ITC19/ ITU & ITC Workshop for Developing Countries 2099-2108
LIANG X.J. and XIN Z.H.(Editors)
V.B. IVERSEN and KUO G.S.(Editors)
Beijing University of Posts and Telecommunications Press

Congestion Control Scheme Aiming at P2P Applications in High-Speed Networks*

SONG Lihua, WANG Haitao and CHEN Ming

Institute of Command Automation, PLA Univ. of Sci. & Tech.
Box 23, Biaoying 2, Yudao Street, Nanjing, China. 210007
minnehaha@126.com

Institute of Communication Engineering, PLA Univ. of Sci. & Tech.
Box 123, Biaoying 2, Yudao Street, Nanjing, China. 210007
haitmail@126.com

Institute of Command Automation, PLA Univ. of Sci. & Tech.
Box 23, Biaoying 2, Yudao Street, Nanjing, China. 210007
mingchen@public1.ptt.js.cn

Abstract: Existing congestion control mechanisms were developed for early low-speed, small scale networks. Today as bandwidth-delay product continues to increase, they suffer from poor performance such as low efficiency, slow convergence, full queue and instability. On the other hand, P2P applications have grown dramatically. Their bandwidth-consuming characteristic puts further pressure on congestion control mechanisms. To cope with this situation, a FAST TCP based novel congestion control scheme is proposed in this paper, which makes use of large scale network measurement and fuzzy logic control technologies. The central idea is to gather network performance metrics periodically to supervise end systems on properly configuring FAST TCP parameter. According to network state, a fuzzy logic inference engine is used to determine the right number of extra packets every connection should maintain in networks. Simulations under both single and multiple bottlenecks indicate that this scheme can achieve high throughput and steady queuing delay in high bandwidth-delay product networks. It is particularly suitable for serving P2P like traffic with high volume, long-lived flows.
Keywords: Congestion Control, P2P Application, Network Measurement, Fuzzy Logic Control, FAST TCP, Extra Packets

1. INTRODUCTION

Congestion Control is one of the critical technologies that ensure network usability, stability and efficiency. In the past two decades, Internet has scaled up by several orders of

* This work was supported by the National Natural Foundation of China under Grant No. 90304016

magnitude in size, speed, load and users. While there were no severe performance decline, it mainly owes to the broadly employment of congestion control mechanisms in TCP protocol. However, as the bandwidth-delay product continues to grow, existing mechanisms present poor performance such as low efficiency and slow convergence. It is generally believed that these mechanisms will eventually become performance bottlenecks themselves.

On the other hand, thanks to bandwidth increase and data compression progress, Peer to Peer applications have multiplied in recent years. P2P applications, for example eDonkey2000 [1] and BitTorrent [2], are among the most popular applications today. People use them to share audio, video or software resources. The shared files varies from several megabytes to hundreds megabytes. Being called elephants, their transmission often consumes huge bandwidth and long time. Research shows that P2P activity has been a significant and growing component of Internet traffic [3-4]. Because of its distinct characteristics comparing to transient Web flows, it puts additional pressure on congestion control mechanisms.

Current researches on congestion control can be categorized into two classes according to the location where they put the main responsibilities. One is the traditional scheme, which relies on end systems to response to congestion and makes no specific assumption on routers. The classical TCP Reno and its variations, as well as TCP Vegas [5] and TCP FAST all belong to this category. Due to their limited locations, end systems can only deduce network states from packet loss events and RTT information. They have some inevitable blindness in making control decisions. Thus it is difficult to improve the congestion control's efficiency merely from end systems. The other category enhances routers' function by making them discard packets at some probability and explicitly or implicitly inform end systems when congestion is going to happen. Examples are various Active Queue Management schemes such as Random Early Discard, Adaptive-RED, Random Exponential Marking, Proportional-Integral controller and Virtual Queue. These schemes perform very well when properly configured. But it is argued that they all become oscillatory and prone to instability in high bandwidth-delay product networks [6]. Furthermore they increase router's overhead, cause difficulties in network investment preservation and continuable development. Their practicability is restricted.

Until nowadays no mathematical model founded is precise enough to characterize Internet activity [7-8]. So network measurement becomes an effective method to understand and control its state. By using network congestion information obtained from measurement, the blindness of control decisions could be eliminated and the control efficiency could be increased. Network is a complicated distributed system, on which measurement results are often with uncertainty and delay. Fuzzy logic is one of the tools of Computational Intelligence. Fuzzy logic control is good at situations where rigorous control theoretic approaches cannot be used due to difficulties in obtaining a formal analytical model, while at the same time some intuitive understanding of the process is available. So it is very suitable for network control.

This paper proposes a novel congestion control scheme based on FAST for P2P applications in high-speed networks. It combines large scale network measurement and fuzzy logic control technologies. This scheme can adjust user activities according to network state without special help from routers. Simulations indicate that it can achieve higher utilization as well as more stable queuing delay than routine mechanisms for P2P applications in high bandwidth-delay product networks. And this depends on neither amount nor location of network performance bottlenecks.

Section 2 illustrates elements of this novel scheme, in company with a suggested implementation frame. Components of its fuzzy control part are to be introduced in section 3. Simulations in section 4 will explore the proposed scheme's performance in both single and multiple bottleneck environments; meanwhile compare it with some other classical congestion control schemes. Section 5 concludes the paper.

2. MEASUREMENT-BASED CONGESTION CONTROL

FAST [9-10] is a congestion control scheme proposed earlier for TCP on the base of control theory. Its main idea is to deduce available bandwidth from queuing information carried in RTT and try to keep constant extra packets in networks for every connection from the source end. By doing this the protocol can avoid unstable congestion state as well as to acquire quickly spare bandwidth. FAST has shown outstanding efficiency, stability and fairness in high-speed network simulations [10]. But its key parameter, the expected number of extra packets, is difficult to set, which might be its main weakness. This is because that the right number of extra packets is related to link bandwidth and the amount of concurrent connections scramble for resources, whereas end systems have no way to get more information than loss event and RTT on network state. Large scale network measurement and the widely deployed measurement infrastructure could ease this problem yet.

The basic idea underlying measurement-based congestion control is to supervise end systems on configuring right number of extra packets depending on network state information which is obtained from distributed measurement. Put it concretely, a macroscopic supervision layer is introduced upon end systems, which performs measurement of network backbone's performance periodically (say in tens of seconds or minutes). According to the gathered metrics such as bottleneck bandwidth and RTT, in conjunction with the target queuing delay, the upper layer comes up with appropriate extra packets amount for contesting connections and informs corresponding end systems. FAST algorithm is used in end systems. The value of extra packets parameter is updated when it is informed by the upper layer. Measurement is

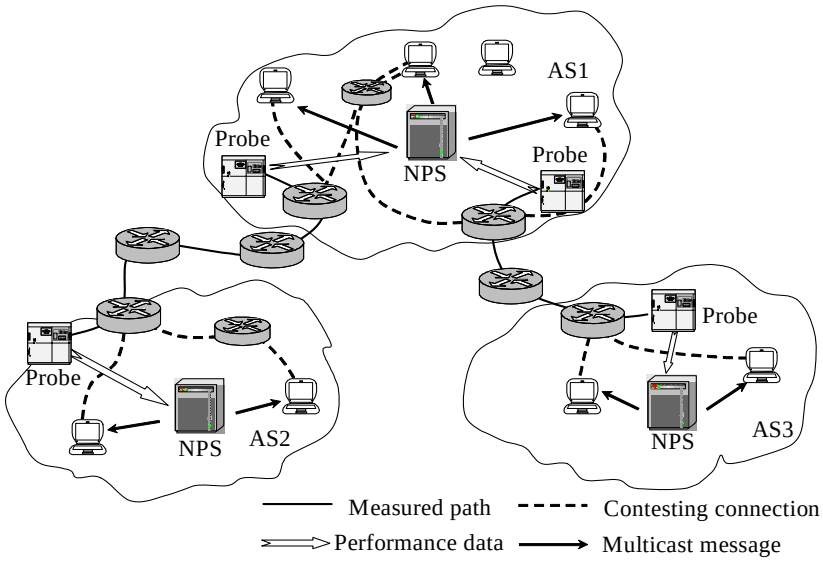


Figure1. An implementation frame of measurement-based congestion control

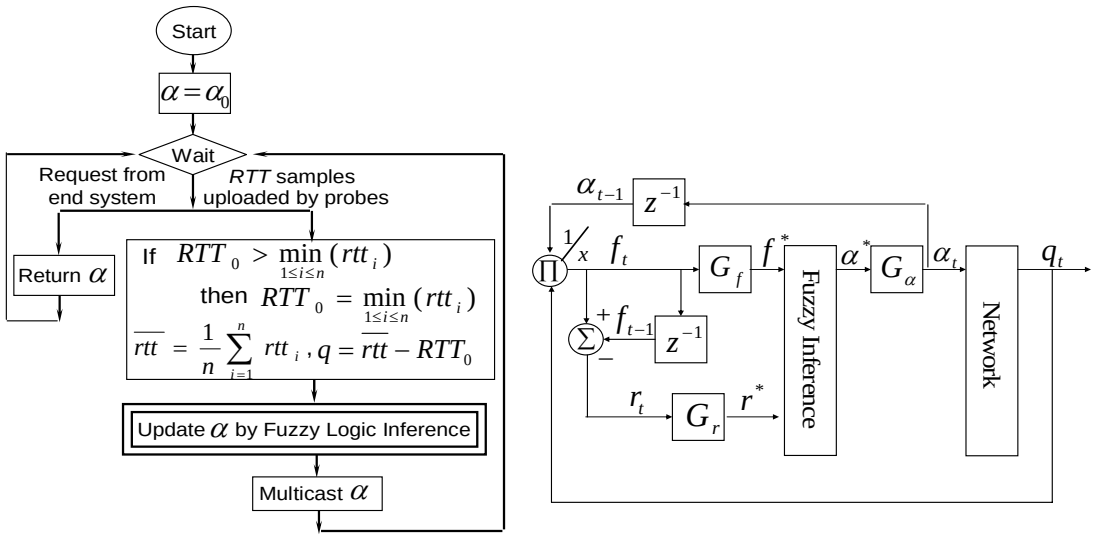
limited to backbone and its frequency is not high. So networks are supposed to get not much intrusion.

Figure 1 shows a possible implementation frame of the suggested scheme. Measurement tools, such as bprobe and Ping, carry out measurement tasks at end points of network backbone. The results are uploaded to Network Performance Servers. NPS is an intelligent component we introduced, which usually locates at the center of a network domain such as Autonomous System. NPSes collectivity constitutes the core of the suggested frame. The task of a NPS is to collect and process measurement data from probes in its domain and communicate processing results to end systems. For example, in this frame, NPSes receive measurement samples uploaded by probes monitoring at backbone end points in their each domain. Basing on these samples and some historical data, they work out the right number of extra packets for correlative end systems in their domains and inform them by multicast. End systems join relevant multicast groups when connection established and leave when terminated, between which they reset their parameters periodically upon received multicast messages.

In our prior work, we have designed and implemented a network monitoring and measuring system following an architecture called Universal Network Measurement Platform [11]. Through the cooperation of a name server, monitoring centers and various probes distributed in networks, this system can both monitor network traffic in a flow fashion in Ethernet and ATM, and acquire end to end performance metrics such as delay, loss rate, bandwidth, jitter and route actively. A preparatory NPS has also been developed in the UNMP system environment. It can process measurement data according to the fuzzy algorithm to be explained in the next section, and then respond to requests from end systems for expected number of extra packets.

3. FUZZY LOGIC CONTROL IN NPS

Figure 2(a) depicts NPS processing flow, where α represents the expected number of



(a) NPS flow chart

(b) Fuzzy logic control system

Figure 2. NPS functional blocks

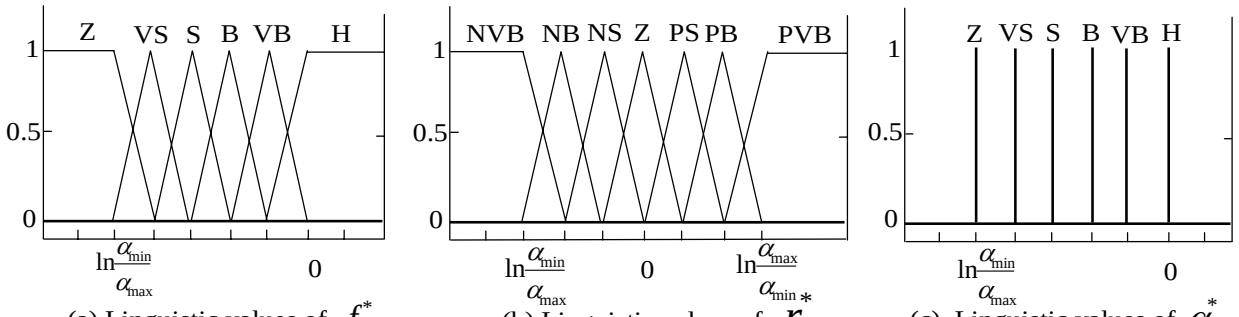
extra packets for contesting connections, with initial value α_0. RTT_0 is the minimum *RTT* having been obtained, which represents the round trip propagation delay of the measured path. And q represents the current average queuing delay. α updating algorithm, the most important part of this flow, is implemented as a fuzzy logic control system. The fuzzy system is designed to regulate the extra packets every connection holds by considering the quantity of concurrent connections and its changing rate, in order to achieve a target queuing delay. The designed fuzzy logic control system is detailed in Figure 2(b). It has two inputs. One is the ratio of average queuing delay at time t and the expected number of extra packets at last instant, i.e. $f_t = q_t / \alpha_{t-1}$, which represents estimation for the quantity of concurrent connections. The other is the difference of f_t, denoted as $r_t = \Delta f_t = f_t - f_{t-1}$, which represents the concurrent connections' changing rate. Output is α_t, the expected number of extra packets at time t. The control cycle is identical to measurement cycle. f^*, r^*, α^* are the scaled values of f_t, r_t and α_t respectively. Their scaling gains are as follows, where $\tilde{q}$ is the target queuing delay and $\alpha_{\max}$, $\alpha_{\min}$ determine high and low limits for α:

$$f^* = \ln\left(\left(\alpha_{\min} \cdot f_t\right)/\tilde{q}\right) \tag{1}$$

$$r^* = \begin{cases} r_t/|r_t| \cdot \ln\left(|r_t| \cdot \alpha_{\max}/\tilde{q}\right) & if\ |r_t| > \tilde{q}/\alpha_{\max} \\ 0 & if\ |r_t| \le \tilde{q}/\alpha_{\max} \end{cases} \tag{2}$$

$$\alpha^* = \ln\left(\alpha_t/\alpha_{\max}\right) \tag{3}$$

As shown in Figure 3, 6, 7, 6 linguistic values are defined for f^*, r^* and α^* in their respective ranges. For the sake of computational simplicity, uniform, full intersectant



(a) Linguistic values of f^* (b) Linguistic values of r (c) Linguistic values of α^*

Figure 3. Membership functions of the linguistic values of input and output variables

Table 1
Linguistic rules – rule base

α^* \ f^*	Z	VS	S	B	VB	H
NVB	H	H	H	H	VB	×
NB	H	H	H	B	VS	Z
NS	H	H	VB	S	VS	Z
r^* Z	H	VB	B	S	VS	Z
PS	×	B	B	S	VS	Z
PB	×	×	S	S	VS	Z
PVB	×	×	×	VS	VS	Z

triangular shaped member functions are used for the input variables, while a uniform singleton member function is used for the output variable. Based on all these linguistic values, a rule base (see Table 1) containing 35 fuzzy rules are defined, which is fine tuned by observing the progress of simulation. Mamdani [12] method is used in the inference engine. And the defuzzification process employs weighted mean method [13].

4. SIMULATIONS AND RESULTS

4.1. Simulations of Single Bottleneck

In the first group of experiments, we compare the performance of measurement-based scheme with other classical congestion control schemes under single bottleneck network environment. These reference schemes include FAST [10], NewReno/adaptive-RED [14] and NewReno/REM [15]. The simulation is conducted in ns-2.7 simulator. Just as illustrated by Figure 4, 4 long-lived TCP connections with large volume data (imitating P2P applications) join the contention for bandwidth of link R1R2 in serial number turn every 400 seconds. When the last one has come in and persisted for 400 seconds, they begin to quit in reverse turn with the same space. Here R1R2 is the only congested link in the whole scene. In the

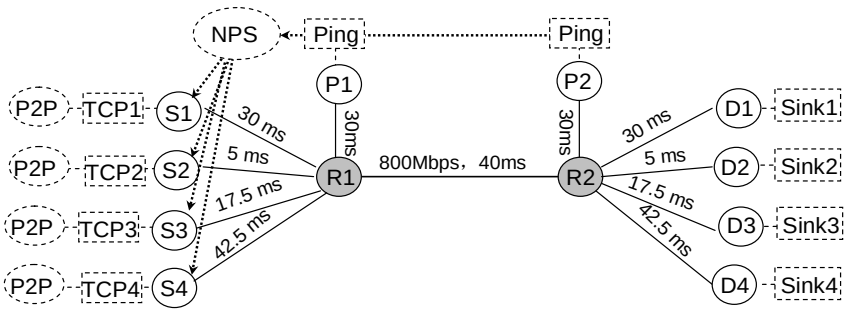


Figure 4. Simulation topology of single bottleneck link

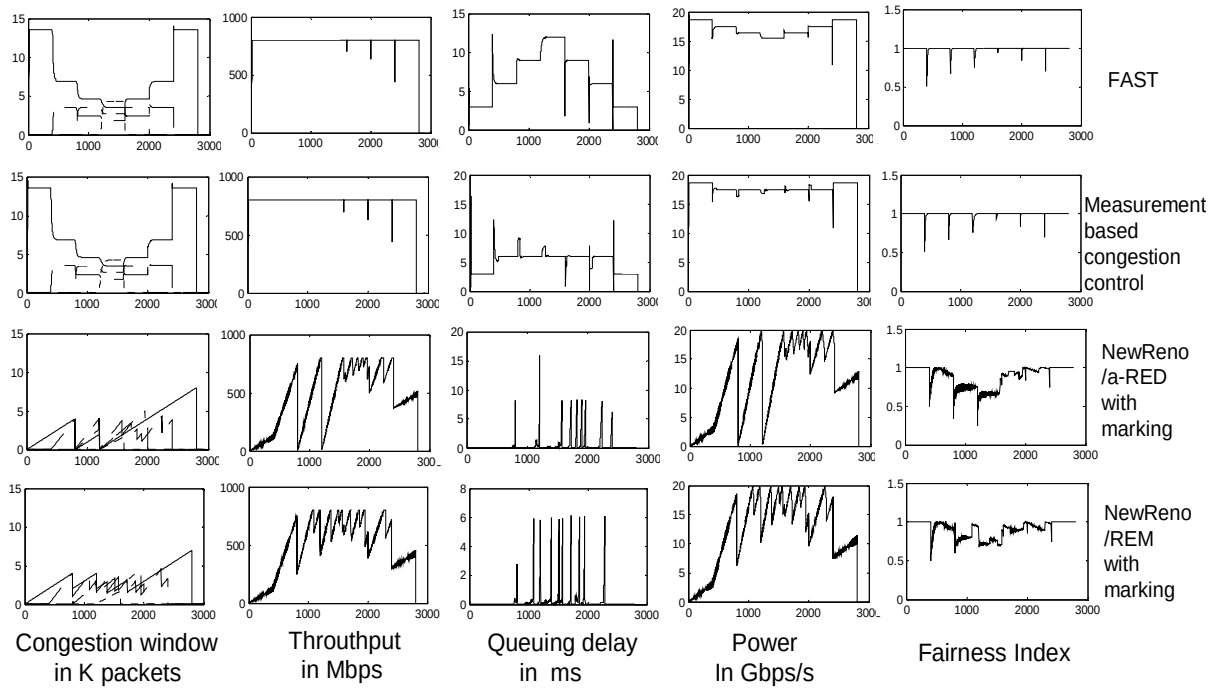


Figure 5. Performance of FAST, measurement-based scheme, adaptive-RED and REM under single bottleneck

measurement-based congestion control scheme, Ping agents on node P1 and P2 perform measurements of RTT on path P1R1R2P2 once every 1 minute. 10 samples are gathered every time and delivered to the NPS on P1, where they get processed according to the algorithm described in last section. Processing results, i.e. the expected number of extra packets, is communicated to the sources. $\alpha_{\max} = 300KBytes$, $\alpha_{\min} = 1500Bytes$. Multicast is replaced by function invocation for simplicity. For a-RED and REM schemes, the experiments are conducted with marking packets as congestion feedback. Their parameters are set to authors recommended values or deduced by complying recommended procedures. The expected number of extra packets in FAST is 200 constant, while all other 3 schemes take 6 milliseconds as target queuing delay. Packets are 1500 bytes length through all simulations.

Figure 5 shows the congestion window, throughput on bottleneck link, queuing delay, Power and fairness index of all 4 schemes as functions of time. Power and fairness index [16] are used to evaluate congestion control mechanism's effectiveness and fairness. Here power is defined as the ratio of throughput and forwarding delay on bottleneck link.

What can be figured out from Figure 5 is that a-RED and REM all exhibit some extent slow convergence and oscillation under high bandwidth conditions. Oscillation brings low utilization, as well as unsatisfactory effectiveness and fairness. Contrasting to a-RED and REM, FAST and the measurement-based scheme both achieve good convergence, high and stable throughput and fairness. However, since every connection tries to maintain 200 extra packets in the router buffer, FAST's queuing delay climbs and Power descends gradually as more connections joining in. It can be deduced from this phenomenon that packets will have to be dropped in a more crowded environment. This situation gets improved after introducing network measurement technique. The figure reveals that in the measurement-based scheme,

the number of extra packets each connection maintains in router buffer get adjusted in time according to network state. When there are a few connections, each one reserve quite a number of extra packets for convergence and quickly acquiring spare bandwidth. When there are pretty many connections, each one reserve relatively small extra packets for keeping queuing delay and Power around constant values.

4.2. Simulations of Multiple Bottlenecks

In the second group of experiments, we investigate the measurement-based scheme's performance when there is more than one bottleneck. In the first experiment of this group, whose network topology and simulation results are shown in Figure 6, there are two congested links at different times. Initiated at time 0, 1 connection (represented by solid arrows) activates in every 200 seconds to transfer large volume data from node S1, S2 and S3 to D1 with the same round trip propagation delay of 220 milliseconds. Here link R1R2 is the bottleneck. 200 seconds hereafter, i.e. at time 600 seconds, 2 connections activate simultaneously to transfer cross traffic from node S4 and S5 to D2. Represented by small dashed arrows, they both have 160 milliseconds round trip propagation delay. Now the bottleneck shifts to link R3R4. After 400 seconds, cross traffic transmission terminates and the bottleneck comes back to R1R2 at time 1000 seconds. Then the original 3 connections quit in turn with 200 seconds space. Through the whole process, probes keep performing measurements of RTT on path R1R2R3R4 periodically. Probes upload their sample data to NPS and through which to inform the expected number of extra packets to all connections. The target queuing delay is 6 milliseconds.

In Figure 6(b), the scheme maintains a very high throughput. The bandwidth of bottleneck links gets almost full utilized. Queuing delay on bottleneck links tends to constant under adjustment of the extra packets parameter (Figure 6(c)). The target queuing delay, 6 milliseconds on the whole path R1R2R3R4, is achieved perfectly well.

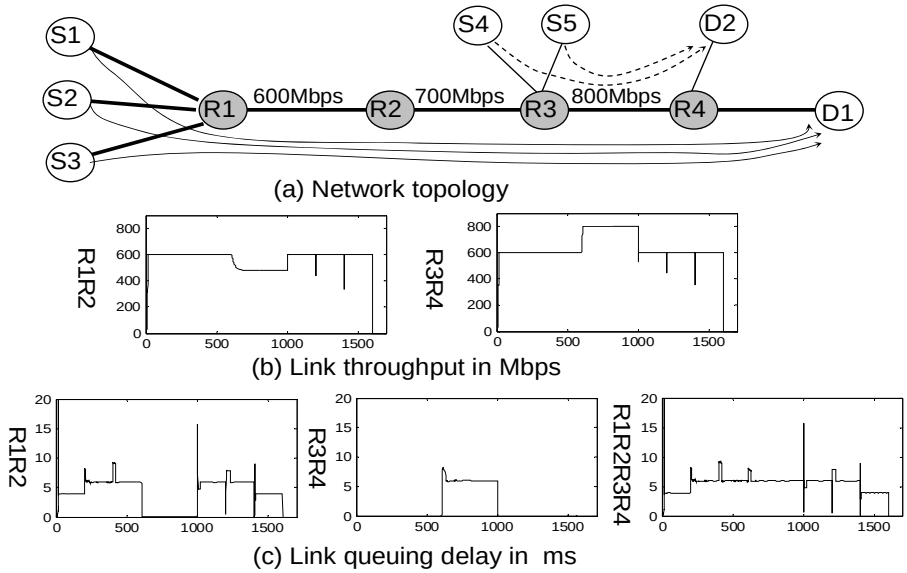


Figure 6. Performance of measurement-based congestion control scheme under two bottlenecks

REFERENCES

1. eDonkey2000. http://www.edonkey2000.com/ .
2. BitTorrent. http://bitconjurer.org/BitTorrent/ .
3. S. Sen, W. Jia. Analyzing Peer-to-Peer Traffic across Large Networks. ACM/IEEE Transactions on Networking. 2004, 12(2): 219 - 232
4. T. Karagiannis, A. Broido, N. Brownlee, et al. File-Sharing in the Internet: A Characterization of P2P Traffic in the Backbone. Technical Report, University of California, Riverside. 2003
5. L. Brakmo, L. Peterson. TCP Vegas: End to End Congestion Avoidance on a Global Internet. IEEE Journal on Selected Areas in Communication. 1995, 13(8): 1465-1480
6. S. Low, F. Paganini, J. Wang, et al. Dynamics of TCP/AQM and a scalable control. Proc. of IEEE Infocom'2002
7. C. Williamson. Internet Traffic Measurement. IEEE Internet Computing. 2001, 5(6): 70-74
8. W. Willinger, R. Govindan, S. Jamin, et al. Scaling Phenomena in the Internet: Critically Examining Criticality. Proc. of Natl. Acad. Sci. USA. 2002, 99(Suppl. 1): 2573-2580
9. C. Jin, D. Wei, S. Low, et al. FAST TCP: From Theory to Experiments. Networking Lab, California Institute of Technology. http://netlab.caltech.edu/pub/papers/fast-030401.pdf, 2003. 12
10. C. Jin, D. Wei, S. Low. FAST TCP: Motivation, Architecture, Algorithms, Performance. Zhang Z. Proc. of IEEE InfoCom'2004. HongKong, China: 2004
11. M. Chen, R. Zhang, L. Song, et al. UNM: An Architecture of the Universal Policy-based Network Measurement System. Rouskas G. Proc. of 13th IEEE Workshop on LANMAN. San Francisco, USA: 2004
12. Y. Zhang, Y.Q. Liu. Soft Computing Methods. Beijing, China: Science Publisher, 2002. 48-54
13. N.Y. Zhang, P.F. Yan. Neural Networks and Fuzzy Logic Control. Beijing: Tsinghua University Publisher, 1998. 192-193
14. S. Floyd, R. Gummadi, S. Shenker. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. Technical Report, Proc. of ICSI'2001
15. S. Athuraliya, S. Low, V. Li, et al. REM Active Queue Management. IEEE Network Magazine. 2001, 15(3): 48-53
16. L. Peterson, B. Davie. Computer Networks: A System Approach. Callifornia, USA: Morgan Kaufmann Publishers, 2000. 454-457