The Effect of Active Buffer Management on TCP Adaptivity

Zoltán Szabó, Sándor Molnár

High Speed Networks Laboratory
Dept. of Telecommunications and Telematics
Budapest University of Technology and Economics
H-1117 Magyar Tudósok krt. 2, Budapest, Hungary
{szabo, molnar}@ttt-atm.ttt.bme.hu

Zsolt Kenesi

Traffic Analysis and Network Performance Lab.
Ericsson Hungary Ltd
H-1037 Laborc u. 1, Budapest, Hungary
zsolt.kenesi@eth.ericsson.se

1 Introduction

Data traffic based on Transmission Control Protocol (TCP) is dominant in IP networks. The detailed analysis of the features and behaviour of TCP is hot topic of recent research programs [3]. (The most important information about TCP can be found in RFC 793 in which TCP originally was defined and RFC 1122 and RFC 2001 that contain further additions.

Since TCP connections are capable of providing and interpreting feedback they adapt to different network conditions that are relevant to the actual scenario they experience. It was published in papers [1] and [2] that if a TCP connection shares a bottleneck link with non-adaptive background traffic flow, TCP adapts to it inheriting and propagating the correlation structure and statistical properties of the background traffic flow above a characteristic time scale. This time scale of adaptation depends on the end-to-end path properties.

Service providers' IP networks support traditionally public Internet service, so a single best-effort serviceclass was adequate to support all Internet application. With different applications requiring different QoS it is important to introduce QoS support into the network. The QoS architecture of IP networks is the Differentiated Service solution [4].

The DiffServ architecture provides QoS by dividing traffic into different categories and ensures differential treatment of traffic within each class. Active queue management techniques and queue scheduling algorithms are currently implemented in most of the routers (Cisco, Juniper) to support different QoS classes in IP networks.

In our research we focus on the analysis of the effects of widely spread quality of service mechanisms on TCP adaptivity.

2 Characteristics of TCP adaptivity

In our work we analysed TCP adaptivity that is described by characteristics of adaptivity firstly published in [2]. The applied topology is depicted in Figure 1.

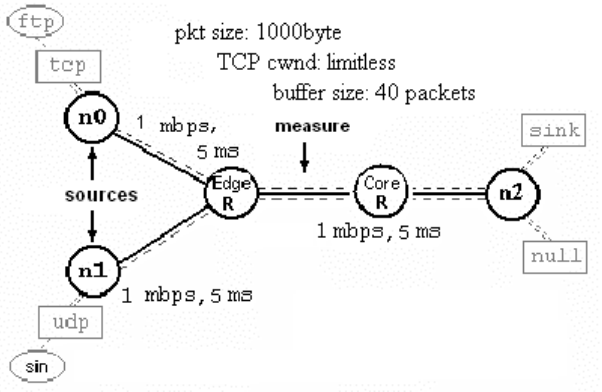

Figure 1: Topology of simulated network

Since TCP connections are capable of providing and interpreting feedback they adapt to different network conditions that are relevant to the actual scenario they experience. It was published in [2] that if a TCP connection shares a bottleneck link with a non-adaptive background traffic flow, TCP adapts to it inheriting and propagating the correlation structure and statistical properties of the background traffic flow above a characteristic time scale. This time scale of adaptation depends on the end-to-end path properties, i.e. round-trip time, widow size, etc.

In the cited paper the measure of adaptivity, i.e., the ratio of power spectra of TCP and background traffic at a given frequency, was investigated. During the analysis the rate of non-adaptive UDP background traffic changed as a function of sine wave with constant amplitude, as we discuss below.

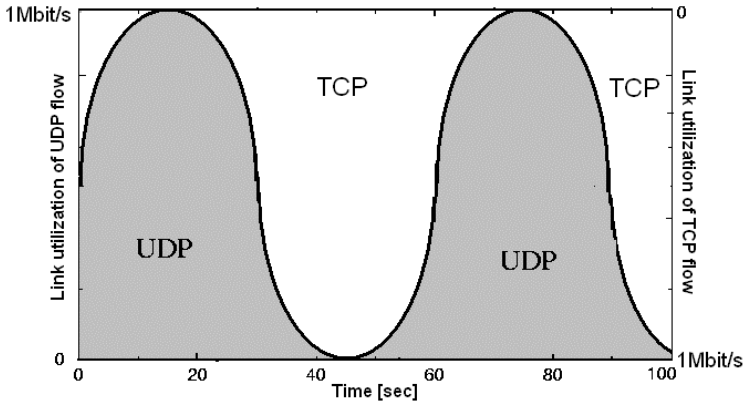

Figure 2: Fluctuated UDP and totally adapted TCP traffic on bottleneck link

The power spectrum of sine wave of a given frequency *f* consists of a single frequency component at *f*. If TCP is able to adapt to the fluctuations of the background traffic flow, it changes as a function of sine wave of a given frequency *f*, see Figure 2. In this case, the same frequency component *f* should appear as a significant spike in the power spectrum of the TCP traffic, as well. Note that the background traffic utilised half of the bandwidth in average at each frequency setting.

The measure of adaptivity is defined as spectral density of the adapting TCP rate process divided by the spectral density of the background traffic at a given frequency.

An important question is how this measure changes with the frequency of background traffic, i.e., with time-scale of fluctuations. It is possible to plot the adaptivity characteristics of TCP by performing the above mentioned experiment for a wide range of time-scales.

Figure 3 shows the characteristics of adaptivity. It can be observed that TCP adaptivity is almost invariant to TCP versions.

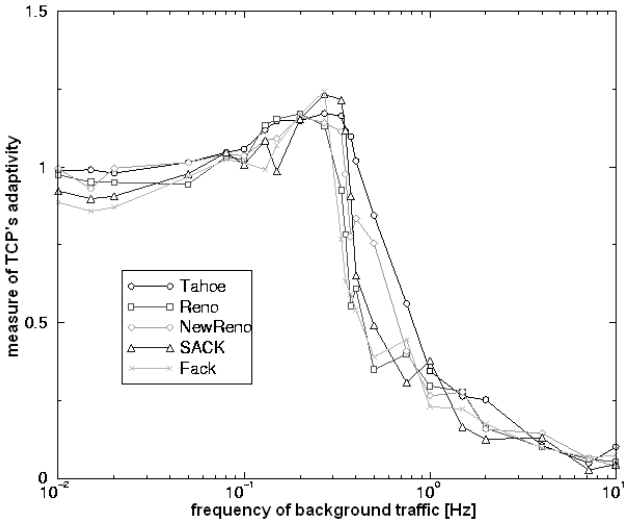

Figure 3: Characteristic of TCP adaptivity

If the time-scale the background traffic fluctuates on is large ($f_{background}$ <0. 1Hz), the measure of adaptivity is close to 1. It means that TCP is able to adapt almost perfectly to the background traffic. However, on smaller time-scales ($f_{background}$ >0. 3Hz) TCP cannot adapt to the changes of the background traffic, therefore the ratio of the TCP and UDP frequency component is very small.

In the range of 0.1-0.3 1/sec a resonance effect can be observed. At these time-scales TCP is more aggressive, and gains even higher throughput than what is left unused by the non-adaptive background flow.

3 QoS MECHANISMS

The *Differentiated Service* architecture [4] provides QoS over IP networks by classifying traffic and applying different QoS classes.

In our measurement we differentiated the TCP and non-adaptive background UDP traffic so that the flows were classified by their transmission protocol. We defined different QoS classes for non-adaptive UDP flow and for TCP-based data traffic.

Previously reported [1] characteristics of TCP adaptivity describes how TCP was able to adapt to periodically changing background traffic while TCP and UDP met in a shared buffer. The tail-drop buffer size is 40 packets in the configuration. There is no queue management algorithm implemented in tail-drop queues, they drop packets when the queue is full. Tail-drop queues treat all traffic in the same way.

In practice, buffer size is an important setting in the network configuration. Figure 4 shows the effect of buffer size on the curve of TCP adaptivity.

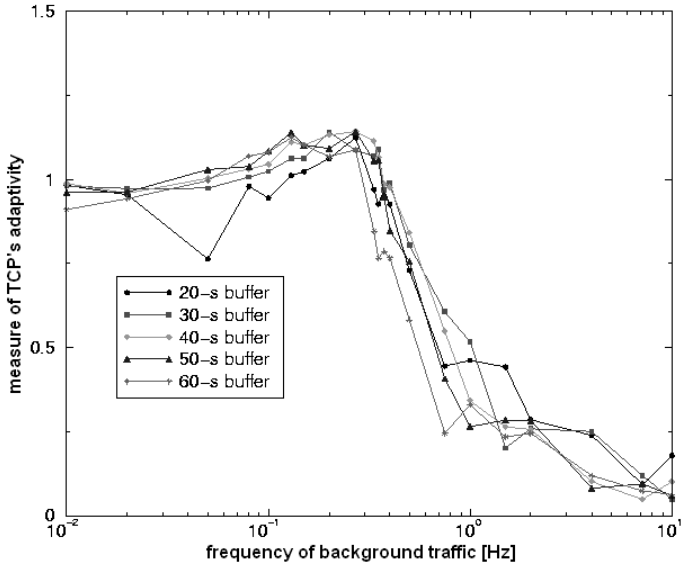

Figure 4: Effect of changed buffer size on TCP adaptivity

It can be observed that TCP adaptivity almost independent of the buffer size therefore we kept 40 packets for the initial buffer size in the following scenarios.

3.1 Random Early Detection (RED)

RED mechanism was proposed by Sally Floyd and Van Jacobson in the early 1990s to address network congestion in a responsive rather than reactive manner. RED offers a widespread, effective congestion-avoidance mechanism. Like Tail Drop, RED treats all traffic equally in a given buffer. But RED takes advantage of the congestion control mechanism of TCP. RED indicates the reduction of the source transmission rate by randomly dropping packets prior to periods of high congestion. RED uses a packet drop profile to control the aggressiveness of its packet discard process. The drop profile defines a range of drop probabilities across a range of queue occupancy states, see Figure 5.

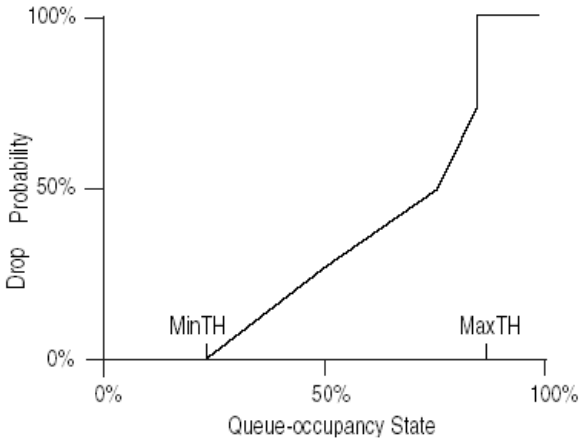

Figure 5: Characteristics of RED profile

In light congestion phase the RED starts to drop packets and it will decrease its transmission rate until all the packets reach their destination which indicates that the congestion is cleared.

Figure 6 represents the effect of RED mechanism on TCP adaptivity. The RED profile shows that RED drops packets from the whole buffer. The packet drop rate increases linearly with

the average queue size. RED does not make difference between transport protocols, thus UDP packets are also dropped from the queue. We found that TCP cannot adapt to the fluctuation of background traffic due to increasing packet loss rate of RED.

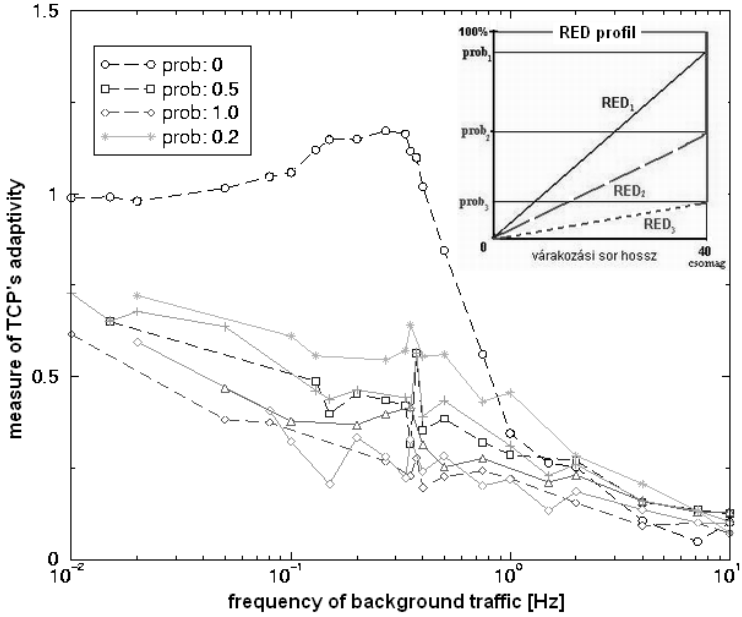

Figure 6: Effect of RED mechanism on TCP adaptivity

The result clearly shows that using RED for UDP traffic is pointless. In the following we analyse the effect of another RED mechanism which differentiate between TCP and UDP protocol.

3.2 Weighted Random Early Detection (WRED)

To support QoS in IP network it is essential to differentiate the TCP-based, loss-sensitive data traffic from real-time, delay-sensitive UDP stream even if these flows meet in a shared buffer. WRED is an extension to RED that allows you to assign different RED drop profiles to different types of traffic in the same queue. In our work we defined different RED profile to TCP and UDP flow.

Applying WRED active queue memory management in the shared buffer queuing we was able to define different QoS classes with different RED parameters for TCP and UDP in the shared buffer.

In this case we analysed the effects of WRED on TCP features. To push an advantage of WRED we defined different RED profiles for TCP and no RED profile for UDP, which means that, UDP flow was able to utilise the whole shared buffer without packet loss.

In the following we present the effects of different WRED configurations on TCP adaptivity. We defined a less aggressive RED profile for TCP and no RED profile for UDP. This setting results in that UDP queuing works as Tail Drop mechanism and TCP flow does not loss packet below the queue occupancy level of 50 percent. Figure 7 illustrates that this WRED configuration has no effect on TCP adaptivity.

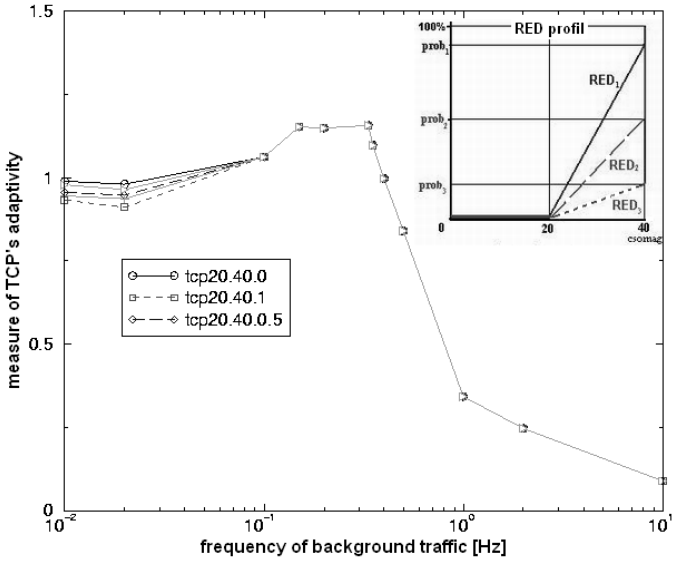

Figure 7: Effect of RED mechanism on TCP adaptivity (less aggressive drop profile for TCP)

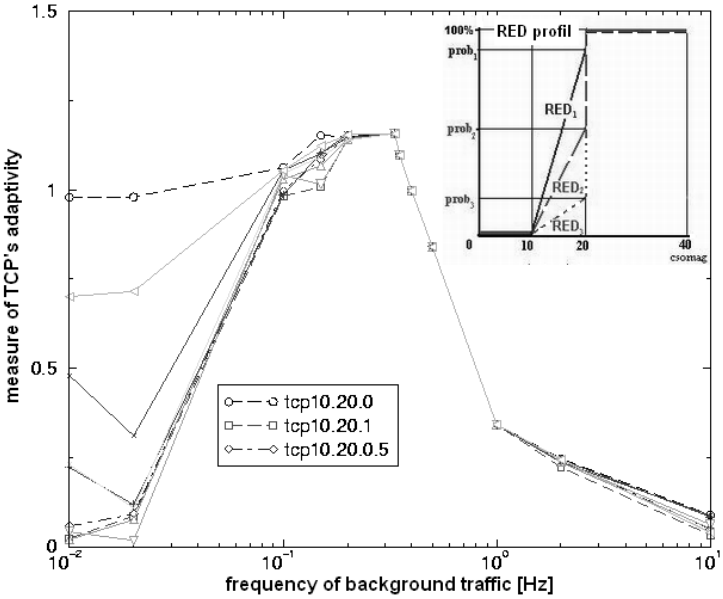

Figure 8: Effect of RED mechanism on TCP adaptivity (more aggressive drop profile for TCP)

In the second scenario more aggressive RED profile is applied. It can be noticed that the adaptivity characteristics is quite robust above cut-off frequency, see Figure 8. This WRED configuration does not affect TCP adaptivity when background traffic fluctuates at these time-scales. However, in small time-scales these features reduced linearly as drop probability increased.

3.3 Weighted Fair Queuing (WFQ)

We showed the results about TCP adaptivity when TCP and UDP used a shared buffer with different queue memory management. In IP networks different queues can be defined for TCP and UDP traffic that are prioritised by queue scheduling disciplines.

WFQ [7] offers dynamic, fair queuing that divides bandwidth across queues of traffic based on weights. WFQ ensures that all traffic is treated fairly determined by its weight.

WFQ is an automated scheduling method that provides fair bandwidth allocation to all network traffic based on the applied priority. WFQ is a flow-based algorithm that simultaneously schedules interactive traffic to the front of a queue to reduce response time and fairly shares the remaining bandwidth among high-bandwidth flows.

We also analysed the effect of WFQ with different priorities for UDP flow on TCP adaptivity.

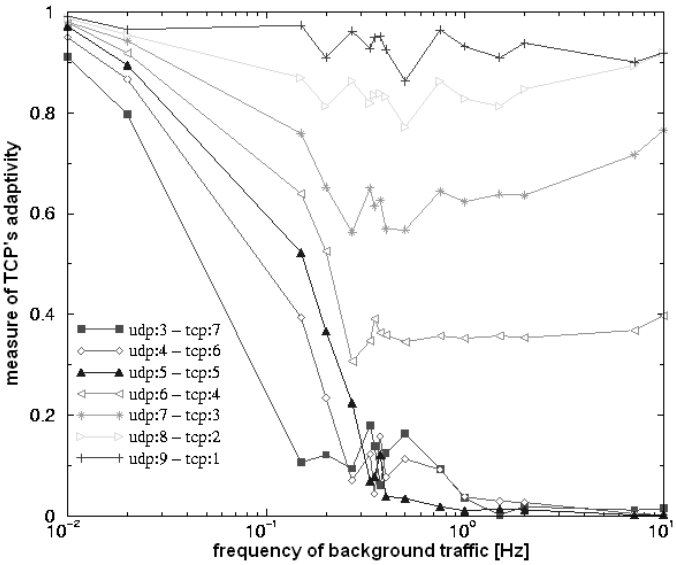

Figure 9: Effect of WFQ mechanism on TCP adaptivity

Figure 9 shows that TCP adaptivity is dependent on priority levels allocated to UDP and TCP applying WFQ. In case of higher priority for UDP serving TCP flow is able to adapt better to background traffic on all time-scales.

Setting WFQ queue scheduling discipline in our configuration we defined different buffer for TCP and UDP flow instead of one shared buffer, which caused variation on TCP adaptivity as Figure 8 shows.

We observed that in case of TCP and UDP flows are separated into different buffers TCP does not suffer heavy packet losses and able to send packets when UDP packets are not transmitted.

4 SUMMARY

In our research we focused on how TCP adaptation is affected by active buffer management techniques. Our results show that in case the TCP shares bottleneck with non-adaptive background traffic flow, i.e., UDP traffic, the characteristics of adaptivity can be modified by applying RED or WRED queuing mechanisms and the measure of variation depends on different RED and WRED settings.

If the traffic is separated into different buffers and served by WFQ, TCP adaptivity is almost perfect at each time-scale.

All results can also contribute to the better understanding of the fractal properties of TCP traffic.

5 REFERENCES

[1] A. Veres, Zs. Kenesi, S. Molnár, G. Vattay, *TCP's Role in the Propagation of Self-Similarity in the Internet, appear in Computer Communications*, Special issue on Performance Evaluation of IP Networks and Services.

[2] A. Veres, Zs. Kenesi, S. Molnár, G. Vattay*, On the Propagation of Long-Range Dependence in the Internet*, ACM Computer Communication Review, Vol. 30, No. 4, pp. 243-254, October, 2000.

[3] Gary R. Wright, W. Richard Stevens, *TCP/IP Illustrated*, Volume 2 The Implementation, Addison-Wesley Publishing Company 1995

[4] S. Blake, D. Black, M. Carlson, *An Architecture for Differentiated Services*, Request for Comments 2475, December, 1998

[5] Sally Floyd, Van Jacobson*, Random Early Detection Gateways for Congestion Avoidance*, IEEE/ACM Transactions on Networking, 1993

[6] Chuck Semeria, *Supporting Differentiated Service Classes: Active Queue Memory Management*, Juniper Networks, White Paper, 2002, http://www.juniper.net

[7] H. Zhang, *Service Disciplines For Guaranteed Performance Service in Packet Switching Networks*, Proceedings of the IEEE, 83(10), Oct 1995.