Faculty of Engineering

ETM3056
Communication Networks
Trimester 1, (2007/2008)

Assignment

Group 13
Quality of Service – Integrated/Differentiated Service vs. Handling of Delay/Jitter

by group

HENG FOOK YAU	1051107330
YAP SHIAO PHEI	1051103646
FAISAL MOHD. JAMAL MADANI	1041111424
RAVINDRA A/L SIVANASVARAN	1041106564
SHAH SIREEN SHABIR	1041103213
ZAED BIN ZAHIR	1041101875
CHEN CHEE YONG	1031132296

(11 SEPTEMBER 2007)

TABLE OF CONTENTS

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ATM	Asynchronous Transfer Mode
ALE	ATM Link Enhancer
BER	Bit Error Rate
BS	Base Station
BE	Best Effort Services
CC	Call Control
DiffServ	Differentiated Services
DSCP	Differentiated Services Code point
DLC	Data Link Control
e2e	End-to-End
EF	Expedited Forwarding
IntServ	Integrated Services
ISP	Internet Service Provider
ITU	International Telecommunication Union
LLC	Logical Link Control
MAC	Media Access Control
MAN	Metropolitan Area Network
MFC	Multifield Classifier
MPLS	Multi Protocol Label Switching
PHY	PHYsical layer of a network model
PHB	Per Hop Behavior
PRMA	Packet Reservation Multiple Access
QoS	Quality of Service
RRM	Radio Resource Management
RSVP	Reservation Protocol

RTD	Round Trip Delay
SIR	Signal-to-Interference Ratio
SLA	Service Level Agreement
UMTS	Universal Mobile Telecommunications System
UPT	Universal Personal Telecommunications
UT	User Terminal
TOS	Type of Service
VCI	Virtual Circuit Identifier
VoIP	Voice over IP
WAN	Wide Area Network
W-CDMA	Wideband CDMA

ACKNOWLEGEMENTS

First, we would like to thank our lecturer, Mr. Khairil Anuar and Md. Zalifah from FOE for being one of the main factors of us being able to complete this research paper successfully. Their recommendations and constructive comments on our pre-draft was a valuable asset as it leads us in the right direction. Not only that, they also put a lot of effort in explaining the details of getting this job done, so that we could have a clearer view about what we are going to do.

Secondly, we also owe a great debt to some of the friends. Some of them that had taken this subject, especially the Panasonic Lab researcher were willing to give us their suggestions and experiences in using the NS-2 simulator for the simulation purpose. Without their help, we would certainly have to spend much more time cracking our head thinking of the next move. Other than that, we would also like to express our appreciation to all the people that were willing to spend their time in order to assist us. The results that we gathered from them made it easier for us to where to find for the secondary resource. Moreover, some of the suggestions that they raised really helped us a lot.

Last but not least, we would like to thank our classmates, as while they were doing the same task as us, they were willing to share their experience and also give us some suggestions on our report. We were able to exchange ideas with them, and this lead us to a better outcome.

1. INTRODUCTION

1.1 Background

Quality of Service (QoS) refers to the capability of a network to provide better service to selected network traffic over various technologies, including Frame Relay, Asynchronous Transfer Mode (ATM), Ethernet and 802.1 networks, SONET, and IP-routed networks that may use any or all of these underlying technologies. The primary goal of QoS is to provide priority including dedicated bandwidth, controlled jitter and latency that required by some real-time and interactive traffic, and improved loss characteristics. Also important is making sure that providing priority for one or more flows does not make other flows fail. QoS technologies provide the elemental building blocks that will be used for future business applications in campus, WAN and service provider networks.

1.2 Objectives

This assignment was intended as a contribution towards end-to-end QoS deployment on the wired and wireless Internet. The work presented in this document has proven right what was stated in the beginning: that enabling QoS in the Internet is a difficult task due to the complexity it introduces in the overall wired and wireless Internet architecture.

The study will focus on the theoretical background of IntServ and DiffServ and jitter/delay in terms of QoS. The QoS problems will be investigated along its resource management. The concept on these two technologies is discussed followed by the impacts. We will offer a detailed description of the services, as the basis for the implementation in real networking environment. Then, the comparison between these two services will also be study. The mathematical algorithms used will be identified as well as the key parameters and factors that influence their performance. A new service classification proposal based on the discussion and findings is described next. Finally, if time is allowed both analytical and practical simulation results using NS-2 for different traffic scenarios will be presented.

1.3 Fundamental Concept

QoS technologies divided into 3 main components. There are Best Effort, Integrated Services (IntServ) and Differentiated Services (DiffServ).

Differentiated Services *(DiffServ)* is a protocol for specifying and controlling network traffic by class so that certain types of traffic get precedence - for example, voice traffic, which requires a relatively uninterrupted flow of data, might get precedence over other kinds of traffic. Differentiated Services is the most advanced method for managing traffic in terms of what is called Class of Service (Cos).

Unlike the earlier mechanisms of 802.1p tagging and Type of Service (ToS), Differentiated Services avoids simple priority tagging and depends on more complex policy or rule statements to determine how to forward a given network packet. For a given set of packet travel rules, a packet is given one of 64 possible forwarding behaviors - known as per hop behaviors (PHBs). Differentiated Services and the Class of Service approach provide a way to control traffic that is both more flexible and more scalability than the Quality of Service approach.

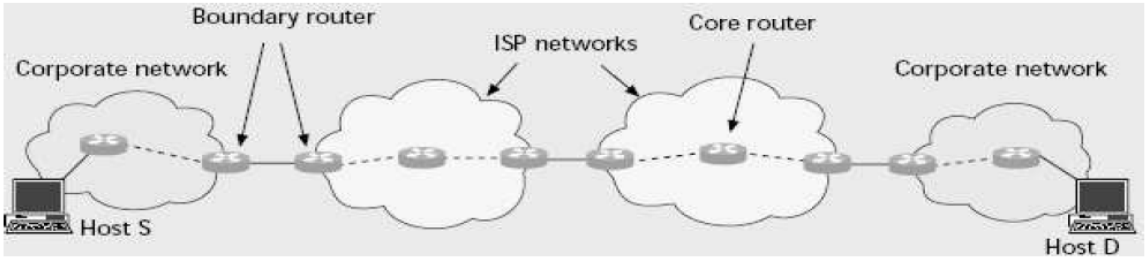


Figure 1.1: End-to-end transport from host S to host D under the Diffserv architecture

Integrated services *(IntServ*) are an architecture that specifies the elements to guarantee quality of service (QoS) on networks. IntServ can for example be used to allow video and sound to reach the receiver without interruption. IntServ specifies a fine-grained QoS system, which is often contrasted with DiffServ coarse-grained control system. The idea of IntServ is that every router in the system implements IntServ and every application that requires some kind of guarantees has to make an individual reservation.

2. Quality of Service

2.1 QoS Concepts

Fundamentally, QoS enables you to provide better service to certain flows. This is done by either raising the priority of a flow or limiting the priority of another flow. When using congestion-management tools, you try to raise the priority of a flow by queuing and servicing queues in different ways. The queue management tool used for congestion avoidance raises priority by dropping lower-priority flows before higher-priority flows. Policing and shaping provide priority to a flow by limiting the throughput of other flows. Link efficiency tools limit large flows to show a preference for small flows [1].

Cisco IOS QoS for instance, is a tool box, and many tools can accomplish the same result [1]. A simple analogy comes from the need to tighten a bolt: You can tighten a bolt with pliers or with a wrench. Both are equally effective, but these are different tools. This is the same with QoS tools. You will find that results can be accomplished using different QoS tools. QoS tools can help alleviate most congestion problems. However, many times there is just too much traffic for the bandwidth supplied. In such cases, QoS is merely a bandage. A simple analogy comes from pouring syrup into a bottle. Syrup can be poured from one container into another container at or below the size of the spout. If the amount poured is greater than the size of the spout, syrup is wasted. However, you can use a funnel to catch syrup pouring at a rate greater than the size of the spout. This allows you to pour more than what the spout can take, while still not wasting the syrup. However, consistent over pouring will eventually fill and overflow the funnel [1].

2.2 Basic QoS Architecture

The basic architecture introduces the three fundamental pieces for QoS implementation (see Figure 2.1). The QoS identification and marking techniques for coordinating QoS is from end to end between network elements. Within a single network element, it does have queuing, scheduling and traffic-shaping tools. The QoS policy, management and accounting functions are to control and administer end-to-end traffic across a network [5].

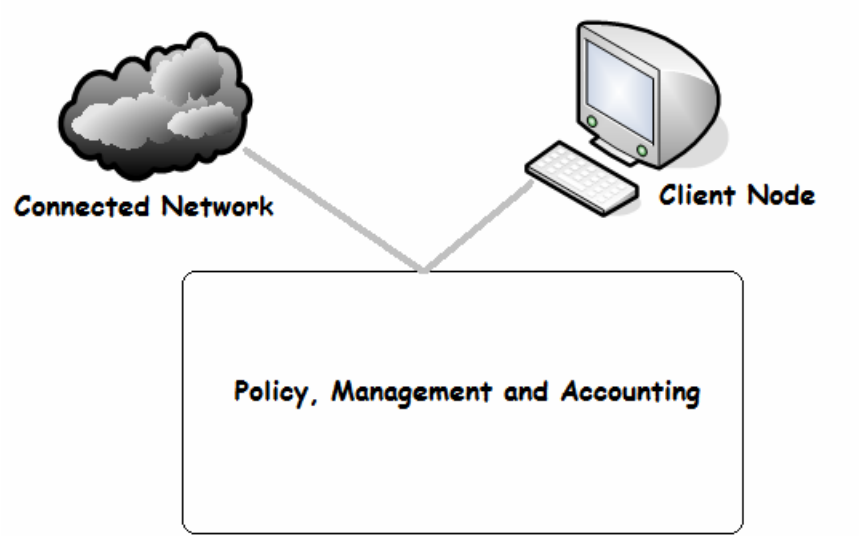


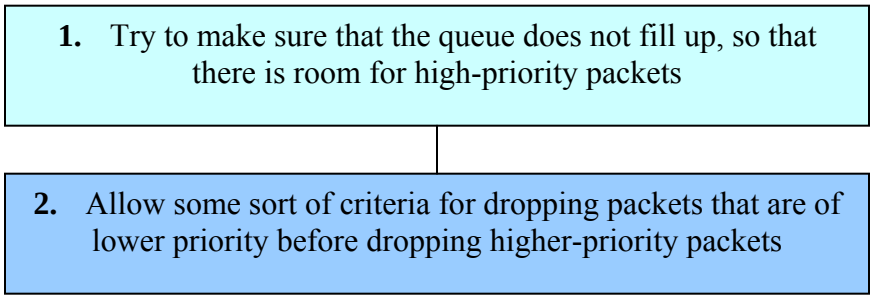
Figure 2.1: Basic QoS Implementation with 3 main components

When come to classification. It to provide preferential service to a type of traffic, it must first be identified. Second, the packet may or may not be marked. These two tasks make up classification. When the packet is identified but not marked, classification is said to be on a per-hop basis. This is when the classification pertains only to the device that it is on, not passed to the next router. This happens with priority queuing (PQ) and custom queuing (CQ). When packets are marked for network-wide use, IP precedence bits can be set (see the section "IP Precedence: Signaling Differentiated QoS") [3] [4]. Common methods of identifying flows include access control lists (ACLs), policy-based routing, committed access rate (CAR), and network-based application recognition (NBAR).

2.3 Congestion and Queue Management

Congestion management, queue management, link efficiency, and shaping/policing tools provide QoS within a single network element [4]. Because of the busty nature of voice/video/data traffic, sometimes the amount of traffic exceeds the speed of a link. Tools include priority queuing (PQ), custom queuing (CQ), weighted fair queuing (WFQ), and class-based weighted fair queuing (CBWFQ). Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and will be dropped. This is a tail drop. The issue with

tail drops is that the router cannot prevent this packet from being dropped even if it is a high-priority packet. So, a mechanism is necessary to do two things:



2.4 Applications requiring QoS

A defined Quality of Service may be required for certain types of network traffic, for example:

- ✓ Streaming multimedia may require guaranteed throughput
- ✓ IP telephony or Voice over IP (VOIP) may require strict limits on jitter and delay
- ✓ Video Teleconferencing (VTC) requires low jitter
- ✓ Alarm signalling (eg. Burglar alarm)
- ✓ Dedicated link emulation requires both guaranteed throughput and imposes limits on maximum delay and jitter
- ✓ A safety-critical application, such as remote surgery may require a guaranteed level of availability (this is also called hard QoS).

These types of service are called *inelastic*, meaning that they require a certain level of bandwidth or a minimum of delay to function - any more than required is unused, and any less will render the service non-functioning. Overall, *elastic* applications can take advantage of however much or little bandwidth is available. For instance, a remote system administrator may want to prioritize variable, and usually small, amounts of SSH traffic to ensure a responsive session even over a heavily-laden link [17].

2.5 Protocols that provide QoS [17]

- ✓ Differentiated services (DiffServ)
- ✓ Frame relay
- ✓ X.25
- ✓ Some ADSL modems
- ✓ Integrated services (IntServ)
- ✓ Resource reSerVation Protocol (RSVP)
- ✓ RSVP-TE
- ✓ Asynchronous Transfer Mode (ATM)
- ✓ Multiprotocol Label Switching (MPLS) provides eight QoS classes
- ✓ IEEE 802.1p
- ✓ IEEE 802.11e
- ✓ IEEE 802.11p
- ✓ The Type of Service (TOS) field in the IP header

2.6 QoS mechanisms

Quality of Service (QoS) can be provided by generously over-provisioning a network so that interior links are considerably faster than access links. This approach is relatively simple, and may be economically feasible for broadband networks with predictable and light traffic loads. The performance is reasonable for many applications, particularly those capable of tolerating high jitter, such as deeply-buffered video downloads. Commercial VoIP services are often competitive with traditional telephone service in terms of call quality even though QoS mechanisms are usually not in use on the user's connection to his ISP and the VoIP provider's connection to a different ISP. Under high load conditions, however, VoIP quality degrades to cell-phone quality or worse. The mathematics of packet traffic indicates that a network with QoS can handle four times as many calls with tight jitter requirements as one without QoS. The amount of over-provisioning in interior links required to replace QoS depends on the number of users and their traffic demands. As the Internet now services close to a billion users, there is little possibility that over-provisioning can eliminate the need for QoS when VoIP becomes more commonplace.

For narrowband networks more typical of enterprises and local governments, however, the costs of bandwidth can be substantial and over provisioning is hard to justify. In these situations, two distinctly different philosophies were developed to engineer preferential treatment for packets which require it. Early work used the "Integrated Service" philosophy of reserving network resources. In this model, applications used the Resource reservation protocol (RSVP) to request and reserve resources through a network. While IntServ mechanisms do work, it was realized that in a broadband network typical of a larger service provider, Core routers would be required to accept, maintain, and tear down thousands or possibly tens of thousands of reservations. It was believed that this approach would not scale with the growth of the Internet, and in any event was antithetical to the notion of designing networks so that Core routers do little more than simply switch packets at the highest possible rates.

The second and currently accepted approach is "Differentiated Services". In the DiffServ model, packets are marked according to the type of service they need. In response to these markings, routers and switches use various queuing strategies to tailor performance to requirements. Routers supporting DiffServ use multiple queues for packets awaiting transmission from bandwidth constrained interfaces. Router vendors provide different capabilities for configuring this behavior, to include the number of queues supported, the relative priorities of queues, and bandwidth reserved for each queue. In practice, when a packet must be forwarded from an interface with queuing, packets requiring low jitter are given priority over packets in other queues.

Additional bandwidth management mechanisms may be used to further engineer performance, to include:

- ✓ Traffic shaping (rate limiting):
  - o Token bucket
  - o Leaky bucket
- ✓ Scheduling algorithms:
  - o Weighted fair queuing (WFQ)
  - o Class based weighted fair queuing

- Weighted round robin (WRR)
- Deficit weighted round robin (DWRR)

✓ congestion avoidance:

- RED, WRED - Lessens the possibility of port queue buffer tail-drops and this lowers the likelihood of TCP global synchronization
- Policing (marking/dropping the packet in excess of the committed traffic rate and burst size)
- Explicit congestion notification
- Buffer tuning

As mentioned, while DiffServ is used in many sophisticated enterprise networks, it has not been widely deployed in the Internet. Internet peering arrangements are already complex, and there appears to be no enthusiasm among providers for supporting QoS across peering connections, or agreement about what policies should be supported in order to do so.

3. Resource Management in *IntServ* / *DiffServ*

3.1 Overview

QoS is the classification of packets for the purpose of treating certain classes or flows of packets in a particular way as compared to the other packets. In the current Internet architecture, a large percentage of the traffic is either multimedia related or a form of real time data that is critical to an application. Typical applications include Voice over IP (VoIP) and video conferencing. These time-critical data require some level of Quality of Service (QoS) guarantee. Various solutions have been proposed to address this problem. These include integrated services (e.g. RSVP), differentiated services and Multi Protocol Label Switching (MPLS).

3.2 Reservation Protocol (RSVP)

Integrated Services are using Reservation Protocol (RSVP) for it resource management. RSVP can be used to communicate with the forwarding network nodes to reserve Quality of Service in advance [7]. It is used for QoS signaling and actual QoS requests is define separately. These reservations are associated with one or multiple data streams. Thereby the nodes are able to schedule the packets accordingly and to reject reservations which exceed their capabilities. By the fact that RSVP messages are completely separated from the actual traffic and that they are issued by the client, RSVP is a highly flexible protocol. Each client can make exact specifications for each transmission, in advance, and choose a network path which grants his QoS. RSVP is client-oriented and session-oriented [6].

In the DiffServ model a packet's "class" can be marked directly in the packet, which contrasts with the IntServ model where a signaling protocol is required to tell the routers which flows of packets requires special QoS treatment [6]. DiffServ achieves better QoS scalability, while IntServ provides a tighter QoS mechanism for real-time traffic. These approaches can be complimentary and are not mutually exclusive. It provides a way to deliver the end-to-end Quality of Service (QoS) that real-time applications require by explicitly managing network resources to provide QoS to specific user packet streams. It uses "resource reservation" and "admission control" mechanisms

as key building blocks to establish and maintain QoS. IntServ uses Resource Reservation Protocol (RSVP) to explicitly signal the QoS needs of an application's traffic along the devices in the end-to-end path through the network. If every network device along the path can reserve the necessary bandwidth, the originating application can begin transmitting. Besides end-to-end signaling, IntServ requires several functions on routers and switches along the path [7]:

- ✓ Admission Control to determine whether a new flow can be granted the requested QoS without impacting existing reservations;
- ✓ Classification to recognize packets that need particular levels of QoS;
- ✓ Policing to take action, including possibly dropping packets, when traffic does not conform to its specified characteristics; and
- ✓ Queuing and Scheduling to forward packets according to the QoS requests that have been granted.

3.3 Network Layer Functionality

In order to allow a client to make a reservation, the client must first know, which servers are available. For this purpose, every sender implementing RSVP periodically sends “Path” messages in the network. These messages are forwarded downstream by the switches and routers along the way. A “Path” message will contain the unicast IP address of the previous node and some additional information as below [7]:

AdSpec	The optional Advertisement Specification is used in a reservation model called “One Pass With Advertising” (OPWA). In this model, the “Path” messages gather information that the receiver can use to predict end-to end service.
FilterSpec	The Filter Specification in a “Path” message must allow to select without ambiguousness the senders packets from others in the same session on the same link. It will contain the sender

	IP address and TCP/UDP port.
TSpec	Transmit Token Bucket - The Traffic specification of the sender. It defines the traffic characteristics of the traffic flow generated by the sender. The necessary QoS required to receive data from this server will be encoded here. The TSpec is used to prevent over-reservation and some other failure situations.

Table 3.1: Additional information in the "Path" message

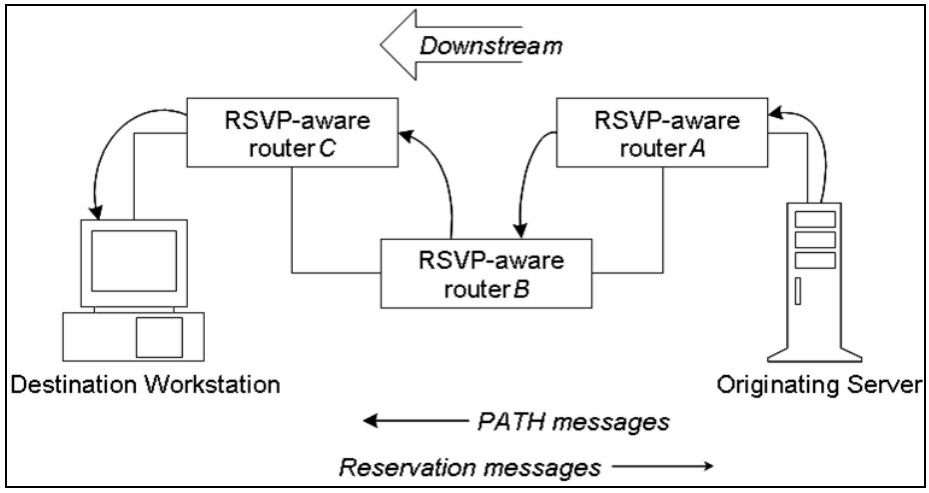


Figure 3.1: The basic convey of RSVP message between server and client

When the "Path" message finally arrives at the client, the client can be sure that every intermediate node has built up the necessary routing information to forward its requests to the server. Also, the client has been informed of an available service. If the client decides that he wants a transmission from the server and that the client can meet the necessary QoS requirements. The client sends a "Resv" request. This request is being forwarded upstream by the nodes to the sender. During this process, requests from different clients may merge into a larger reservation when they are using the same link. Both "Path" and "Resv" messages must contain a timeout value. If these are not refreshed in a constant rate, the network expects the service to have ceased. This relieves the nodes from having to handle connection failures and drop-outs [7].

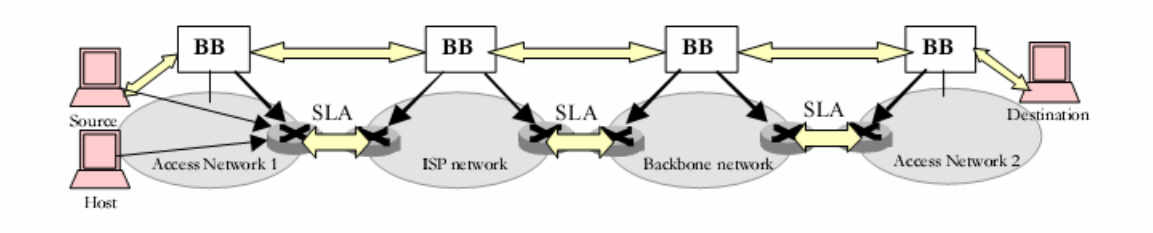


Figure 3.2 Two-tier resources Management Model for Differentiated Services Networks

3.4 Reservation Styles

The FilterSpec of such a "Resv" message may match streams from multiple servers. Depending on the desired treatment of each stream and the distribution of servers in hierarchical networks, different reservation styles are possible as table below:

Sender Aggregation:	Distinct	Shared
Explicit Sender Selection:	Fixed-Filter (FF)	Shared-Explicit (SE)
Wildcard Sender Selection:	(not specified)	Wildcard-Filter (WF)

Table 3.2: Various reservation styles

The reservation may contain an explicit list of IP addresses, one for each sender. However, if the senders are located within the same subnet, a local area network for example, wildcards may be used. Another distinction can be made between *distinct* and *shared* reservations: In a distinct reservation, each server is associated with a separate reservation specification. In a shared reservation, the streams from multiple servers are merged and receive an overall regulation. This discrimination leads to three possible reservation styles [7].

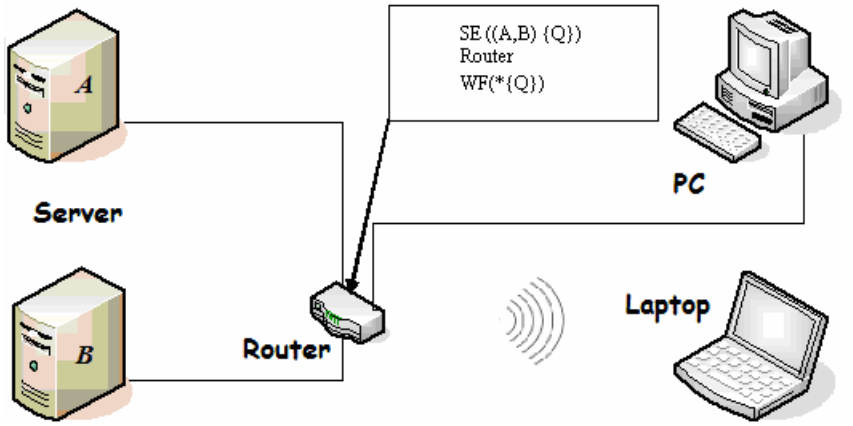


Figure 3.3: A fixed-filter style

Combining an explicit selection with a distinct aggregation leads to the most common reservation style: A *fixed-filter* style is always useful when servers may not be grouped together and you cannot make any estimation about interdependencies in their sending behavior [7]. An example would be a global video conference, gathering IP addresses from all over the globe. Also, during video transmission, a constant stream of data has to be sent all the time. In a fixed-filter reservation, the QoS requirement for a reservation may be easily computed as the sum of the separate RSpecs. Another typical situation in which advance reservation may be used is audio communication. At first sight, one would expect a similar behavior but with lower bandwidth requirements. Vocal communication between humans however has an interesting relation: As rarely more than one speaker talks simultaneously, the other streams are idle most of the time. This

phenomenon can be used for saving QoS, which is then available for other users. Sharing a reservation specification between multiple senders is possible within RSVP.

3.5 Node Level Functionality

Before that a reservation is accepted, every node performs two kinds of checks: *Policy Control* verifies that the client is allowed to communicate via this node and to receive the requested information. *Admission Control* checks whether the machine is at all capable to provide the QoS requirements defined in the reservation. If both controls are fined, parameters are set in the Classifier and the Scheduler to allow the data transfer. If not, a negative answer is given to the routers and clients downstream [7].

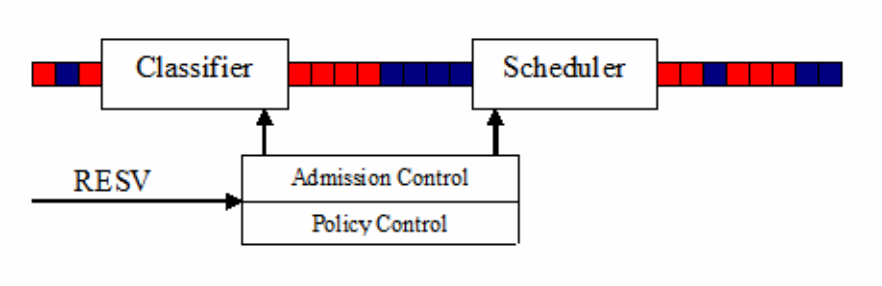


Figure 3.4: Actual moves of Classifier and the Scheduler

The Classifiers task is to identify incoming packets and to match them with the existing reservations. For this purpose, also "Resv" messages contain a FilterSpec similar to the one sent out by the sender. Identified packets are then scheduled in a way that should allow every QoS reservation to be fulfilled. This is the task of the scheduler. Non-matched packets are handled using Best Effort Delivery 4. QoS requirements are encoded in the "Resv" message's Reservation Specification (RSpec) and Traffic Specification (TSpec) [7].

3.6 RSVP in Practical

Even though RSVP provides interesting concepts for advance reservation, it was not able to propagate widely. This is due to some severe drawbacks: As every reservation specifies its own QoS requirements, classifying and scheduling the packets has become an immense task [7]. Servers in a large network or even an Internet-style network have to handle several thousand of clients at once. If incoming packets would have to be filtered

through a thousand of filters, and then scheduled in concurrence with the same amount of requests, the management would require much too much effort to allow any of them to be delivered at time. It must also be considered that every node on the path groups different reservations and thus has to do as much work. There are two typical user groups for whom RSVP was intended: Inter-domain traffic engineering and real-time streaming applications. RSVP has proved unsuitable for the first one, but attempts are made to integrate it in IP Telephony applications. The success is unclear, as sender-initiated transmissions are also imaginable.

4. QoS: Problems using *IntServ* and *DiffServ*

4.1 The Concepts

In *IntServ*, this is an architecture which specifies the elements in order to have guaranteed quality of service (QoS), as used for transmission of video and speech without interrupt. The main idea for using integrated service is that every router in the networking system implements the service, and individual reservation has to be made to obtain guaranty. Two key terms for the integrated service are “Flow Specs” which describes the reservation, and “RSVP” as the underlying mechanism to signal it across the network.

In *DiffServ*, this is an architecture that specifies a simple, scalable and coarse-grained mechanism for classifying, managing network traffic and providing quality of service (QoS) guarantees on modern IP networks. It can be used to provide low-latency, guaranteed service to critical network traffic such as voice or video while providing simple best-effort traffic guarantees to non-critical services such as web traffic or file transfer [13].

4.2 The Problems

In *IntServ*, The very first problem come into this topic would be the complexity in the routers. The RSVP protocol is quite complex and consumes quite a lot of memory spaces, and also the processing capabilities on the network routers. The routing became more complicated due to the packet classification and scheduling functions. In order to have QoS service using RSVP signaling technique, those functions have to be implemented in all routers [8].

Then, the scale of the network which limited only to that of small scaled ones has made the application less popular. Many states must be stored in each router and number of flows in the backbone may be large. As a result, integrated service works on a small-scale, but as you scale up to a system the size of the Internet, it is difficult to keep track of all of the reservations [10].

Other than that, the application of integrated service need a concept of “Virtual Paths” or aggregated flow groups for the backbone. First, the application must

characterize its traffic source and the resource requirements. The network then uses a routing protocol to find a path based on the requested resources. Next reservation protocol is used to install the reservation state along that path. At each hop admission control checks whether sufficient resources are available to accept the new reservation. Once the reservation is established, the application can start to send traffic over the path for which it has exclusive use of the resources [9].

Another matter to be paid attention which make the integrated service troublesome is that policy controls are needed, in terms of reservation, security and accounting. For reservation, an administrator is needed to decide who can make reservation [10], who is the important client in the network has always to be admitted. For the aspect of security, an integrated service network often converges the important data of different enterprises [11]. This requires the carrier to be powerful in secure the data. Lastly, for the accounting control, the model for usage feedback, another important issue in integrated service, as the prevention to abuse of network resources [12].

In *DiffServ*, the problem arises as the designing of end-to-end services with weighted guarantees at individual hops is difficult [10]. The end-to-end behavior of differentiated service is not predictable as the details on the method of a router dealing with the type of service field are not fixed. It is more complex when a packet gone across more differentiated cloud before reaching its destination [13]. The Peering problem is a problem arose due to unassociated services that vary provided by different Internet operator. Internet operator could have fixed the classes of end-to-end connectivity by enforcing a standardized policy across the network. However, it is not in the preference to make the complex peering agreement more complicated [13].

For knowledge sharing, peering is voluntary interconnection of administratively separate Internet networks for the purpose of exchanging traffic between the customers of each network. The pure definition of peering is settlement-free or "sender keeps all," meaning that neither party pays the other for the exchanged traffic; instead, each derives revenue from its own customers. Marketing and commercial pressures have led to the word peering routinely being used when there is some settlement involved, even though that is not the accurate technical use of the word. The phrase "settlement-free peering" is

sometimes used to reflect this reality and unambiguously describe the pure cost-free peering situation.

Peering requires physical interconnection of the networks, an exchange of routing information through the Border Gateway Protocol (BGP) routing protocol and is often accompanied by peering agreements of varying formality, from "handshake" to thick contracts [14]. On the other hand, QoS is for the aggregate and not micro-flows. Individual packet flows are aggregated to one large flow which is then treated uniformly [15]. Thus, large numbers of short flows are better handled by aggregates. However, it is not suitable for long or high flows applications like voice and video session as per-flow guarantees are needed [10].

Lastly, open loop control approaches used by IETF is kind of undesirable. Data packets are dropped according to different level of different levels of best-effort service at times of network congestion [16], rather than waiting for feedback at the source [10]. This may lead to inefficiency of channel utilization and no assurance for data transmission.

5. *IntServ* and *DiffServ* Architectures

5.1 *IntServ* Architectures

The architecture described 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 characterized by parameters such as bandwidth, packet delay and packet loss. 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 [23].

5.2 *IntServ* Model

The 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, example 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. Furthermore, the Integrated Service model consists of a set of service commitments, related to the service requests. The network commits to deliver service either to individual flows or to collective-aggregate flows [22]. In order to avoid the danger of failures in providing the agreed service, IS model provides several scenarios where the traffic control is provided implicitly by dropping the packets which are marked as pre-emptily, example, less valuable packets within a flow.

5.3 Implementation Reference Model

For realization of the Integrated Services model the Implementation Reference model defines several mechanisms that encompass the layer 3 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, i.e. the service that determines the way the scheduler should handle them [22]. 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 simplified RSVP/IntServ framework is shown in Figure 5.1. As it is shown every RSVP aware router in the IntServ will perform RSVP signaling, admission control, scheduling and policing [23].

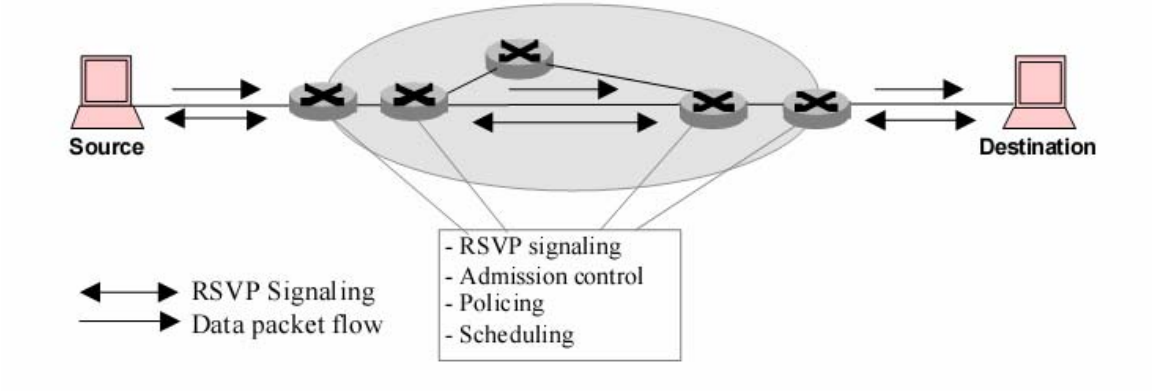


Figure 5.1 RSVP/IntServ frameworks

5.4 *DiffServ* Architecture

The Differentiated Services (DiffServ) architecture 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 Behavior (PHB) as main building blocks. At each node packets are handled according to the PHB invoked by the DS byte in the packet header [22]. The

DiffServ divides the entire network into domains, where DiffServ domain is a contiguous set of nodes which operate with a common set of service provisioning policies and PHB definitions [22]. 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 neighboring boundaries domains.

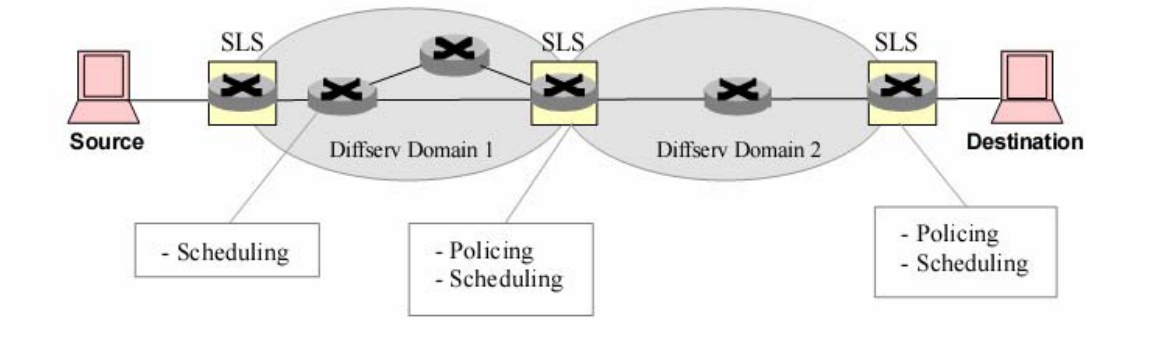


Figure 5.2 DiffServ Framework

5.5 IntServ over DiffServ

As described above both *IntServ* and *DiffServ* 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 Integrated Services operation over Differentiated Services 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 [23].

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. On the other hand DiffServ itself will profit by using RSVP as a mechanism to properly provision quantitative services across the networks [23]:

- In DiffServ, admission control is applied in a relatively static way by provisioning policing parameters at network elements. Using RSVP will enable DiffServ to apply resource-based admission control, which will optimize the use of resources in the network. Furthermore, it will enable DiffServ to apply policy-based admission control

on users/applications traffic.

- In DiffServ the DSCP code point can be set either at the host or at the router and by means of an explicit mechanism such as RSVP DCLASS , DiffServ will be able to perform traffic identification/classification straightforwardly.
- IntServ network elements perform per-flow traffic conditioning. Pre-conditioning traffic in this way before they enter DiffServ, enhances the ability of DiffServ to provide quantitative services using aggregate traffic control.

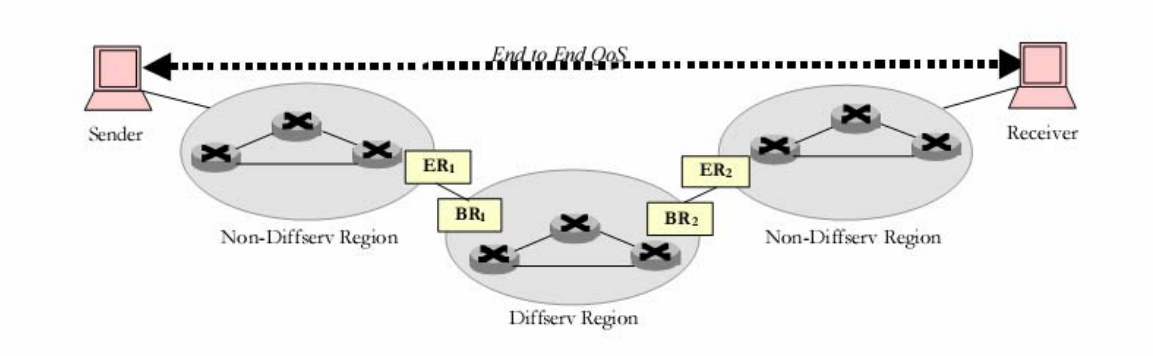


Figure 5.3.: The reference network for the IntServ/RSVP over DiffServ framework

As shown above the framework consists of the following elements:

- Sender and Receiver are RSVP aware hosts, which can initiate an RSVP process based on the requirements of QoS-aware applications running on the hosts.
- Edge and Border routers are the edge devices whose functionality depends on the specific realization of the framework.
- Non-DiffServ region is an IntServ capable region containing hosts and other network elements that are IntServ capable and also other network elements.
- DiffServ network region may or may not contain RSVP-aware routers.

6. Utilization bounds of *IntServ* vs. *DiffServ*

6.1 Background

To guarantee a required Quality of Service when transporting multimedia is quite a challenge which has receive a lot of attention in recent years [24].To overcome this issue to approaches have been developed which is IntServ and DiffServ. IntServ which is also known as "hard QoS" makes strict bandwidth reservations where as DiffServ as the name suggests simply differentiates between multiple traffic flow. Basically IntServ reserves network resources between individual flows while DiffServ provisions network resources.

Engineers were faced with challenges to develop algorithms that would enable efficient traffic control while allocating or provisioning resources for both IntServ and DiffServ since the issues of VBR (variable bit rate) video is very sensitive to delay and degree of burrstones is quite high [24]. IntServ and DiffServ offer different class of Services. IntServ by right should have a better deterministic guarantee as compared to DiffServ however DiffServ should have bandwidth utilization and this is what our topic of discussion is about. Deterministic bandwidth and resource reservation is achieved by IntServ through admission control however the down side is that a lot of bandwidth gets wasted. DiffServ give priority based on service class and this gives us better bandwidth utilization as compared to DiffServ but the down side is it has a higher degree of uncertainty since no admission control is performed .A higher degree of uncertainty here translates to we being unable to guarantee the flow since maximum delay and jitter will be harder to calculate and packet losses become more apparent.

Our objective is to compare how much utilization of bandwidth can be achieved by IntServ and DiffServ approach by using network calculus and realistic workload. Network calculus has shown us that IntServ cannot guarantee us a utilization of more then 40% but when network calculus was applied on DiffServ to our surprise achieves less then that and from here we can question the validity of the EF's utilization bound provided by network calculus. This bring us to our second objective is to use simulation results to investigate the accuracy of the formulas provided by the network calculus.

6.2 Results using Network Calculus

6.2.1 GPS scheduling and *IntServ*

The per flow management approach of the workload that meets the QoS requirements is used by IntServ. Its based on admission control test, a RSVP which reserves the bandwidth that travel the path which one connection crosses and GPS scheduling of different flows [1]. *Parekh* and *Gallager* introduces an equation $\alpha(t) = \sigma + \rho(t)$ where σ is the bucket depth and $\rho(t)$ is the drain rate. This is used to calculate maximum delay when traffic is leaky bucket regulated at the input [24].

$$Di = \sigma + nLi/R + \Sigma n \; Lmax/Cj \qquad (1)$$

For session i, L_i is the maximum packet size for session I and Lmaxj is the maximum packet size in node j. , Cjth bandwidth of the link j, and n the number of nodes. In order to simplify this delay expression, we can use the Ctot and Dtot. Ctot is nLi and D_{tot} is Σn Lmax/Cj. When bandwidth reservation R is smaller than the peak reservation p, equation 1 is used $R \leq p$ but when the bandwidth reservation is bigger we use another equation. The delay equations are [24],

$$D = (\sigma + Ctot)(Dtot)/R \qquad R \leq p \qquad (2)$$

$$D = (M + Ctot)(Dtot)/R \qquad R \geq p \geq \rho \qquad (3)$$

For a new connection I with a given end-to-end delay Di, it is necessary to calculate the bandwidth reservation that makes the equation 2 or 3 less than D, and the sum of the bandwidth for all channels at the node less than the total link bandwidth C_j.

6.2.2 *DiffServ* and EF traffic

DiffServ is based on traffic aggregates [24]. The aggregate that crosses a router belongs to a (PHB). Per hop Behaviour indicates the behavior of individual routers rather then end to end services. The 2 types of PHB ‘s are Expedited Forwarding (EF) and Assured Forwarding(AF).EF ‘s arrival rate is based is limited by the link speed of the router and it can be implemented wit strict priority over other packets and should be

forwarded wit and should be forwarded with minimum delay and loss. Although AF offers better then best effort basis EF can offer services that meet QoS requirements. Le bounded provided us an equation to obtain the maximum delay when each micro flow of the aggregate leaky-bucket regulated at the input [24]. In this formula the end to end delay bound is a function of the traffic aggregate characteristics and the maximum number of hops the connection can cross. The equation for the delay bound is:

$$D_I = e + \tau / 1 - (h - 1)\, \nu \qquad (4)$$

e is the latency and (h-1) is a bound on the number of hops used by any flow [24].

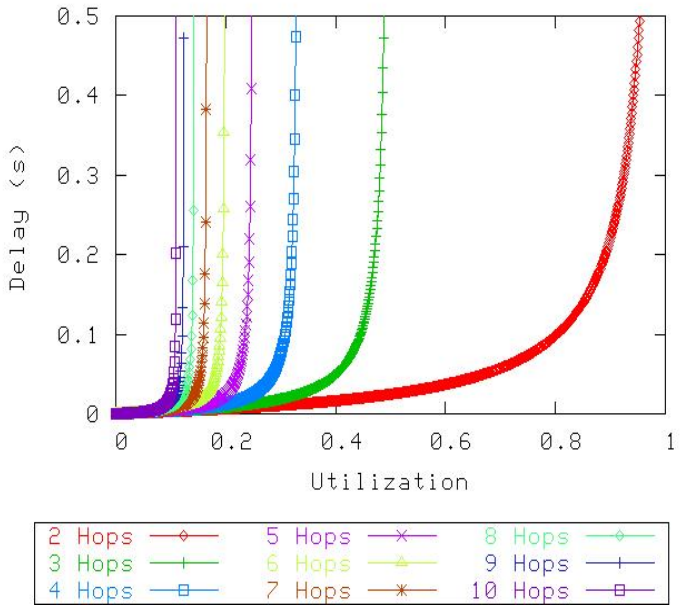


Figure 6.1: Bound vs. Utilization factor

Figure 6.1 shows how varies the bound D (vs. the utilization factor) in Equation (4) with the number of hops (H). The parameters of the simulation are,

a) e = 2 1500B/rm'

b) Lmax=1000b

c) σi= 100B for all flows

d) ρ_i= 32kb/s for all flows

e) rm = 149, 760M b/s

f) $Cm = +\infty$

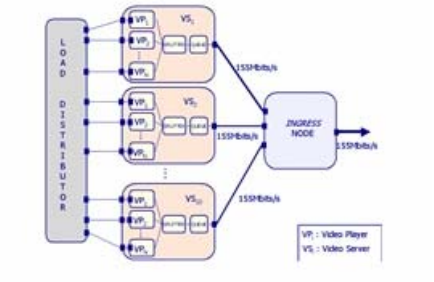
For 10 hops the bound is valid for small utilization factors but does not mean the worst case delay goes to infinity.[1]In general cases network is unbounded however in some cases like a unidirectional ring there is a finite bound when v<1. When we restrict the admitted workload it is possible to guarantee a maximum delay bound for an aggregate despite DiffServ does not employ admission control. For a new connection with an end to-end delay $Dmax_i$it is necessary to calculate D_1 (adding the terms (σi, ρi) to τ and ν respectively) and in order to guarantee all the flows [24].

$$\min(Dmaxi) \geq hDi \qquad (5)$$

6.3 Real Traffic Comparison

Traffic is labelled as EF so as to have low delay, low jitter and low loss rate. Ingress node will schedule this traffic at the highest priority and the only limit to the traffic aggregate is that the node will start dropping packets when the arrival rate is higher than the link capacity. Workload generated by the RTNOU simulator in the as an input file to the DUSTIN simulation tool, before it pass through the admission control test. This makes that the offered workload is not guarantee but we are able to reproduce the same scenario of the first experiment and making the results comparable [24].

Figure 6.2: Load description file



7. Dimension of QoS: Delay Handling and Jitter

7.1 Discussion

There are basically five dimensions to QoS that impact the end-user experience. There are *Availability*, *Throughput*, *Delay or latency*, *Delay variation*, including *jitter* and *Losses*. For our case here, we will discuss about the *Delay and Jitter.*

Delay or latency is the average transit time of a service from the ingress to the egress point of the network. Many services, especially real-time services such as voice and video communications are highly intolerant of delay [21]. Delay variation is the difference in delay exhibited by different packets that are part of the same traffic flow. High frequency delay variation is known as *jitter*, while low-frequency delay variation is called wander. Jitter is caused primarily by differences in queue wait times for consecutive packets in a flow, and is the most significant issue for QoS. Certain traffic types – especially real-time traffic such as voice and video – are very intolerant of jitter. Differences in packet arrival times cause choppiness in the voice or video. All transport systems exhibit some jitter. As long as jitter falls within defined tolerances, it does not impact service quality. Excessive jitter can be overcome by buffering, but this increases delay, which can cause other problems [21].

Application	Availability	Throughput	Delay	Jitter	Loss
Circuit Services	High	High	High	High	High
Interactive Video	High	High	High	High	Medium
Voice Telephony	High	Low	High	High	Low
Broadcast Video	High	High	Low	Medium	Medium
SNA	High	Medium	Medium	Low	High
File Transfer	Medium	Low	Low	Low	High
E-mail	Low	Low	Low	Low	High

Table 7.1: The following table shows the sensitivity of different applications to these QoS attributes

Internet, it lacked of the ability to provide QoS guarantees due to limits in router computing power. It therefore ran at default QoS level, or "best effort". There were four "Type of Service" bits and three "Precedence" bits provided in each message, but they were ignored. These bits were later re-defined as Differentiated Service Code Points (DSCP) and are largely honored in peered links on the modern Internet [17]. When looking at circuit-switched networks, Quality of service is affected by various factors, which can be divided into "human" and "technical" factors. Human factors include: stability of service, availability of service, delays, user information. Technical factors include: reliability, scalability, effectiveness, maintainability, Grade of Service and etc. Many things can happen to packets as they travel from origin to destination, resulting in the following problems as seen from the point of view of the sender and receiver [17]:

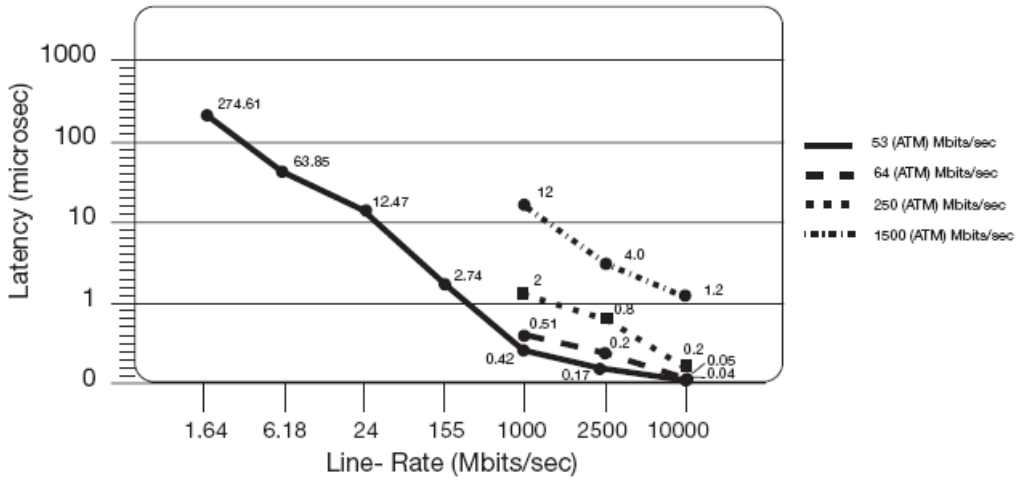


Figure 7.1: Packet Latency vs. Line Rates

Dropped packets

The routers might fail to deliver (*drop*) some packets if they arrive when their buffers are already full. Some, none, or all of the packets might be dropped, depending on the state of the network, and it is impossible to determine what will happen in advance. The receiving application may ask for this information to be retransmitted, possibly causing severe delays in the overall transmission [17].

Delay

It might take a long time for a packet to reach its destination, because it gets held up in long queues, or takes a less direct route to avoid congestion. In some cases, excessive delay can render an application, such as VoIP, unusable [17].

Jitter

Packets from source will reach the destination with different delays. A packet's delay varies with its position in the queues of the routers along the path between source and destination and this position can vary unpredictably. This variation in delay is known as jitter and can seriously affect the quality of streaming audio and/or video [17].

Out-of-order delivery

When a collection of related packets is routed through the Internet, different packets may take different routes, each resulting in a different delay. The result is that the packets arrive in a different order than they were sent. This problem necessitates special additional protocols responsible for rearranging out-of-order packets to an isochronous state once they reach their destination [17].

Error

Sometimes packets are misdirected, or combined together, or corrupted, while en route. The receiver has to detect this and, just as if the packet was dropped and ask the sender to repeat it [17].

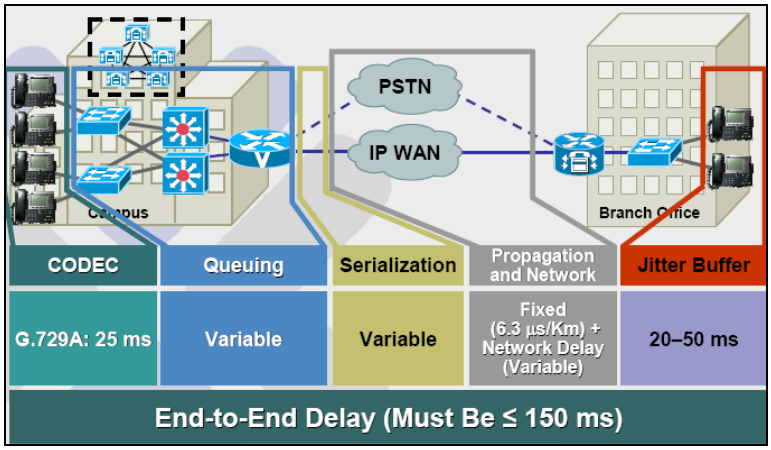


Figure 7.2: QoS Requirement – Elements that affect lantency and jitter

7.2 Example: Low-Complexity Handling of Delay Jitter for VoIP

Using data networks to transport digitized voice in data packets has changed the way we communicate. The advantages of this approach, such as cost savings and ease of network management, have convinced even the skeptics [18]. In some cases, data networks with digitized audio in data packets provide the entire audio communication sys tem for a business or institution. Voice over Internet Protocol (VoIP) has not only transformed the way we communicate; it has also brought with it many possibilities that we never previously thought possible. Along with advantages and new possibilities, new technology also brings new challenges. Delay jitter is one of the major challenges that affect the quality of service (QoS) in VoIP. We discuss methods for handling delay jitter in VoIP. We propose a practical, effective, low-complexity technique for handling delay jitter. Our technique targets the business environment, which commonly uses embedded digital devices in company telephone systems. The data network for such a system could range from a simple local area network (LAN) for a small office with a few users to a complex wide-area network (WAN) for a large company with many users to telecommuting over a general public network such as the Internet [18].

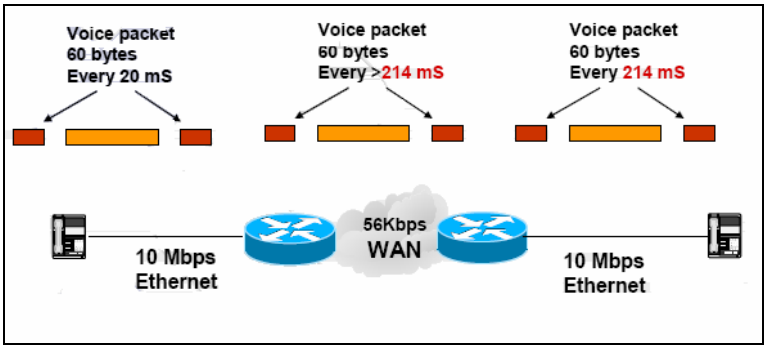


Figure 7.3: Jitter for voice packet

8. End To End Delay & Jitter In *DiffServ*

8.1 Overview

An end to end packet delay jitter has a negative impact on the offered QoS in IP networks. Therefore, in this paper we clarify this passive impact, and discuss the delay jitter that is based on the analysis done in [25]. However, we focus on the expedited forwarding (EF) class in the differentiated services network (DiffServ). Therefore, playout buffers must be added to the DiffServ network for handling the EF delay jitter problem.

Nowadays the networks with guaranteed quality of service (QoS) are greatly paid attention. These networks will offer alternatives to the existent ones, which offer a single service class called best effort. An example of such networks is the Internet network, where the end-users have no guarantee on the quality of their required services. Therefore, service models such as asynchronous transfer mode (ATM), integrated service (IntServ), and DiffServ have been developed to support new service classes with varying traffic characteristics and QoS requirements. Recently, the efforts in the world have been intensified on redefining, and improving the design of the DiffServ, so that it supports better the QoS of the supported service classes. Hence, the new demand of QoS required for real time applications, such as video streaming, audio, and IP telephony are abstracted in the real time service classes offered by these service models [25].

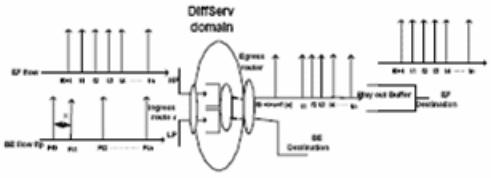


Figure 8.1: Delay Jitter in DiffServ

In DiffServ network as shown in Figure 8.1; EF flows are statistically multiplexed with the BE flows. Thereby, packets belong to the EF flows experience different individual delays at the servers along their paths toward their destinations, which cause distortion in the timing sequence of the real time applications flows serviced as EF flows,

arising from their packets delay variations. This can be measured at each node traversed by EF flow through computing the difference between the inter departure and inter arrival times of two consecutive packets belong to the EF flow. Also, packets delay variations can be measured by another parameter, rate jitter, which is the rate of difference between the minimum inter arrival and maximum inter arrival times. This parameter is very convenient for measuring video broadcast over the DiffServ domain, since a slight deviation of rate is translated to only a small playout delay [27].

Thus, playout buffers at the end users are needed to reorganize the time sequence of the interactive applications flows. These de-jitter buffers have been considered the main solution for compensating the packets delay variations in their flows. However, their design still causes a major challenge in dealing with the delay jitter. This challenge has been represented in two points: the first point is the playout buffer mechanism choice, so that if it is a static playout buffer, then the playout instants are chosen according to the first packet arrival in its flow, however if it is adaptive playout buffer, then the playout instants change according to the packets arrivals instants. The second point in the challenge is the playout buffer depth; for example if it is small, then this increases the packets loss probability. However, if it is long, then this adds a significant delay to the flows delay budget, which is not tolerated by the real time flows. Therefore, an optimal choice for the playout buffer's depth, and play out buffer mechanism are required. Nevertheless, the choice is an application dependent [28].

Network traffic is divided into three classes *{CEF, CAF, CBE}*, where *CEF*, and *CAF* contains a determined number of flows *{*1*,* 2*..,NEF }* and *{*1*,* 2*..,NAF }* respectively. However, the number of flows in class *CBE* is kept unknown, and they don't receive a special treatment by the DiffServ nodes. Therefore, this type of traffic can be injected by any network user at any time. Nevertheless, we have kept it under control along the simulations, to see their effects on the EF flows. Along our analysis we focus on the EF class under the varying effects of BE characteristics, burrstones and peak rate on this class.

8.2 Jitter in Single Class Service Discipline

Scheduling service disciplines (S.S.D) have been considered the main tool in reducing, and improving the delay variations of the traffic flows. Therefore, some S.S.D's have been supplied with further equipments, such as clocks for stamping the different packets. Then, the packets stamps are used by next S.S.D's along their paths to hold the delayed or earliest packets in their regulators to become eligible for transmission. This keeps in turn the flows sequence time of the different flows, which eliminates the jitter of these flows. However, there is a big gap between what have been theoretically discussed about the scheduling service disciplines, and what is really implemented in the real networks.

FIFO scheduling service discipline is mostly used in the actual networks. This service discipline is used due to its simplicity and availability. However, it causes distortion for a real time traffic class, whenever it supports this class with another one. This distortion is represented in the delay variations of packets belonging to the real traffic flows, which is explained deeply in [1].However, in this paper we adopt the results of this important reference for analyzing the delay jitter in DiffServ network that support the EF traffic class and others through FIFO S.S.D.

EF packets are given a priority higher than BG packets. Thus, EF packets are served a head of BG packets that arrive at the same time. Therefore, the delay jitter of EF packets at *mth* node equals to the difference of BG backlog times measured at two EF consecutive packets instants, if we consider the fluid queuing system. However, in this paper, we consider the discrete queuing time model, where FIFO server serves a packet each time unit. Consequently, EF delay jitter is the difference of backlogged BG packets numbers seen by two EF consecutive packets (*QnBGn*+1*,mth* ,*Qn*+1*BGn,mth*), and their inter arrival time is *In, n*+1

$$J_{EF}^{m^{th}} = Q_{BG_{n+1},m^{th}}^{n+1} - Q_{BG_n,m^{th}}^{n} + I_{n,n+1} \qquad (3)$$

Henceforth, delay jitter analysis is coincided with the analysis of the variations of the queue size at the arrival instants of EF packets. From [25], The queue size in Z-domain can be expressed at the arrival instants of EF packets as follows:

$$Q(z) = G_{EF}^{'}(1)(1-\rho_{total})\frac{(1-z^{-1})(B(z)/z)^{k}}{(1-zG_{EF}(B(z)/z)} \times \frac{\pi_{k=1}^{K}\left[((z/B(z))-(r_k/(B(r_k))\right]}{\pi_{k=1}^{K}\left[1-(r_k/B(r_k))\right]} \quad (4)$$

Where, *G' EF* (1) = (*∂GEF* (*z*)/*z*)*z*=1. *k* = *Gmax* − 2, *ρtotal* = *ρBG* + *ρEF* is the total load at the server.The inter arrival time (*I*) of EF flows has finite supports; *Gmin*, and *Gmax*, where 1 ≤ *Gmin,*≤ *Gmax* ≺ ∞. *B*(*z*) is the probability generating function (p.g.f) of the random variable corresponding to the BG batch size. And, *GEF* (*z*) is the (p.g.f) of the integer random variable *I*, which can be expressed as follows [25] :

$$G_{EF}(z) = \sum_{i=G_{min}}^{G_{max}} Pr(I=i)z^{i} = \sum_{i=G_{min}}^{G_{max}} g_i z^{i} \quad (5)$$

$$J_{EF}^{k^{th}}(Z) = \sum_{i=G_{min}}^{G_{max}} g_i J_i(z) \quad (6)$$

where,

$$J_i(z) = z(B(z))^{i} + (B(z))^{i-1}(z-1)\sum_{k=1}^{i-1}(B(z)/z)^{-k}\phi(z^{-1};k) \quad (7)$$

$$\phi(z^{-1};k) = \sum_{L=0}^{k-1} z^{L}\pi_k(0;L)Pr(Q=L) \qquad \forall 1 \le k \le i-1 \quad (8)$$

where, *πk* (0;*L*) is the probability that the queue is empty at the *Kth* time instant following the arrival of EF packet, after all the new arrivals at this time instant have been counted.It is immediate that *πk*(0;*L*) = 0 for 1 ≤ *k* ≤ *min*(*L,Gmax* − 1).Hence, *πk*(0;*L*) = *Pr*(*Q*(*k*) + *B* 0|*Q*(0) = *L*) [25].

8.3 Jitter in Multi-Class Service Discipline

Multi-class S.S.D's are often employed in multi-class networks to isolate the different traffic classes from one to another, and manage the output link capacity among the backlogged traffic flows belonging to the different traffic classes according to their QoS requirements. However, the real time traffic flows are subjected at any node in a network to be delayed or their sequence times to be distorted due to other traffic classes' characteristics.

EF flows still suffer from the problem of delay jitter as we are going to see in the simulation section. This problem can be analyzed or interpreted in the following three scenarios: First, when the packets belong to EF class 1 in the DiffServ network shown in Figure 2 arrive to the server of core router 1, while it is busy in servicing a packet belongs to BE transmitted by BE source 1 or 2.Then, as long as this server doesn't finish servicing the BE packet, other EF packets belong to other EF sources probably arrive, where two sources of delay variations can be underlined in this situation: one due to the BE packet that is being currently serviced by the server. Which, results in a worst delay jitter equivalent to $\left[\frac{L_{min}}{C_0-C_{EF}}, \frac{L_{max}}{C_0-C_{EF}}\right]$. This jitter source forms the second scenario which is similar to that we have explained in section 4.However, in this case the back ground for EF flows generated by EF source 1 are other EF packets generated by other EF sources 2,3 in the DiffServ. However, we can apply the same analysis of the previous section, but we have to change the values of *Gmax*, *Gmin* according to the following tow equations respectively:

$$G_{max} = max\left[\frac{L_{min}}{C_0-C_{EF}}, \frac{L_{max}}{C_0-C_{EF}}\right], \text{ and } G_{min} = min\left[\frac{L_{min}}{C_0-C_{EF}}, \frac{L_{max}}{C_0-C_{EF}}\right].$$

The BG distribution can be characterized by the Taylor series around $\rho BE = 0$.Hence, the EF queue size $Q(z, \rho BG)$ at the EF packets arrival instants can be characterized through the expression clarified in the Theorem 1, [28]:

Theorem 1 (Light Back Ground Traffic). *Given* $B(z,\rho) = 1 + \rho(a(z) - 1) + O(\rho)^2$,*and* $a(z) = \sum_{i=0}^{A_{max}} a_i z^i$,*then if* $A_{max} \prec G_{min} \Rightarrow \quad Q(z,\rho) = 1 + \rho\left[\frac{(a(z)-z)}{z-1}\right] + O(\rho^2)$

Where, *Amax* is the upper bound on the minimum spacing of arriving EF packets. Which, is determined by the characteristics of the BG traffic.Then, the EF flows jitter can be characterized through the expression clarified in the Theorem 2, [28]?

Theorem 2 (EF Flows Jitter). *If* $A_{max} \prec G_{min} \Rightarrow J(z) = G(z)\left[1 + \frac{\rho}{2}(f_1(z) - 1)\right] + O(\rho^2)$.*where,* $f_1(z) = \frac{a(z)-1}{z-1}$

In our simulations, we have used FTP application, who's Poisson distribution as a back ground traffic. Z-transform of Poisson distributed batches can be described as follows:

$$B(z) = e^{\rho(z-1)} \qquad (9)$$

Then, by using Taylor series we can expand this as follows: Then from theorem 1, and background Taylor expansion 10, we get a(z) = z,

$$B(z) = 1 + \rho(z-1) + \rho^2\frac{(z-1)^2}{2} + O(\rho^3) \qquad (10)$$

Then from Theorem 2, f(z) = 1, thus

$$J_{EF}(z) = G(z) + O(\rho^2) \;\; if \;\; G_{min} \succ 1 \qquad (11)$$

Where, we see no effect for the first order term appeared in expression 10. So far, we have focused on a single node analysis. Nevertheless, the previous results form the base in the analysis of jitter in a multiple nodes. Since, if we approximate the departure process of the EF traffic flows from any node in the DiffServ domain as a renewal process, the marginal distribution of the departure process of EF flows from node *k* is approximated by a renewal process with the inter-arrival time distribution identical to *Jk*(*i*).Denote the sequence *Gk*+1(*i*) as the probability of inter arrival time of EF flows entering node *k* + 1 to be *i* time service units [26].

$$G_{k+1}(i) = J_k(i) \qquad 1 \le k \le N-1 \qquad (12)$$

and in the Z-domain, we have

$$G_{k+1}(z) = J_k(z) \qquad 1 \le k \le N-1 \tag{13}$$

Consequently, once the EF arrival process at the first node of the network is periodic, then we simply have $G1(z) = ZT$ .Therefore, if we track the EF flows' jitter from their sources to destinations, then their exact marginal jitter distribution can be obtained [26].

9. End To End Delay & Jitter In *IntServ*

On factor of great importance to interactive applications is end-to-end delay. This is certainly true for real-time applications using voice or video streams. Across a network such as the Internet, the end-to-end delay is made up of many components.

Propagation delay: this is also called "speed-of-light" delay, and is a physical constraint that is linked to the propagation of a physical signal. In general, the speed of light, c is taken to be approximately 3.0*10^8 m/s, and this is often used in calculations. However, it should be noted that in copper the propagation speed of an electrical signal is nearer 2.5* 10^8 m/s, whilst in fiber the propagation speed of an optical signal is nearer 2.0* 10^8 m/s [29].

Transmission delay: this is the delay in getting bits onto the wire due to the speed of the link. Do not confuse this with propagation delay. A 1000byte packet is transmitted "faster" on a 10Mb/s line than on a 64Kb/s link, i.e. the bits are placed onto the wire quicker, but the electrical signal is subject to the same propagation delay in both cases [29].

Network element processing delay: a packet arriving at a network element may be queued an input buffer, then it will be read and processed (e.g. forwarding decisions made at routers), and finally queued to an output buffer while it waits to be transmitted [29].

End-system delay: delay may be introduced at the sender or receiver for various reasons. The input may not be processed immediately or transmitted immediately at the sender, for example due to end-system load. At the receiver, delay may be introduced in order to compensate for network affects, e.g. the use of de-jitter buffers in real-time audio tools. The end-to-end path that traffic follows across the Internet is never fixed for any application. In general, the application neither knows nor cares about the actual end-to-end path. Changes to the path occur due to the dynamic nature of the IP routing protocols, and do not forget that paths may not be symmetric delay may be asymmetric. Traffic patterns may be observed to have effects that are localized, e.g. localized congestion, as well as "time -of-day" effects [28].

End-to-end jitter

- Variation in delay:
  - per-packet delay changes
- Effects at receiver:
  - variable packet arrival rate
  - variable data rate for flow
- Non-real-time:
  - no problem
- Real-time:
  - need jitter compensation

Causes of jitter

- Media access (LAN)
- FIFO queuing:
  - no notion of a flow
  - (non-FIFO queuing)
- Traffic aggregation:
  - different applications
- Load on routers:
  - busy routers
  - localised load/congestion
- Routing:
  - dynamic path changes

Jitter is the delay variation observed at the receiver. Packets do not arrive with constant delay so the timing of packet generation at the sender is perturbed and timing needs to be re-constructed at the receiver – this is called synchronization [28]. The effects at the receiver are application dependent, but what is visible is a variable packet arrival rate, and therefore a variable data rate for the flow. This is not suitable for application such as audio which produce flows that may have a constant data rate. For non-real-time applications, jitter is typically not an issue. For real-time applications, jitter compensation needs to be applied at the receiver.

Jitter is caused by a number of factors. At the sender, use of LAN technology like Ethernet may lead to packet transmissions being unevenly spaced. In routers, the use of FIFO queuing and traffic aggregation may lead to packet spacing being perturbed. Some routers may also use non-FIFO techniques, perhaps prioritizing certain traffic and so disrupting the normal spacing of other flows. Traffic aggregation, with many different size packets sharing the same buffers/output-link may also cause jitter.

As router load increases, buffers/queues in routers may start to fill up, adding queuing delay. Very busy routers may lead to congestion, and this may lead to further delay. Where congestion or router failure leads to dynamic routing changes, packets may find themselves traversing different network paths between the same source and destination. This causes delay variation. Congestion in the network can lead to routing instability and route flapping [29].

10. Conclusions

In this paper, we have presented the basic of DiffServ/IntServ and Jitter/Delay in the network of QoS. Nowadays, voice, audio and video traffic put increasing pressure on both LAN and WAN networks. DiffServ promises a lot but yet there are many questions unanswered. Differentiated Services seems to have the potential to become the long awaited universal service differentiation approach to Internet.Users are accustomed to the high reliability and high quality of standard voice and video technologies. Although the transport medium is changing our expectations remain the same. For the time being, DiffServ helps administrators prioritize different types of traffic without any resource reservations.

RSVP a hard QoS technique will help reserve a required level of capacity to support QoS effort. Likewise, QoS has to be planned from end-to-end so the bottlenecks are identified and removed. QoS will not relieve the responsibility of the network managers to plan, and allocate resources accordingly. But the various elements that comprise QoS can offer powerful tools to enable network managers to improve network performance. Delay can come from a variety of sources that include the end system, the local LAN, the access to the transport network and the transport network itself. Excessive delay can cause difficulties in normal conversation, and its effects increase with the level of interaction in the conversation and the amount of delay.

We have analyzed the e2e delay jitter in the DiffServ/IntServ network. we thought that the EF flows would not suffer from any delay jitter, since through this configuration they will be isolated from other traffic classes. We found out that delay jitter of EF flows depend on the back ground traffic intensity in the network, when the different traffic flows meet at the core router 1 some spikes are formed in the three scenarios. Therefore, the background traffic intensity must be controlled to guarantee the EF delay jitter.

11. Recommendations

Due to time limitation, we un-successful to simulate the model using Ns-2 as we proposed early. For future work, the recommendations will be concentrated on the simulation on the QoS using NS-2 to investigate the system performance evaluation. The simulation model will basically model with a new scheme and compared with type of Differentiated Services and Integrated Services in term of it jitter and delay variation. Then, the performance of the new algorithm is to be discussing by presenting the simulation results under different traffic scenarios and various parameters. Although based on the work presented in this document the objectives are almost fully completed, still in order to achieve end-to-end QoS in a wired and wireless Internet architecture requires a lot more research. There can be a number of research directions derived from this document for future work.

REFERENCES

[1] Cisco Systems. *Cisco IOS 12.0 Quality of Service*. Indianapolis: Cisco Press, 1999.

[2] Ferguson, Paul, and Huston, Geoff. *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*. New York: John Wiley & Sons, 1998.

[3] Lee, Donn. *Enhanced IP Services*. Indianapolis: Cisco Press, 1999.

[4] Vegesna, Srinivas. *IP Quality of Service for the Internet and the Intranets*. Indianapolis: Cisco Press, 2000.

[5] Leonard Franken. Quality of Service Management: A Model-Based Approach. PhD thesis, Centre for Telematics and Information Technology, 1996.

[6] Thomas Hirsch (July 2001). *Integrated & Differentiated Services*. From http://www.a-n-t-s.de/thomas/uni/kbs/

[7] MESHAVI BHATIA. (no date). *QoS over the Internet: The RSVP Protocol*. From http://www.cis.upenn.edu/~bhatia/rsvp.html

[8] Pekka Pessi. *RSVP and the Internet Integrated Services*. (1997). Retrieved August 28, 2007, from http://www.tml.tkk.fi/Opinnot/Tik-

110.551/1997/rsvp.html

[9] Zheng Wang. *Internet QoS: Architectures and Mechanisms for Quality of Service.* (1999). Retrieved August 28, 2007, from http://www.isoc.org/oti/articles/0601/wang.html

[10] Raj Jain. *Recent Trends in Networking Including ATM and Its Traffic Management and QoS.* (n.d.). Retrieved August 28, 2007, from http://www.cis.ohio-state.edu/~jain/talks/atm_mty.htm

[11] Huawei Technologies Co., Ltd. *NE20 Router Series: The Union of Smart and Solid — On Application in an Integrated Service Network.* (n.d.). Retrieved from http://www.huawei.com/products/datacomm/catalog.do?id=365

[12] RFC 1633, see http://www.faqs.org/rfcs/rfc1633.html

[13] Wikipedia. *Differentiated services*. (August 2007). Retrieved August 28, 2007, from http://en.wikipedia.org/wiki/Differentiated_services

[14] Wikipedia. *Peering.* (August 2007). Retrieved August 28, 2007, Retrieved from http://en.wikipedia.org/wiki/Peering_agreement

[15] Florian Baumgartner, Torsten Braun and Pascal Habegger. *Differentiated Services: A New Approach for Quality of Service in the Internet.* (n.d.) Retrieved August 28, 2007, from http://www.iam.unibe.ch/~baumgart/pubs/hpn98.pdf

[16] Liping Zhang. A Differentiated Services Architecture for the Internet (n.d.).

[17] Retrieved August 28, 2007 from http://pages.cs.wisc.edu/~lhl/cs740/DiffServ.ppt.

[18] QoS, see www.wikipedia.org.

Low-Complexity Handling of Delay Jitter for VoIP , see

[19] http://www.actapress.com/PaperInfo.aspx?PaperID=17434

[20] Parijat Garg & Rahul Singhai. *GoS in VoIP* (n.d.).

[21] VoIP Performance Management. (January 2006). *Impact of Delay in Voice over IP Services*.

[22] Mika Ilvesmäki. (n.d.) *Differentiated Services –architecture*

[23] IETF. *An Architecture for Differentiated Services*. RFC 2475. (Dec 1998)

[24] Utilization bounds of IntServ vs DiffServ

, See www.comp.brad.ac.uk/het-net/HET-NETs04/CameraPapers/P10.pdf

[25] Jacobson,V.: Congestion avoidance and control. Proc. ACM SIGGCOMM,

August. 1998, pp.314-329.

[26] Fujimoto,K.,Ata,S.,Murata,M.. Adaptive Playout Buffer Algorithm for Enhancing Perceived Quality of Streaming Applications. To appear in Telecommunication Systems. January 2004.

[27] Mansour,Y.,Shamir,B.P.: Jitter Controlin QoS Networks. IEEE/ACM Trans. on Networking. 9(4):492-502, 2000.

[28] Belenki,S.: An Enforced Inter-Admission Delay Performance- Driven connection Admission ControlAl gorithm. ACM SIGCOM,Computer communication review. Vol.32, No.2, April 2002.

[29] Landry,R.,Stavrakakis,I.: Study delay Jitter with and without peak rate enforcement. IEEE/ACM Trans. on Networking. Vol.5, No.4, August 1997.

APPENDIX

(SEE NEXT PAGE)