

Advanced Modelling of DiffServ Technology

Jiří Hošek, Lukáš Růčka, Karol Molnár

Abstract—The Differentiated Services are currently one of the most used mechanisms for the quality of service assurance in IP networks. The basic principle of this network architecture is to separate the network traffic into several classes with the specific priority. This paper is focused on the advanced modelling of this technology in the simulation environment OPNET Modeler. Basic DiffServ parameters such as the Per Hop Behavior index, queuing mechanisms and queue management algorithms are simulated and analyzed. The simulation results and the final evaluation are presented at the end of this paper.

Index Terms—Assured Forwarding, DiffServ, DSCP, Expedited Forwarding, Fair Queuing, OPNET Modeler, Priority Queuing, Random Early Detection, Round Robin

I. INTRODUCTION

THE IP (Internet Protocol) networks in their fundamental form are not able to provide different handling for different network traffic types, which is important especially for real-time network services like VoIP (Voice over Internet Protocol) or videoconference. Therefore there is need to implement specific network mechanisms that are able to distinguish data flows according to the source applications and then classify them to separate traffic classes. It is possible to determine the optimized traffic policy in dependence on the statistic data about created traffic queues. This policy will guarantee assurance of the required QoS (Quality of Service) for applications with the different way of network management.

There are different types and concepts of the architectures which implement the quality of service standards and classify data units in accordance with their priority. The DiffServ (Differentiated Services) technology is one of the most often used QoS architectures implemented on the network layer of the reference model [1]. The DiffServ mechanism is based on the basic idea when the data flow incoming into the network is classified into several classes with specific priority level. The network resources are then reserved for these traffic classes depending on the priority [1], [2].

The DiffServ mechanism is composed from several functional blocks and every block has many control

parameters. The suitable configuration of these parameters is important for the right function and efficiency of this QoS technology. Therefore the main attention in the DiffServ mechanism should be paid to proper policy and network management settings.

II. DIFFERENTIATED SERVICES

The Differentiated Services (DiffServ, DS) are nowadays the most extended solution for quality of service assurance in IP networks. Because of its relative simplicity and stateless packet processing, DiffServ offers high scalability and can efficiently operate in large data networks too. This is the reason why DiffServ is practically the only world-wide extended QoS support technology in use [3].

The architecture of DiffServ is based on a simple model where incoming data flows are classified and sorted into traffic classes according to predefined rules, as type and port number of a transport layer protocol and IP address. Every packet entering the network is marked by a specific mark DSCP (DiffServ Code Point) which determines the treatment of the given packet, i.e. it defines the assigned traffic class. This packet marking is executed only at the network bounds. During packet forwarding over the core network routers only read the mark and manage the corresponding treatment. The mark is stored in IP packet header [4], [5].

A. DiffServ Domain

The DiffServ domain is a continuous part of an IP network with the same packet handling rules and administration authority. The uniform administration ensures the requirements validity evaluation, queue based flow assignment and packet marking. Fig. 1 shows the DiffServ domain and its two key elements [3], [4].

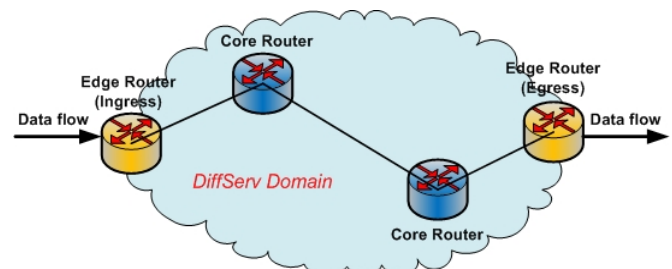


Fig. 1. DiffServ Domain

The edge (boundary) router interconnects the DiffServ domain with other DS domain or non-DiffServ-capable part of a network. This router classifies measures and marks the incoming traffic and subsequently sent them into the DS

J. Hošek¹, L. Růčka² and K. Molnár³ are with the Department of Telecommunications of the Faculty of Electrical Engineering and Communication at the Brno University of Technology, Purkyňova 118, 612 00 Brno, Czech Republic (phone: +420 541149106; e-mail: hosek@feec.vutbr.cz¹, rucka.lukas@phd.feec.vutbr.cz², molnar@feec.vutbr.cz³).

domain. The edge router can work as an ingress or egress node, depending on its location. If there are two DiffServ domains connected by one router then this router works as the egress router for the first domain and as the ingress router for the second DS domain respectively. It also has to perform all packet processing functions for both directions [4].

The second element – core router ensures the packet forwarding and treatment according to the DSCP stored in packet headers. This treatment is realized by systems of queues and packet schedulers [4].

B. DiffServ Architecture

The reference model of the DiffServ technology is defined in the specification RFC 2475 [4]. This model is composed of several blocks that are the basement of the whole mechanism. The DiffServ reference model can be divided into two main parts. The Classification is the first part. This block is responsible for packet classification and marking by DSCP. The second part is called Conditioning. It performs traffic conditioning based on the network parameters and DSCP values [6].

The first block of the classification part is called Classifier. This block identifies the incoming packets and subsequently classifies into several classes according to predefined rules. There are two basic types of classifiers – Behaviour Aggregate (BA) and MultiField (MF). The more detailed description of these classifiers can be found in [6].

The Marker is the second block of the classification part. This block assigns a DSCP mark to every IP packet. The DSCP mark defines the packet treatment within the network – so called Per Hop Behavior (PHB) index. There are three PHB types – Expedited Forwarding (EF), Assured Forwarding (AF) and the default PHB called Best Effort (BE) [2], [7], [8].

The traffic conditioning part is composed of four blocks – Meter, Remaker, Shaper and Dropper. The functions of these blocks are described in [4].

III. DIFFSERV CONTROL MECHANISMS

The DiffServ architecture is based on several control mechanisms. The most important of these mechanisms that were modelled and analyzed in the OPNET Modeler simulation environment are described in following text.

A. Per Hop Behavior

As mentioned above the data flow is adjusted at the DiffServ boundary router and assigned to specific type of aggregation. Each type of aggregation is identified by the DSCP. Within the network packets are passed on according to the PHB type associated with specific DSCP. More DSCP tags can be mapped to the one PHB type. The PHB type represents externally observable forwarding scheme used in DS nodes (e.g. routers) applied to a particular DS aggregation schemes. The PHB is a technical description internal to a network and it is not observable for an end user. The PHB does not specify the internal implementation mechanisms of the DS node. The classification based on the BA is one of the

simplest DiffServ classifier which uses only the DSCP mark [5].

As mentioned above there are three PHB types. First, the Best Effort type does not provide any guarantees that data are delivered or that a user is given a guaranteed level of the QoS. The default Best Effort packet forwarding in traditional IP networks correspond to the BE PHB type. Packets belonging to this PHB can be forwarded in any manner without any restrictions. It means that the network will deliver as many of these packets as possible and as soon as possible. For the BE PHB a value of 000000 DSCP is recommended [4], [5]. The BE is suitable service for data transfers which do not request any level of the QoS.

The second PHB type is called Expedited Forwarding (EF) PHB. The EF PHB type is defined as a forwarding treatment for a DiffServ aggregate [7]-[9]. The outgoing transfer rate of aggregated packets from any DS node must equal or exceed a configured rate. The EF PHB ensures that every DS node in the DiffServ domain sends packets included in the EF PHB average rate of at least equal to the configured rate. For EF PHB is necessary to carry out inspection on the edge of a DS domain according to the values agreed by a service level agreement (SLA).

There are many ways how to implement the EF PHB. The EF PHB requires a guaranteed amount of outgoing resources of the DS node (e.g. bandwidth, priority). This means that packets with the EF PHB are treated with an allocated amount of outgoing resources of the DS node. That will guarantee forwarding behavior with the minimum delay, minimum jitter and minimum packet loss. The EF PHB can be used for circuit emulation, private leased line emulation and real-time services (e.g., voice, video). These services are not tolerant to high values of loss, delay and jitter. It is recommended to use the EF only in limited way. The recommended DSCP value for EF PHB is 101110 [5], [9].

The last is the Assured Forwarding (AF) PHB type. AF provides assurance that the IP packets are forwarded with a high probability. This high probability is provided that the aggregate traffic does not exceed the subscribed information rate (profile). If aggregate traffic exceeds the subscribed profile than the excess traffic is delivered with lower probability as the traffic that is within the profile [8], [10]. For the AF PHB is objective to deliver the packet reliably. It means that the packet loss is important and both delay and jitter are unimportant.

The AF PHB provides four different classes of forwarding behaviors. For each AF class specific amount of outgoing resources (e.g., bandwidth, buffer space) is allocated in each DS node. On the edge of the DiffServ domain the packet flows are marked and assigned to a given AF class. Within each AF class there are three possible values of drop precedence. This drop precedence determines the relative importance of the packet. Packets with the lower drop precedence are dropped before packets with a higher drop precedence which the DS node tries being lost. In a DiffServ node, the level of the forwarding assurance depends on the amount of available resources that are allocated for a particular AF class. It is important to mention that DS node

must not aggregate two or more AF classes together. The AF PHB is appropriate for non real time network services. The detailed description of recommended code points for AF classes can be found in [8].

B. Queuing Mechanisms

A queue scheduling allows managing access to a fixed amount of the output bandwidth by selecting of the next packet transmitted into the output queue. The queuing is the optimal point to introduce QoS differentiation mechanisms. Each queue scheduling mechanism attempts to find the right balance between complexity, control, and fairness. Queuing elements determine the packet ordering, possibly storing them temporarily or discarding them.

A queue scheduling is applied on output ports of routers. For each output port, the packets are classified and queued. This chapter is focused on following types of queue scheduling methods:

- Round-Robin (RR),
- Priority Queuing (PQ),
- Fair Queuing (FQ),
- Weighted Round Robin (WRR),
- Weighted Fair Queuing (WFQ).

The further described methods provide basic principles. The actual implementation of the queue scheduling in a DiffServ router is specified by manufacturer.

The Round-Robin (RR) is simple scheduling algorithm which assigns part of whole quantum (e.g. time slices) to each queue in equal portions. This is happening in circular order. All queues are handled without a priority.

The basic principle of the next methods – the Priority Queuing (PQ) is shown in Fig. 2 [2].

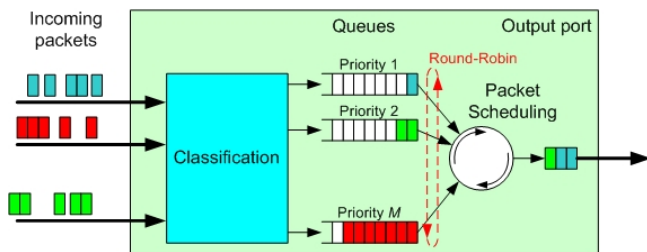


Fig. 2. Priority Queuing

Several distinct queues with assigned relative priority levels are created in this algorithm. Packets are scheduled from a particular priority queue in First-In-First-Out (FIFO) order only when all queues of a higher priority are empty [2]. The result is that the highest priority traffic receives a minimal delay. But all other priority levels may experience resource starvation if the highest precedence traffic queue remains occupied. PQ is designed to provide a relatively simple method of supporting differentiated service classes. The apparent disadvantage of this type the queue scheduling algorithm is the possibility of complete suppression traffic with a lower priority.

The Fair Queuing (FQ) is another general principle of the traffic class distinction. It is also referred to as per-flow or flow-based queuing. In this system, incoming packets are sorted into M queues as is shown in Fig. 3 [2].

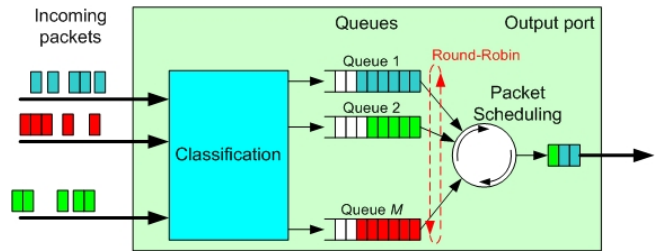


Fig. 3. Fair Queuing

Each queue is allocated $1/M$ of the output port bandwidth. The packet sending management cyclically serves individual queues that have packets to send (Round-Robin algorithm). During each cycle, at most one packet from any queue is transmitted. The main advantage of the FQ mechanism is its simplicity. The main drawbacks of FQ are follows. First, each queue is served with the same regularity. This means that the system is unable to allocate available width relative to data flows that have a different bandwidth requirement. Second, during the queue processing a packet is transmitted, regardless of the packet size. That has impact on the actual bandwidth distribution among the queues.

The limitations of the FQ and PQ models are removed by Weighted Round Robin (WRR) queuing model. WRR is able to ensure the allocation of different amounts of the bandwidth for a various service classes according to their bandwidth requirements. Each of the queues is serviced in a Round-Robin order (see Fig. 4) [2].

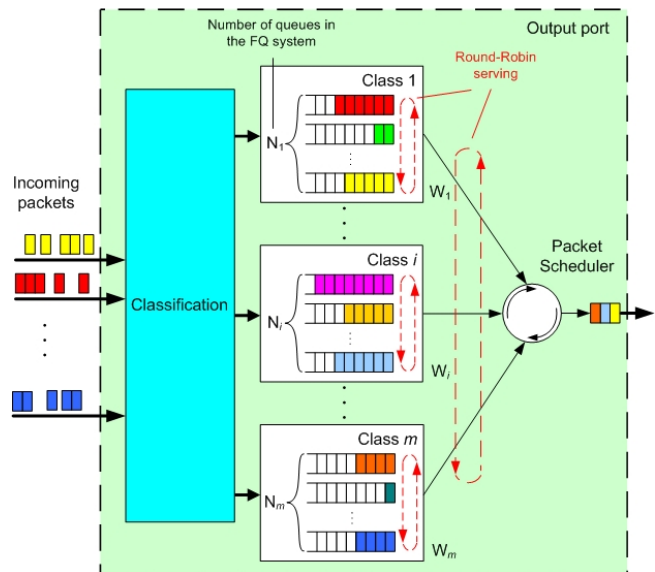


Fig. 4. Weighted Round Robin

This mechanism is sometimes identified as the class-based queuing (CBQ) or custom queuing. The WRR queuing model allows each queue to assign a different percentage of the

available bandwidth of the output port. The principle of the WRR queuing model function is as follows. Bandwidth of the outgoing port is divided into the m classes in proportion to appropriate weights of classes. Weights are determined by the bandwidth requirements of the classes and should add up to 100%. The allocation of bandwidth between each class can provide the percentage distribution of the time that is used by manipulating of this class. Individual flows are scheduled by the FQ within each class [2]. The WRR queuing model did not solve the problem of the influence of the packet length to the bandwidth.

The WRR queuing model involves two layers of round robin scheduling. The first level represents the choice of classes from 1 to m . The second level is used for the selection of the queue within that class.

The Weighted Fair Queuing (WFQ) is queue scheduling designed to address limitations of the FQ model. In the WFQ the incoming packets are sorted into N queues. Based on the queue weights a different percentage of the output port bandwidth is assigned. The weights add up to 100%. WFQ also eliminates the problem of the influence of the variable length of the packet to the bandwidth. The WFQ attempts to approximate a theoretical model referred to as the weighted bit-by-bit Round-Robin scheduler [2]. Actually WFQ sends out the packets from the queues in time which is based on the calculated order of packets according to the theoretical model (see Fig. 5) [2]. The support of the fair bandwidth allocation adds the significant computational complexity of the queue scheduling algorithm.

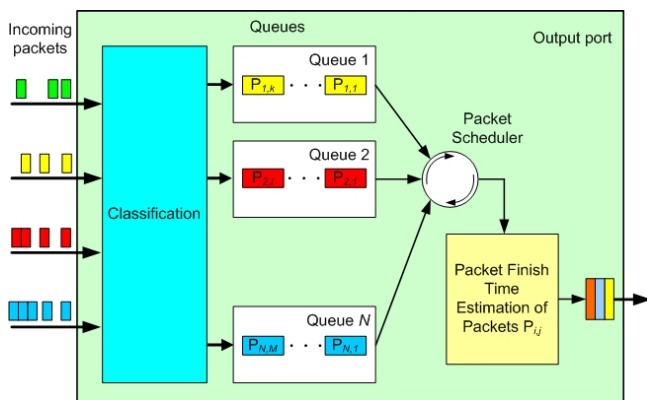


Fig. 5. Weighted Fair Queuing

C. Queue Management Mechanisms

Packets are usually assigned to the queue because of a lack of resources (e.g. bandwidth). The traditional technique for queue management accepts packets until the maximum length of queue is reached (congestion). Then subsequent incoming packets are dropped (rejected). This state takes until the packet queue number is decreased because of a packet has been transmitted. This technique is known as "tail drop" [11]. However, the tail drop method causes the phenomenon referred to as the "TCP global synchronization". To prevent the global TCP synchronization the Active Queue

Management (AQM) was introduced. AQM seeks to anticipate an onset of congestion and take action to prevent or mitigate the effect of the congestion. Two types of AQM methods are discussed in this section.

The Random Early Detection (RED) is a QAM algorithm for queue management. As mentioned the tail drop method drops all packets when a queue threshold is exceeded (congestion). This is very unfair. The RED algorithm drops arriving packets probabilistically. The probability of dropping increases as the estimated average queue size grows. The probability of a packet being dropped from a particular connection is proportional to its bandwidth usage rather than the number of packets it transmitted.

The RED algorithm itself consists of two main parts: a congestion prediction algorithm and a packet drop profile as you can see in Fig. 6 [2].

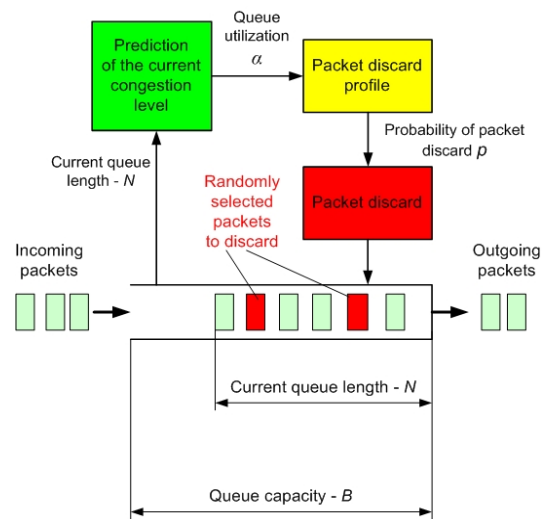


Fig. 6. Random Early Detection Operation

The traffic in the queue behaves over time as the congestion prediction module. This module also detects any build up of congestion. Based on the queue size B and the instantaneous queue length N the module calculates average queue utilization α . On the basis of the average queue utilization the module estimates the congestion state.

The packet drop profile (see Fig. 7) determines whether the packet will be marked. Whether the packet is marked depends on the average queue utilization α and a probability p . The marked packets are in fact dropped. The average queue utilization α is compared with two thresholds α_{MIN} and α_{MAX} . The minimum threshold α_{MIN} sets the minimum queue utilization before packet marking will begin. The maximum threshold α_{MAX} is a soft maximum that the algorithm will attempt to stay under. When the average queue utilization is greater than the maximum threshold, every incoming packet is marked (the queue begins to operate in a "virtual" tail drop mode).

When the average queue utilization is between the minimum and the maximum threshold, each arriving packet is marked with a probability p . The packet drop probability p is

determined by a pre-defined function. The probability function, in general, can be in any form (such as a linear function shown in Fig. 7). The probability that a packet is marked is roughly proportional to how the corresponding connection shares the bandwidth.

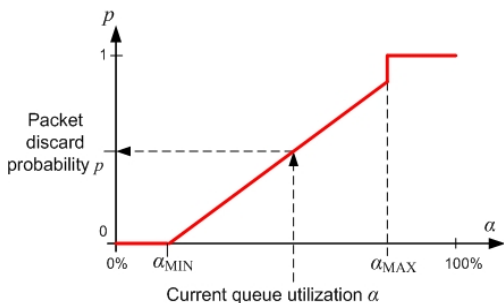


Fig. 7. Packet Drop Profile

The Weighted Random Early Detection (WRED) uses the same parameters as RED, but it has the ability to perform RED on traffic classes individually [2]. WRED is an extension to RED that allows assigning multiple (different) RED drop profiles to different types of traffic (class) in the same queue. In addition, it allows defining multiple drop profiles within a single queue. In this way, can be achieved a different packet dropping probability within a single queue (different priorities of service).

IV. MODELLING OF DIFFSERV TECHNOLOGY

The modelling of DiffServ technology was realized in the simulation environment OPNET Modeler (OM) which is well known software tool enabling design, simulation and analysis of different type of network technologies, architectures and protocols [12].

In OPNET Modeler we created a DiffServ domain model which is shown in Fig. 8. This DS domain is composed of two edge routers and two core routers. The network model also contains three configuration elements and some application servers and client stations that generate specific network traffic.

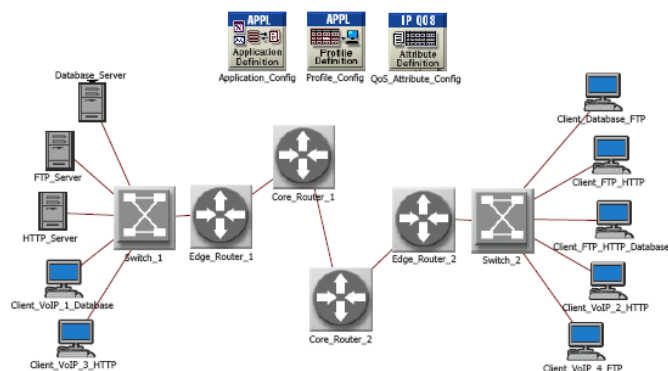


Fig. 8. Simulation Model of DiffServ Domain

There are two basic components – Application Config and Profile Config which are used for selection and configuration of network applications in OM. The QoS Attribute Config

component is used for the global configuration of the quality of service. The network application VoIP (Voice over Internet Protocol), FTP (File Transfer Protocol), HTTP (HyperText Transfer Protocol) and database access were set up in the simulation model.

The DiffServ technology was configured on the edge and core routers. The configuration of edge routers was more complicated because these routers execute the classification and also packet marking. By contrast, core routers perform only the packet sorting into specific queues according to their priority defined by the DSCP mark. The packets incoming into the edge DS domain node were classified based on their application type and then sorted into classes defined by ACL (Access Control List) rules. In our model we defined four classes with corresponding priorities and DSCP marks. The VoIP traffic was put into the class with the highest priority and on the other hand the FTP traffic was placed into the lowest priority class. On the basis of this classification the DSCP mark was set up for every packet. The DSCP determined the packet priority which is used for sorting of packets into specific queue in DS core routers.

V. SIMULATION RESULTS

The created network model was used for simulating and analyzing of the DiffServ control mechanisms and their settings impact on the final QoS parameters. We focused on three basic DS control blocks – PHB types, queuing mechanisms and queue management mechanisms. Further, the most important results of our research are described.

First, we tested different type of queuing mechanisms and their impact on the quality of network applications. The real-time applications such as the VoIP are delay-sensitive so the low delay and jitter are very important for good quality of this application type. We chose three types of queuing mechanisms – Priority Queuing, Weighted Round Robin and Weighted Fair Queuing. The comparison of the queuing delay for the highest priority queue that was used for the VoIP traffic is shown in Fig. 9. It can be seen that all three mechanisms ensure comparable and sufficient queuing delay for VoIP traffic.

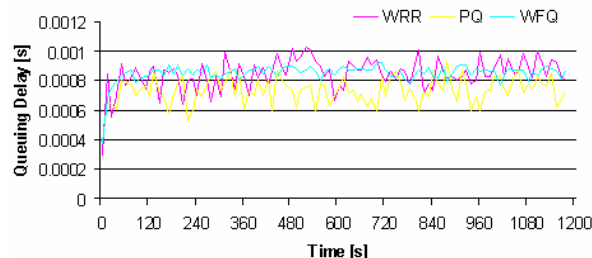


Fig. 9. Queuing Delay for VoIP traffic class

The difference between these mechanisms is in the way how they manipulate with the network traffic in the lowest priority queue. The Fig. 10 shows dropped traffic for the lowest priority queue which was used for FTP data in our model. It is clear from the graph that the WFQ mechanism

drops the largest amount of FTP traffic. It means that this mechanism inhibits the low priority traffic mostly.

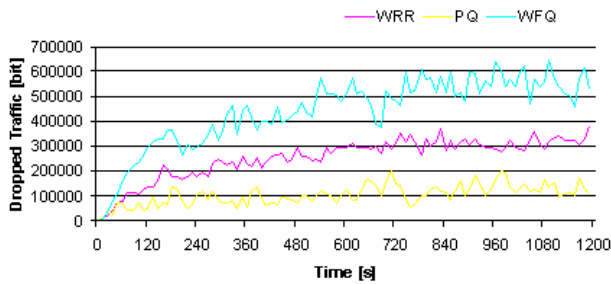


Fig. 10. Dropped Traffic for FTP class

The second DiffServ control mechanism which we analyzed was the PHB type. We compared Assured Forwarding (AF) PHB and Expedited Forwarding (EF) PHB. The Fig. 11 shows the dropped traffic rate for the lowest priority queue. The amount of dropped bits is higher with the EF PHB type that is why the EF reserves most of the network resources for the high priority traffic (e.g. VoIP) and for the low priority traffic remains small amount of network resources contrary of the AF PHB type.

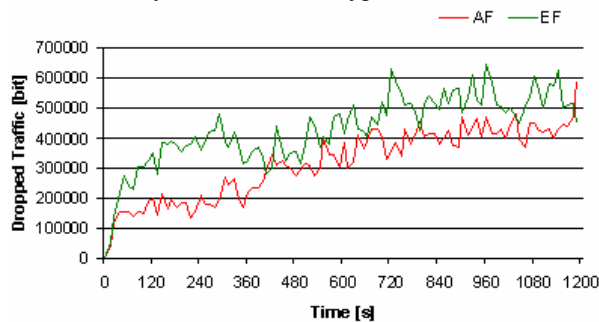


Fig. 11. Dropped Traffic for FTP class

The queue management mechanism was the last tested parameter. There are two most used mechanisms – Random Early Detection (RED) and Weighted Random Early Detection (WRED). We discovered that in an average loaded network the queue management mechanism selection has not significant impact on the final quality of network applications. The Fig. 12 shows the average queue size in packets calculated by WRED algorithm for all defined queues. This average queue size convergence depends on the exponential weight factor.

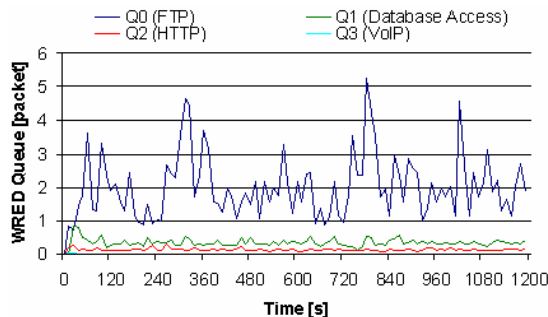


Fig. 12. Average WRED queue size

The graph confirms hypothesis that the packet queuing and dropping is applied mostly for low priority data (e.g. FTP).

VI. CONCLUSION

The DiffServ technology is currently the most used method for quality of service assurance in IP networks. The DS architecture is composed from several blocks with many control mechanisms. The main goal of this paper was to analyze impact of these control mechanisms on the result quality of network services. The network administrators should pay attention to suitable settings of DiffServ parameters. Based on our research we suggest the following combination of DS parameters – WFQ or PQ queuing mechanism, AF PHB type and WRED queue management mechanism. These preferences are the most suitable for average loaded networks.

ACKNOWLEDGMENT

This paper has been supported by the Grant Agency of the Czech Republic (Grant No. GA102/09/1130) and the Ministry of Education of the Czech Republic (Project No. MSM0021630513).

REFERENCES

- [1] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*, San Francisco, CA: Morgan Kaufmann, 2001, ISBN: 1-55860-608-4.
- [2] K. I. Park, *QoS In Packet Networks*, Boston, 2005, ISBN: 0-387-23389-X.
- [3] Molnar, K., Vrba, V., DiffServ-Based User-Manageable Quality of Service Control System, *Proceedings of the Seventh International Conference on Networking*, 2008, pp. 485-490.
- [4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, *An Architecture for Differentiated Service*, RFC2475, 1998.
- [5] Blake, S., Black, D., Nichols, K., Baker, F., Cisco Systems, Torrent Networking Technologies, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, RFC2474, 1998.
- [6] Barker F., Chan, K., Smith, A., *Management Information Base for the Differentiated Services Architecture*, RFC3289, 2002.
- [7] B. Davie, A. Charny, J. C. R. Bennett, K. Benson, J. Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, *An Expedited Forwarding PHB (Per-Hop Behavior)*, RFC3246, 2002.
- [8] J. Heinanen, F. Barker, W. Weiss, J. Wroclawski, *Assured Forwarding PHB Group*, RFC2597, 1999.
- [9] V. Jacobson, K. Nichols, Cisco Systems, K. Poduri, Bay Networks, *An Expedited Forwarding PHB*, RFC2598, 1999.
- [10] D. Grossman, Motorola Inc, *New Terminology and Clarifications for DiffServ*, RFC3260, 2002.
- [11] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang, *Recommendations on Queue Management and Congestion Avoidance in the Internet*, RFC2309, 1998.
- [12] Opnet Technologies, *OPNET Modeler Product Documentation Release 14.0*, 2007.