INTEROPERABILITY OF INTEGRATED SERVICES AND DIFFERENTIATED SERVICES ARCHITECTURES

Radu DOBRESCU[1], Denisa CÂRCIUMĂRESCU[2]

Tendinţele actuale în dezvoltarea aplicaţiilor Internet în timp real şi creşterea rapidă a sistemelor mobile, indică faptul că viitoarea arhitectură Internet va trebui să suporte variate aplicaţii cu diferite cerinţe QoS. Acest document prezintă arhitectura Serviciilor Integrate versus Servicii Diferentiate, fiecare cu cerinţele sale specifice. Serviciile Integrate şi Serviciile Diferentiate sunt două dintre abordările curente care prezintă garantarea QoS în Next Generation Internet. Int Serv oferă resurse sigure prin rezervarea acestora pentru aplicaţii individuale ale specificaţiilor fluxurilor, pe când, DiffServ foloseşte o combinaţie de politici extreme, previziuni şi prioritizări ale traficului. În final, am prezentat o comparaţie între IntServ şi DiffServ raportat la tipurile de servicii, controlul admisie şi aplicarea politicilor de control.

The current trends in the development of real-time Internet applications and the rapid growth of mobile systems, indicate that the future Internet architecture will have to support various applications with different Quality of Service (QoS) requirements, regardless of whether they are running on a fixed or mobile terminals. This document presents a general Integrated Services / Differentiated Services architecture design with specific requirements. Integrated Services (IntServ) and Differentiated Services (DiffServ) are two of the current approaches to provide Quality of Service (QoS) guarantees in the next generation Internet. Integrated Services provide resource assurance through resource reservation for individual applications flows, whereas Differentiated Services use a combination of edge policing, provisioning and traffic prioritization. Finally, we present a comparison between Int Serv and DiffServ related to the applications, the type of services, the admission control and policy control.

Keywords: Quality of service, differentiated services, integrated services

1. Introduction

Internet was designed for non-real time applications and hence does not provide Quality of Service (QoS) guarantees to applications. With the proliferation of the Internet, there is a strong interest in providing QoS to real-time applications in the next generation Internet. QoS includes guaranteed bandwidth, bounded packet delay, jitter and packet loss. QoS is generally implemented by

[1] Professor, Faculty of Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: radud@isis.pub.ro.
[2] PhD student, Faculty of Electrical Engineering, University Valachia of Targoviste, Romania

different classes of service contracts for different users. A service class may provide low-delay and low-jitter service for customers who are willing to pay a premium price to run real-time applications such as video conferencing. Another service class may provide predictable services for customers who are willing to pay for reliability. Finally, the *best-effort* service provided by the current Internet will remain for those customers who only need connectivity. The Internet Engineering Task Force (IETF) has proposed a few models to meet the demand for QoS. Notable among them are the Integrated Services (IntServ) and Differentiated Services (DiffServ) models. The IntServ model is characterized by resource reservation; before data is transmitted, applications must set up paths and reserve resources along the path. IntServ aims to support applications with different levels of QoS within the TCP/IP (Transport Control Protocol/Internet Protocol) architecture. IntServ however, requires the core routers to remember the state of a large number of connections giving rise to scalability issues in the core of the network. It is therefore *suitable at the edge network* where the number of connections is limited. The DiffServ model is currently being standardized to provide service guarantees to aggregate traffic instead of individual connections. The model does not require significant changes to the existing Internet infrastructure or protocol. The DiffServ model utilizes six bits in the Type of Service (TOS) field of the IP header to mark a packet for being eligible for a particular QoS. DiffServ does not suffer from scalability issues, and hence is *suitable at the core of the network*. It is therefore believed that a significant part of the next generation Internet will consist of IntServ at the edge and DiffServ at the core of the network. As a result, architectures with IntServ at the edge and DiffServ at the core to provide QoS to end applications have been proposed at the IETF. Interconnection of IntServ and DiffServ, in order to exploit the individual advantages of IntServ (per flow QoS guarantee) and DiffServ (good scalability in the backbone), requires a mapping from IntServ traffic flows to DiffServ classes to be performed at the ingress to the DiffServ network.

2. Integrated Services Architecture

The Integrated Services (IntServ) architecture described in detail in [1] recommends a set of extensions to the Internet architecture in order to enable services that go beyond the traditional best – effort service, aimed for addressing the real-time applications QoS requirements. QoS in terms of Intserv is associated with the time-of-delivery of packets and is characterised by parameters such as bandwidth, packet delay and packet loss rate [2]. The IntServ architectural design is based on the notion that in order to fulfill the QoS requirements of the applications, network resources should be managed and controlled, which implies that the admission control and resource reservation are the key building block of

this architecture. As such the Intserv architecture provides mechanisms by means of which applications can choose between different services for their traffic and explicitly signal QoS requirements per individual flow to network elements (hosts, routers). The functionality of the Integrated Service architecture can be seen as a composition of two basic elements, Integrated Service model and the reference implementation model, which provides the necessary kit and the accompanied terms for realisation of the Integrated Service model. Each of these elements encompass a certain number of functional entities, which are described below:

- **Integrated Service model**

The Integrated Service model defines two types of services the Controlled Load Service and the Guaranteed Service for usage by the real-time applications. The specific service is invoked by the applications QoS requirements. The application's generated traffic, depending on these QoS requirements, will get the one of two existing service treatment, either the Controlled Load Service or Guaranteed Service. QoS requirements depend on the nature of different applications, that is, whether they are elastic, non-adaptive or adaptive real-time applications. Reservation model in Intserv describes scenarios on how the reservations are made and managed.

- **Implementation reference model**

For realisation of the Integrated Services model the Implementation Reference model defines several mechanisms that encompass the layer 3 (router) scheduling, classification, admission control and resource reservation. The classification, scheduling and admission control are part of traffic control tools. The classifier determines to which class each packet belongs according to their QoS requirements,. the service that determines the way the scheduler should handle them. The scheduler processes these packets based on their QoS requirements. Each network element in the network performs admission control and policy control to the incoming flows in order to determine whether there are enough resources and whether the flow has permissions to request the specific service. The RSVP (Resource Reservation Protocol) signalling protocol [3] was designed as a dynamic mechanism for explicit reservation of resources in Intserv, although Intserv can use other mechanisms as well. The IntServ architecture and RSVP can also function independently of one another. And, even though IntServ was designed and provides the means for end-to-end QoS, it is not widely deployed. As it is emphasised so many times by now, due to maintenance and control of per-flow states and classification, reserving resources per-flow introduces severe scalability problems at the core networks, where the number of processed flows is in a millions range. Consequently the usage of the Integrated Services architecture is limited to small access networks where the number of flows using reservations is modest.

The simplified RSVP/IntServ framework is shown in Figure 1. As it is shown every RSVP aware router in the IntServ will perform RSVP signalling, admission control, scheduling and policing.

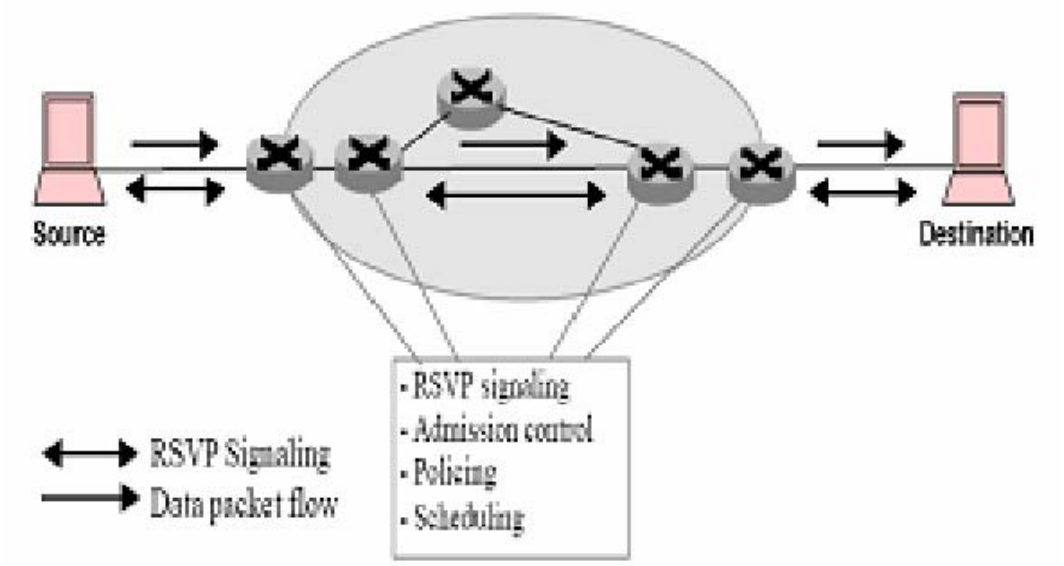

Fig.1. A simplified RSVP/IntServ framework

3. RSVP (Resource Reservation Protocol)

The Resource Reservation Protocol (RSVP) is a signalling protocol that can be used by an application to convey its QoS requirements to network elements. RSVP is used only for communication of QoS parameters and it doesn't provide any QoS related functions, that is the RSVP protocol itself has no understanding of the information it carries on QoS requests. RSVP is initiated by an application at the beginning of a communication session. A communication session is identified by the combination of the IP destination address, transport layer protocol type and the destination port number [4]. Each RSVP packet contains details of the session they belong. The resource provisioning is independent of RSVP; that is the admission/rejection of the required resources by means of RSVP for a particular flow is a function of IntServ in this case. Once the requested resources are reserved, they will be used by the particular data flow.
RSVP protocol defines seven types of messages, of which the fundamental ones are the PATH and RESV messages. PATH and RESV messages carry out the basic operation of RSVP. The rest of the RSVP messages are used to either provide information about the QoS state or to explicitly delete the QoS states along the communication session path. The RSVP messages and their functions as given in [5] are presented in fig. 2:

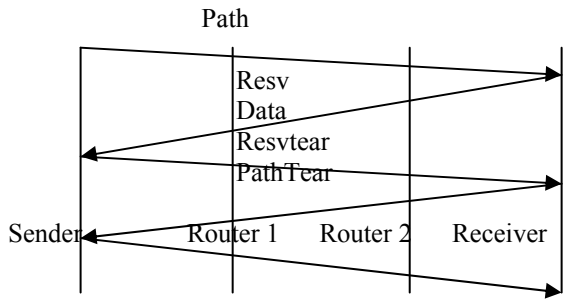

Fig.2. Typical RSVP messages

- The PATH message - is sent by a source that initiates the communication session and it explicitly binds the data path of a flow. Furthermore, it describes the capabilities of the source.
- The RESV message - is issued by the receiver of the communication session and it follows exactly the path that the RSVP PATH message has followed hop by hop back to the communication session source. The RESV message on its way back to the source may install QoS states at each hop. These states are associated with the specific QoS resource requirements of the destination.
- The PATH Error message – It is used to report errors that are occurring during the installation of a path from the source to the destination of a communication session.
- The RESV Error message – It is used to report errors that are occurring during the installation of the reservation states along the communication session path.
- The RESV Confirm message - It provides a positive indication to the initiator of the communication session informing that all nodes along the communication session path accepted the reservation request. The RSVP Confirmation messages are typically sent by the source of the communication session directly to the destination of this communication session. Intermediate nodes do not process RSVP confirmation messages.
- The PATH Tear message – It is sent by the source of the communication session and it explicitly deletes the stored QoS state information on all nodes included in a communication session path.
- The RASV Tear message - Is sent by the destination of the communication session and it explicitly deletes the stored QoS state information on all nodes included in a communication session path.

4. Controlled load service and guaranteed service

Controlled Load (CL) Service [6] is intended for adaptive real-time applications, which are highly sensitive to overloaded conditions in the network. The controlled load service offers only a single function to these applications, it provides the traffic delivery within the same bounds as would have been the case in the environment of "unloaded" (not heavily loaded or congested networks. CL does not accept nor use the specific QoS parameters such as packet loss and delay as control parameters. In requesting CL service applications may expect, under the assumption that the network is functioning correctly, that their traffic will be delivered successfully and that the transit delay induced by the network is close to the minimum transit delay of successfully transmitted traffic. The QoS disruption in the delivery service depends on the "burst time". Burst time is defined as the time needed for the transmission of the maximum flow's burst size at the required transmission rate. These parameters are defined in the TSpec of the requester's flow. The short duration of QoS disruption events occur when the average queuing delay is significantly larger than the burst time and they are considered as "normal operation". If the congestion loss is significantly larger than the burst time, that is considered as a failure of the resource allocation schemes. The concrete Controlled Load TSpec parameters as given in [6] are:
r – token bucket rate (measured in bytes/second)
b – token bucket size (measured in bytes)
p – peak rate (measured in bytes/second)
M – maximum datagram size (measured in bytes)
m - minimum policed unit (measured in bytes)
The network elements receiving a CL request must provide the necessary bandwidth and packet processing resources for handling the requested level of traffic as given in the TSpec of the requestor. The method a network element uses to determine whether the request can be accommodated is a local matter, and can be implementation dependent as long as the control parameters and message formats are interoperable. It may employ measurement-based approaches or it may employ appropriate scheduling mechanisms.

4.1 Guaranteed Service

The Guaranteed Service (GS) [7] is an quantitative service which provides bandwidth guarantees and delay bounds and as such it is intended for non-adaptive real time applications with strict QoS requirements. The GS service controls only the maximum delay; thus it does not control the minimum delay or control or minimise the jitter. The delay consists of the fixed delay and the queuing delay. Fixed delay is a path property and is determined by the setup mechanism (e.g. RSVP) during the path set-up, while queuing delay is determined

by the GS service. In order to determine specific end-to-end delay bounds, GS service relies on the behaviour of each network element in the path starting from the source. The end-to-end delay bound as given in [7] is:

$$D=\begin{cases}\dfrac{(b-M)(p-R)}{R(p-R)}+\dfrac{(M+C_{tot})}{R}+D_{tot} & p>R\geq r\\ \dfrac{(M+C_{tot})}{R}+D_{tot} & r\leq p\leq R\end{cases} \quad (1)$$

where: r – token bucket rate (measured in bytes/second); b – token bucket size (measured in bytes); p – peak rate (meas--red in bytes/second); M – maximum datagram size (measured in bytes); R – flow service rate (or bandwidth) (measured in bytes/sec); Ctot - end-to-end calculation of C (see below); Dtot – end-to-end calculation of D.

By means of the token bucket model, in particular the token bucket size b and the rate r, the applications (which in fact controls these parameters) has an a priori knowledge about the queuing delay that the guaranteed service will provide. The application's source is allowed to transmit data as long as there are tokens available in the bucket. The packets are released at the token bucket rate r, although it may happen that the source produces more packets than this rate r. In this case the packets are stored in the buffer and than released at rate r. The transmission rate is limited by the peak rate p. When there is no transmission the bucket can accumulate tokens up to size b. The bucket is filled at a constant rate with tokens, until it is full. In case the delay exceeds the expectations the application can modify its token bucket and data rate to reduce the delay. The guaranteed service relies on the fluid model conforming to the token bucket model to ensure that the queuing delay of any packet in the flow is less than the total delay computed along the flow path:

$$D_t = \frac{b}{R} + \frac{C}{R} + D \quad (2)$$

b—the maximum number of tokens; R—the service rate the packets are served;
c—additional delay which is rate -dependent, referred also as packet serialisation (unit is in bytes); D—additional delay which is not-rate – dependent, is a result of time spent waiting for transmission through a node (unit is in microseconds)

5. Differentiated Services Architecture

The Differentiated Services (Diffserv) architecture [8] was introduced as a result of the efforts to avoid the scalability and complexity problems of Intserv. Scalability is achieved by offering services on aggregate basis rather than per-flow and by forcing as much as possible the per-flow states to the edges of the network. The service differentiation is achieved by means of Differentiated Service (DS)

field in the IP header and the Per-Hop Behaviour (PHB) as main building blocks. At each node packets are handled according to the PHB invoked by the DS byte in the packet header. The Diffserv divides the entire network into domains, where Diffserv domain as defined in [9] is a contiguous set of nodes which operate with a common set of service provisioning policies and PHB definitions. The Diffserv domain consists of the interior nodes and boundary nodes, which connect the Diffserv domain to other domains and are responsible for conditioning the traffic according to the service agreement that is in effect between neighbouring boundary domains. The Diffserv domain will provide to its customer, which is a host or another domain, the required service by complying fully with the agreed Service Level Agreement (SLA). A SLA is a bilateral agreement between the boundary domains negotiated either statically or dynamically. The transit service to be provided with accompanying parameters like transmit capacity, burst size and peak rate is specified in the technical part of the SLA,. the Service Level Specification (SLS). The simplified Diffserv framework is given in Figure.3.

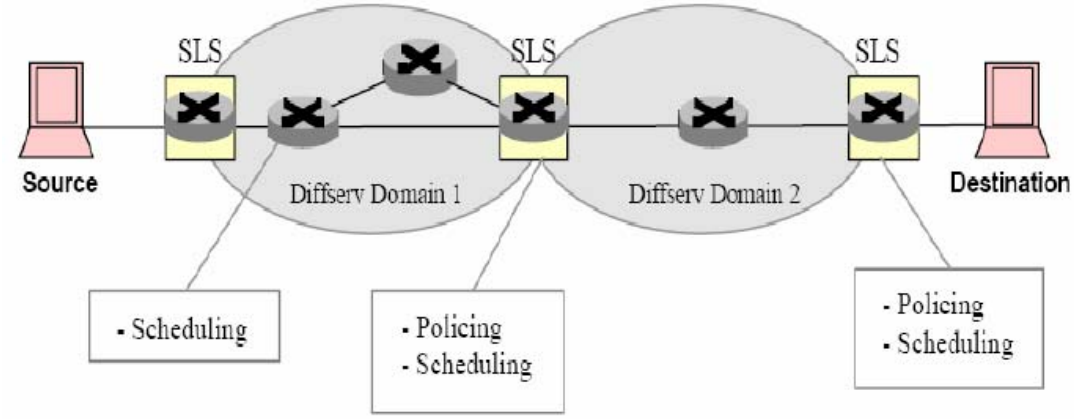

Fig.3. A simplified Diffserv framework

The specific realisation of the Diffserv architecture depends on whether the resources are statically or dynamically provisioned and on resource allocation mechanisms. The two-tier resource management model for Diffserv proposes an approach allowing each individual Diffserv domain to make their own choices on the mechanisms and protocols to be used for internal QoS support, while for the external QoS support they rely on the SLA. By means of this model the end-to-end QoS can be achieved through a concatenation of inter-domain and intra-domain resource allocations, as long as those allocations match the level of the aggregated demand [10]. Each domain in this model will have a resource manager.

6. Integrated services and differentiated services:- a comparison

Integrated Services and Differentiated Services represent two different solutions. Integrated Services provide resource assurance through resource reservation for individual applications flows, whereas Differentiated Services use a combination of edge policing, provisioning and traffic prioritization. In the following we present a comparison between Int Serv and DiffServ related to the applications, the type of services, the admission control and policy control and the classifiers.

6.1 Applications

In order to use the QoS services, an application must supply the network with the necessary information to receive the desired treatment. So a QoS architecture impacts the way applications are designed .

IntServ. The Resource reSerVation Protocol (RSVP) has been designed as a signaling protocol that applications use to convey QoS specs in IS routers. (Braden, *et al*., 1997). The IntServIS architecture requires that network's nodes are notified about each application's QoS requests, therefore a strong overhead is introduced in RSVP routers which manage lots of QoS data flows. RSVP is the most widely used mechanism to transmit QoS requests to all routers in a QoS path, though it is not mandatory in the IS architecture (Wroclawski, 1997). The access to the local RSVP process by applications is simplified through calls to a standard RAPI (Resource API).

DiffServ. Though no end-to-end signaling is required in the DS architecture, datagrams must be marked somewhere with the proper Differentiated Services Code Point (DSCP) value (Blake, *et al*., 1998), and this can be made in two different ways:
Host marking: Each host marks datagrams for each application. In order to pursuit the proper DSCPs, the hosts should maintain local tables provided by routers of the network they belong to.
Router marking: Datagrams are marked by routers in the local domain's network, maybe the first one. A mechanism like RSVP is therefore needed for the first-hop signaling, to allow applications to notify the router. More interestingly, packets can be remarked as they move to a different DS domain.

6. 2. Type of services

In the IntServ architecture an application specifies the QoS needs choosing a service and a related set of parameters, therefore a network must be ready to treat an arbitrary number of different QoS requests. In the DS architecture an

application specifies a service selecting a Per Hop Behaviour (PHB) in a limited set of choices: the QoS demand can't be defined in the absolute terms of basic parameters (data rate, delay and so on). A DSCP is served with a PHB; interestingly, a set of DSCPs has a scope that is local to a DS domain.
IntServ. A best-effort service is applied to datagrams when no service is used; in this case, no kind of signaling is then required. Two different services are now standardized:
Controlled Load Service. Applications are allowed to specify some statistic parameters of the flow (data rate, peek rate, token bucket size). This service is used to obtain an unloaded path with an high throughput and low packet loss rate.
Guaranteed Service. Applications can also specify the end-to-end delay: real-time applications are a natural target of this service.
DiffServ. Three classes of PHB are now standardized:
Class Selector Compliant PHB Groups This group of PHB is a way to keep a backward compatibility with the Precedence bits, coded with the three leftmost bits of the ToS field in IPv4. To this purpose, 8 DSCP are then used to give high priority to network (e.g. routing) traffic. Inside this class, a Default PHB is applied to best-effort datagrams and the 0 code is used in the DS field.
Expedited Forwarding PHB. The Expedited Forwarding (EF) PHB is used to obtain a service with low packet losses, low delay, low jitter. To do this, the EF queue in routers is low-sized and served with an high priority.
Assured Forwarding PHB . The Assured Forwarding (AF) PHB group is a way to offer different levels of forwarding assurances for IP packets inside a DS domain. Four AF classes are defined and each class is so assigned a defined amount of resources; inside each class to a packet is assigned one of four different drop precedence. So, 16 DSCP are used for the AF PHB group.

6.3. Admission control and policy control

In QoS networks, the usage of network's resources is restricted to the available amount of resources and sometimes planned. An allocation request takes place only if there are enough free resources in the network, and if the requester is permitted to use them. Thus, the Admission Control module verifies that the network has a sufficient amount of resources to accept the user's request. The Policy Control module verifies that the user has sufficient administrative permissions to request resources. Otherwise, if one of the two controls fails the request is rejected.
IntServ. The Admission Control and Policy Control are made at each hop of the End-to-End QoS path. This leads to the scaling problem in core routers because of the intense activity they carry on.

DiffServ. The Admission Control and Policy Control are made in the network's border. A Border Router verifies the correctness of the incoming requests at the ingress of a DiffServ domain; in these same points traffic is shaped (and maybe reshaped). Internal Routers (that perhaps concentrate lots of the traffic) don't verify the amount of used resources. In this way, some of the QoS complexity is bounded to border of a DiffServ domain.

6. 4. Classifiers

A QoS network treats datagrams in different ways depending upon the kind of traffic they are used for, to this purpose a QoS architecture must provide a classifier in the forwarding path inside a router. A classifier selects datagrams and forward them in different service queues.
IntServ. The most straightforward way to serve IntServ data flows is to dedicate a scheduler's queue to each one. A classifier switches each datagram in the proper scheduler's queue looking at the 5-ple (Source IP address, Destination IP address, Source Transport Port, Destination Transport Port, Transport Protocol ID): traffic is identified with a per-application granularity. A router looks for these fields in the IP and Transport headers of all datagrams to find IP packets belonging to allocated flows, and this leads to a high amount of work. Moreover, fragmentation must be avoided because transport-level information is used.
DiffServ. All datagrams with the same DSCP have the same treatment. The DS field is 6-bits-long, so the total number of queues/behaviours is limited to a number of 64, and different data flows are served by the same queue. Datagrams are distinguished by the DiffServ field. This field is contained in the IPv4 ToS field, or in the IPv6 Traffic Class field. A single field is checked to this purpose.

7. Conclusions

Integrated and differentiated services do not necessarily have to be considered as competing concepts. It is rather advisable to combine both approaches. As described above both Integrated and Differentiated Services architecture are designed to deploy QoS on the best effort Internet, by means of different mechanisms for differentiation of services and each having their own advantages and disadvantages. The framework for InteServ operation over DiffServ views the two architectures as complementary towards deploying end-to-end QoS. As noted in this framework Intserv provides means for end-to-end QoS over different heterogeneous networks and it must be supported in different network elements, thus Diffserv network is just a network element in this end-to-end path. It is primarily intended to support the quantitative (guaranteed) services end-to-end, which has not been deployed yet by RSVP/Intserv, due to the lack of scalability. The benefits of this framework for Intserv is thus rather obvious, since

Diffserv aggregate traffic control scalability fills in the lack of scalability of the RSVP /Intserv. IntServ and DiffServ networks are both proposed to promote quality of service (QoS) for multimedia applications in the Internet. However, the complexity of communication between diverse applications and underlying QoS architectures leads to one of the deployment problems which decreases the utility of QoS provisioning.

R E F E R E N C E S

[1] *R. Braden, D. Clark, S. Shenker,* "Integrated Services in the Internet Architecture: An Overview", IETF RFC 1633, 1994.

[2] *R. Dobrescu, B. Droasca, R. Grigorescu* - QoS strategies for satellite communication networks, Proc. of the 12-th Int. Conf. SIMSIS, 2004, p. 117-123

[3] *R. Dobrescu, M. Dobrescu* - Reactive Congestion Control for Multipoint Video Services, 8th International Workshop on Systems, Signals and Image Processing, Bucuresti, p. 95-99, June 2001

[4] *G. Eichler, H. Hussmann, G. Mamais, I. Venieris, C. Prehofer, S. Salsano,* "Implementing Integrated and Differentiated Services for the Internet: A practical approach," IEEEcommunication Magazine 38, p. 132–141, Jan 2000

[5] *M. Jacobsson, S. Oosthoek, G. Karagiannis,* "Resource Management in Differentiated Services: A Prototype Implementation". Seventh IEEE Symposium on Computers and Communications, 01-04 July 2002, p. 21-28

[6] *J. Wroclawski,* "Specification of the Controlled-Load Network Element Service, RFC2211, IETF INTSERV, September 1997

[7] *S. Shenker, C. Partridge, R. Guerin,* "Specification of Guaranteed Quality of Service", RFC 2212, September 1997

[8] *J. Harju, P. Kivimaki,* Co-operation and comparison of diffserv and intserv: Performance measurements, in Proc. 25th IEEE Conference on Local Computer Networks, 2000, p. 177–186

[9] *K. Kilkki,* Differentiated Services for the Internet, Macmillan Technical Publishing, 1999

[10] *T. Chahed, G. Hebuterne, C. Fayet,* On mapping of QoS between Integrated services and Differentiated Services, in International Workshop on Quality of Service, 2000, p. 173–175