck networks. The iXCP outperforms P-XCP [11] in terms of stability and utilization. S-XCP [17], an XCP bandwidth compensation algorithm based on state feedback, can achieve efficient and fair bandwidth allocation in a multi-bottleneck environment. Moreover, the performance of XCP in GEO satellite networks is evaluated in [18]. [19] compares some aspects of XCP and CUBIC with Quick-Start mechanism [20].

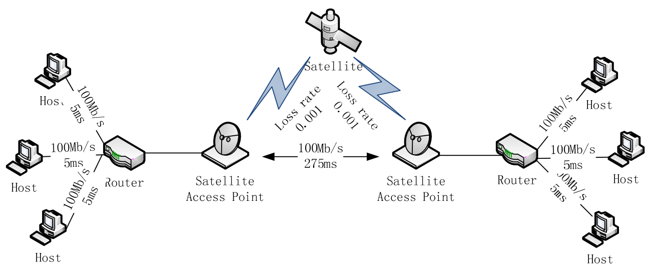


Fig. 3 Simulation Topology

3. Simulation-Based Comparisons and Analysis

In this section, the fairness of FAST TCP and XCP, as well as throughput, queue length and friendliness to TCP are evaluated. The simulation topology is shown in Fig. 3 and the default parameters of the GEO satellite networks are listed in Table 1.

Table 1: Default Parameters

Item	Bottleneck	Side link
RTT(ms)	550	10
Bandwidth(Mb/s)	100	100
Bit Error Rate	0.001	0.0
Flow Number	10	1

3.1 Fairness

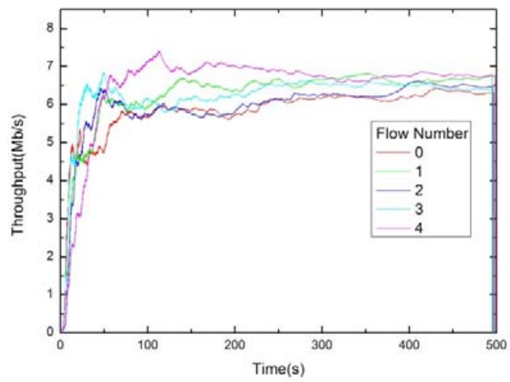


Fig. 4 Fairness of FAST TCP.

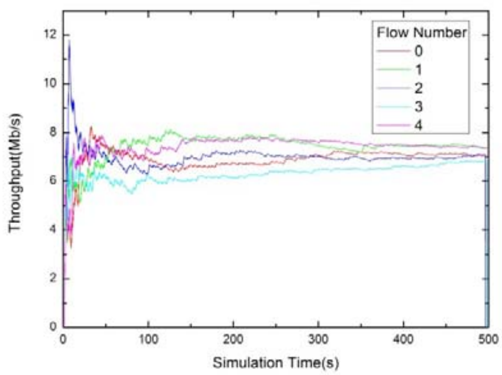


Fig. 5 Fairness of XCP.

To evaluate the fairness of FAST TCP and XCP, we set the number of flows is 5 and compare the throughput of every flow. Figure 4 and Fig. 5 show that in steady state, both FAST TCP and XCP maintain good fairness. But the convergence time of FAST TCP is much longer than that of XCP. The reason is that XCP uses explicit feedback to probe the available bandwidth and can acquire the fair share during several RTTs. However, FAST TCP uses the window adjustment approach as TCP, so it needs more time to reach equilibrium.

3.2 Throughput and queue length of the bottleneck link

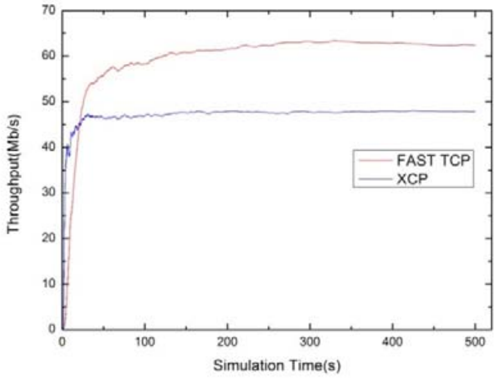


Fig. 6 Throughput of the bottleneck link.

Figure 6 shows that both FAST TCP and XCP cannot achieve high utilization due to high BER in satellite links. High BER results in constant retransmissions and more bandwidth is needed to deal with the packet losses. In steady state, the throughput of FAST TCP is 20% higher than that of XCP, but FAST TCP still has the long convergence time problem.

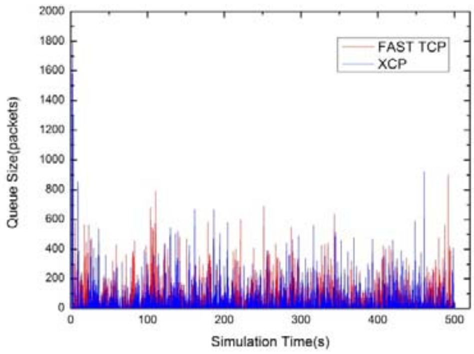


Fig. 7 Queue length of the bottleneck link.

Figure 7 shows that both FAST TCP and XCP have oscillation in queue length, but the average queue length is below the reasonable limit the buffer can afford. When packets lost, FAST TCP halves the window and enter loss recovery phase, and XCP uses the same policy to reduce the window to 1 as TCP doses. Constant packet losses make FAST TCP and XCP switch between dealing with packet loss and re-probing the available bandwidth, which results in queue oscillation.

3.3 Efficiency under various BERs

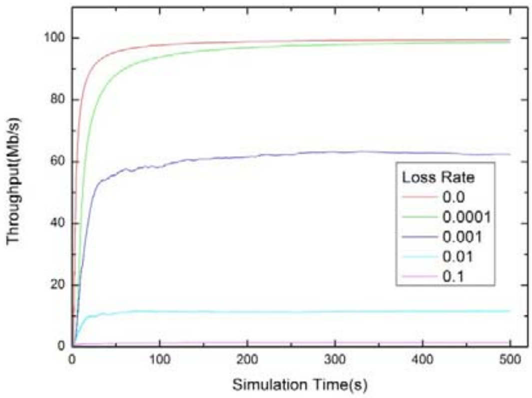


Fig. 8 FAST TCP under Various BERs.

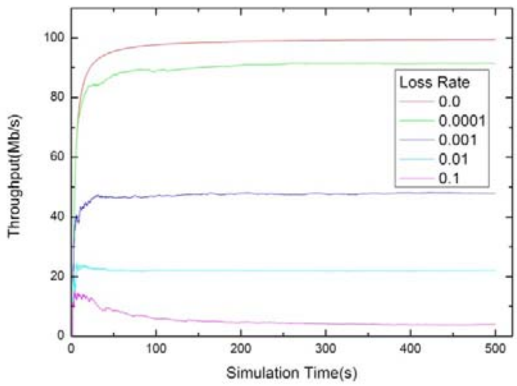


Fig. 9 XCP under Various BERs.

Figure 8 and 9 show that the throughput of FAST TCP and XCP declines sharply as the BER increases because more retransmission is needed and more bandwidth is wasted. Especially when the BER is 0.1, the throughput of FAST TCP is almost 0. In spite of constant packet losses, XCP can reach the equilibrium faster than FAST TCP due to the explicit feedback information from the routers. When the BER is higher than 0.001, XCP outperforms FAST TCP remarkably.

3.4 Friendliness to TCP

When coexisting with FAST TCP flows or XCP flows, TCP flows cannot acquire the fair share of bandwidth because TCP uses the AIMD policy to adjust the window. Meanwhile, TCP misinterprets the packet loss due to high BER as congestion loss, which degrades the TCP drastically. No matter FAST TCP flows or XCP flows, they coexist with TCP flows can acquire much higher throughput than TCP. The results are depicted as Fig. 10 and Fig. 11.

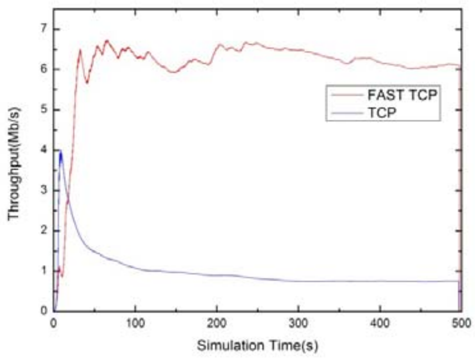


Fig. 10 FAST TCP vs.TCP

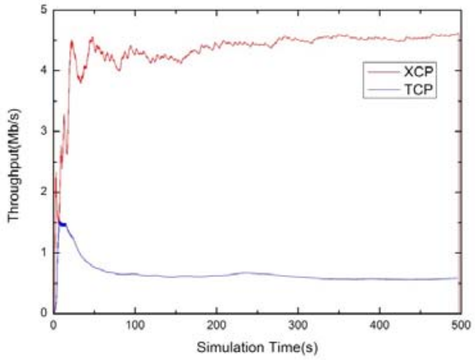


Fig. 11 XCP vs. TCP

4. Conclusion and Future Work

Through extensive simulations, the performance of FAST TCP and XCP such as fairness, the throughput and queue length of the bottleneck link, throughput under various BERs and friendliness to TCP is evaluated in the paper. The simulations show that FAST TCP and XCP maintain good fairness, the throughput of FAST TCP bottleneck link is 20% higher than that of XCP bottleneck link under the same conditions. But the convergence time of FAST TCP is longer than that of XCP because XCP uses explicit feedback approach to achieve fair share faster. Both FAST TCP and XCP keep almost the same reasonable queue length in addition to the queue oscillation. The results also show that BER has drastically negative impact on the performance of these two protocols. The throughput decreases sharply as the BER increases. The reason is that both FAST TCP and XCP interpret all packet losses as congestion loss and congestion control measures are taken. When coexisting with FAST TCP flows or XCP flows, TCP flows suffer severe throughput degradation because of AIMD policy. FAST TCP flows or XCP flows do not grab bandwidth from TCP flows.

Allowing for the characteristics of the satellite networks, the congestion control mechanism in such environment should deal with not only high bandwidth-delay but high BER. Our future work is focused on the design of an explicit congestion control mechanism with measures dealing with high BER in satellite networks.

References

[1] S. Floyd, “HighSpeed TCP for large congestion windows,”RFC 3649. [Online]. Available: http://www.ietf.org/rfc/rfc3649.txt, Dec. 2003.

[2] C. Jin, D. X. Wei, and S. H. Low, “Fast TCP: motivation,architecture, algorithms, performance,” In Proc. IEEE INFOCOM 2004, Hong Kong, Mar. 2004, pp.2490-2501.

[3] D. Katabi, M. Handley, and C. Rohrs, “Congestion control forhigh bandwidth-delay product networks,” in Proc. ACM SIGCOMM 2002, Pittsburgh, Aug. 2002, pp. 89-102.

[4] L. Xu, K. Harfoush, and I. Rhee, “Binary increase congestioncontrol (BIC) for fast long-distance networks,” In Proc. IEEE INFOCOM 2004, Hong Kong, Mar. 2004, pp. 2514-2524.

[5] R. King, R. G. Baraniuk, and R. Riedi, “TCP-Africa: anadaptive and fair rapid increase rule for scalable TCP,” In Proc. IEEE INFOCOM 2005, Miami, Mar. 2005, pp.1838-1848.

[6] David X. Wei, Cheng Jin, Steven H. Low, and Sanjay Hegde,“Fast tcp: Motivation, architecture, algorithms, performance,” IEEE/ACM Transactions on Networking, vol. 14, no. 6, pp. 1246–1259, Dec. 2006.

[7] L. S. Brakmo and L. L. Peterson, “TCP Vegas: end-to-endcongestion avoidance on a global Internet,” IEEE Journal on Selected Areas in Communications, vol. 13, no. 8, pp. 1465–80, Oct. 1995.

[8] C. Yuan, L. S. Tan, L. H. Andrew, et,al., "A Generalized FAST TCP scheme", Computer Communications, Volume 31 Issue 14, September, 2008, pp. 3242-3249.

[9] H. Y. Zhang, X. Y. Zhang, B. H. Fan, et. al., "Adaptive FAST TCP", Future Networks, 2010, Hainan, China, pp. 114-118. Jan. 2010.

[10] X. Jun, Y. Lu, L. Zhu, "A New Congestion Control Algorithm for TCP Based on Integrated Feedback", in Proc. ICCT 2011, Jinan, China, 2011, pp. 515-519.

[11] K. Y. Zhu, K. L. Yeung, V. O. K. Li, "P-XCP: a transport layer protocol for satellite IP networks", Dallas, USA, in Proc. Globacom 2004, Nov. 2004, pp. 2707-2711.

[12] Y. Sakumoto, H. Ohsaki, M. Imase, "Increasing Robustness of XCP (eXplicit Control Protocol) for Dynamic Traffic", GLOBALCOM 2007, Washington, USA, Nov. 2007, pp. 2025-2030 .

[13] Y. Su and T. Gross, “WXCP: explicit congestion control for wireless multi-hop networks,” in Proc. IWQoS 2005, Passau, Jun. 2005, pp. 313-326.

[14] D. M. L. Pacheco, C. Pham, L. Lefevre, “XCP-i: explicit control protocol for heterogeneous inter-networking of high-speed networks. in Proc. GLOBECOM 2006, San Francisco, Nov. 2006, pp. 1-6.

[15] Y. R. Yang, C. T. Chan, Y. C. Chen and Y. C. Chang, “Smart window adjustment for XCP's congestion management,” in Proc. ICACT 2006, Phoenix Park, Feb. 2006, pp. 476-479.

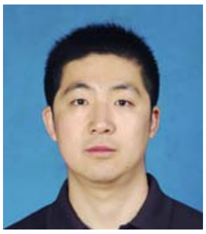
[16] X. W. Yang, Y. B. Lu and L. Zan, “Improving XCP to achieve max–min fair bandwidth allocation,” Computer Networks, vol. 54, issue 3, pp. 442-461, Feb. 2010.

[17] H. R. Zhou, G. Z. Dai, F. H. Ye and H. X. Zhang, “S-XCP: improving XCP to achieve efficient and fair bandwidth

allocation," in Proc. GLOBECOM 2009, Honolulu, Nov. 2009, pp. 1-6.
[18] J. Fasson, F. Hobaya and F. Arnal, "XCP in a geostationary satellite context," in Proc. IWSSC 2008, Toulouse , Oct. 2008, pp. 145-149.
[19] M. Welzl, A. Abfalterer and S. Gjessing, "XCP vs. CUBIC with Quick-Start: observations on implicit vs. explicit feedback for congestion control," in Proc. IEEE ICC 2011, Kyoto, Jun. 2011, pp. 1-6.
[20] S. Floyd, M. Allman, A. Jain and P. Sarolahti, "Quick-Start for TCP and IP," RFC 4782. [Online]. Available: http://www.ietf.org/rfc/rfc4782.txt, Jan. 2007..

Yandong Sun was born in 1978. He received B.E. and M.E. degrees in computer science and technology from Harbin Institute of Technology, China, in 2004 and 2006, respectively. He is currently a Ph.D. candidate in computer architecture of Harbin Institute of Technology. His research interests include computer networks architecture, wireless networks protocol design especially congestion control mechanisms in satellite networks.

Zhenzhou Ji was born in 1965. He received Ph.D. degree in computer architecture from Harbin Institute of Technology, China, in 2000. Since 2003, he is a professor in computer science and technology with Harbin Institute of Technology. His research interests include advance computer architecture, parallel processing technology and computer networks. He is the director of T&R Division of Computer Hardware Fundamental School of Computer Science & Technology of Harbin Institute of Technology, and also the vice-director of Professional Committee of Computer Architecture of China Computer Federation. He anticipated in or presided many researches, such as projects of National Natural Science Foundation of China, prearranged projects of General Armament Department and projects of 863. He was awarded one Third Prize of the Science & Technology Progress by the Spaceflight Department of China one Third Prize of the Science & Technology Progress by the Science & Technology Committee of Heilongjiang Province and one 3rd-class prize of the Science & Technology Progress by Commission of Science Technology & Industry for National Defense.



Hui Wang was born in 1983. He received B.E. degree in computer science and technology from Hebei University of Technology in 2005 and M.E. degree in computer science and technology from Harbin Institute of Technology, China, in 2008, respectively. He is currently a Ph.D. candidate in computer architecture. His research interests include network modeling and congestion control in satellite networks.