6^{th} *International Conference on Applied Informatics*
Eger, Hungary, January 27–31, 2004.

Differentiated Services simulations using traditional scheduling algorithms*

Miklós Lengyel[a], **János Sztrik**[b]

[a]Department of Informatics Systems and Networks, University of Debrecen
e-mail: mlengyel@inf.unideb.hu

[b]Department of Informatics Systems and Networks, University of Debrecen
e-mail: jsztrik@inf.unideb.hu

Abstract

The aim of our investigation is to consider a simple dumbbell Diffserv network topology in which performance comparison (in terms of throughput, delay and queue length) is made between the traditional traffic scheduling algorithms: Priority (PRI), Weighted Round Robin (WRR) and Weighted Interleaved Round Robin (WIRR) schedulers. Random Early Detection (RED) is used as active queue management algorithm. An earlier version of this paper can be found in [6]. The aim of the present investigations is to show how the above mentioned performance measures vary if we change the packet size.

In the core of the network there is a bottleneck link and the consideration is performed on that node. All of our traffic generators are Constant Bit Rate (CBR), the transport protocol is User Datagram Protocol (UDP). We used Network Simulator (NS, version 2) for our simulation experiments.

Key Words and Phrases: AF PHB, differentiated service, EF PHB, queue management, scheduler, UDP

1. Introduction

The history of the Internet has been of continuous growth in the number of hosts, the number and variety of applications, and the capacity of the network infrastructure. A scalable architecture for service differentiation must be able to accommodate this continuous growth. The Differentiated Services (Diffserv or DS) architecture [1] provides a more flexible, scalable architecture than the existing models of service differentiation. The specification of Diffserv architecture appeared in 1998, but the current research is still expanding it. The architecture

*Supported by Hungarian Scientific Research Found OTKA T0-34280/2000.

is based on a simple model where traffic entering a network is classified and possibly conditioned at the boundaries of the network, and assigned to different DS codepoints. Within the core of the network, packets are forwarded according to the per-hop behaviour associated with the DS codepoint. A per-hop behaviour (PHB) is a description of the externally observable forwarding behaviour of a DS node applied to packets with a particular DS codepoint. PHBs are implemented in nodes by means of some buffer management and packet scheduling mechanisms. Two different PHBs were developed: the Assured Forwarding (AF) PHB [2] and the Expedited Forwarding (EF) PHB [3]. The AF PHB group provides delivery of IP packets in four independently forwarded AF classes. Within each AF class, an IP packet can be assigned to one of three different levels of drop precedence. EF PHB is intended to provide low delay, low jitter and low loss services by ensuring that the EF packets are served at a certain configured rate.

The Diffserv architecture achieves scalability by implementing complex classification and conditioning functions only at network boundary nodes, and by applying per-hop behaviours to aggregates of traffic which have been appropriately marked using the DS field in the IPv4 or IPv6 headers. This architecture only provides service differentiation in one direction of traffic flow and is therefore asymmetric.

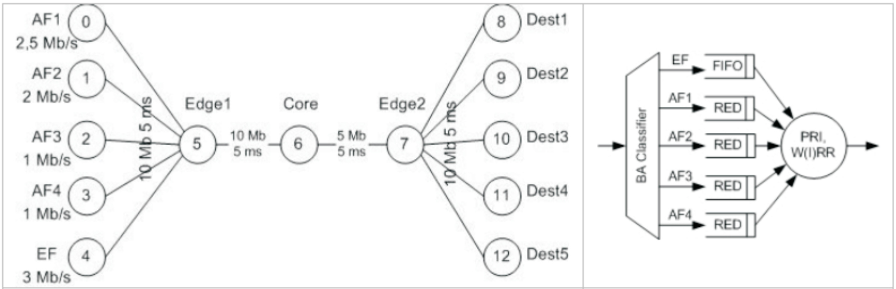
While many studies have addressed issues on the Diffserv architecture (e.g., dropper, marker, classifier and shaper), there have been few attempts to analytically understand a flow's behavior in a Diffserv network.

In this paper we enhance our earlier paper [6], in which a performance comparison was made between the traditional traffic scheduling algorithms (PRI, WRR, WIRR) in a dumbbell Diffserv topology using packets with size of 1000 bytes. We consider how the performance measures vary if we use packets with the size of 500 bytes in the same environment.

Section 2 presents the results of the simulations. Conclusions are drawn in Section 3.

2. Simulation results

Simulations were performed using Network Simulator [7] (NS, version 2.1b9a), which was developed at the University of California. NS is an event-driven network simulator, which is implemented in C++ and uses OTcl (Object Tool Command Language) as the command and configuration interface. We considered the simple dumbbell topology shown in Fig. 1/a. All links have the same fixed delay of 5 ms. The consideration is performed on the Core node where there is the bottleneck link. The structure of the output interface of the core node is shown in Fig. 1/b.



a.) Simulated network topology b.) Simulated output interface

Figure 1. Simulation scenario

We ran each simulation for 60 seconds. The traffic generators are CBRs over UDP. The nodes 0-3 generate AF1-AF4 traffic, while 4. node generates EF. The i-th node sends packets to i+8-th node, i = 0,1,2,3,4. An AF class is implemented in the nodes as a RED physical queue with three virtual queue, while EF as a droptail (FIFO) queue. Figure 2,3 show the Sent packet ratio and the Received packet ratio, which was set up such that they are equal in case of the three schedulers. We make throughput, delay and queue length comparison between the scheduling algorithms PRI, WRR and WIRR and we confront it with our earlier results [6].

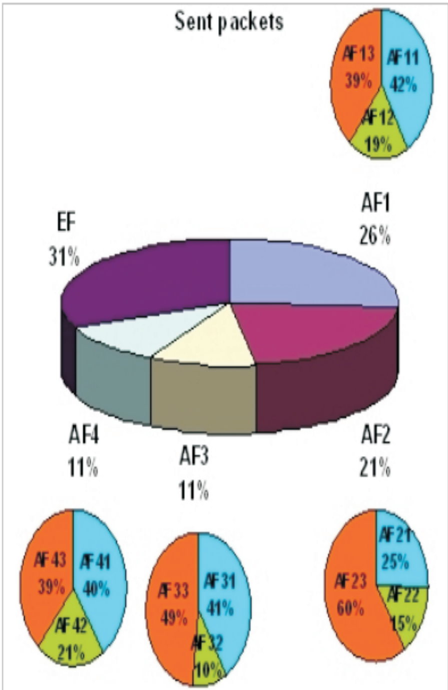


Figure 2. Sent packet ratio

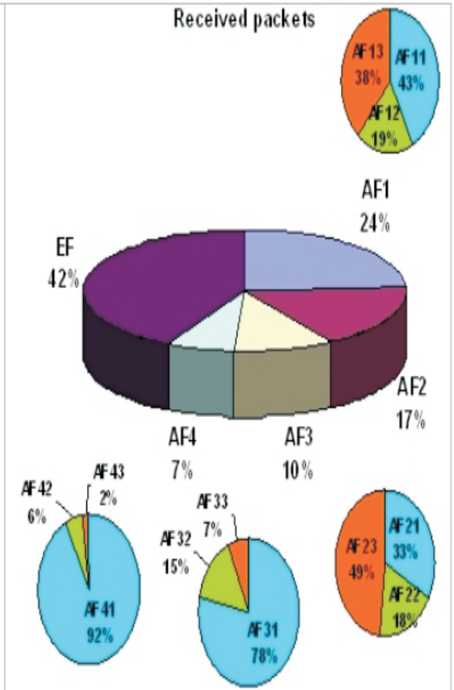


Figure 3. Received packet ratio

The whole simulation scenario is the same like in our previous investigation [6], the only difference is that we use packets with size of 500 byte instead of 1000 byte. This means that the link buffer must have a capacity (in packets) which is equal with the buffer length (in packets) in original simulation multiplied by two. Since the RED algorithm in our simulation works in "packet mode"(not in "byte mode") we have to change the adequate RED parameters in the simulation script regarding to new packet size.

First of all we consider the queue length. The next three figures show how the queue length varies in case of the three schedulers. Currently the maximum queue length can be 100 packets, because we use packet size which is equal with the packet length in the original simulation divided by two.

All the observations (see [6] for details) which were taken in case of the original simulation are relevant to this simulation also. We can see that in case of 500 byte packet size simulation the queue length variation density is twice time bigger than in case of 1000 byte packet size simulation. This is because in case of 500 byte packet size simulation the number of generated packets (by source nodes) is twice time bigger than the number of generated packets in case of 1000 byte packet size simulation.

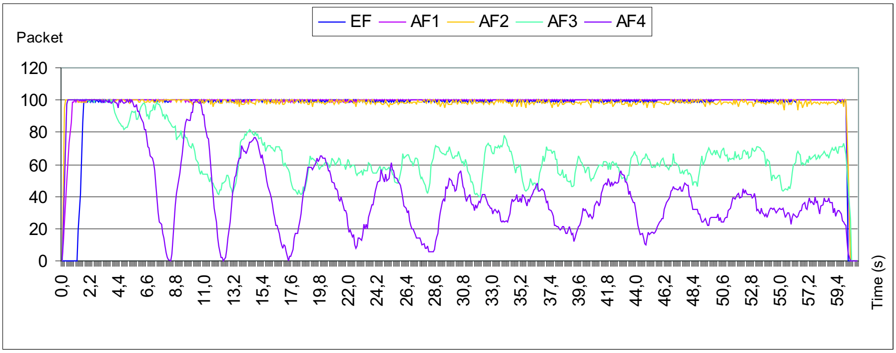


Figure 4. Queue length in case of PRI scheduler

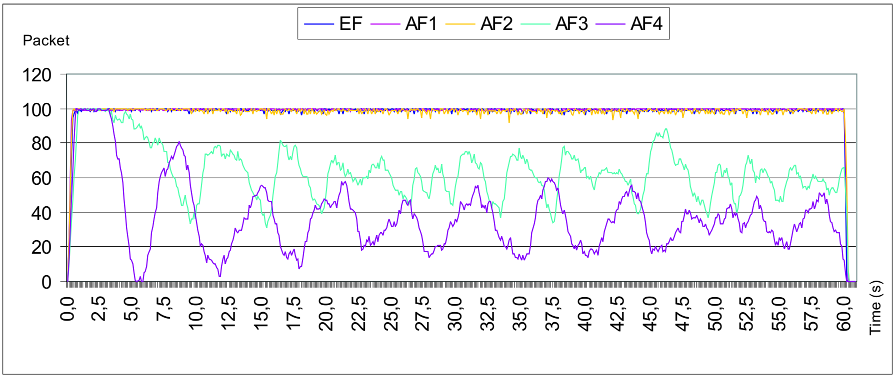


Figure 5. Queue length in case of WIRR scheduler

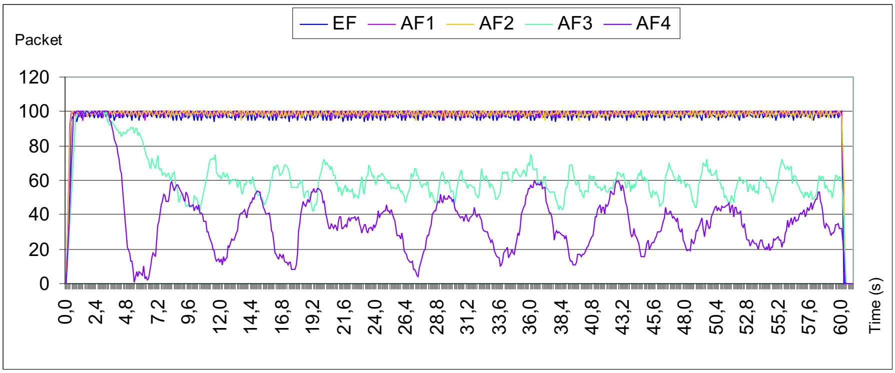
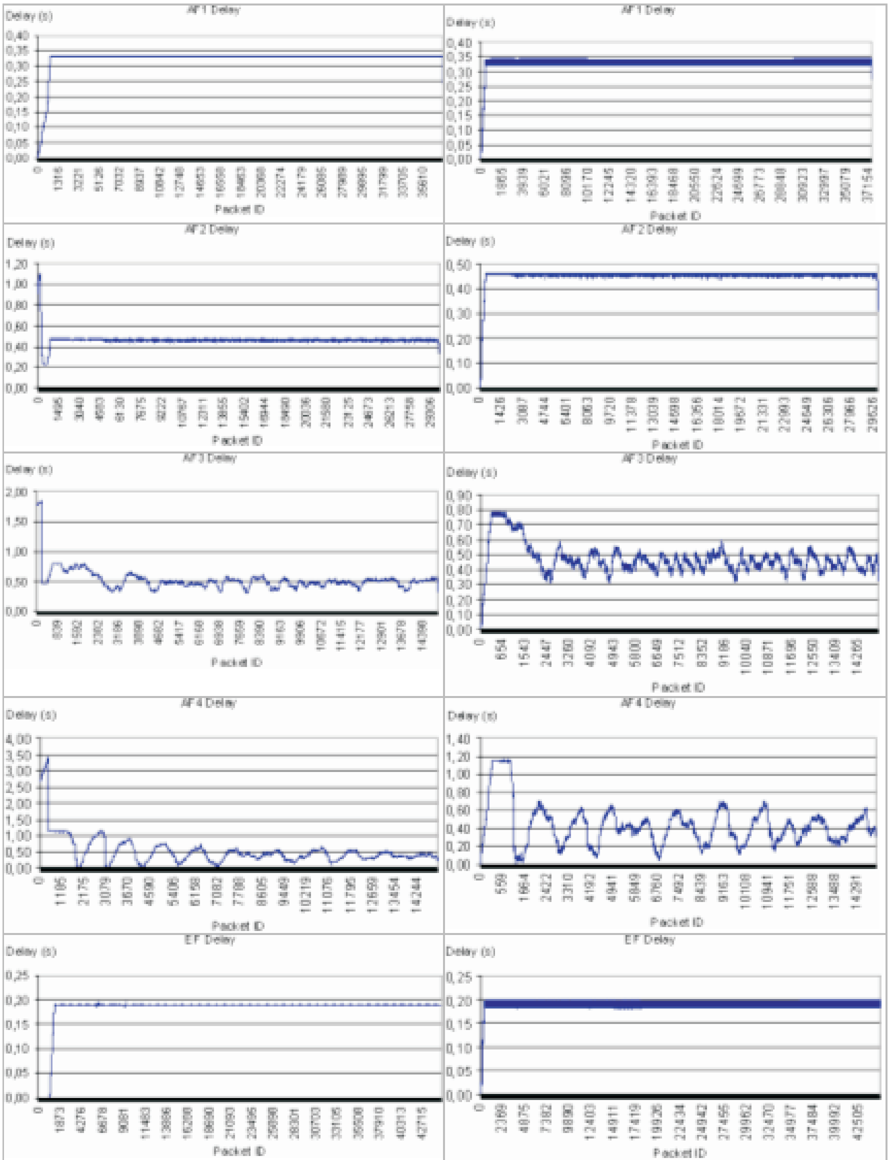


Figure 6. Queue length in case of WRR scheduler

Figure 7 shows the delay variation of packets. The delay varies exactly as the queue size varies, conform to the well known Little-formula ($Q = \lambda$*W). This means that the delay variation density is also twice time bigger than in the case of original (1000 byte packet size) simulation [6].



a.) Delay in case of PRI scheduler | b.) Delay in case of WRR scheduler

Figure 7. Delay of packets

Because of page number limitations we only present the delay of packets in case of PRI and WRR scheduler, but the WIRR scheduler also holds the above criteria.

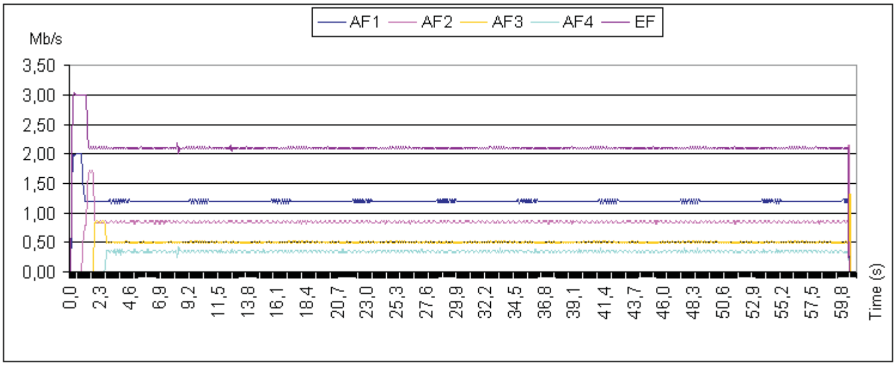


Figure 8. Throughput in case of PRI scheduler

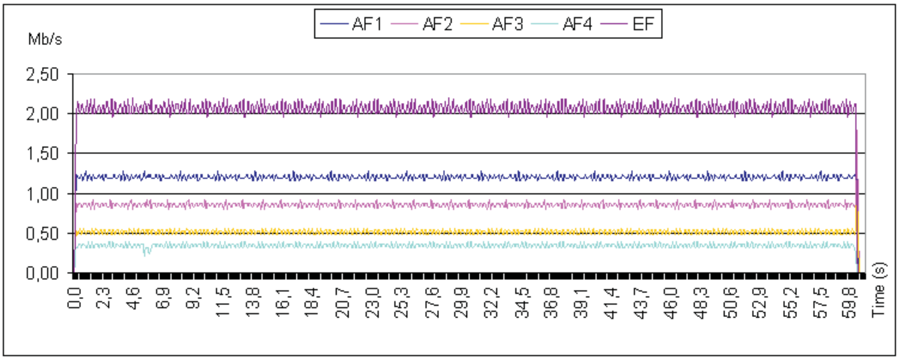


Figure 9. Throughput in case of WIRR scheduler

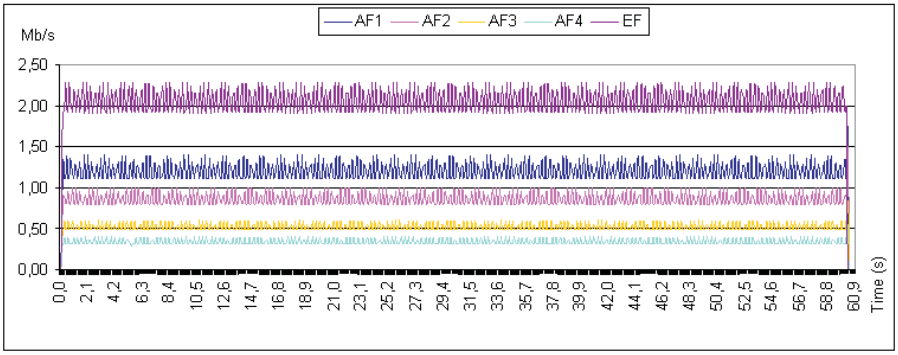


Figure 10. Throughput in case of WRR scheduler

Figure 8, 9, 10 show the throughput variation, which has the same characteristic like queue size (or delay), namely that the variation density is bigger (twice time) than in the case of 1000 byte packet size simulation.

Similar to the original simulation, the average realized throughput per class is the same for all schedulers, but the deviation (jitter) from the mean is the smallest in case of PRI scheduler and the greatest in case of WRR (while WIRR is between them).

The next figures show a comparison between the original (1000 byte) and the actual (500 byte) simulation in terms of arithmetic mean delay per class in case of the three schedulers.

It can be observed that the packet size changing does not have significant effect to the average delay.

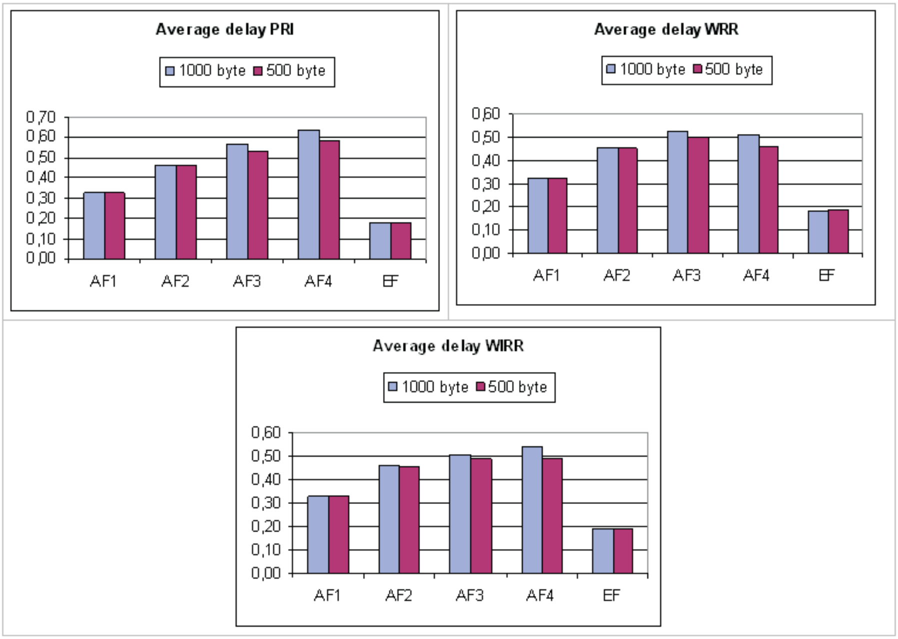


Figure 11. Arithmetic mean delays

3. Conclusions

A performance comparison was made between the traditional traffic scheduling algorithms in a simple dumbbell Diffserv topology. We enhanced our earlier paper [6] and we investigated how the performance measures vary if we change the packet size.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", IETF RFC 2475, December 1998.

[2] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", IETF RFC 2597, June 1999.

[3] B. Davie, A. Charny, J. Bennett, K. Benson, J. Boudec, W. Courtney, S. Davari, "An Expedited Forwarding PHB", IETF RFC 3246, March 2002.

[4] B. Braden, D. Clark, B. Davie, S. Floyd, V. Jacobson, J. Wroclawski, L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", IETF RFC 2309, April 1998.

[5] S. Floyd, V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, pp. 397-413.

[6] M. Lengyel, J. Sztrik, "Simulation of Differentiated Services in Network Simulator", Annales Universitatis Scientiarium Budapestinensis de Rolando Eötvös Nominatae, Sectio Computatorica, (accepted for publication)

[7] K. Fall, K. Varadhan, "The ns Manual", available via http://www.isi.edu/nsnam/ns/ns-documentation.html, April 2002.

[8] R. Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modelling", John Wiley and Sons, Inc., New York, 1991.

Postal addresses

Miklós Lengyel
Depart. of Informatics Systems
and Networks
University of Debrecen
4010 Debrecen, P.O. Box 10
Hungary

János Sztrik
Depart. of Informatics Systems
and Networks
University of Debrecen,
4010 Debrecen, P.O. Box 10
Hungary