

Evaluating Performance Among Different TCP flows in a Differentiated Services Enabled 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

virpi.laatu@tut.fi, jarmo.harju@cs.tut.fi, pekka.loula@tut.fi

Abstract

This paper examines the interaction among short-lived and long-lived TCP flows in a test network that can be operated as an ordinary best effort or as a Differentiated Services (DS) enabled network, and shows how TCP service can be improved by isolating TCP flows based on their different characteristics. The target of the measurements was to investigate the interaction among TCP flows having similar priority classification but different traffic volume. The flows are originating from a single customer network and destined to the ISP's DS enabled network. With our measurements, we have shown the benefits resulting from the separation of such TCP flows into two categories: large-volume long-lived FTP-type flows and small-volume short-lived interactive (e.g. HTTP-type) flows. For FTP flows goodput was increased and for HTTP flows decreased transfer time was obtained.

1. Introduction

The rapid transformation of the Internet into a commercial infrastructure raises the need to support service differentiation and different user pricing policies. This is reflected in the importance of SLAs between Internet Service Providers (ISPs) and their customers, which specify various levels of service guarantees that are to be met.

The Differentiated Services architecture [1] is a respectable approach for providing service differentiation in an IP networks. Service discrimination is based on the value of the DS field in IP packet. Six bits of the DS field are used as a codepoint (DSCP) to mark a packet to receive a particular forwarding treatment (Per-Hop Behavior, PHB) at each edge of the network node. PHBs are implemented in nodes by means of some buffer management and packet scheduling mechanisms. After the packets have been marked, policed and shaped according to a SLA at the boundary of the network, they are forwarded through the core network according to the per-hop behavior associated with the DS codepoint. [2]

We have focused our research on a particular DS mechanism called Simple Integrated Media

Access (SIMA) model, which is an implementation of Dynamic RT/NRT PHB Group suggested in [3]. SIMA offers simple and intuitive mechanisms for providing service differentiation and a feasible charging scheme in an IP networks. In the SIMA service, a nominal bit rate (NBR) forms the basis of charging, and defines how the network capacity is divided among different connections during overload situations.

There are many difficulties encountered in a DS capable network, such as the dynamics of different TCP based applications over differentiated services network and the interaction of TCP and UDP flows. It has been seen throughout a number of studies [e.g. 4, 5, 6, 7, 8] that it is difficult to guarantee requested throughput to the TCP flows in a such network. Most of these studies have concluded that the throughput attained by a customer is affected, not only by the marking strategies at the edge router, but by the presence of other flows in the same bottleneck link. The problem of TCP flows in a DS architecture is mainly caused by TCP's congestion control mechanisms. The TCP flow reduces its transmission rate when a packet loss occurs in the network. In such situation, UDP flows, which never reduce their transmission rate quickly takes over the bandwidth that becomes available. The reader is referred to [9] for a detailed discussion on TCP congestion control mechanisms.

Yeom and Reddy in [4] proposed simple steady-state throughput models for TCP flows in terms of reservation rate, packet drop rates and RTTs (Round Trip Time). Sahu et al. in [5] derived a simple analytical model for determining the achieved rate of a TCP flow when edge routers use token bucket marking and the core routers use active queue management for preferential packet dropping. As shown in [5], it is not feasible to achieve ideal service differentiation across a set of TCP with token bucket marking. In their study, they focused on a single TCP flow, i.e. the interference of other flows was modeled by induced losses in the flow under study at the bottleneck path. Moreover, most of these other studies deal only with homogenous traffic flows or an individual flow.

Some studies concerning the interaction among flows with different characteristics in a DS capable

network have been recently presented in [6, 8, 10, 11, 12, 13, 14, 15, 16], but most of them suffer from the same problem. They have mainly focused on achieving targeted throughputs and performance guarantees for long TCP transfers (i.e., they analysed steady-state performance of TCP flow) in the presence of competing UDP flows in the same bottleneck link. Nandy et al. in [8] proposed the intelligent traffic conditioning strategies to alleviate the unfairness due to different RTTs and target rates, and UDP/TCP interactions, and evaluated these strategies through simulations. However, the proposed strategies may not be scalable due to the usage of per-flow state information at each edge nodes and communication requirements between them.

A particularly interesting case concerns the interaction among short-lived and long-lived TCP connections. Due to TCP congestion control mechanisms (slow start) short-lived TCP connections (e.g. Web browsing) have much different characteristics and behavior than bulk FTP transfers. Some studies [14, 17, 18] have investigated the fairness problem between short-lived and long-lived TCP flows. As argued in [18], short-lived TCP flows, which transmit small documents, suffer from reduced throughput compared with long-lived ones, which transmit large documents. The authors in [17] and [18] proposed to employ the RIO-based and the hash-RED- based queue management policies at routers, respectively to improve fairness between short-lived and long-lived flows.

Things will become more complex, because in a differentiated services network, for scalability reasons service contracts (SLAs) will not typically be on a per end-user (per-connection agreement) basis. One common scenario will be the case where a company contracts a target rate with an ISP, i.e. the agreements cover the aggregate rate sent by the company, so that at any one time a large number of source flows originating from the company would then compete for the aggregate target rate. There has recently been done some simulation works on aggregate marking [19, 20, 21]. However, these markers are so called per-aggregation markers and do not look at fairness issues within an aggregate, which is the problem addressed in this study. Maintaining individual, per-flow state information (e.g. such as proposed in [22]) at each edge node can, of course solve unfairness problem within an aggregate. However, these schemes are not scalable or even very feasible when the number of flows entering a marker increases and is varying all the time.

There are various issues to be addressed before a reasonable level of service differentiation between TCP users can be offered, such as interaction and performance issues among differently behaving traffic flows. This interaction requires further work such as to study how short-lived and long-lived TCP

connections can be better integrated into QoS architectures. In this paper, we focused on the unfairness problem between TCP flows having similar priority classification but different traffic volume within an aggregation. The performance of our DS scheme was evaluated in term of transfer latency for Web pages and achieved goodput for FTP flows. This paper extends the results of our earlier measurements concerning TCP performance [10, 23] and gives further insight into the problems of TCP support in QoS schemes based on the DS architecture.

The remainder of the paper is organized as follows: In Section 2, we describe briefly the architecture of SIMA service model that is used in this study. In Section 3, we describe the measurement scenario, i.e. what things were measured, how, and in what kind of conditions. In Section 4, we present the performed tests and the results of these tests. Finally, Section 5 summarizes our findings.

2. The SIMA service model

Differentiated Services is one way of satisfying the diverse needs of the Internet users by providing QoS in the Internet. We have focused our research on a particular DS mechanism called Simple Integrated Media Access (SIMA) model. As with all DS architectures, the building blocks of this service consist of two main parts: access nodes (boundary nodes) and core network 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, 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. The MBR calculation is based on

the principle of exponentially weighted moving average and it uses a different weighting factor for real time and non-real time traffic. Once the MBR has been measured (see [24] for more details), the DP value for the packet is determined by the formula (1).

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

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

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 [24].

In the core network nodes the DSCP field is used to determine how packets are treated. The treatment in these nodes includes 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 [25].

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

The goal of the measurements was to investigate the interaction among different TCP flows originating from a single customer network. We create a single bottleneck link in the network and examine how congestion on this link affects the performance of the FTP flows and HTTP flows in terms of transfer latency for HTTP flows and achieved goodput for FTP flows.

3.2. Performance metrics

In a DS capable network service guarantees are often specified using aggregate measures, e.g. a fixed bandwidth guarantee for traffic originating from a single company, but the performance measures of real interest are usually the level of performance that individual users and applications experience (e.g. the per-flow goodput).

In these experiments, the performance measures taken into account are the average goodput obtained by a FTP flow at the receiver. Since the average goodput is meaningless for short-lived TCP connections, we used per-transfer latency instead and the total number of connections completed as our metrics for short-lived TCP connections. The goodput measures the rate of successfully transmitted packets, whereas the transfer latency measures the time interval starting when the Web server sends out the first packet of a response and ending when the Web client receives the last packet of the response. In addition, we measured the number of retransmitted packets for both traffic types. In these experiments, we consider not only the throughput achieved by individual sources, but also the throughput achieved by aggregated sources.

3.3. Test network

Figure 1 shows the test network topology used in this study. 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. 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 (FTP Client and Web Client), 2 server nodes (FTP Server and Web Server), and an Adtech AX/4000 acting as source for background traffic. Border1 and Border2 nodes had both the access and core node functionality, whereas Core1 and Core2 nodes were built only with the core node functions. The bottleneck link is between Core1 and Core2 nodes. The bandwidth of all the links except the link between an Adtech AX/4000 and Core1 node were set to 10 Mbps. The link between Adtech AX/4000 and Core1 node was built with the 100Base-TX Ethernet technology in the half-duplex mode. All the network nodes were Pentium II PCs 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. We generated several TCP flows destined to the other end of the network through a 10 Mbps bottleneck link. The traffic flows traversing through to the core network during the measurements are shown in Figure 1.

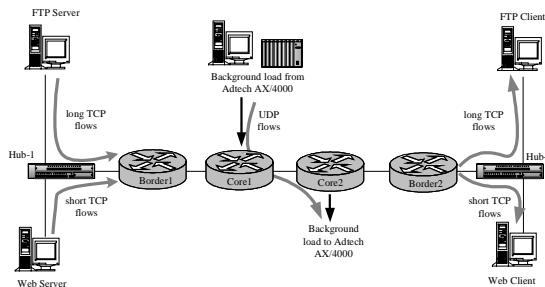


Figure 1. The test network topology

3.4. Configurations

The following parameters of the core nodes concerning the RT and NRT queues and classes were used respectively: $[9800, 0.01; 20000, 0.001]^1$. The core node parameters were not modified during the measurements, whereas access node configurations were varied in each different test case by varying the access rules (i.e. NBR values and RT/NRT class selection). The edge nodes in the test network classify packets based on source or destination port number (port 80 for WWW).

3.5. Test methodology

Two different scenarios were created in order to evaluate the interaction among short-lived and long-lived TCP flows. For analysing FTP and HTTP traffic behaviour, we used TCP sender/receiver pttcp utility [26]. 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 utility enables a single client to initiate multiple parallel HTTP requests to a Web server. We also used tcpdump to capture packet flows, and analysed tcpdump traces with the help of tcptrace utility [27]. The TCP version used in our measurements was NewReno with SACK option. The MTU (Maximum Transfer Unit) was set to 1448 bytes. For modelling HTTP traffic, we adopted the HTTP traffic model according to a latest study on HTTP traffic measurement [28]. The distribution of interpage and interobject time, number of objects within a Web page and size of an object are given in Table I. The sessions, once running are assumed to contain an “infinite” number of Web pages.

Table I. The attributes of HTTP session and corresponding distributions

<i>Number of sessions</i>	5
<i>Number of connections per session</i>	1
<i>Number of Web pages per session</i>	Infinite
<i>Time interval between Web pages [sec.]</i>	Exponential mean: 10

¹ [queue size in bytes, the weighting factor for non-real time and real time traffic]

<i>Number of Web objects per page</i>	Exponential mean: 3
<i>Web object size [kilobytes]</i>	Pareto mean: 12 shape: 1.2
<i>Time interval between Web objects [sec.]</i>	Exponential mean: 0.01

The background traffic was used to create bottleneck on the link between Core1 and Core2 nodes. 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 (Figure 2) and consisted of several separate UDP flows with different DP values. The background traffic was marked using the TOS option of Adtech AX/4000. 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.

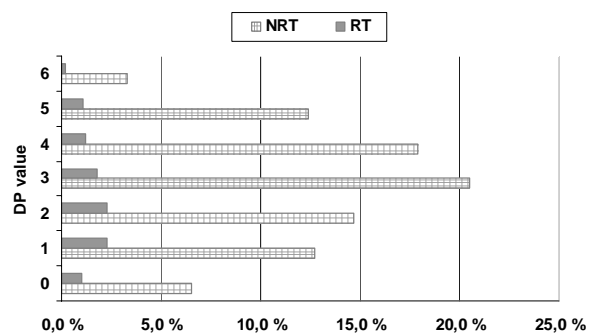


Figure 2. Background traffic DP-level distribution

Each of the tests was carried out five times with the same parameters to gain confidence in our results. All tests lasted for 40 seconds. The presented results are averages of five individual measurements (except the throughput graphs).

4. Test cases and results

In this section we present the measurements that we performed on our test network, and the obtained results. Two different scenarios were created in order to zoom out on the details of the performance of TCP flows before and after isolating short-lived and long-lived TCP flows into separate service classes. The scenarios were composed of several short-lived TCP flows and long-lived TCP flows with UDP/CBR background flows. During the measurements, the FTP workload consisted of 15 FTP flows, while the number of HTTP flows generated by 5 sessions varied according to the network congestion level. In all the cases each single FTP file transfers was 1.8 Megabytes in length. We also present the results that we performed without QoS enabled.

4.1. Baseline measurements

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 baseline measurements were made both in over-provisioned and under-provisioned network conditions in order to study the effect that the background load level had on the performance of FTP and HTTP flows. We can see from Figure 3 that the FTP flows shared the bandwidth in a fair way in both over-provisioned and under-provisioned case, as one would expect due to the basic properties of TCP.

In Table II we have shown the summary of the baseline measurements in the under-provisioned case (i.e. background load on) for reference purposes with the QoS enabled cases (in Section 4.2 and 4.3). Figure 4 depicts the average transfer latency of all of the existing HTTP flows with respect to their traffic volume. We observed that in the under-provisioned case, the largest flow sizes have suffered relatively more than smaller flows.

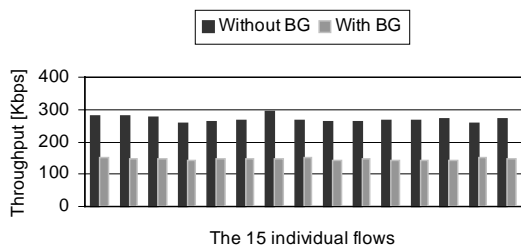


Figure 3. Achieved throughputs by individual FTP flows with and without background load

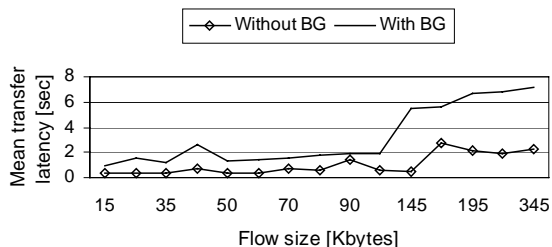


Figure 4. Mean transfer latency without QoS enabled

Table II. Summary of the results of the baseline measurements

Traffic type	Avg. aggregate goodput [Kbps]	Unique data sent [Kbytes]	Rexmt data sent [Kbytes]	Number of conn.
FTP	157.7	27956.3	5214.8	15
HTTP	19.2	1553.2	64.5	37

Table II lists the results of the baseline measurements. The achieved average goodput for both aggregates is indicated in the second column. The third column tells the unique data bytes sent by aggregate. The fourth column in Table II is the total number of bytes retransmitted (Rexmt) for an FTP

aggregate and a HTTP aggregate during the measurement. The last column tells the total number of completed transfers.

4.2. Short-lived and long-lived TCP flows competing for the aggregate NBR rate

After the baseline measurements, we enabled the DS functionality of the core nodes and the access nodes, and made measurements in the case where the network was under-provisioned.

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. A large number of source flows originating from the company would then compete for the aggregate target rate.

In this section, we consider the case where both FTP flows and HTTP flows are mixed together to share the same aggregated contract rate (target NBR= 600 kbps) within a single (non-real time) service class. In this case the marking behavior differs from the marking of individual flows. In the marking of individual flows, the NBR rate for the 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. Figure 5 shows the throughputs achieved by individual FTP flows within an aggregation. The results of this experiment show that all the FTP flows share the available bandwidth in a relatively fair manner. We also observed improvement in average goodput obtained by the FTP aggregate when compared to the case where QoS was disabled and network was under-provisioned (Avg. aggregate goodput of 217.8 kbps as compared to 157.7 kbps). Moreover, the number of retransmitted bytes is also reduced in the FTP case.

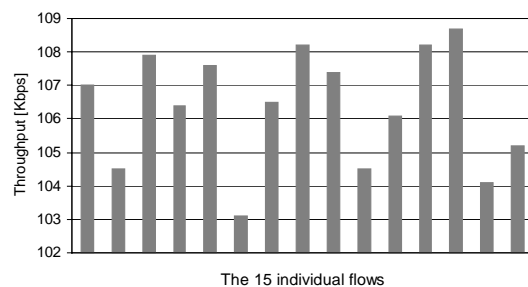


Figure 5. Individual throughputs for aggregated FTP flows

However, the improvement in average aggregate goodput seems to have come at the expense of HTTP connections' performance, in terms of the mean transfer latency and the number of completed connections. Figure 6 shows the results of the mean transfer times for this experiment computed across the 5 HTTP sessions as a function of flow

size. Table III summarizes the results of the measurements.

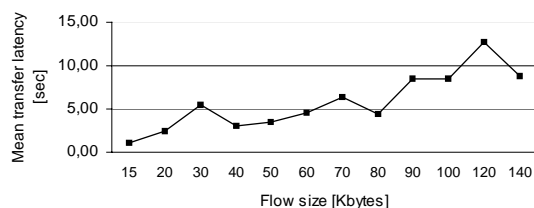


Figure 6. Mean transfer latency for aggregated HTTP flows

Table III. Summary of results for aggregated flows

Traffic type	Avg. aggregate goodput [Kbps]	Unique data sent by agg. [Kbytes]	Rexmt data sent by agg. [Kbytes]	Number of connections
FTP	217.8	27963.8	3747.9	15
HTTP	15.4	1488.4	52.3	33

4.3. Short-lived and long-lived TCP flows mapped to separate classes

In this section, we study the effects achieved by separating the short-lived and long-lived TCP flows into two different service classes. In this measurement scenario, there are two sets of aggregated TCP flows, each having the same target NBR (300 kbps), but all FTP flows are set to non-real time service class and all HTTP flows are set to real time service class. The results show (Figure 7) that the FTP flows are again pretty close to equally sharing the available bandwidth. In addition, we observed the improvements in throughput obtained by an individual FTP flow to be more than 50 % compared to the case where the flows are competing for the same aggregate NBR. Moreover, we observed that isolating the flows reduces the mean transfer latency for most of the HTTP flows. Considering the isolated case versus the aggregated case, the improvements in mean transfer latency range from 40 % for Web page (size 80 Kbytes) to 80 % for Web page (size 120 Kbytes). Most Web pages have improvements ranging from 50 % to 77 %. The detailed results of the measurement are shown in Table IV.

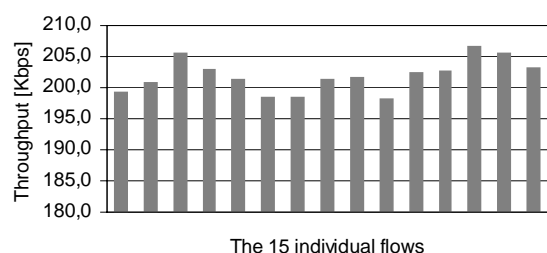


Figure 7. Individual throughputs for isolated FTP flows

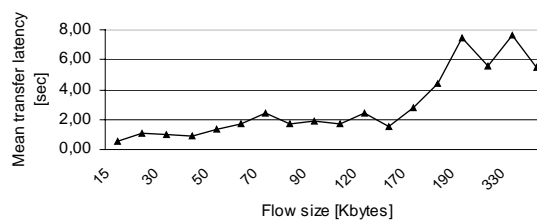


Figure 8. Mean transfer latency for isolated HTTP flows

Table IV. Summary of results for isolated short-lived and long-lived flows

Traffic type	Avg. aggregate goodput [Kbps]	Unique data sent by agg. [Kbytes]	Rexmt data sent by agg. [Kbytes]	Number of connections.
FTP	194.2	27973.0	3419.2	15
HTTP	33.0	2307.1	68.1	45

5. Conclusions

In this study, we examined the effects of a particular DS mechanism has on the behavior of TCP flows which can be considered equally important (i.e. their priority is the same) but having different traffic volumes. The behavior was mainly evaluated by observing the goodput of FTP flows and transfer latencies for HTTP flows. We isolated those flows into two classes based on their traffic volume, by allocating an interactive class for the short-lived HTTP flows and a non-real time class for the long-lived FTP flows.

With our measurements, we have shown that our QoS scheme results in better transfer latencies for short-lived flows, without significantly affecting long-lived ones. This also shows that, by properly classifying traffic based on the characteristics and requirements of applications, user-perceived performance can be significantly improved, in terms of achieved goodput by an individual flow and the transfer latencies of Web pages. The measurement results show that the transfer latency of short-lived flows is reduced for more than 45 %. This seems to be due to the fact that separating short-lived and long-lived flows into different service classes decreases the amount of retransmissions. 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.

The devised method works in such DS environments where differentiation can be made with respect to interactive vs. non-real time characteristics of the flows. The SIMA model and its implementation used in the measurements is just one example of such DS environments. By proper configuration similar behavior could be achieved e.g. in the Assured Forwarding model specified in the

RFC 2597. Future work will include experiments in that environment.

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. Nichols, S. Blake, F. Baker, D. Black, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, *Network Working Group, RFC2474*, December 1998.
- [3] K. Kilkki, *Differentiated Services for the Internet*, Macmillan Technical Publishing, 1999, pp. 245-256.
- [4] 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.
- [5] 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.
- [6] N. Seddigh, B. Nandy, P. Piedad, "Bandwidth Assurance Issues for TCP Flows in a Differentiated Services Network", *Proceedings of IEEE Globecom'99*, Rio De Janeiro, December 1999.
- [7] H. Ritter, K. Wehrle, L. Wolf, "Improving the Performance of TCP on Guaranteed Bandwidth Connections", *Proceedings of KiVS 2001*, Hamburg, Germany, February 2001.
- [8] B. Nandy, N. Seddigh, P. Piedad, J. Ethridge, "Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks", *Proceedings of IFIP High Performance Networking*, Paris, June 2000.
- [9] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmission, and Fast Recovery Algorithms", *Network Working Group, RFC2001*, January 1997.
- [10] 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.
- [11] I. B. H. de Alves, J. F. de Rezende, L. M. de Moraes, "Evaluating Fairness in Aggregated Traffic Marking", *Proceedings of the IEEE Globecom'00*, November 2000.
- [12] S. H. Lee, S. J. Seok, S. J. Lee, C. H. Kang, "Study of TCP and UDP Flows in a Differentiated Services Network Using Two Markers System",
- [13] I. Andrikopoulos, L. Wood, G. Pavlov, "A Fair Traffic Conditioner for the Assured Service in a Differentiated Services Internet", *Proceedings of IEEE International Conference on Communications (ICC2000)*, New Orleans, LA, June 2000, pp. 806-810.
- [14] W. Noureddine, F. Tobagi, "Improving the Performance of Interactive TCP Applications Using Service Differentiation" *Proceedings of IEEE Infocom 2002*, New York, USA, June 2002.
- [15] V. Sander, I. Foster, A. Roy, L. Winkler, "A Differentiated Services Implementation for High-Performance TCP Flows", *Computer Networks: The International Journal of Computer and Telecommunications Networking*, December 2000, Vol. 34, Issue 6.
- [16] A. Sang, H. Zhu, S. Li, "Weighted Fairness Guarantee for Scalable Differentiated Services (DiffServ)", *Proceedings of IEEE International Conference on Communications*, Helsinki, Finland, June 2001.
- [17] L. Guo, I. Matta, "The War Between Mice and Elephants",
- [18] K. Tokuda, G. Hasegawa, M. Murata, "Analysis and Improvement of the Fairness between Long-lived and Short-lived TCP Connections", *Proceedings of Seventh International Workshop on Protocols for High-Speed Networks (PfHSN2002)*, Germany, April 2002.
- [19] 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.
- [20] H. Su, M. Atiquzzaman, "ItswTCM: A New Aggregate Marker to Improve Fairness in DiffServ", *Proceedings of IEEE Globecom'01*, San Antonio, Texas, November 2001.
- [21] Y. Chait, C. V. Hollot, V. Misra, D. Towsley, H. Zhang, J. C. S. Lui, "Providing Throughput Differentiation for TCP Flows Using Adaptive Two-Color Marking and Two-Level AQM", *Proceedings of IEEE Infocom'02*, New York, USA, June 2002.
- [22] I. Yeom, A. L. N. Reddy, "Adaptive Marking for Aggregated Flows", *Proceedings of IEEE Globecom'01*, San Antonio, Texas, November 2001.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] I. Pratt, Pratt's Test TCP (pttcp), <http://www.cl.cam.ac.uk/Research/SRG/netos/netx>
- [27] S. Ostermann, TCP dump file analysis tool (tcptrace), <http://www.tcptrace.org>
- [28] F. D. Smith, F. H. Campos, K. Jeffay, D. Ott, "What TCP/IP Protocol Headers can tell us about the Web", *Proceedings of the ACM SIGMETRICS*, Cambridge, MA, June 2001, pp. 245-256.