

COVERPAGE

Measurements Based Analysis of the Characteristics of TCP in a Differentiated Services Capable Network

Principle author

Virpi Laatu

Tampere University of Technology, Pori

P.O. BOX 300

FIN-28101 PORI, FINLAND

Tel. +358 2 627 2743, Fax +358 2 627 2727

Email: virpi.laatu@pori.tut.fi

Other authors:

Jarmo Harju

Tampere University of Technology, Tampere

P.O. BOX 553

FIN-33720 TAMPERE, FINLAND

Tel. +358 3 365 3854, Fax +358 3 365 3808

Email: harju@cs.tut.fi

Pekka Loula

Tampere University of Technology, Pori

P.O. BOX 300

FIN-28101 PORI, FINLAND

Tel. +358 2 627 2740, Fax +358 2 627 2727

Email: virpi.laatu@pori.tut.fi

Keywords: Congestion control, quality of service, performance measurements, network management

STATEMENT:

If this paper is accepted, I will attend the conference to present it.

Virpi Laatu

Measurements Based Analysis of the Characteristics of TCP in a Differentiated Services Capable Network

VIRPI LAATU, JARMO HARJU AND PEKKA LOULA

Tampere University of Technology, Pohjoisranta 11 P.O Box 300, FIN-28101 Pori, Finland

Tel +358 2 627 2743, fax: +358 2 627 2727

laatu@pori.tut.fi, harju@cs.tut.fi, loula@pori.tut.fi

Abstract — This paper presents results from performance measurements in a test network that can be operated as an ordinary best effort or as a Differentiated Services (DS) capable network. The target of the measurements was to analyse the effects that the DS mechanism has on the behaviour of the TCP flow control mechanism. The network implements service differentiation by using eight drop precedence levels and separate classes for real time and non-real time traffic.

Performed tests verified that our QoS (Quality of Service) implementation allows QoS -controlled TCP senders to recover faster from congestion than what is possible for TCP sessions in purely best-effort network. When QoS is enabled, the access node helps smooth traffic bursts flowing into the Diffserv core network. As a consequence, the congestion window oscillates moderately and keeps TCP operating in the congestion avoidance phase, making the rate adjustment scheme more controllable.

Keywords: Congestion control, quality of service, performance measurements, network management

1. Introduction

The rapid growth of the Internet has led to the situation where the services that the Internet provides have been changing. Traditional best-effort service does not meet the delivery needs of these new time and performance-critical services. The Differentiated Services (DS) architecture [1] is a respectable approach for providing service differentiation in an IP networks.

A significant amount of the Internet traffic today is carried by the TCP transport protocol [2]. However, in the future DS enabled Internet, the coexistence of TCP and UDP is becoming particularly important with the increasing use of streaming video and Internet telephony using UDP transport over the Internet. There are various issues to be addressed before any sophisticated end-to-end services can be offered, such as fairness and performance issues among non-responsive traffic (UDP) and responsive traffic (TCP). These issues need to be resolved before quantitative assurances of some form can be specified in SLA (Service Level Agreement) contracts.

Many studies concerning the performance of TCP and interactions of TCP and UDP flow in a Differentiated Services capable network have been recently presented [3], [4], [5], [6], [7], [8], [9], [10], [11] and [12]. Nandy et al. [7] have proposed an Aggregate Flow Control (AFC) technique with Diffserv traffic conditioner to improve the bandwidth and delay assurance of differentiated services. Sahu et al. [4] discovered that the throughput of a TCP connection in a DS network does not only depend on the sender but also on the receiver. They showed that the appropriate choice of the ACK class has a great influence on the throughput. In [5], [9] and [6], it has been observed that it is difficult to guarantee requested throughput to the TCP flows in a differentiated services network. This is mainly due to congestion avoidance technique in TCP, which makes it difficult for flows to achieve performance goals. Seddigh et al. [9] studied the effects of five different factors on the throughput rates of TCP flows in a RIO (RED with In and Out) -based Differentiated Services network. These factors include RTT (Round Trip Time), packet size, number of micro-flows in a target aggregate, size of the target rate and existence of UDP flows. The authors indicate that in an over-provisioned network, all target rates are achieved regardless of these factors. However, the factors clearly cause uneven distribution of excess bandwidth among TCP flows. Further, they reported that as the network approaches an under-provisioned state, these factors play an important role in determining the extent to which aggregates of TCP flows achieve or do not achieve their target rates. In [8] it is shown that while the use of concurrent connections increases throughput for those applications that break a TCP connection into multiple connection, it also increases the packet drop rates in the Internet. Results of our earlier measurements [13] showed that our QoS implementation guarantees quite fair treatment for the competing flows when TCP and UDP flows are in the same bottleneck link. However, the measurements showed that the TCP flow aggregate will be punished for its good behaviour when the target rate of the UDP flow increases.

The problems concerning the performance of TCP and the dynamics of TCP and UDP are the topics that still need more research for a better understanding. This paper

extends the results of our earlier measurements [13], [14], [15] and [16] providing further insight on the properties of QoS schemes based on the Differentiated Services architecture. This paper increases the understanding of TCP behavior and can be used to predict TCP throughput with its contract rate in a differentiated services network. In this paper, we are focusing our research on the characteristics of bulk data transfer within our DS scheme. The performance of our DS scheme was evaluated on the basis of data transfer times and the change of the congestion window size measurements.

In TCP, the window size for the connection, called a congestion window (cwnd), is changed according to the network congestion. Upon connection startup phase, TCP enters the slow start phase where cwnd is initialized to one MSS (Max. Segment Size). TCP then probes for the available bandwidth inside the network by exponentially increasing its congestion window per round trip time (RTT). Once a packet loss occurs in the network, TCP assumes that there is congestion in the network and therefore reduces the congestion window size to half of its current value (fast recovery). In network congestion situations, UDP flows will not back off, and quickly they take over the bandwidth they need. The reader is referred to [17] for a detailed discussion on TCP congestion control mechanisms.

The remainder of the paper is organized as follows: In Section 2, we describe the measurement scenario, i.e. what things were measured, how, and in what kind of conditions. In Section 3, we present the performed tests and the results of these tests. Finally, Section 4 summarizes our findings.

2. The SIMA service model

Differentiated Services is a one way of satisfying the diverse needs of the Internet users by providing QoS in the Internet. The Differentiated services framework has been proposed within the IETF to provide different levels of service for network traffic over IP networks. The DS architecture is based on a simple model where traffic entering a network is classified and possibly conditioned at the boundaries of the network according to a service level agreement, and assigned to different behavior aggregates. This is done by marking the DS fields of the packets' IP headers with the corresponding Differentiated Services Code Point (DSCP). The DSCP value initiates a specific Per-Hop Behavior (PHB) treatment at each network node. After the packets have been classified at the boundary of the network, they are forwarded through the core network according to the per-hop behavior associated with the DS codepoint. [1]

We have focused our research on a particular DS mechanism called Simple Integrated Media Access (SIMA) model [18]. The SIMA service model is an

implementation of Dynamic RT/NRT PHB Group suggested in [19] and [20]. As with all DS architectures, the building blocks of this service consists of two main parts: access nodes (boundary nodes) and core network nodes (interior nodes). The traffic metering and classification of every flow are performed at the access nodes in order to determine a DSCP for each packet. SIMA provides service discrimination with 16 PHBs and two service classes: one class for real time (RT) and the other for non-real time (NRT) traffic, both having 8 priority values (DP, drop precedence).

In SIMA, the QoS offered to a customer is characterised by the service class (RT or NRT) and one parameter, Nominal Bit Rate (NBR). When using the network, the customer is not tightly bound to the agreed NBR, but the experienced service level is dependent on the ratio MBR/NBR, where MBR stands for Momentary Bit Rate. When the customer exceeds the NBR (resulting in high value of the ratio) the packets are labelled with lower priority. In a lightly loaded network this may still result in a reasonable service, while in a congested network it will lead into packet drops. On the other hand, low ratio means that packets get high priority values decreasing the possibility of packet drops, and thus increasing reliability. This provides the customer and the service provider with an easy and intuitive way to trade off QoS against price in their mutual Service Level Agreement.

The MBR is the only parameter that keeps changing in an active flow. The other two (NBR and service class) are static parameters that are agreed upon in the SLA. Once the MBR has been measured (see [21] for more details), the DP value for the packet is determined by the formula (1).

$$x = 4.5 - \frac{\ln\left(\frac{MBR}{NBR}\right)}{\ln(2)} \quad (1)$$

$$DP = \begin{cases} 6 & \text{if } x \geq 6 \\ \text{Int}(x) & \text{if } 0 < x < 6 \\ 0 & \text{if } x \leq 0 \end{cases}$$

The access nodes are responsible for measuring the index x for individual flows and marking IP packets with correct DP values based on x . The operation of the access node is discussed in more detail in [21].

Inside the network (i.e., in the core network nodes) the DSCP field is used to determine how packets are treated. The treatment in these nodes include packet scheduling, buffering and forwarding functions. The operation of a core network node is based only on the information in the DSCP field. A key variable in the scheduling and buffering unit of a core network node is

the allowed drop precedence (DP_a) value. When the core network node receives a datagram, it calculates the DP_a and compares it to the datagram's DP value. The DP_a calculation is based on the occupancy levels of the real time and non-real time buffers. If an incoming packet has smaller precedence than DP_a it is dropped before the actual buffering. The operation of the core network node is discussed in more detail in [22].

3. Measurement arrangement

The details of the measurements are presented in this section. First, we give a brief review of the goal of the measurements. After that, we present the measurement methodologies and the test network used in the measurements. We also describe the software tools and traffic workloads used in the study and explain how these were used in each experiment.

3.1. Goals of the measurements

We create a single bottleneck link in the network and examine how congestion on this link affects the congestion control of TCP. We mainly focus on the time-dependent behaviour of cwnd, since the change of the window size has a significant impact on TCP behaviour.

3.2. Performance metrics

Concerning data transfer, performance metrics can be divided into two major categories: user metrics and operator metrics. Although the two are related, an end user is typically interested in the time it takes to complete the data transfer (goodput) and predictability of service, whereas the service provider is concerned more about network resource utilization and aggregate service quality metrics.

In these experiments we measured the time required to transfer 1 MB of data in our test network. While this is a very telling metric, it is not necessarily the only measurement that indicates how TCP flow behaves. In addition, we measured the average throughput obtained by a TCP flow and the number of segments dropped by the network (retransmitted packets). TCP performance is well-known to be sensitive to a connection's round trip time (RTT). The larger the RTT, the more time is needed to recover after a packet loss. That's why we were also interested to see if this effect is diminished by the use of our DS scheme. We calculated both the mean of RTT and the standard deviation of RTT.

The flow rate (BW) of a connection of TCP is equal to the ratio of the window size W of the connection divided by its total round trip delay RTT. Equation 2 [23] defines the minimum window size W (in bytes) required for a TCP to fully utilize the given amount of available

bandwidth. The flow rate of connection is a function of not only the window size W of the connection but also of the window sizes of the other connections. When the network is uncongested, the flow rates are proportional to the window sizes. However, as congestion occurs, the rates are no longer linear in the window sizes.

$$BW = \frac{W}{RTT} \quad (2)$$

3.3. Test network

The test network used for the measurements is shown in Figure 1. The test network can be considered to consist of three sections: two access networks built around hubs, and the DS capable core network that connects these two access networks. The nodes that were located at the border of the core network (Border1 and Border2), had both the access and code node functionality. The core routers (Core1 and Core2) were built only with the core node functions. The bottleneck link is between Core1 and Core2 nodes. Bandwidth of every link except the link between Adtech AX/4000 and interface eth2 of Core1 node, is limited to 10 Mbps. The link between Adtech AX/4000 and the interface eth2 of Core1 node was built with the 100Base-TX Ethernet technology in the half-duplex mode. The source and the sink for the background traffic were directly connected to Core1 and Core2, respectively. For TCP traffic, the source and the sink were connected to Hub-1 and Hub-2 respectively. All the network nodes were Pentium II PC's running Linux RedHat 6.2 distribution, including kernel version 2.2.9. An Adtech AX/4000 traffic generator was used to generate UDP background traffic to the network. Adtech AX/4000 was equipped with two 10/100 Ethernet interfaces such that background traffic could be originated from a given network card and received by the second one. The traffic flows traversing through to the core network during the measurements are shown in Figure 1.

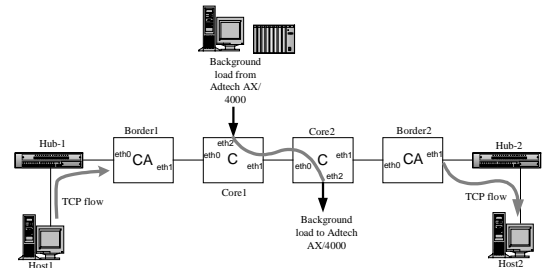


Figure 1. The test network

3.4. Configurations

During the measurements we utilized the following nodes of the test network: 4 core nodes (Border1, Core1, Core2 and Border2), 2 access nodes (Border1 and

Border2), 2 host nodes (TCP traffic source and sink) and an Adtech AX/4000 Analyzer/Generator acting as UDP source and sink for many simultaneous UDP flows.

Configuration parameters of the core nodes concerning queue sizes were configured with the following settings: RT queue size was set to 9800 bytes and NRT queue size was set to 20000 bytes. The weighting factor for real time class was set to 0.01 and for non-real time to 0.001. The core node parameters were not modified during the measurements. Access node configurations were varied in each different test case by varying the NBR value.

3.5. Test methodology

Several different scenarios were created in order to evaluate the characteristics of a TCP flow with bulk data contents. For analysing TCP behaviour, we used TCP sender/receiver pttcp [24] utility. The pttcp utility is a publicly available benchmarking tool. It is a simple TCP client-server application, which consists of a data-sending client and data-receiving server. The pttcp is fully configurable to allow user to set various parameters such as the send and receive socket buffer sizes and the amount of data to send. We used the NewReno [25] with SACK [26] version of TCP. In our tests TCP source sent only non-real time packets. The MTU (Maximum Transfer Unit) was 1448 bytes.

We also used tcpdump to capture packets flows, and tcptrace software tool [27] was used to summarize tcpdump data set. Tcptrace can produce several different types of output containing information on each connection seen, such as elapsed time, bytes and segments sent and received, retransmissions, round trip times, window advertisements and throughput. It can also produce a number of graphs for further analysis.

While the test traffic occupies a relatively small portion of the link capacity, background traffic was needed to create congestion to the network. During the measurements (except the baseline measurements) the background load was always turned on. The background traffic used during the measurements were generated by the Adtech AX/4000 traffic generator and looped back to the same apparatus through the bottleneck link. Background load was provided with a pre-defined DP level distribution. The background traffic was marked using the TOS option of Adtech AX/4000. All UDP sources transmitted only non-real time packets. For background traffic we used the periodic (CBR, constant bit rate) setting to generate streams of packets where packet arrivals are periodic. Periodic arrivals provide a more stable comparison basis. The background load consisted of several separate UDP flows. A set of well-behaved UDP/CBR sources was used to load the network in the presence of persistent congestion so that, in

principle, it only accepts for delivery the packets that have been provided with, at least, the DP value 4. The total sum of these well-behaved UDP flows (DP=4) was about 8.5 Mbps. In addition, UDP flows with lower DP value were used to keep the allowed DP level as steady as possible.

Each test was repeated five times with the same parameters to estimate the variance between the runs. The presented results are averages of five individual measurements (except the graphs).

4. Test cases and results

In this section we present the measurements that we performed on our test network, and the obtained results. Several different scenarios were created in order to zoom out on the details of the behavior of TCP's window before and after incorporating QoS mechanism. The scenarios were composed of a single or multiple TCP flow(s) with and without UDP/CBR background flows.

We first consider a single connection behavior, analyzing how congestion control mechanisms of TCP work in our QoS capable network. In the case a single TCP source, 1 Mbyte of data is sent. We then analyze the effects that multiple simultaneous connections have on our QoS capable network.

4.1. Baseline tests

For the reference purposes, we first disabled the DS mechanisms, in order to test the behavior of the network operating in the best-effort mode. The achieved results provided a baseline for our performance measurements. The baseline measurements were made under different network load conditions. Figure 2 shows typical behaviour of the congestion window sizes as a function of time in the case where QoS was disabled and the network was over-provisioned. When a packet loss occurs, the slow-start mechanism will decrease the congestion window by half. This leads to the classic "saw tooth" pattern shown in Figure 2. The same behavior of the congestion window was observed when the QoS functionality was enabled and the network was over-provisioned. The results showed that the behavior of a single TCP flow was not depending on whether QoS was enabled or not, when the network was over-provisioned.

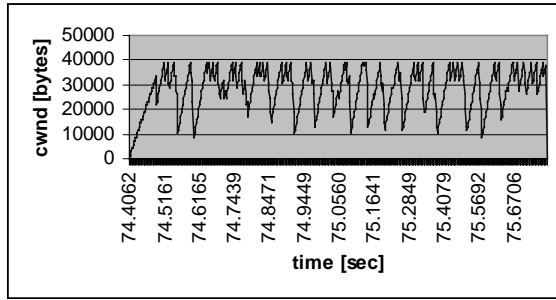


Figure 2. Congestion window as function of the time without QoS enabled and without additional background load

Next we studied the effect that the background load level had on the congestion window behavior. We varied the background load level from 8.5 Mbps to 10 Mbps. In this experiment the QoS functionality was disabled.

Figure 3 shows the behavior of the TCP flow in the case when the background load level was 9.6 Mbps. When we loaded the network so that the background load reached the link capacity, the aggressive UDP flows left no bandwidth for the TCP connection.

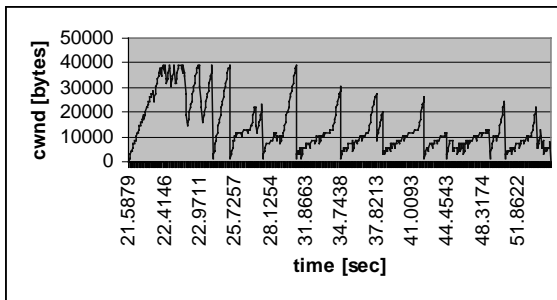


Figure 3. Congestion window as function of the time without QoS enabled and under 9.6 Mbps background load

After that we enabled the DS functionality of the core nodes and the access nodes, and made measurements in the case where the network was over-provisioned (background load level was 0 Mbps). When there was not background load no remarkable differences could be measured between the case where QoS is enabled or disabled (Figure 4).

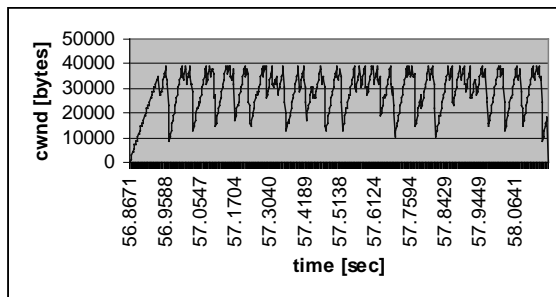


Figure 4. Congestion window as function of the time in the case where QoS is enabled and the network was over-provisioned

In over-provisioned QoS network, although the achieved throughput is high as 6.05 Mbps, most of the

packets are marked with lower DP value than four (Figure 5), since the NBR was specified to be 1.2 Mbps. At times of instantaneous congestion in the network, the packets marked with smaller DP than four are more likely to be dropped.

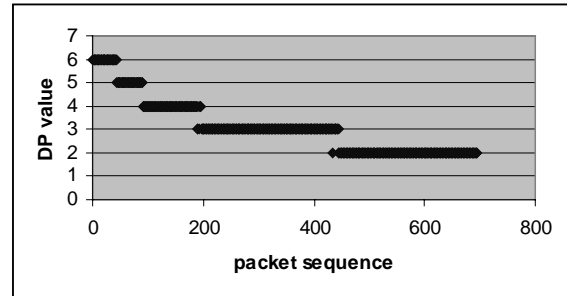


Figure 5. The DP value distribution of TCP flow

When the network is under-provisioned, a non-congestion controlled UDP sources get almost all its packets through the network at the expenses of TCP connections' throughput. Table I summarizes the results of the baseline measurements. The third column in Table I shows the average stream bandwidth achieved by a single TCP stream. The sixth column in Table I is total number of bytes retransmitted (Rexmt) for a single TCP stream during the transmission period.

Table I. Summary of the results of the baseline measurements

Back-ground load [Mb/s]	Sima on /off (NBR)	Avg. stream BW [Mb/s]	Trans. time [sec]	RTT mean/mean deviation [msec]	Rexmt data bytes
0	Off	5.99	1.34	44.4/18.4	0
8.5	Off	0.98	8.16	192.2/90.8	135545
9.6	Off	0.16	50.21	168.6/152.6	276568
9.7	Off	0.05	148.14	261.6/156.6	385168
10	Off	0	Inf.	-	-
0	On (1.2 Mb/s)	6.05	1.32	43.7/17.3	0

4.2. A single TCP flow with different NBR

In the second experiment, the characteristics of a single TCP flow were evaluated by varying the size of NBR value. In this experiment, the total sum of the background sources with DP value 4 was 8.5 Mbps. A single TCP flow was transmitted through the core network and the throughput, transfer time, RTT, and the size of the congestion window were measured. Also the DP values were recorded during the connection.

This experiment studied the impact of NBR values on the behavior of the congestion window attained by the TCP flow. We observed that in all cases our QoS scheme falls short of providing the assurance of the rate they aimed to provide. In the case where the TCP has low

NBR value (400 kbps) the average throughput of a TCP flow remains low, whereas the UDP sources (with DP value four) get 98 percents of its packets through the network at the expenses of a TCP connections' throughput. Actual TCP throughput is persistently much less than load of the network would indicate is available (1.5 Mbps for packets with DP value 4). If the NBR value is assigned too low, the core router queues will oscillate as congestion window does its characteristic sawtooth. This behavior is seen in the Figure 6.

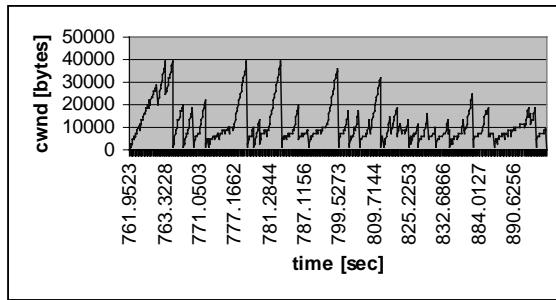


Figure 6. Congestion window as a function of time in the case where NBR rate is assigned too low

Throughput decrease is due to the fact that too many packets are receiving low DP values, thus limiting the TCP performance. The DP value distribution of TCP flow during the connection is shown in Figure 7.

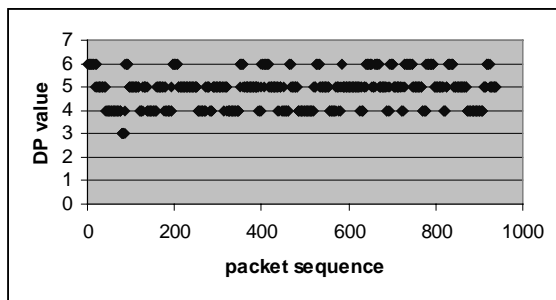


Figure 7. The DP values of TCP flow in the case where NBR rate is assigned too low

Figure 8 shows the situation where NBR value is more close to optimal (1600 kbps). In the case the congestion window oscillates moderately and the number of retransmitted packets remains low.

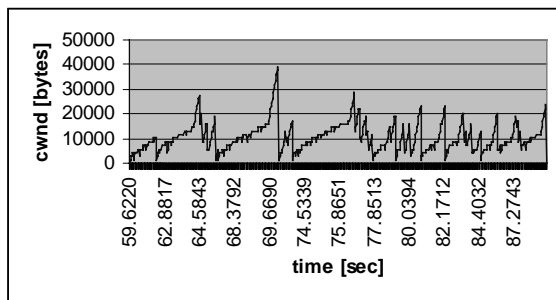


Figure 8. Congestion window in the case where NBR value is assigned more optimally (1600 kbps)

As Figure 9 shows, when the NBR value is assigned high enough, larger fraction of TCP segments are marked with priority higher than four. However, it should be noted that the fraction of packets with priority higher than four should be kept as small as possible. The priority of a packet is only meaningful relative to other packets of different priorities.

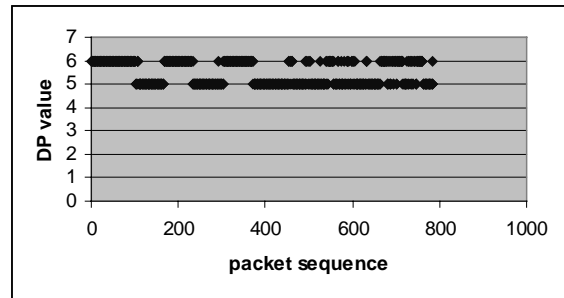


Figure 9. The DP value distribution of TCP flow in the case where NBR value is 1600 kbps

When the NBR value is assigned too large (Figure 10), the TCP packets are marked with the highest DP, thus producing a large burst of data in the beginning of a transfer.

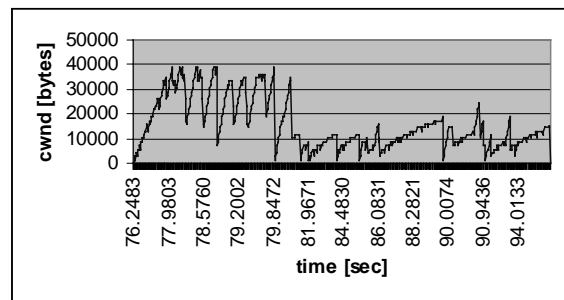


Figure 10. The behavior of congestion window in the case where NBR rate is assigned too high

We can observe from Table II that the average RTT for a TCP flow in each experiment lies somewhere between 44 and 180 milliseconds, while those for the QoS disabled case lies somewhere between 170 and 260 milliseconds. If the assigned NBR value is too low or high, TCP performance suffers from large variations in round trip time estimates. Table II summarizes the results of the measurements of a single TCP connection behavior.

Table II Summary of measurement results for a single TCP connection

<i>Sima</i> <i>NBR</i> [kb/s]	<i>Avg.</i> <i>stream</i> <i>BW</i> [kb/s]	<i>Trans.</i> <i>time</i> [sec]	<i>RTT</i> <i>mean/mean</i> <i>deviation</i> [msec]	<i>Rexmt</i> <i>data</i> <i>bytes</i>	<i>BG load</i> <i>with DP4</i> <i>reduction</i> [kb/s]
400	61.0	131.14	44.4/18.4	354760	148.9
600	137.5	58.18	129.6/125.2	237472	315.7
800	180.5	44.31	143.3/124.8	205049	450.5
1000	235.5	33.96	118.0/104.0	155817	502.6
1200	250.9	31.89	136.0/102.4	130321	309.9
1400	242.0	33.06	140.7/100.9	129754	358.7
1600	420.9	19.01	179.0/114.6	95568	178.4
1800	345.8	23.14	170.5/149.7	72401	326.0

4.3. Multiple TCP flows sharing the same NBR

In a differentiated services network, service contracts will not typically be on a per end-user basis. One common scenario will be the case where a company contracts a target rate with an ISP (Internet Service Provider). A large number of source flows originating from the company would then compete for the aggregate target rate.

In this section, we study the behavior of the congestion window when having multiple parallel connections in the aggregate. First we evaluated how throughput varies with the number of parallel connections when sharing a contract rate. In this experiment, each TCP source sent 250 kB of data. In this case the marking behavior differs from the marking of individual flows. In the marking of individual flows, the NBR rate for an individual flow is fixed. In the marking of aggregated flows, however, the NBR rate consumed by an individual flow is not fixed even though the aggregated NBR rate is fixed. In this scenario we considered the case where four parallel connections share NBR value of 800 kbps. The results of this experiment show that none of the TCP flows achieves its shared target rate (200 kbps). The results of the measurement is shown in Table III.

Table III. Summary of measurement results for multiple TCP connections

	<i>Flow1</i>	<i>Flow2</i>	<i>Flow3</i>	<i>Flow4</i>
<i>Transfer time [sec]</i>	14.39	14.70	23.86	25.17
<i>Avg. Stream Bandwidth [kbps]</i>	139.0	136.1	83.8	79.5
<i>Rexmt data bytes</i>	75297	84929	63712	57920
<i>RTT mean [msec]</i>	144.5	147.6	120.5	172.3
<i>RTT mean deviation [msec]</i>	139.1	98.7	95.6	135.7

However, for four parallel TCP streams, the aggregate bandwidth is increased from 180 kbps (see Table II, row NBR=800 kbps) to 440 kbps. While the use of parallel connections increases throughput, it comes at the cost of degraded congestion control behavior, thus leading to the increased number of retransmitted packets. In this experiment, it is also observed that the variation of RTT becomes larger. In addition, the contract rate shared by the aggregated flow is consumed unfairly within the aggregation. Figure 9 shows the congestion window size evolutions of four TCP connections.

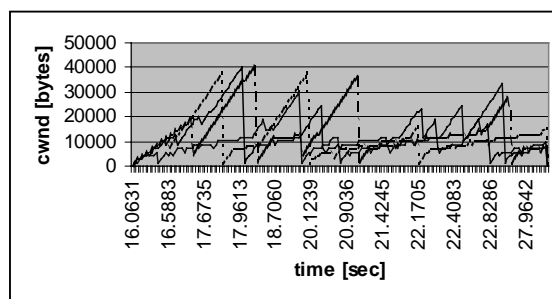


Figure 11. The behavior of congestion window in the case where four TCP connections are sharing NBR value of 800 kbps

5. Conclusions

While WWW traffic and file transfers dominate today, there are many possible new applications that could take their place. These diverse applications with different needs are driving the current efforts to differentiate the treatment of network traffic. In this study, the characteristics of TCP in a differentiated services network was considered. We measured the effects that a particular DS mechanism has on the congestion control properties of TCP. The behavior was evaluated by observing the throughput, RTT, file transfer time and the number of segments dropped by the network (retransmitted packets).

Our measurements show that the congestion window oscillates moderately and keeps TCP operating in the congestion avoidance phase, making the rate adjustment scheme more controllable, when QoS is enabled. We also observed that there happens less retransmissions with optimally assigned NBR values, which means less traffic in the network. Although the relative amount of bytes retransmitted in our case is small, the effect can be more significant when there are more contributing end systems. The absence of retransmitted bytes also implies an improvement in the goodput. This becomes very important for low bandwidth links, such as ADSL lines. However, based on the measurements, it appears that it is not possible to guarantee requested NBR rate to the TCP flows in a heavily congested differentiated services network. A further increase of NBR value does not bring any additional gain with respect to the obtained

throughput. This behavior should be taken into consideration when specifying pricing models for the DS networks.

Our measurements raised several issues, which are the focus of our future work in the area. From the performed measurements, we can conclude that our QoS scheme is able to protect TCP flows against aggressive UDP traffic and can guarantee a certain minimum bit rate. Nevertheless, the detection and policing of aggressive flows should be deployed in order to be able to provide a certain bandwidth profile to TCP connections and a reasonable level of service differentiation between TCP and UDP users.

References

- [1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", *Network Working Group, RFC2475*, December 1998.
- [2] K. Thomson, G. Miller, R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics", *IEEE Network*, November/December 1997, pp. 10-23.
- [3] I., Yeom, A. L. N. Reddy, "Modeling TCP Behavior in a Differentiated Services Network", *IEEE/ACM Transactions on Networking*, Vol. 9, No. 1, February 2001, pp. 23-33.
- [4] S. Sahu, P. Nain, D. Towsley, C. Diot, V. Firoiu, "On Achievable Service Differentiation with Token Bucket Marking for TCP", *Proceedings of ACM SIGMETRICS 2000*, Santa Clara, August 2000, pp. 372-376.
- [5] I. Yeom, A. L. N. Reddy, "Realizing Throughput Guarantees in a Differentiated Services Network", *Proceedings of ICMCS*, June 1999, pp. 108-117.
- [6] W. Feng, D. D. Kandlur, D. Saha, K. G. Shin, "Adaptive Packet Marking for Providing Differentiated Services in the Internet", *Proceedings of International Conference on Network Protocols (ICNP'98)*, October 1998, pp. 1340-1349.
- [7] B. Nandy, J. Ethridge, A. Lakas, A. Chapman, "Aggregate Flow Control: Improving Assurances for Differentiated Services Network", *Proceedings of IEEE Infocom 2001*, Alaska, USA, April 2001.
- [8] S. Floyd, K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", *IEEE/ACM Transactions on Networking*, May 3, 1999.
- [8] N. Seddigh, B. Nandy, P. Piedad, "Bandwidth Assurance Issues for TCP Flows in a Differentiated Services Network", *Proceedings of Global Internet Symposium, GLOBECOM99*, Rio De Janeiro, December 1999.
- [10] H. Ritter, K. Wehrle, L. Wolf, "Improving the Performance of TCP on Guaranteed Bandwidth Connections", *Proceedings of KiVS 2001*, Hamburg, Germany, February 2001.
- [11] P. Piedad, N. Seddigh, B. Nandy, "The Dynamics of TCP and UDP Interaction in IP-QoS Differentiated Services Networks", *Proceedings of the 3rd Canadian Conference on Broadband Research (CCBR)*, Ottawa, November 1999.
- [12] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, M. Stemm, R. H. Katz, "TCP Behavior of a Busy Internet Server: Analysis and Improvements", *Proceedings of IEEE Infocom 98*, San Francisco, March 1998.
- [13] V. Laatu, J. Harju, P. Loula, "Measurement Based Analysis of the Fairness of TCP and UDP Flows in a Differentiated Services Capable Network", *Proceedings of the Conference on Internet and Multimedia Systems and Applications (IMSA'01)*, Honolulu, Hawaii, USA, August 2001, pp. 114-119.
- [14] J. Harju, J. Laine, S. Saaristo, "Performance Measurements and Analysis of TCP Flows in a Differentiated Services WAN", *Proceedings of LCN2000*, Tampa, Florida, November 2000, pp. 296-303.
- [15] J. Laine, J. Harju, J. Karjalainen, J. Lemponen, S. Saaristo, "Real-Time Traffic Measurements in a Differentiated Services Network", *Proceedings of ICC2001*, Helsinki, Finland, June 2001, 127-131.
- [16] K. Kilkki, J. Ruutu, "Simple Integrated Media Access (SIMA) with TCP", *Proceedings of the 4th INFORMS Telecommunications Conference*, Boca Raton, Florida, USA, March 1998.
- [17] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmission, and Fast Recovery Algorithms", *Network Working Group, RFC2001*, January 1997.
- [18] K. Kilkki, J. Ruutu, SIMA homepages, <http://www-nrc.nokia.com/sima/>
- [19] K. Kilkki, *Differentiated Services for the Internet*, Macmillan Technical Publishing, 1999, pp. 245-256.
- [20] M. Loukola, K. Kilkki, J. Ruutu, "Dynamic RT/NRT PHB Group", *draft-loukola-dynamic-00.txt*, November 1998.
- [21] J. Laine, S. Saaristo, J. Lemponen, J. Harju, "Implementation and Measurements of Simple Integrated Media Access (SIMA) Network Nodes", *Proceedings of ICC2000*, New Orleans, USA, June 2000, pp. 796-800.
- [22] J. Laine, S. Saaristo, J. Harju, "Core Node Implementation for Differentiated Services", *Proceedings of 5th EUNICE Open European Summer School*, Barcelona, Spain, September, 1999, pp. 29-34.
- [23] J. Postel, "Transmission Control Protocol", *Network Working Group, RFC793*, September 1981.
- [24] I. Pratt, Pratt's Test TCP (pttcp), <http://www.cl.cam.ac.uk/Research/SRG/netos/netx>
- [25] S. Floyd, T. Henderson, U. C. Berkeley, "The NewReno Modification to TCP's Fast Recovery Algorithm", *Network Working Group, RFC2582*, April 1999.
- [26] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, "TCP Selective Acknowledgement Options", *Network Working Group, RFC2018*, October 1996.
- [27] S. Ostermann, TCP dump file analysis tool (tcptrace), <http://www.tcptrace.org>