

QoS MODELLING IN OPNET MODELER

Karol Molnár

Department of Telecommunications, Faculty of Electrical Engineering and Communication,
Brno University of Technology, Purkyňova 118, 612 00 Brno, Czech Republic,

email: molnar@feec.vutbr.cz,

Quality of Service (QoS) support represents one of the most important features of modern multimedia networks. Since QoS support technologies are still objects of intensive development and fine tuning, simulations and modelling are highly required in this field. This paper describes how to build a simulation model in the industry's leading Opnet Modeler environment.

Keywords: QoS, Opnet Modeler, Differentiated Services

1. INTRODUCTION

Several QoS support technologies have been developed during the last ten-fifteen years. Currently, the technology of Differentiated Services (DiffServ) is the most widely implemented solution. Although, there are quite standardized configuration tools to deploy DiffServ in corporate and campus networks, the derivation of configuration parameters for these commands is still empirical and the parameters are quite often obtained as a result of intensive experiments. On other hand, such experiments in real networks are quite risky and alternative solutions are welcome. A sophisticated simulation can be an alternative.

Opnet Modeler is an up-to-date simulation environment capable of simulating the behaviour of network processes (communication protocols), network components (servers, workstations, switches, routers, etc.), applications (http, ftp, email, VoIP, database, etc.) and their extended combinations (subnetworks, fixed and wireless networks, etc.). It also supports Differentiated Services with the configuration process quite close to the configuration of real systems. In the paper this process is described together with several simulation results.

2. DIFFERENTIATED SERVICES

DiffServ is a QoS support technology, where network resources are reserved for classes of services. According to the concept of Differentiated Services, traffic evaluation and marking are realized at the network edges. The incoming traffic is sorted into these traffic classes immediately at the edges of the DiffServ domain and the corresponding DiffServ Code Point (DSCP) is assigned to the packets. In general, packet classification is based on TCP/UDP port numbers and network addresses, which are usually able to identify exact network services, so that different classes can be defined for real-time, business and best-effort data. QoS support in the core network is then based on the DSCP assigned to the packet, which is much faster than analyzing the whole packet header in all core-routers. The DiffServ technology is stateless and it is relatively easy to manage.

By now there are two standardized packet treatment technologies, so called per-hop behaviours (PHB) for DiffServ: Assured Forwarding (AF) and Expedited Forwarding (EF). Assured Forwarding [2] introduces prioritization and controlled resource sharing between traffic classes. It defines four traffic classes with three different packet drop-priorities in each of them. Expedited Forwarding [3] ensures absolute guarantees on bandwidth, transmission delay and jitter, which is required by modern multimedia applications. More information about Differentiated services can be found, for example, in [1], [4] and [6].

3. DIFFERENTIATED SERVICES IN OPNET MODELER

As mentioned earlier, Opnet Modeler offers sophisticated support to simulate the behaviour of Differentiated Services. This chapter shows the configuration workflow used to build a DiffServ simulation model.

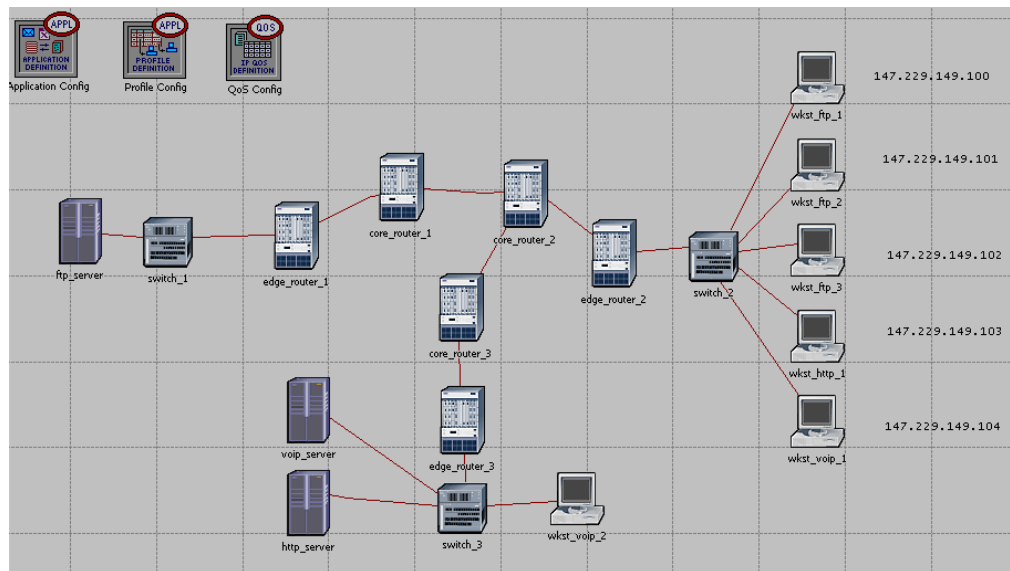


Fig. 1 Diffserv domain simulation model

In the first step the network components (servers, workstations, switches and routers) are placed into the workspace and interconnected with the required type of communication links (Ethernet and DS1 in our case).

Then the Application_Config and the Profile_Config objects are brought onto the scene. These are independent simulation objects. Application_Config is used to configure the parameters of simulated applications. An example of an FTP application is shown in Fig. 2. There are several predefined application profiles like Low Load or High Load, but it is also possible to use custom-defined configurations.

Attribute	Value
? name	Application Config
? model	Application Config
? Application Definitions	(...)
? rows	1
row 0	
Name	FTP High Load
Description	(...)
Custom	Off
Database	Off
Email	Off
Ftp	High Load
Http	Off
Print	Low Load
Remote Login	Medium Load
Video Conferencing	High Load
Voice	Edk...
Voice Encoder Schemes	All Schemes

Fig. 2 Configuration of an FTP application

Timing control for the defined applications, including start time, duration, serial or simultaneous operation, etc. is configured using the Profile_Config object. These profiles are then assigned to the servers and workstations in the scene. In our case the QoS_Config object must also be added to the scene. This component is the most important one from the point of view of DiffServ configuration. An example of a simulation model is shown in Fig. 1.

The queuing system used in the simulation model is one of the parameters configured through the QoS_Config object. In our example the Weighted Fair Queuing (WFQ) system was configured and assigned to the traffic.

Next, the queuing system defined in the QoS_Profile object is assigned to the corresponding interfaces of the core routers inside the DiffServ domain.

It is also required to configure traffic measuring and classification for the edge routers. For this purpose access control lists (ACL) are first defined to classify the incoming packet streams. The ACLs used in our model are based on address patterns. In the next step the traffic classes must be defined. This can be done by configuring the attributes of the edge routers.

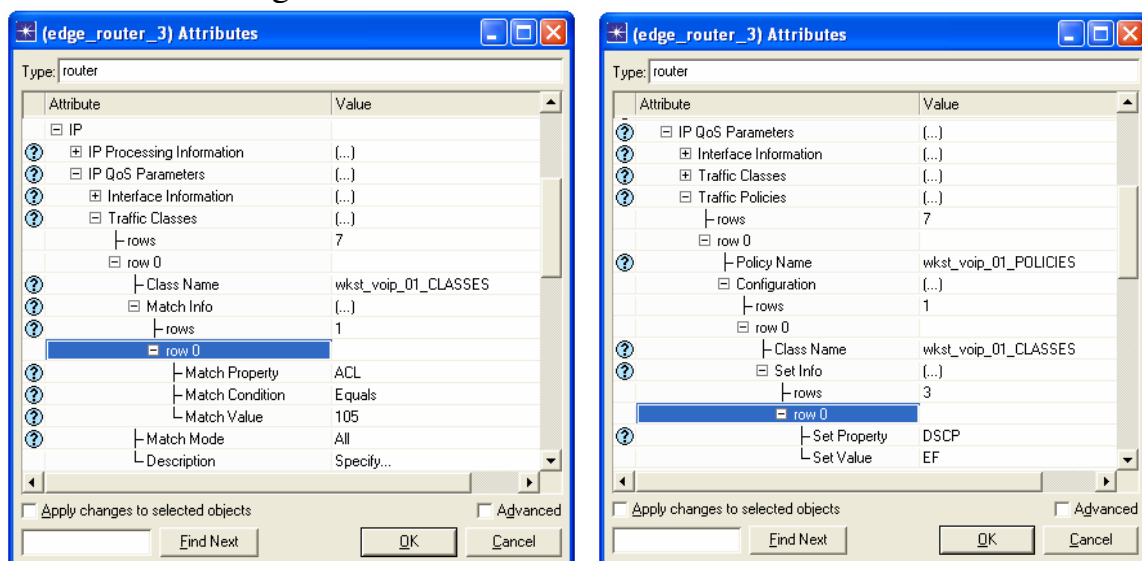


Fig. 3 a) Traffic class configuration, b) Traffic Policies configuration

Traffic classes are defined using the previously configured ACLs. If the condition in the ACL is satisfied, the corresponding ID is assigned to the packet. An example of traffic class configuration is shown in Fig. 3 a).

Furthermore, traffic policies must be configured. It is also done as a configuration of the edge routers' attributes. Traffic policies define what type of PHB is used for the given traffic class, Fig. 3 b). In our example both the Expedited Forwarding and Assured Forwarding PHBs were configured.

In the last step the queuing profiles are configured for the given traffic classes. It is done through the attributes of the QoS Config object. Two queuing profiles were used during our simulations: Priority Queuing and Weighted Fair Queuing.

Finally, the QoS configurations must be assigned to the corresponding router interfaces, as shown in Fig. 4.

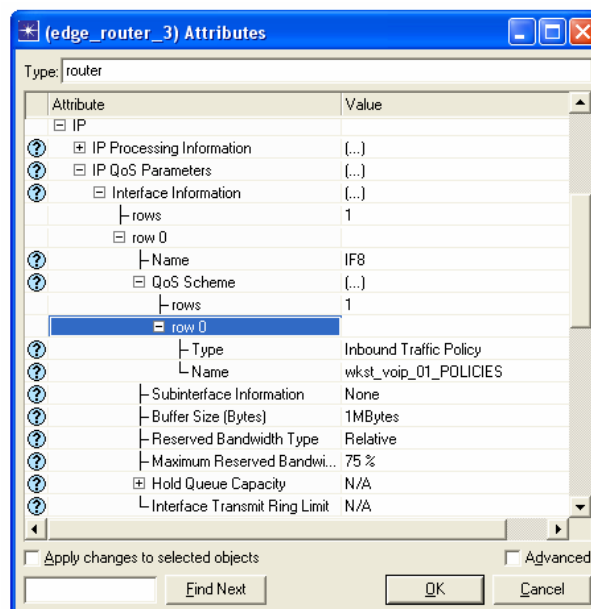


Fig. 4 Assignment of QoS configuration to the router interface

For comparison, two scenarios are created: one with pure test applications and QoS configuration and one with additional background traffic. This allows us to analyze the effect of non-controlled background traffic on the processing of the classified traffic.

After the scenes have been configured, the statistics, which are important in the evaluation of the results, are selected. These statistics can be global, considering all network components or local, considering only one selected network component.

After configuring all the parameters, the simulation can be started. During this phase an executable file is generated using a c++ compiler, which is then used to simulate the configured behaviour.

The last step contains the displaying and evaluation of the simulation results.

4. SIMULATION RESULTS

A simulation model according to Fig. 1 was built and analyzed by our team. During the analysis we focused on transmission delay and buffer usage. These results are shown in Fig. 5 and Fig. 6.

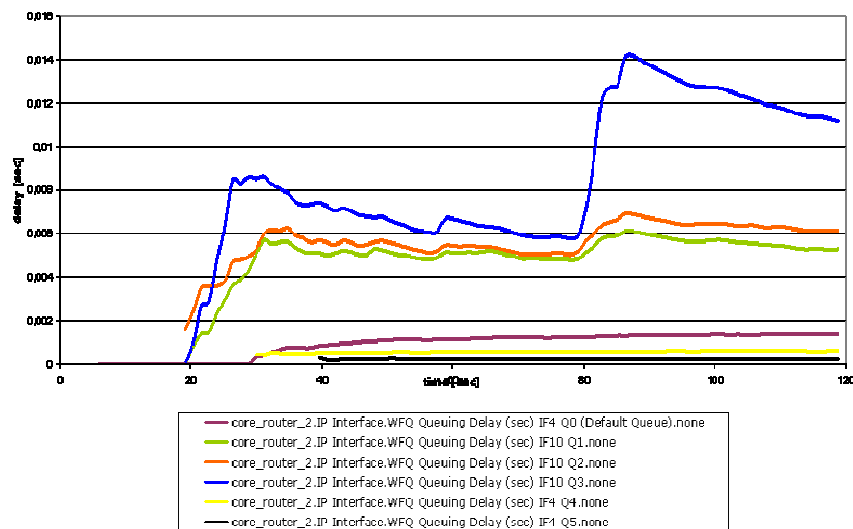


Fig. 5 Queuing delays of the queues in core router 2

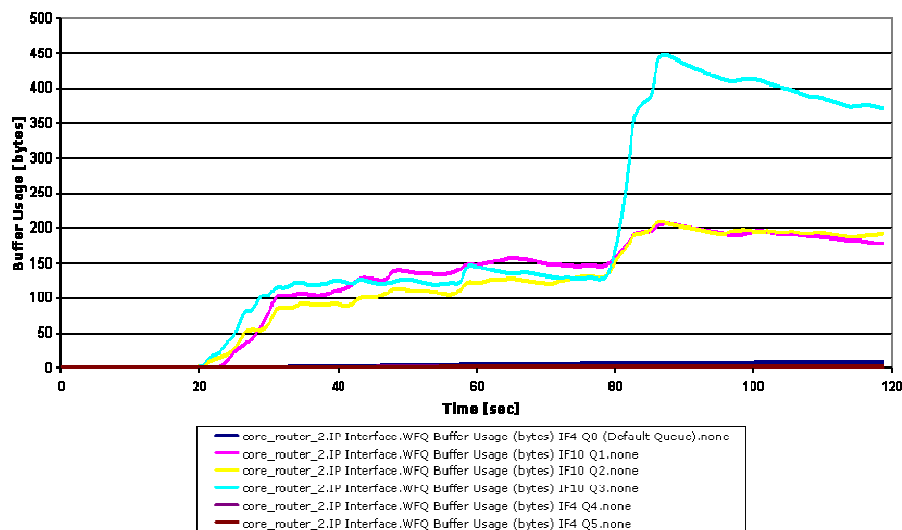


Fig. 6. Buffer usage by the queues in core router 2

Fig. 7 displays the results of traffic shaping achieved by using three packet drop priorities as defined for the AF PHB.

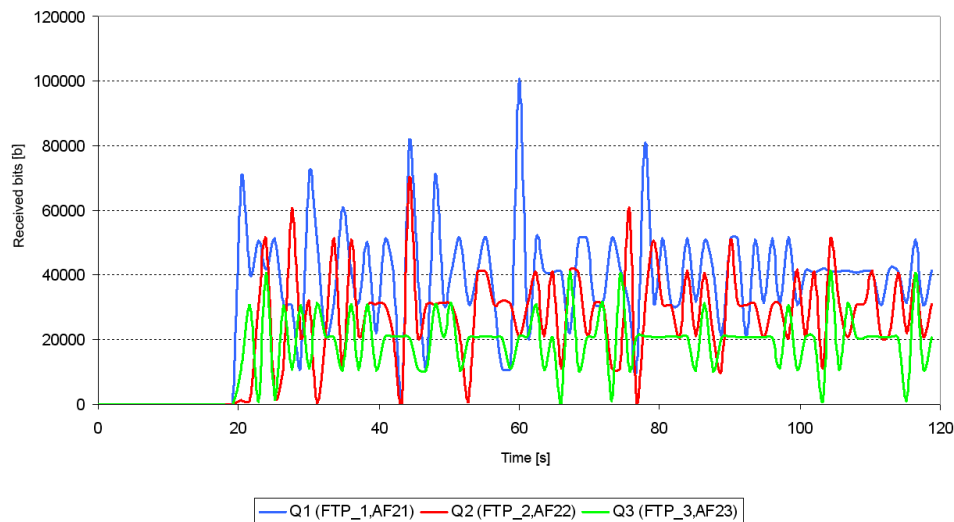


Fig. 7 Traffic shaping according to the three drop-priorities of the AF PHB

5. CONCLUSION

During our work we proved that the Opnet Modeler environment is suitable for extensive simulation of QoS related functionalities in data networks. We also defined an effective workflow for the DiffServ configuration. All the main steps of this workflow were presented in this paper to help others in solving similar tasks. The expected packet treatment was confirmed by the results of our simulations.

6. REFERENCES

- [1] Blake S., Black D., et al., *An Architecture for Differentiated Services*, RFC 2475, <<http://www.ietf.org/rfc/rfc2475.txt?number=2475>>.
- [2] Heinanen J., Baker F., et al., *Assured Forwarding PHB Group*, RFC 2597, <<http://www.ietf.org/rfc/rfc2597.txt?number=2597>>.
- [3] Davie B., Charny A., et al., *An Expedited Forwarding PHB (Per-Hop Behavior)*, RFC 3246, <<http://www.ietf.org/rfc/rfc3246.txt?number=3246>>.
- [4] Wang Z., *Internet QoS - Architectures and Mechanisms for Quality of Service*, 2001
- [5] OPNET Technologies, *OPNET Modeler Product Documentation Release 11.0*, OPNET Modeler, 2005
- [6] Kun I. Park, *QoS In Packet Networks*, ISBN: 0-387-23389-X, 2005 Boston

Acknowledgement

This paper has been supported by the Grant Agency of the Czech Republic (Grants No. 102/06/1569 and 102/05/P585) and the Ministry of Education of the Czech Republic (projects No. 1K04116 and No. MSM0021630513).