University of Information Technology

Network Design for a Research Institute

Myint Myat Aung

UIT-1391

Network Design & Engineering (CST-32406)

September 2023

Table of Contents

Overview of the need for QoS

Ever since its creation, the internet has been enabling humans to exchange information across the world. While initially only used in academia for sharing research knowledge, it has now found (previously unexpected) applications in business, entertainment, education and many other fields. Along with these new use-cases came new challenges and demands, including the need for low latency communication for real-time applications and high bandwidth for large data transfers.

While using better or simply more hardware can solve the problem (see "Overprovisioning"), this is often not feasible due to costs and physical constraints. Thus, the need to carefully provision and prioritize scarce network resources led to the development of Quality of Service (QoS) mechanisms.

Overprovisioning

By building a network with enough hardware resources (more cables, better routers) to support the maximum possible traffic, applications will never need to worry about packet loss, high latency or low bandwidth.

An example of this is the **telephone system** [1], where so much network capacity is available that it is impossible for a telephone call to not go through due to high demand.

However, this is a solution that is not nearly scalable enough as exponentially more (expensive) hardware is needed as the network gets bigger. Futhermore, if the expected amount of traffic increases, it will not be able to guarantee maximum performance anymore.

QoS mechanisms can offer consistent performance regardless of hardware quality and traffic spikes.

Quality of Service (QoS)

Most networks must provide service to applications that want this **real-time delivery** at the same time that they provide service to applications that want **high throughput**.

QoS is the name given to mechanisms that reconcile these competing demands.

It is a performance-related guarantee where some resources are set aside for a particular application. e.g:

- at least this much bandwidth

- at most this much delay

In contrast to systems with QoS mechanisms, **best effort** systems provide equal treatment to all packets/hosts/applications.

There are 4 different network resource requirements that QoS intends to manage [2]:

Bandwidth

How much data can be transferred at a time. This usually translates to upload/download speed.

Delay

How long it takes a packet to travel from source to destination.

Jitter

Variance in the time it takes a packet to arrive. The packets of a data stream may arrive unordered, often earlier or later than its expected time.

Loss

Chance of packets being lost in the network and never arriving.

The table below shows the resource requirements of different applications.

Table 1. Applications' varying resource requirements.

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Voice chat	Low	High	High	Low
Videoconferencing	High	High	High	Low

We make a few interesting observations:

- Email being a messaging system not concerned with fast or large data delivery can be served well by a network without much prioritization for it.
- File sharing doesn’t need to be timely, but it needs a high bandwidth for large files.
- Voice chat, in contrast, is real-time and needs to have as low a latency as possible, but not as much bandwidth due to audio transfer not needing much data.

- Videoconferencing like voice chat is a real-time application but requires much more bandwidth. In addition, loss is acceptable as users don’t mind a few frames dropping.

As such, different applications have different network requirements to achieve proper performance. This resource management will be done using the QoS mechanisms discussed in the next chapter.

Improving QoS in wired networks

Quality of service (QoS) is not provided by a single mechanism but a collection of separate mechanisms, each of which can be used alone or in combination with another technique. These mechanisms can be categorized into 2 broad categories [3]:

Fine-grained approaches

which provide QoS on an individual basis, e.g. applications. e.g.

- Integrated Services with RSVP

Coarse-grained approaches

which provide QoS on a collective basis, e.g. a class of traffic. e.g.

- Differentiated Services

Fine-grained approach: Integrated Services with RSVP

Integrated services, or IntServ, use the mechanisms listed below to provide QoS [4]:

Flowspec

Firstly, applications request what kind of resources they need from the network, e.g. "I need a maximum delay of 100 ms". This information is provided to the network in the form of a **flowspec**, where the word *flow* means a set of packets associated with a single application.

Admission control

Secondly, before the network provides the application with its requested service, it will check if it actually has enough resources. This is known as **admission control**.

Resource reservation (signaling)

We need a communication mechanism for the above requests and responses to take place. This is called **resource reservation**. The communication protocol used for this purpose is the *Resource Reservation Protocol (RSVP)* (see below).

Packet scheduling

Finally, after the flowspec has been described and admission control decisions have been made, the hardware (routers and switches) perform the requested resource allocation, termed **packet scheduling**.

Resource Reservation Protocol (RSVP)

The Resource Reservation Protocol (RSVP) is a transport layer protocol designed to reserve resources across a network using the integrated services model. RSVP operates over an IPv4 or IPv6 and provides receiver-initiated setup of resource reservations for multicast or unicast data flows. [5]

RSVP has a few characteristics suitable for signaling resource reservation information [cnasa525]:

Robustness

RSVP uses soft-state, relying little on state stored in routers. Unused state is deleted automatically following a time period.

Multicast support

RSVP supports multicast flows (one to many receivers) just as well as unicast flows. This is done by allowing receivers to keep track of senders in a *receiver-oriented* design.

Coarse-grained approach: Differentiated Services

Differentiated services, or DiffServ, rely on traffic classifying themselves instead of applications reserving resources for their traffic. This is achieved by marking packets with special bits according to their priority.

Differentiated services define the layout of the TOS (Type of service) byte (DS field) in the IPv4 header according to RFC 2474 [6]. The standard also defines a base set of *per-hop behaviors*, or PHBs, which govern how different types of packets are to be treated. Two prevalent per-hop behaviours, defined by the IETF, are:

- The Expediated forwarding (EF) PHB
- The Assured Forwarding (AF) PHB

The Expedited Forwarding (EF) PHB

The idea behind expedited forwarding is very simple. Two classes of service are available: regular and expedited. The vast majority of the traffic is expected to be regular, but a limited fraction of the packets are expedited. The expedited packets should be able to transit the network as though no other packets were present. In this way they will get low loss, low delay and low jitter service—just what is needed for VoIP. A symbolic representation of this ‘‘two-tube’’ system is given in the figure below. Note that there is still just one physical line. The two logical pipes shown in the figure represent a way to reserve bandwidth for different classes of service, not a second physical line. [7]

[fig1] | *fig1.png*

Figure 1. Expediated vs regular packets

The Assured Forwarding (AF) PHB

Assured forwarding specifies that there shall be four priority classes, each class having its own resources. The top three classes might be called gold, silver, and bronze. In addition, it defines three discard classes for packets that are experiencing congestion: low, medium, and high. Taken together, these two factors define 12 service classes. [8]

The steps for AF are:

1. Classify the packets into one of the four priority classes by a **classifier**.
2. Determine the discard class for each packet. This is done by passing the packets of each priority class through a traffic **policer** such as a token bucket.
3. Processed by **routers** with a packet scheduler that distinguishes different classes, usually using weighted fair queuing.

[fig2] | *fig2.png*

Figure 2. Steps of assured forwarding

Critical evaluation of QoS mechanisms

Integrated services vs Differentiated services

Integrated services was founded on the concept that all flow-related state information should be in the end systems [9]. The problems with the integrated services architecture are:

- The amount of state information increases proportionally with the number of flows. This places a huge storage and processing overhead on the routers. Therefore, this archiecture does not scale well in the Internet core.
- The requirement on routers is high. All routers must have RSVP, admission control, M F classification, and packet scheduling.
- Ubiquitous deployment is required for guaranteed service. If some parts of the network do not support RSVP, then the whole system fails to work as intended.

Differentiated services, developed after integrated services, aim to address these shortcomings.

Differentiated services is significantly different from integrated services. First, there are only a limited number of service classes indicated by the DS field. Since service is allocated in the granularity of a class, the amount of state information is proportional to the number of classes rather than the number of flows. Differentiated services is therefore more scalable. Second, sophisticated classification, marking, policing, and shaping operations are only needed at the boundary of the networks. ISP core routers need only to have behavior aggregate (BA) classification. *Therefore, it is easier to implement and deploy differentiated services*.

Integrated services + differentiated services

IntServ model can provide a QoS (quality of service) guarantee for end-to-end service, there is a limitation in extensibility of the resource reservation process according to RSVP (resource reservation protocol). DiffServ model cannot provide a QoS guarantee for end-to-end service in the whole network, but it can meet requirements of practical application for extensibility. The interoperation of these two approaches seems to be a promising solution to provide end-to-end QoS in a scalable way, as researched by Almesberger in [10] and Liu in [11].

The basic idea is to use the DiffServ approach in the core network and the IntServ + RSVP in the access network [10]. In this scenario, a key-role is played by interworking devices, called Edge Devices (ED), placed at the borders between these domains (see figure).

[fig3] | *fig3.png*

Figure 3. Combining IntServ and DiffServ

As in Figure 4, in the control plane the ingress ED receives RSVP PATH messages from the RSVP sources, stores the “PATH state” and forwards the messages towards the destination. The DiffServ routers in the core simply ignore the RSVP messages and forward them transparently (i.e. without processing them). When the PATH message reaches the egress ED, that is again RSVP capable, it is interpreted and forwarded toward the final destination. Note however that the concept of ingress or egress ED depends upon the flow’s direction; an ED can be contemporarily ingress ED for a flow and egress ED for another one (in the opposite direction). The same procedure applies to RSVP RESV messages, which are received by the egress ED and sent upstream to the ingress ED. Upon the reception of the first RESV message related to a given flow the ingress ED performs a flow admission control procedure related to the DiffServ cloud. If the admission procedure is successful, the ingress ED sends back the RESV message towards the sender host. On the other side, in the user plane, the ingress ED receives the IP packets related to a given flow, maps them into the appropriate DiffServ class, and forwards them into the DiffServ network towards the Egress ED. The routers within the DiffServ cloud route the IP packets toward the Egress ED with DiffServ-based scheduling (i.e. without changes with respect to a classical DiffServ router). Finally the egress ED behaves as a normal RSVP router and manages IP packets on a per flow basis.

[fig4] | *fig4.png*

Figure 4. RSVP paths in the combined network

Performance of the combined system

We can observe Harju’s work [12] in order to get an idea of the effectiveness of a combined IntServ and DiffServ network.

[fig8] | *fig8.png*

Figure 5. Harju’s test setup

The experiment was performed by monitoring a 42 kbit per second flow from a workstation 1 to workstation 5. The 3 figures below measure delay over time, with high values meaning high *delay* and flunctuating values meaning high *jitter*.

Figure 5 presents the worst case delay characteristics. The best- effort network is overloaded and both FIFO queues, at the access node and at the core node, are constantly full.

[fig5] | *fig5.png*

Figure 6. Delay of the test flow in the overloaded best-effort network

In the RSVP configuration, Figure 6, we can see a considerable decrease in delay.

[fig6] | *fig6.png*

Figure 7. Delay of the test flow in the overloaded all-IntServ network

Finally, both delay and jitter are reduced in the combined network scheme.

[fig7] | *fig7.png*

Figure 8. Delay of the test flow in the (overloaded) combined IntServ and DiffServ network

The measurements showed that the combination of IntServ and DiffServ architectures can provide the same level or even better QoS than pure IntServ model. In addition, the scalability problems of IntServ were realized and the different schedulers were studied. The test network was very simple and the traffic patterns were not realistic, representing the worst case situation. In addition, the impacts of the real-time traffic to the background traffic were not studied. This should be taken into account in the real network provisioning.

Conclusion

We have given a brief introduction to QoS, the need for QoS and the mechanisms with which we can provide QoS, along with an analysis of a combined QoS technique. More efficient QoS techniques are being demanded as our global internet grows and newer services and applications are being developed. Concerns such as bandwidth, scalability and latency are becoming bigger issues as we move more towards a digital-dominant world, as is evident from the growth of videoconferencing during the COVID-19 pandemic. It is therefore necessary to develop new, or combine existing QoS mechanisms in novel ways, to meet the demands of modern applications.

References

- [1] A. S. Tanenbaum and D. Wetherall, *Computer Networks* (Pearson, 2014), 404.
- [2] Tanenbaum, *Computer Networks*, 405.
- [3] L. L. Peterson, B. S. Davie, *Computer Networks: A Systems Approach* (Morgan Kaufmann, 2021), 518.
- [4] Peterson, *Computer Networks: A Systems Approach*, 518.
- [5] Wikipedia, *Resource Reservation Protocol*, https://en.wikipedia.org/wiki/Resource_Reservation_Protocol.
- [6] K. Nichols et al., *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, RFC 2474 (Network Working Group, 1998).
- [7] Tanenbaum, *Computer Networks*, 422.
- [8] Tanenbaum, *Computer Networks*, 423.
- [9] X. Xiao and L. M. Ni, *Internet QoS: A Big Picture* (1999).
- [10] W. Almesberger et al., *Combining IntServ and DiffServ under Linux* (2000).
- [11] J. Liu, *Design and Implementation of Vo IPQoS Model Combining IntServ and DiffServ Based on Network Processor IXP2400* (IEEE, 2021).
- [12] J. Harju and P. Kivimäki, *Co-operation and Comparison of DiffServ and IntServ: Performance Measurements* (IEEE, 2000).