

## Performance assessment of DIFFSERV and INTSERV services in QOS on an academic network using NS2

### Evaluación de desempeño en QOS para servicios DIFFSERV e INTSERV sobre una red académica mediante el uso de NS2

Carlos Andrés Martínez<sup>1</sup>, Danilo López<sup>2</sup>, Jhon Jaime Ramírez<sup>3</sup>, Rubén Darío Gómez<sup>4</sup>.

<sup>1</sup>Universidad Distrital Francisco José de Caldas, Bogotá, Colombia, [camartineza@correo.udistrital.edu.co](mailto:camartineza@correo.udistrital.edu.co)

<sup>2</sup>Universidad Distrital Francisco José de Caldas, Bogotá, Colombia, [jjramirezo@correo.udistrital.edu.co](mailto:jjramirezo@correo.udistrital.edu.co)

<sup>3</sup>Universidad Distrital Francisco José de Caldas, Bogotá, Colombia, [rdgomez@correo.udistrital.edu.co](mailto:rdgomez@correo.udistrital.edu.co)

<sup>4</sup>Universidad Distrital Francisco José de Caldas, Bogotá, Colombia, [dalopez@udistrital.edu.co](mailto:dalopez@udistrital.edu.co)

Received: 01 May 2013

Accepted: 11 June 2013

Published: 30 July 2013

#### Abstract

This article provides a performance evaluation between DiffServ and IntServ, two computer architectures that allow the implementation of quality of service (QoS) to a given network. A simulated scenario in NS2 is proposed and the two types of QoS are implemented to evaluate variables such as jitter, bandwidth, delay, and packet loss, all with the aid of graphs and with statistical results.

First, the simulation is done without implementing QoS; subsequently, IntServ and DiffServ are implemented to compare between the two architectures and define the advantages and disadvantages of each type of service and determine the most appropriate for a specific program.

**Key words:** Bandwidth, quality of service, DiffServ, IntServ, Jitter, Loss, Delay, Throughput.

#### Resumen

En el presente artículo se realiza una evaluación de desempeño entre DIFFSERV e INTSERV, dos arquitecturas que permiten implementar calidad de servicio (QoS) a una red determinada. Se propone un escenario simulado en NS2 y se implementan los dos tipos de calidad de servicio, con el fin de evaluar variables como Jitter, el ancho de banda, el retardo y la pérdida de paquetes, todo esto con la ayuda de gráficos y resultados estadísticos.

En primer lugar se realiza la simulación sin implementar QoS, posteriormente se implemente DIFFSERV e INTSERV con el fin de realizar una comparación entre las dos arquitecturas y poder definir las ventajas y desventajas de cada tipo de servicios y determinar cuales es el más apropiado para una aplicación en particular

**Palabras clave:** Ancho de banda, calidad de servicio DIFFSERV, INTSERV, Jitter, Pérdidas, Retardos throughput

### 1. Introduction

Systems and telecommunications networks must be able to meet the needs that arise constantly, even though critical situations exist that hinder this process,

such as declines in communication links, saturation within the network, interference in the communication media, and others.

Currently, technological progress has allowed the

emergence of programs like VoIP transmission, streaming multimedia content, and IPTV systems among others. However, these lead to improving network response in terms of bandwidth optimization, delays and data loss, which is not a simple process due to the physical limitations of the elements that make up the network. [1]

Quality of service (QoS) is an alternative to make better use of physical resources, allowing the traffic that flows in an IP network to be managed efficiently. Because all data being transmitted need not be processed in the same way, for example, traffic due to web access or file transfer tolerates higher delays than audio and video programs would; QoS can grant certain privileges to a particular traffic to minimize delays and losses.

The issue related to multimedia programs on IP is one of the most researched due to the requirements it has in terms of width of delays and loss of information and QoS becomes crucial, in fact, analytical models are being proposed on home environments focused on the provision of QoS in order to have a base model on which to plan future network construction focused on multimedia systems [2]

The two architectures to analyze regarding IP networks are Diffserv and Intserv. DiffServ is based on packet marking with which a priority value is assigned to them in the queuing process, while IntServ is based on resource reservation in all nodes of the network [3]

Currently, investigations have already been made between these two architectures, evaluating loss and delay as outlined in the publication of the Universal Multiservice Networks magazine titled "Mapping of loss and delay between intserv and diffserv" [4] [1] this has generated the motivation to conduct a study based on said previous research where variables such as jitter, delay, throughput and losses are evaluated through NS2 simulation on a proposed topology. Later, the results will be analyzed and respective conclusions will be reached.

## 2. Theoretical framework

The IP protocol, described in RFC 791, defines in the header of the IP v4 protocol an octet called ToS [5], which is divided into two parts, three bits to

differentiate eight priority levels called precedence, and the remainder acting as flags to indicate the preferred route of the package:

- Minimum delay
- Maximum Performance (Throughput)
- Maximum reliability (reliability)
- Minimum cost (cost)

PRECEDENCE	D	T	R	C	X
------------	---	---	---	---	---

*Figure 1.* ToS byte of the IPv4 header, 1981

In Figure 1, ToS byte is shown with the main bits, the first three the precedence, D: Delay, or minimum delay, T: Throughput or Maximum performance, R: Reliability or maximum reliability, C: Cost or minimum cost x, which is a reserved bit.

### 2.1 Intserv

Problems such as only a sending priority being available and not one of dismissal gave rise to new architectures. In 1994, the RFC 1633 emerged, which defines the IntServ [6] where the user requests in advance the resources needed, each router notices that request and books what is required.

The resource reservation is carried out by the Resource Reservation Protocol (RSVP), which guarantees the requested QoS and in case the necessary bandwidth may not be ensured that request would be dismissed immediately. A sequence of datagrams, also called flow, is performed and this is important because it defines a reserve by a flow and not a datagram [6]. The flow consists of the IP address, ports of origin and destination, and the transport protocol used.

The RSVP protocol books the capacity requested by a flow in all the routers, creating state information in each router. It is, despite being used in IP, connection-oriented and it is considered mainly in multicast traffic. If conditions are not secured, the call is rejected; this process is known as admission control.

### IntServ provides three services:

#### 2.1.1 Guaranteed:

Where a minimum flow and maximum delay are guaranteed, each router in the way does it, sometimes

not implemented due to limitations in the physical environment.

### 2.1.2 Controlled load:

The delay is low but no guarantees are given, quality similar to a lightly loaded datagrams network.

### 2.1.3 Best Effort:

No guarantees are given. [7]

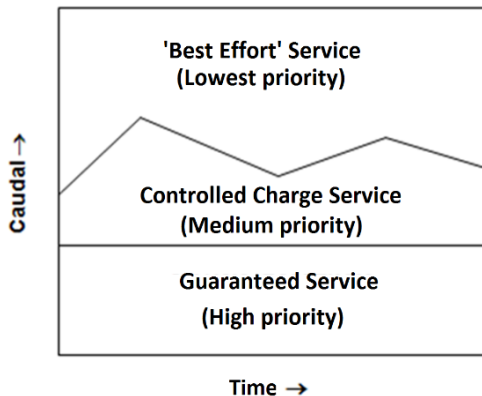


Figure 2. Distribution of resources in IntServ; figure taken from [7]

After 1997, some limitations appeared; this was due to scalability problems because of the need to maintain state information in each router and, therefore, cannot be used in large networks like the internet.

This huge flow of information causes problems in implementing the hardware to keep the algorithms; however, it continues because of improvements that have been made and an interest in using the RSVP protocol in Multi-Protocol Label Switching (MPLS) where the flows are small [8]

## 2.2 Diffserv

The RFC 2472 came in 1998 regarding the DiffServ model [9] trying to solve scalability issues by marking packets with a label read by the routers detailing the treatment and priority they should be given by the routers. In this model, there are no reservations so the routers do not see the flows, which imply that there is no signaling protocol or status information that caused so many problems to the IntServ model.

Quality guarantees are not so strict, which may occasionally be tolerable. Packets are classified into classes, also called Class of Service (CoS); for each class, a Service Level Agreement (SLA) is assigned specifying values of QoS parameters for each class.

Classes are limited and independent of the number of flows; for this reason, the complexity does not depend on the number of users and does not involve problems of ascending information making the architecture scalable.

QoS information is in the datagrams in a DS field; this information is recognized by the routers by configuration and they give the treatment to each class.

Figure 3 shows six bits dedicated to Differentiated Service code point (DSCP), which indicate the treatment and two CU bits that are not used (currently used in congestion control). With the six bits, it has 64 categories of traffic but they have been divided into three groups.

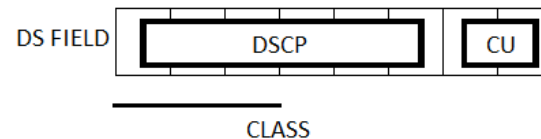


Figure 3. Field DS (RFC 2474)

Table 1 shows the ranges dividing the traffic, where the highest priority range is 40 to 47.

Table 1. Class meaning of the DSCP

Rank	Value	Meaning	Priority
0-7	000xxx	Better effort	0
8-15	001xxx	Preferential	1
16-23	010xxx	Preferential	2
24-31	011xxx	Preferential	3
32-39	100xxx	Preferential	4
40-47	101xxx	Preferential	5
48-55	110xxx	Network Control	-
56-63	111xxx	Network Control	-

## 3. Proposal development

The topology shown in Figure 4 is available to carry out the task. 3 sources, 3 intermediate nodes and 3 receptors; the topology will be implemented in the NS2 software to verify the performance of both integrated and differentiated services. Values of packet loss, jitter, delay, bandwidth usage will be determined as parameters for comparison between the two services mentioned.

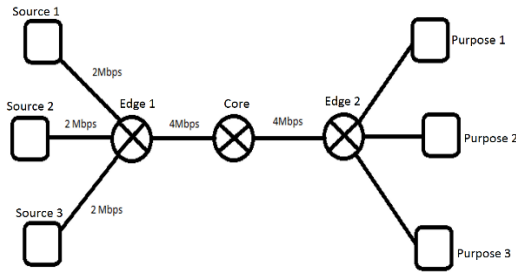


Figure 4. Topology proposed

First, we obtain a graph (Figure 5) of the bandwidth used by each traffic, where the trunk capacity is high enough to allow transmission of all packets sent by the 3 sources to succeed; this is done to compare with later results. We must point out that the transmission rate of each transmitter was stipulated at 1.6 Mbps.

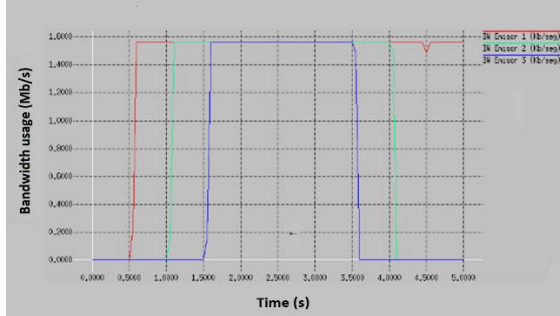


Figure 5. Use of bandwidth of the three traffics with unlimited bandwidth

Considering the actual conditions of the topology, due to the settings shown in Figure 4, it is clear that the capacity of each channel from the source to the Edge is 2Mbps, and the Edge Core channel has a capacity of 4Mbps, meaning that when the sources are using the maximum capacity, there will be a packet loss because the capacity of the backbone should be at least 4.8Mbps.

### 3.1 Ns2 simulation

To verify this phenomenon, nine nodes are simulated:

```
Set n0 [$ ns node] # this is done for the 9 nodes
```

We proceed to specify the mode of transmission between nodes, the capacity of each channel, channel delay, and the type of tail loss. In the backbone links SFQ was used, where packet loss is equitable for the three traffics.

```
$ns duplex-link $n0 $n3 2Mb 10ms DropTail
$ns duplex-link $n1 $n3 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 2Mb 10ms DropTail
$ns duplex-link $n3 $n4 4Mb 10ms SFQ
$ns duplex-link $n4 $n5 4Mb 10ms SFQ
$ns duplex-link $n6 $n5 2Mb 10ms DropTail
$ns duplex-link $n7 $n5 2Mb 10ms DropTail
$ns duplex-link $n8 $n5 2Mb 10ms DropTail
```

Now, a UDP agent is set up for each source and, thus, the type of traffic for each source is specified as follows:

```
set udp1 [new Agent/UDP]
$ns attach-agent $n0 $udp1
set udp2 [new Agent/UDP]
$ns attach-agent $n1 $udp2
set udp3 [new Agent/UDP]
$ns attach-agent $n2 $udp3
```

Traffic of source 1 is set as constant bit rate (CBR), as well as with the other 2 traffics. Packets will have a size of 500 bytes and will be sent every 0.0025 seconds, *i.e.*, at a rate of 400 packs /s, equivalent to a transmission rate of 1.6 Mbps.

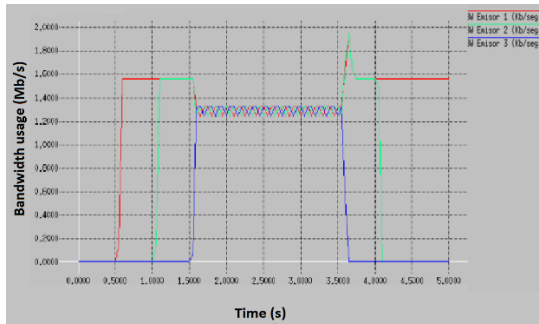
```
set trafico1 [new Application/Traffic/CBR]
$trafico1 set packetSize_ 500
$trafico1 set interval_ 0.0025
$trafico1 attach-agent $udp1
$udp1 set class_ 1
```

The definition of the other two traffics is similar to that described above. Finally, receiving agents are created and the sources and receiver agents are associated. The definition of the packets may be null or Loss Monitor. In this case, Loss Monitor is chosen to obtain readings of some variables of interest like bytes arriving at the receiver, packet loss, among others.

```
set recep1 [new Agent/LossMonitor]
$ns attach-agent $n6 $recep1
set recep2 [new Agent/LossMonitor]
$ns attach-agent $n7 $recep2
set recep3 [new Agent/LossMonitor]
$ns attach-agent $n8 $recep3
```

```
$ns connect $udp1 $recep1
$ns connect $udp2 $recep2
$ns connect $udp3 $recep3
```

The result, in terms of bandwidth usage, can be seen in Figure 6; when two emitters transmit, the bandwidth is sufficient to meet the transmission rate of these (1.6 Mbps); however, when the third transmitter begins transmission, the intended bandwidth for each traffic decreases approximately by 1.33Mbps due to the limitation of the channel and that is where the losses become evident.



**Figure 6.** Bandwidth usage of the network without QoS implementation

### 3.1.1 Diffserv

This is the source of the problem, which questions whether all packets sent through the channel are of equal importance to the "receptor users" or if the system tolerates some packet loss, but ensures that "high-priority" packets reach their destination. For this purpose, quality of service is provided to the system. For differentiated services, packets that have a priority level are "marked" as high-priority packets: the medium and low priority packets are also marked. Thereby, when packets arrive to the buffer they are not sent in order of arrival, but by their level of priority. In fact, when the bandwidth is used to the maximum, medium- and normal-priority packets can be discarded, once the high-priority packets are sent, it continues to send those of lower priority.

This implies the existence of three queues with which packets with the appropriate mark will be given special treatment. It is also helpful to monitor the queue sizes as it is known that in case the threshold reached in each queue is exceeded, it immediately proceeds to discard packets from the queues, indicating to the source that the package rate delivery should be decreased.

Using differentiated services for the topology of Figure 4 is proposed, so that the packets sent by source1 are of greater importance for the receiver. In this case, marking the packets takes place; packets from source1 will be "marked" with 10, packets from source2 with "20", and packets from source 3 "30", upon having clear discrimination of packets, a special treatment is offered to every marking.

In the description of the script, the mode of transmission between nodes is specified, as well as the capacity of each channel, the channel delay and type of tail drop. Notably, the mode of transmission among

nodes n3 - n4 - n5 is simplex, and further nodes n3 and n5 are defined as EDGE (edges), and node n4 as CORE.

```
$ns duplex-link $n0 $n3 2Mb 10ms DropTail
$ns duplex-link $n1 $n3 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 2Mb 10ms DropTail
```

```
$ns simplex-link $n3 $n4 4Mb 10ms dsRED/edge
$ns simplex-link $n4 $n3 4Mb 10ms dsRED/core
$ns simplex-link $n4 $n5 4Mb 10ms dsRED/core
$ns simplex-link $n5 $n4 4Mb 10ms dsRED/edge
```

```
$ns duplex-link $n6 $n5 2Mb 10ms DropTail
$ns duplex-link $n7 $n5 2Mb 10ms DropTail
$ns duplex-link $n8 $n5 2Mb 10ms DropTail
```

Four variables are created, linked to the respective links of nodes n4, n5, and n6, then access to the configuration of the connection properties will be allowed (in this case, the treatment will be given to queues).

```
set qE1C [[ $ns link $n3 $n4 ] queue]
set qE2C [[ $ns link $n5 $n4 ] queue]
set qCE1 [[ $ns link $n4 $n3 ] queue]
set qCE2 [[ $ns link $n4 $n5 ] queue]
```

Now the link (qE1C) is configured as follows: Packet size in bytes is determined, in this case the packet size is 500 bytes.

```
$qE1C meanPktSize 500
```

The number of virtual queues and physical queues is determined; in this case, three virtual queues and one physical queue will be worked.

```
$qE1C set numQueues_ 1
$qE1C setNumPrec 3
```

The algorithm with which "marking" of packages will be made is set, this will be done through the TokenBucket algorithm, packets sent from the n0 node to the n6 node will be marked with "10", packets sent from the node n1 to node n7 will be marked with "20", and packets sent from the node n2 to node n8 will be marked with "30".

```
$qE1C addPolicyEntry [$n0 id] [$n6 id]
TokenBucket 10 2000000 50000
$qE1C addPolicyEntry [$n1 id] [$n7 id]
TokenBucket 20 2000000 50000
$qE1C addPolicyEntry [$n2 id] [$n8 id]
TokenBucket 30 2000000 50000
```

Packets with respective markings are sent to a different virtual queues, as specified, packets with a 10 marking to physical queue 1 and virtual queue 1, those with a 20 marking to physical queue 1 and virtual queue 2,

and those with marking 30 are sent to the physical queue 1 and virtual queue 3.

```
$qE1C addPolicerEntry TokenBucket 10 11
$qE1C addPolicerEntry TokenBucket 20 21
$qE1C addPolicerEntry TokenBucket 30 31
$qE1C addPHBEntry 10 0 0
$qE1C addPHBEntry 20 0 1
$qE1C addPHBEntry 30 0 2
```

The priority of each queue is specified, in this case the packets marked 10 have the highest priority, the other packets have low priority; this is specified by the probability of packet loss. A higher priority means lower probability of packet loss.

```
$qE1C configQ 0 0 1 10 0.000002
$qE1C configQ 0 1 11 20 0.40
$qE1C configQ 0 2 11 20 0.20
```

This configuration is carried out three times, one for each link between Edge1 - Core - Edge2.

### 3.1.2 Intserv

Currently, the two architectures coexist to ensure quality of services. They depend on many factors, such as the program, the integrated services were proposed some time before DiffServ and are still in force, which is why it is necessary to evaluate them in this topology.

For simulation in NS2, it is necessary to install a patch in the main folder where the RSVP protocol is included, which has the function of booking the resources requested. The topology is the same as implemented in subparagraph (A) (Figure 7); for this reason, the code used for the creation of the nodes is the same. But when defining connections, integrated services properties must be assigned as follows:

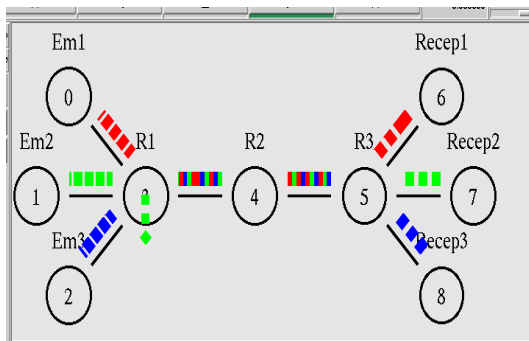


Figure 7. Topology generated by ns-2

A process to define the properties of the links is set up, which defines the transmission type, associated with the RSVP protocol and other properties like delay, packet size, and others. This process creates the links n0, n1, n2 with node n3 and links n6, n7, n8 to node n5.

```
proc create_link {src_node dst_node} {
    global ns
    set rate 2Mb
    set delay 10ms
    set reservable 0.8
    set rsvp_rate 100
    set bo_queue_size 5000
    $ns duplex-rsvp-link $src_node
    $dst_node $rate $delay $reservable $rsvp_rate
    $bo_queue_size Param Null;
}
```

A second process is set created where links n3-n4 and n4-n5 are found. Two processes are developed because each process defines a different channel capacity (2 Mbps and 4Mbps, respectively).

```
proc create_link2 {src_node dst_node} {
    global ns
    set rate 4Mb
    set delay 10ms
    set reservable 0.8
    set rsvp_rate 100
    set bo_queue_size 5000
    $ns duplex-rsvp-link $src_node
    $dst_node $rate $delay $reservable $rsvp_rate
    $bo_queue_size Param Null;
}
```

To create the links, the process is invoked and within the parameters the source terminal and the destination terminal are defined.

```
create_link $n0 $n3
```

Each node is assigned a connection and subsequently assigned an RSVP agent.

```
set rsvp0 [$n0 add-rsvp-agent]
```

As for traffic and other receptor and display agents, the same commands are used for simulation purposes.

```
set flow_id0 1
set rsvp_session0 [$rsvp2 session $n8
$flow_id0]
```

The timeline of the simulation would be as follows:

```
$ns at 0.01 "$rsvp2 sender $rsvp_session0
+700000 2000 20"
```

To calculate the missing values, a Loss monitor agent is created in the source monitor and one on the receiver in such a way that both the sent and received packets are monitored. This is in order to calculate for packages lost through the network. All the information related to the implementation of the IntServ is located in [10].

## 4. Analysis of results

Clearly, when no QoS is implemented within the network, some of the packets sent by the various sources are likely to be discarded, even if the recipients have preference for specific information. This event usually occurs when the channel capacity is not sufficient to meet the transmission needs of the sources it covers; then, there is a clear decrease in the use of bandwidth by the sources, which results in packet loss. This phenomenon can be observed in Figure 6. The channel is for 4Mbps, and each source transmits at a rate of 1.6Mbps, when two sources are transmitting information is not lost; however, when the third source starts transmitting, the channel cannot cope and the three sources suffer loss.

### 4.1 Diffserv

When implementing the QoS (DiffServ), one gets a result more in accordance with the presented need, where particular source packets were more important for the receiver, this is evident in Figure 8, where packets of higher priority (source 1) are those that have a higher percentage of bandwidth use. The traffic generated by source 2 had medium priority, and the traffic generated by source 3 had low priority, in this case the three sources are transmitting continuously.

Saturation in the trunk is clearly observed, however, the bandwidth for source 1 is greater than the bandwidth of the other sources.

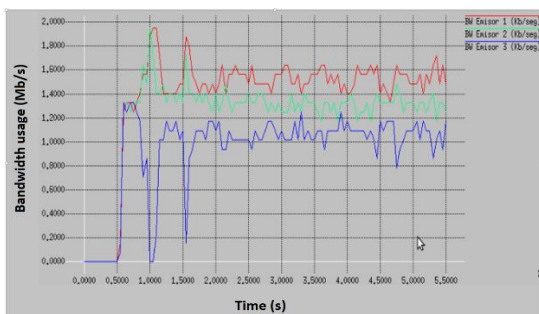


Figure 8. Bandwidth usage topology with the implementation of quality of service (DiffServ)

In Figure 9, we can see how the channel bandwidth is managed, when different sources leave and return transmission. In principle, source 1 begins to transmit at a rate of 1.6 Mbps, later source 2 transmits at a rate of 1.6Mbps, and the channel is capable of supplementing these rates; however, when source 3 transmits, the channel is no longer able to assign the same bandwidth. With the implementation of DiffServ the higher priority packets (red source) are transmitted mostly by the channel, the packets sent by channel 2 (green source) have medium priority and packets sent by source 3 (blue) are mostly discarded when required to give preferential treatment to the other packages. Clearly, when source 3 stops emitting the channel bandwidth is managed again to achieve transmission of the source packets; once an event like the aforementioned happens again, higher priority packets will be given special treatment.

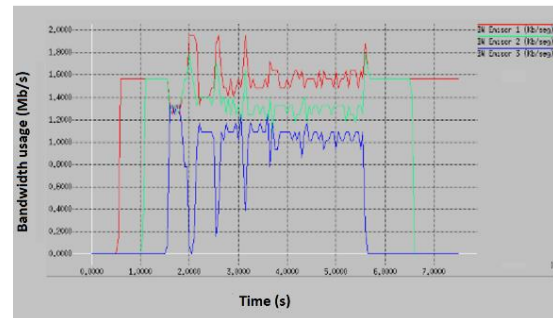


Figure 9. Use of bandwidth implementing DiffServ Where some sources do not transmit continuously

Table 2. Packets lost by each source

CP	TotPkts	TxPkts	ldrops	edrops
All	4788	4105	152	531
10	1796	1796	0	0
20	1596	1409	0	187
30	1396	900	152	344

A graph of the three traffics shown in the topology was made and the packet loss with respect to time was plotted (samples were taken every 0.5 units of time), as can be seen in Figure 10. Table 2 contains the data taken halfway through the simulation, the number of packets transmitted up to that instant can be observed, as well as the number of discarded packets.

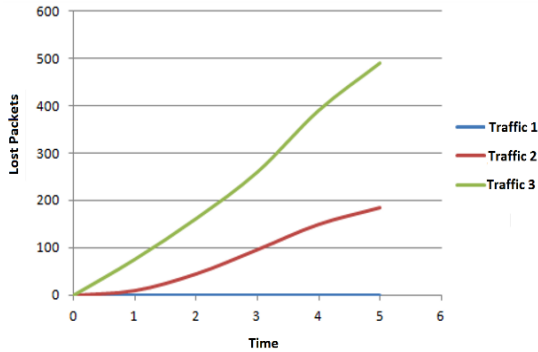


Figure 10. Number of packets lost by each source

Figure 10 shows the number of packets discarded by each source when traffic is continuously present (Figure 8). Lower-priority packets are discarded at higher rates than packets with medium and high priorities. High-priority packets are not lost unlike other packets. According to the figure it does not mean that packet loss rate increases, but it shows lost packets accumulated over time.

Jitter is another parameter that allows evaluating the performance of the implemented quality of service; jitter defines that the amount of delay experienced by each plot can differ [11]. This is caused by the amount of queuing delay and processing time, which may vary depending on the traffic loaded on the network. Figure 11 demonstrates that jitter retrieved for traffic with high priority between certain values of time (4 and 5.4 simulation time units) is variable. The value of this is between 0.001 and -0.001, a small value that could be tolerated in various programs.

Real-time communications (like VoIP), for example, tolerate jitter less than 100 ms, given that this can be appropriately compensated. The implementation of DiffServ on traffic of this kind would be an appropriate option to provide QoS [10]

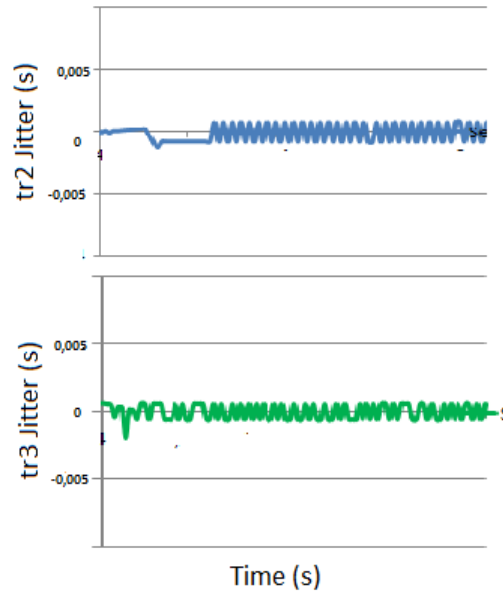
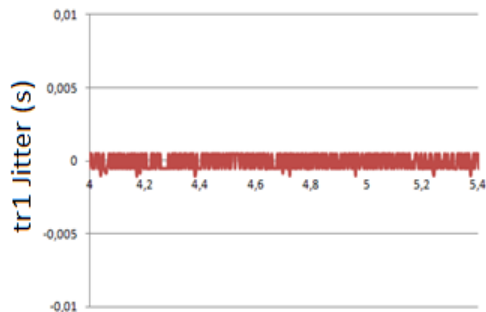


Figure 11. Measuring jitter in DiffServ

Finally, the measurement of the value of delay in sending a packet from the sender is considered (end-to-end delay), the value of the delay has three components: the value of intrinsic transmission delay of the channel, the value of propagation delay, and processing delay (pasting among others). Figure 12 shows the value of delays in a time range for sent packets, this value varies between 75 and 55 ms.

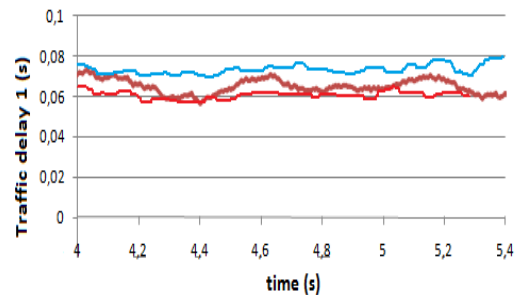


Figure 12. Measurement of extreme delay (end-to-end delay) in DiffServ

#### 4.2 Intserv

Upon simulation, one can see that previous packets are sent; these are the packets generated by the RSVP protocol. In theory, these packets send requests to the devices, making a request to each router so that the required quality of service can be booked that is why



the allocation of time is what will determine which one has the highest priority.

In the IntServ simulation, three consecutive signals were obtained that begin with some delay from one another. Figure 13 shows the output generated in the Xgraph showing the bandwidth used for each channel.

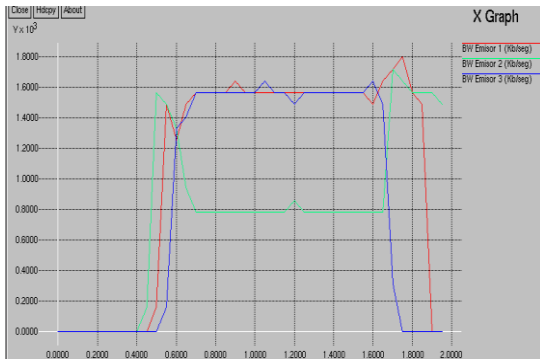


Figure 13. Bandwidth in IntServ

It is important to note that the signal that makes the request first is the red one, then blue and, finally, the green one; despite that, in this simulation, the one that first begins transmitting is the node of the green signal, so at first it can be seen that it occupies the channel, in the moment that the red signal begins to be transmitted it occupies the entire bandwidth it needs and does not allow loss of packets at least at the end, although at the beginning a few are.

The channel has a 4 Mb bandwidth, which is enough to meet and satisfy the requirements reserved by the blue and red signals then the green signal can count on the surplus to send its packets.

Regarding the Jitter in IntServ, it is caused by the amount of queuing delay and processing time that may vary depending on traffic loaded onto the network.

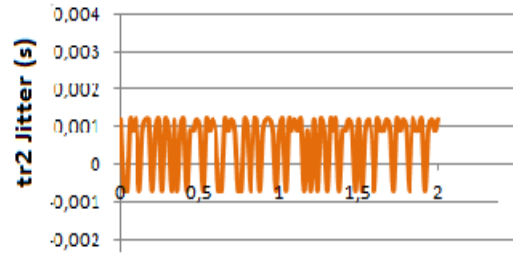
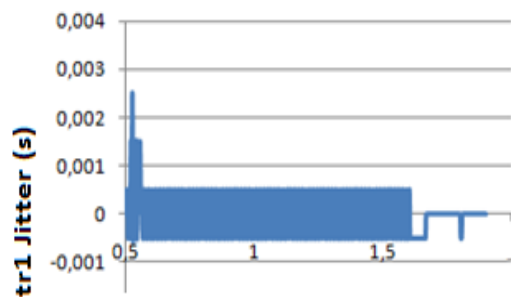


Figure 14. Jitter in IntServ

In Figure 14, we can see the jitter values obtained between certain time values; it can be observed that the jitter is variable, which means that the delay between the different arriving packets is different. However, this variable value is between 0.005 and -0.005, approximately.

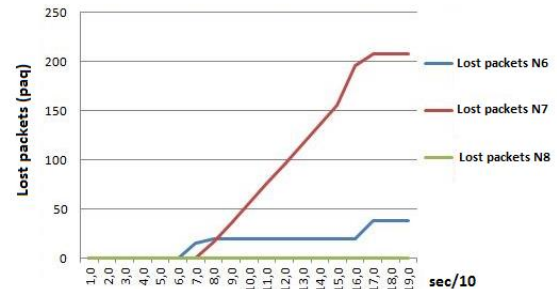
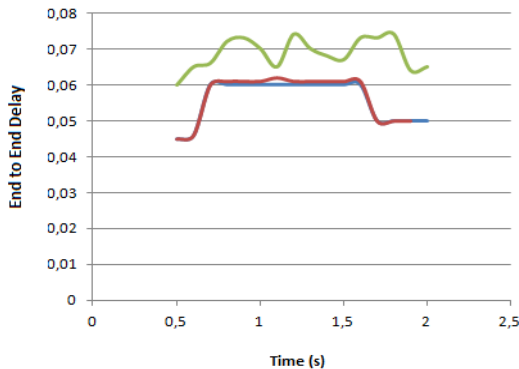


Figure 15. Packets lost in IntServ

Figure 15 shows the number of packets discarded by each source when traffic is continuously present; packets with lower priority are discarded in greater proportion than medium- and high-priority packets. This does not mean that the lost-packet rate increases, but shows lost packets accumulated over time.



**Figure 16.** End to end in IntServ

Finally, the measurement of the delay value in sending a packet from the sender is considered (end-to-end delay), the value of the delay has three components: intrinsic transmission delay value of the channel, the value of propagation delay, processing delay value (pasting among others). Figure 16 shows the delay value in a time line; for sent packets, this value ranges from 45 to 70 ms.

Tables 3, 4, and 5 show the values used in bandwidth, jitter, delay, and number of packets per traffic. The values in the tables were obtained when the backbone was at saturation. The jitter and delay value was obtained as the maximum value of the values obtained in the simulation.

**Table 3.** Comparison between diffserv and intserv for traffic 1

-----	<b>Diffserv</b>	<b>Intserv</b>
BW	1.6 Mbps (variable)	1.6Mbps
Lost packets	0	0
Jitter (max.)	0.001 s	0.00065 s
Delay (max.)	0.07 s	0.06

**Table 4.** Comparison between diffserv and intserv for traffic 2

-----	<b>Diffserv</b>	<b>Intserv</b>
BW	1.4Mbps (variable)	1.6Mbps
Lost packets	41	38
Jitter (max.)	0.0017 s	0.0012 s
Delay (max.)	0.079 s	0.06 s

**Table 5.** Comparison between diffserv and intserv for traffic 3

-----	<b>Diffserv</b>	<b>Intserv</b>
BW	1Mbps (variable)	0.8Mbps
Lost packets	155	208
Jitter (max.)	0.00172 s	0.0023 s
Delay (max.)	0.081 s	0.78 s

From the tables, several factors can be analyzed, the first of which is that both in Intserv and Diffserv it is guaranteed not to lose any high-priority packet. However, the number of lost packets for other traffics is higher in intserv, which means that in Intserv it is more likely that low priority packet will be lost compared to Diffserv.

This is a key factor in the implementation of certain programs, for example, there are some in which no packet loss is tolerated, such as multimedia streams and video conferencing, among others; so this would be a first parameter to be evaluated to select a particular QoS.

On the other hand, jitter and delay implementing Intserv are lower than obtained in Diffserv, this is valuable in programs where the traffic is susceptible to jitter or delay, for example, VoIP. In fact, delay reduction and uniformity of such (ideally zero jitter) are key objectives pursued by the quality of service.

Obviously, the delays for low-priority customers are greater than the delay experienced by higher-priority clients, as can be observed both in the graphs and tables above. This is because the high-priority packets, either in diffserv (special treatment for high-priority queues) or intserv (reserving necessary bandwidth on the trunk line) are sent almost immediately, so they do not experience delay caused by queuing, but only the intrinsic delay of the channel through which they are transmitted. However, the delay implementing Intserv is lower than that obtained with Diffserv. This is because in Intserv a specific bandwidth is reserved, so that it could be analyzed as if the high-priority traffic had a particular channel for transmission; while in Diffserv, despite prioritizing certain packets, a slight delay takes place due to queuing.

It is clear that to ensure the quality of service in an IP network either IntServ or DiffServ can be used. The need to consider the advantages and weaknesses of each of these services is evident. Based on the results obtained, some applications will be mentioned with

their respective needs and the respective comparisons will be performed.

According to Table 6, if the critical factors were the delay and jitter (for telephony or video streaming), the most adequate choice in the first instance would, therefore, be Intserv; however, as it is well known, a problem of this service is that it is only applicable in small networks, unlike the differentiated services. The main reason is the scalability due to the need to maintain state information for each router of each flow. This means that it is not sufficient to assess only parameters of jitter, delay etc. For the selection of a service, there are external factors like the physical viability of implementing the service, operating costs, and network complexity, among others.

**Table 6.** QoS requirements of some applications [12]

Software	BW	Delay	Jitter	Loss
Ap. web	low	Low-medium	medium	Medium-High
telephone	low	low	low	low
Interactive Video	wide	low	low	low
Video streaming	wide	medium	low	low

## 5. Conclusions

Both Intserv and Diffserv seek to ensure that high-priority traffic will transmit all packets to their destination. The difference between the two lies in how they achieve said objective. Factors like jitter, delays, use of bandwidth, and packet loss are starting points to substantiate the comparison and their choice in particular software. But other factors exist that influence the selection, such as network complexity with a particular service, for example, Intserv is more complex to implement (physical environment limitations), as opposed to Diffserv, which has a simple way to classify and prioritize traffic.

Software applications that are susceptible to both delay and jitter, for example, VoIP software, interactive video among others should be treated with a quality of service that provides guarantees for their needs. According to the topology simulated in this study, favorable results were obtained in terms of delay and jitter implementing Intserv which at first preferred choosing this service; however, in larger networks, the feasibility of implementation would be another factor to consider.

As a final conclusion, in this type of topology integrated service implementation provides acceptable results, which in this case make it more feasible to use this type of architecture. For DiffServ, it does not

guarantee the best quality, but it is still a good way to manage priorities in channels because routers do not see the packets but labels they carry and, therefore, it is the most widely used given that IntServ can lead to a problem of oversized and inefficient use of networks, *i.e.*, problems of scalability, but – if used properly – it allows guaranteeing quality of service in terms of Jitter, delay, loss of packets and in the use of bandwidth optimization.

## References

- [1] C. H. E. S. M. G. y. V. L. R. J. Bril, "Multimedia QoS in consumer terminals. Signal Processing Systems," in *2001 IEEE Workshop*, 2001.
- [2] J. Z. y. H.-S. X. L. Tang, "A Case for Multimedia QoS Provisioning in Home Networks," in *Communications (ICC), 2011 IEEE International Conference on*, Kyoto, 2011.
- [3] C. Vogt, "Admission control and resource reservation on the internet," *ACM SIGSOFT Software Engineering Notes*, vol. 27, no. 3, pp. 80-87, 2002.
- [4] T. Chahed, C. Hebuteme and C. Fayet, "Mapping of loss and delay between Intserv and DiffServ. Universal Multiservice Networks," in *ECUMN 2000 1st European Conference on*, Colmar, 2000.
- [5] "RFC 791: Internet Protocol," September 1981. [Online]. Available: [www.ietf.org/rfc/rfc791.txt](http://www.ietf.org/rfc/rfc791.txt).
- [6] C. Min, X. Junqing, W. Yun and G. Guanqun, "An Implementation Model of IntServ/RSVP Based CORBA A/V Stream Service," in *Proceedings of the 36th International Conference on Technology of Object-Oriented Languages*, Washington, 1994.
- [7] R. Montaña, "Universidad de Valencia," [Online]. Available: [www.uv.es/montanan/ampliacion/amplif\\_6.ppt](http://www.uv.es/montanan/ampliacion/amplif_6.ppt). [Accessed 05 June 2013].
- [8] S. Schinitter, F. Hartleb and M. Horneffer, "Quality-Of-Service Class Specific Traffic Matrices in," in *Internet Measurement Conference*, San Diego, 2007.
- [9] K. Nichols, S. Blake and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," IETF.org, December 1998. [Online]. Available: <http://www.ietf.org/rfc/rfc2474.txt>.
- [10] "QoS- experiment," 05 June 2013. [Online]. Available: [http://www.pdfdownload.org/pdf2html/view\\_online.php?url=http%3A%2F%2Fwww.tu-ilmenau.de%2Ffakia%2Ffileadmin%2Ftemplate%2FstartIA%2Fihss%2Fdat%2Flehre%2FDipl\\_WI%2FQoS-experiment.pdf](http://www.pdfdownload.org/pdf2html/view_online.php?url=http%3A%2F%2Fwww.tu-ilmenau.de%2Ffakia%2Ffileadmin%2Ftemplate%2FstartIA%2Fihss%2Fdat%2Flehre%2FDipl_WI%2FQoS-experiment.pdf).
- [11] "QoS Quality Of service VoIP," 3CX 3CX Centralita Telefónica para Windows, 05 June 2013. [Online]. Available: [http://www.voipforo.com/QoS/QoS\\_Jitter.php](http://www.voipforo.com/QoS/QoS_Jitter.php).
- [12] R. Montaña, "Calidad del servicio," [Online]. Available: <http://www.docstoc.com/docs/34977565/Calidad-de-Servicio-%28QoS%29>. [Accessed 05 June 2013].