

DiffServ Extension Allowing User Applications to Effect QoS Control

JIRI HOSEK, KAROL MOLNAR, LUKAS RUCKA

Department of Telecommunications

Faculty of Electrical Engineering and Communication, Brno University of Technology

Purkynova 118, 612 00 Brno

CZECH REPUBLIC

hosek@feec.vutbr.cz , molnar@feec.vutbr.cz , rucka.lukas@phd.feec.vutbr.cz

Abstract: The technology of Differentiated Services (DiffServ) is currently one of the most used QoS technologies. The initial place where the QoS control mechanisms of this technology are applied is the edge router of the DiffServ domain. Right this property is one of the biggest disadvantages of DiffServ because it doesn't support the direct interaction with the user application that can define its QoS requirements most precisely. This paper deals with the proposal of a new QoS control system where the user application is able to cooperate with the edge node of the DiffServ domain and can affect the allocation of network resources. As a communication protocol between the user application and the edge node the well-known SNMP (Simple Network Management Protocol) has been chosen. On the basis of the theoretical proposal the basic simulation model was created in the Opnet Modeler simulating environment. This model is composed from a special SNMP manager and SNMP agent. The simulation model was focused on the evaluation of the SNMP-based communication process.

Key-Words: DiffServ, External System Domain, Management Information Base, Opnet Modeler, QoS, SNMP

1 Introduction

Recent trends in data network development and especially in the structure of network applications in use show that real-time applications are becoming dominant. Examples of such applications are the voice transfer over IP (VoIP) or videoconferencing. These real-time network services have specific transfer requirements that were not taken into account during the design and development of older computer networks. Therefore these requirements have to be satisfied by additional mechanisms and technologies [1].

Among of all QoS technologies the technology of differentiated services (DiffServ) is the most widespread [1, 2]. Its simplicity and scalability within the scope of wide area networks in comparison with other technologies are the main reasons of its expansion.

One of the basic characteristics of the DiffServ mechanism is that the edge node is responsible for the classification, metering, marking, queuing and scheduling of the incoming packet flow. This feature is also the main disadvantage of this technology because it prohibits the end application to effect the process of QoS management. Therefore the application is absolutely dependent on the QoS configuration of the edge node. The efficiency of the whole DiffServ technology therefore depends on the level of the conformity of end-user application requirements with network resources assigned by the edge router [3].

This paper offers a solution to the problem described in the previous paragraph by introducing a system which is supposed to allow networking applications to exactly

define their demands on network resources and to select the most suitable service class for their data. This solution could provide the right communication between the user application and the edge components of the DiffServ domain which could significantly increase the efficiency of this QoS technology.

2 The Technology of Differentiated Services

The DiffServ architecture is based on a simple model, where the entering data flows are aggregated into classes according to the corresponding type of service. Therefore the network nodes (excepting edge routers) do not have to care about every single data flow. Every IP packet entering to the network is marked by a specific identifier which determines the treatment of the given packet, i.e. it defines the traffic class assigned to this packet. This packet marking is executed only at the network bounds. During packet forwarding the core routers only read the mark assigned to the packet and manage the corresponding treatment. The mark is stored in the DSCP (DiffServ Code Point) field of the IPv4 packet header or in the Traffic Class field of the IPv6 packet header [4, 5].

2.1 DiffServ Domain

Wide area computer networks are usually composed of several smaller networks whereas every network can be

organized differently and managed by different subjects. Therefore every network can have different packet treatment rules and QoS support methods. Hence, the wide area networks are divided into several areas with independent administration. These areas are called DiffServ (DS) domains [4]. The example of a DS domain is shown in Fig. 1.

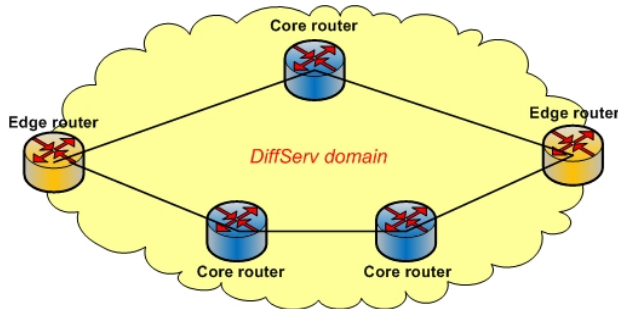


Fig. 1 DiffServ Domain

Every DS domain has its own administration rules which often requires the evaluation of traffic demands, queue based flow assignment and new packet marking. A DiffServ domain consists of two types of routers [2, 4].

2.1.1 Edge Router

The edge router connects two DiffServ domains or lies between a DS domain and the remaining part of the network which does not use packet marking. This router classifies, measures and marks the incoming traffic and subsequently sent them into the DS 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].

2.1.2 Core Router

The core router does not provides classification, metering or marking, but it ensures the forwarding of marked packets using controlled queue servicing and so it guarantees the specific service quality defined by the DSCP placed in the IP header [4].

2.2 DiffServ Reference Model

The reference model of the DiffServ technology is defined in the specification RFC 2475. 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 [4].

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 [4].

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) [6].

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

3 Description of the Proposed QoS System

As mentioned earlier, the absence of interaction between the end application and the edge router is the main disadvantage of the DiffServ technology. However, our QoS support system offers a solution to this problem.

The basic idea of our user-manageable QoS support system is based on the detection of DiffServ configuration parameters from the edge router by the end application. This detection is realized before the users start to send their data. Using these parameters the application will be able to choose the service class that best fits its requirements and traffic type. The block diagram of the proposed system is shown in Fig. 2 [7].

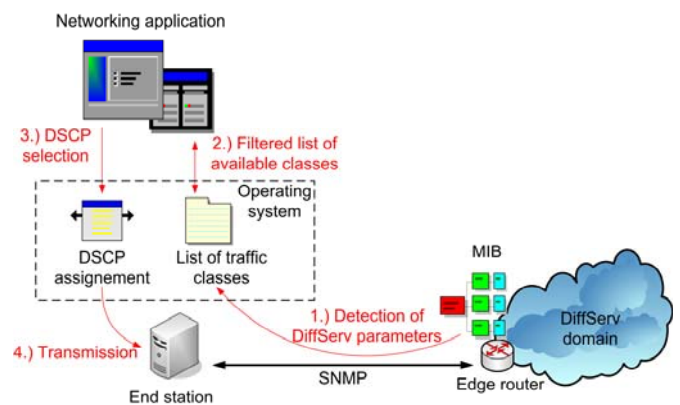


Fig. 2 Block diagram of the proposed user-manageable QoS support system

During the first step the proposed system obtains a set of DiffServ parameters from the edge router of the DS domain. These configuration entries are usually stored in a special database, called MIB (Management

Information Base), at the edge router [8]. The retrieved information is processed by the end station which represents the second step. As a result of the second step the networking application selects an appropriate service class which is supported by the DiffServ domain and corresponds to user application QoS requirements the most.

In the third step the networking application selects the corresponding DSCP mark and sets it for the traffic flow. The assigned DSCP value identifies the PHB type which will be assigned to the traffic flow. The PHB type then specifies the relevant priority level provided to the given data type during the transmission through the network.

Finally, in the fourth step the packet marked by the selected DSCP is sent to the destination node. When the marked packet is delivered to the edge router of the DiffServ domain the standard DiffServ packet processing is accomplished. It means that the incoming network traffic is firstly classified and then measured. If there are sufficient network resources and the specific data flow does not violate the agreed network conditions defined in the SLA (Service Level Agreement) then the edge router leaves the DSCP value, assigned by the application, unchanged. If the incoming data flow violates the predefined traffic conditions in any way then the edge router has the right to meter and reclassify the traffic. The final decision about the DSCP assignment is always made by the edge router of the DiffServ domain.

3.1 Communication Protocol

The system presented before requires a suitable protocol for the communication between the end station and edge node of DiffServ domain. After a detailed analysis the SNMP (Simple Network Management Protocol) was chosen for this purpose. This protocol was originally designed for remote monitoring and administration of network devices [8].

SNMP is a communication platform between the SNMP Agent and the SNMP Manager. The Agent is located at the managed device and makes accessible the configuration information and statistics of this devices. The SNMP manager is placed on the device that manages the network nodes. The Manager sends requests, encapsulated into the SNMP operations, to the agent. As a response the agent usually sends back the requested data. In the case of critical events the agents can inform the manager without any previous polling using a special message called trap [8].

The availability of configuration information through the MIB database is the main reason why we selected the SNMP protocol as a communication platform of our QoS support system. The MIB database stores these information in a hierarchical structure and its statistic

and configuration entries are accessible through OIDs (Object Identifier). For the technology of Differentiated Services the IETF (Internet Engineering Task Force) has defined the DiffServ-MIB that contains standardized configuration methods designed for remote management of this QoS support mechanism [9].

Due to this feature the SNMP protocol can easily obtain the information from the edge router which is required by the proposed user-manageable QoS support system. The way of utilization of the SNMP protocol in our system is shown in Fig. 3 [7].

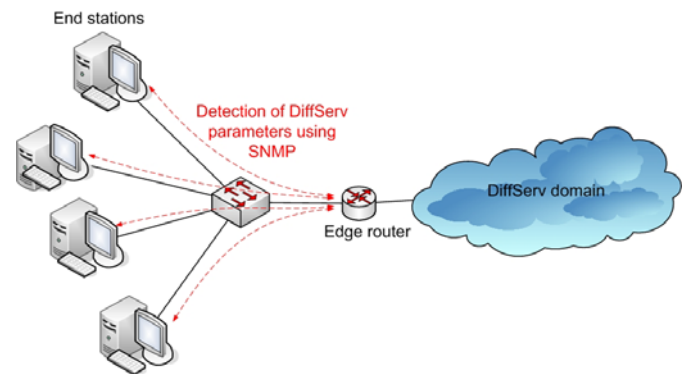


Fig. 3 Utilization of the SNMP protocol in the user-manageable QoS support system

4 The Opnet Modeler Simulation Model

We chose the Opnet Modeler (OM) simulation environment to create a simulation model of the method proposed. We designed a model, which allows us to analyse and evaluate our system before its realization on real network devices. OM contains many predefined network component models to build large, very sophisticated and complex simulation scenarios. The source code of every model is written in programming language C and can be modified if required. User also can create new models of network components or protocols [10].

The most of application protocols available in Opnet Modeler are represented by a parametric traffic generator creating a mathematically described pattern of network traffic. It means that the communication on the application level is simulated by dummy data units with application specific traffic parameters. On top of that, the SNMP protocol is not implemented in the current version of Opnet Modeler. We had two possibilities how to model our proposed system. First, we could programme the whole SNMP model in Opnet Modeler in language C. The second solution lied in the creation of an external application defining the SNMP messages and performing the SNMP commands. We decided to choose the latter way. This solution can be realized in OM through the co-simulation process that enables a cross

connection between Opnet Modeler and an external system. Another advantage of the usage of external application is the possibility to communicate with the real devices, such as routers.

4.1 The Architecture of the Simulation Model

The model of our user-manageable QoS support system is shown in Fig. 4.

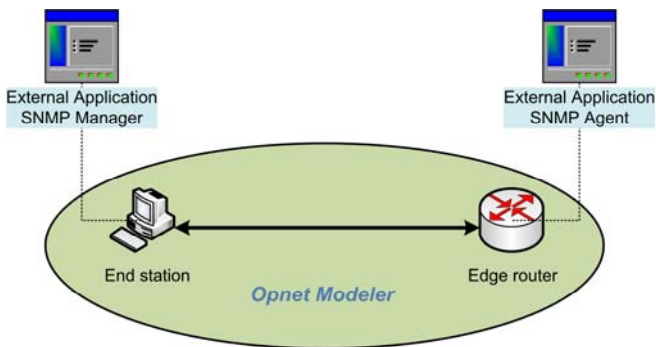


Fig. 4 Design of simulation model

This model consists of the end station and the edge router of the DiffServ domain. To model these components we used the predefined Ethernet workstation model and the IP gateway model with four Ethernet hub interfaces and eight serial line interfaces. These models are included in the standard Opnet Modeler model library. The workstation and gateway are connected with a 10BaseT point-to-point duplex link.

The external applications performing the SNMP commands and sending the SNMP messages were linked to the end station and the edge router. These applications were developed in language C and are based on pre-prepared open source C-based libraries, which contain the definitions of SNMP commands and messages. These libraries facilitated the development of user application. The end station performs tasks known from SNMP Manager, like periodical polling of the SNMP agent and processing the corresponding MIB information. The external application performing the function of the SNMP Agent was linked to the Edge router. The main function of this application is to provide access to the MIB database and to reply to requests received from the manager.

4.2 Communication Interface between External Application and Opnet Modeler

In spite of the fact that the implementation of the SNMP protocol is made by external applications the Opnet Modeler is the main controlling element of the whole simulation model. The simulation is executed as a co-simulation containing two external SNMP applications

and the OM model. When required the OM passes the control to the external application that carries out the requested operations. Then the application returns the control of the co-simulation together with the resulting data back to the Opnet Modeler.

A special interface is required that ensures the control and information exchange between Opnet Modeler and the external applications. For this purpose OM defines the esys interface. This interface belongs to the ESD (External System Domain / Definition) system. The external system is represented in Opnet Modeler as a model whose behavior is determined by an external code. Such a model can represent anything from a microchip to a user application. Opnet Modeler passes data to the external system and receives data from it without having any implicit knowledge of the method of processing these data [10].

The ESD system can contain any number of external system (esys) interfaces, accessible from both the Opnet Modeler and the external code. This interface is the basic component of the whole co-simulation process. The exchange of control between the OM and an external application is based on interruptions. Each interruption invokes the corresponding action and its execution is indicated by another interruption [10]. The basic principle of co-simulation in Opnet Modeler is shown in Fig. 5.

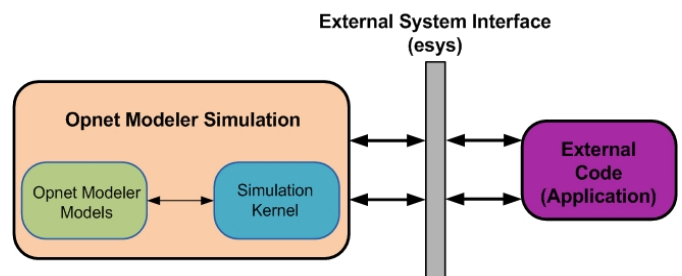


Fig. 5 Co-simulation process in Opnet Modeler

For the purpose of information exchange two esys interfaces were defined between the end station and the SNMP manager application and between the edge router and SNMP agent application correspondingly.

The co-simulation consists of the following phases. After the initialization of the models and variables the end station generates a request for the edge router. Then the end station in the Opnet Modeler passes the control to the SNMP Manager external application which constructs the corresponding SNMP request. This SNMP message is the output of SNMP manager application. This message is then returned back to the end station in OM. Next, the end station encapsulates the SNMP message into a UDP (User Datagram Protocol) datagram and sends it to the edge router. On the edge route side the SNMP message is decapsulated from the packet received. Then the SNMP message is passed through the

esys interface to the SNMP Agent external application. The agent generates a reply to this request. The response contains a set of specific DiffServ configuration parameters obtained from the MIB database. For this purpose the SNMP manager must have sufficient rights to access the MIB database of the router. In the current state of our simulation the DiffServ-MIB is stored in a text file. However, the structure of the simulation model allows real devices to be easily integrated into the simulation scenario.

After the external SNMP agent obtains the required information from the DiffServ-MIB it sends this data back to the OM model of the edge router. The edge router encapsulates the response into a UDP packet and forwards it to the end station. The end station evaluates the received data and derives the most suitable DSCP mark for the network service. This DSCP mark is then set for the IP packets containing application data.

5 Conclusion

The non-existing cooperation between the user applications and the components of the DiffServ domain is the main disadvantage of the DiffServ technology. A solution to this limitation is offered by our user-manageable QoS support system. This system allows the networking applications to affect the process of quality of service assurance. The application will be able to select the QoS class which is the most suitable for its network traffic and which can guarantee the network resources required. The network application indicates the selected class by using the corresponding DSCP mark during the data transmission. The assigned DSCP value can ensure the right packet enqueueing and scheduling at the edge router. However, the final decision about keeping or changing the assigned DSCP mark is still up to the edge router. This is an important requirement to prevent the network from any violation of security or SLA conditions defined in the network.

The Opnet Modeler simulation environment was chosen to evaluate and analyze the proposed system. A simulation model was created in this environment, which models the process of communication between the end station and edge router. A modified SNMP Manager which extends the end station was implemented as an external application. This SNMP manager detects the DiffServ configuration parameters available at the edge node of the DS domain. The required parameters are obtained from the MIB database of the router that is managed by the SNMP agent. The model of the SNMP agent was also realized as an external application. We used the esys interface to exchange data between Opnet Modeler and the two previously mentioned external applications.

The QoS support system described in this paper is

relatively complex and is still in the development phase. It is obvious that the SNMP-based communication method between the end station and the edge router is essential but it is still only a partial segment of the whole system. There are several additional tasks, which must be solved before the integration of this system into a real environment. Some of the most important tasks are the evaluation method of the DiffServ configuration parameters obtained by the end station or the implementation of the DSCP assignment.

References:

- [1] Park, I., K., *QoS in Packet Network*, Springer, 2005.
- [2] Wang, Z., *Internet QoS: Architecture and Mechanisms for Quality of Service*, Morgan Kaufmann, 2001.
- [3] Szegedi, T., Hattingh, C. *End-to-End QoS Network Design*, Cisco Press, 2004.
- [4] Blake, S., Black, D., Carlson, M., Davies, E., Wang Z., Weiss, W., *An Architecture for Differentiated Service – RFC 2475*, IETF, 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 – RFC 2474*, IETF, 1998.
- [6] Babiarz, J., Chan, K., Nortel Networks, Baker, F., Cisco Systems, *Configuration Guidelines for DiffServ Service Classes – RFC 4594*, IETF, 1998.
- [7] 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.
- [8] Mauro, D., Schmidt, K., *Essentials SNMP, Second Edition*, O'Reilly Media, 2005.
- [9] Barker F., Chan, K., Smith, A., *Management Information Base for the Differentiated Services Architecture – RFC 3289*, IETF, 2002.
- [10] Opnet Technologies, *Opnet Modeler Product Documentation Release 14.5*, Opnet Technologies Inc., 2008.

Acknowledgement:

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).