1

The QOS World

Quality of Service (QOS) has always been in a world of its own, but as the technology has been refined and has evolved in recent years, QOS usage has increased to the point where it is now considered a necessary part of network design and operation. As with most technologies, large-scale deployments have led to the technology becoming more mature, and QOS is no exception.

The current trend in the networking world is convergence, abandoning the concept of several separate physical networks in which each one carries specific types of traffic, moving toward a single, common physical network infrastructure. This is old news for the Internet and other service providers, however, a novelty in other realms such as the Data Center. The major business driver associated with this trend is cost reduction: one network carrying traffic and delivering services that previously demanded several separate physical networks requires fewer resources to achieve the same goal.

One of the most striking examples is voice traffic, which was previously supported on circuit-switched networks and is now delivered on the "same common" packet-switched infrastructure. Also, in modern Data Centers the operation of a server writing into the hard drive, the disk, is done using a networking infrastructure that is shared with other traffic types.

The inherent drawback in having a common network is that the road is now the same for different traffic types, which poses the challenge of how to achieve a peaceful coexistence among them since they are all competing for the same network resources.

QOS-Enabled Networks: Tools and Foundations, Second Edition. Miguel Barreiros and Peter Lundqvist.

Allowing fair and even competition by having no traffic differentiation does not work because different types of traffic have different requirements, just like an ambulance and a truck on the same road have different needs. There is always the temptation of simply making the road wider, that is, to deploy network resources in an over-provisioned manner following the logic that although the split of resources was not ideal, so many free resources would be available at all times that the problem would be minimized. However, this approach has some serious drawbacks. First, in certain networks, the traffic flows and patterns are not predictable making it impossible to know the required resources beforehand. Secondly, it works against the major business driver behind network convergence, which is cost reduction. And third, such over-provisioning needs to be done not only for the steady state but also to take into account possible network failure scenarios.

QOS does not widen the road. Rather, it allows the division of network resources in a nonequal manner, favoring some and shortchanging others instead of offering an even split of resources across all applications. A key point with QOS is that a nonequal split of resources implies that there cannot be "win–win" situations. For some to be favored, others must be penalized. Thus, the starting point in QOS design is always to first select who needs to be favored, and the choice of who gets penalized follows as an unavoidable consequence.

In today's networks, where it is common to find packet-oriented networks in which different types of traffic such as voice, video, business, and Internet share the same infrastructure and the same network resources, the role of QOS is to allow the application of different network behaviors to different traffic types.

Hence, for a specific traffic type, two factors must be considered, characterizing the behavior that the traffic requires from the network and determining which QOS tools can be set in motion to deliver that behavior.

1.1 Operation and Signaling

The QOS concept is somewhat hard to grasp at first because it is structurally different from the majority of other concepts found in the networking world. QOS is not a standalone service or product but rather a concept that supports the attributes of a network by spanning horizontally across it.

QOS can be split into two major components: local operation and resource signaling. Local operation is the application of QOS tools on a particular router (or a switch, a server, or any QOS-capable device).

Resource signaling can be defined as the tagging of packets in such a way that each node in the entire path can decide which QOS tools to apply in a consistent fashion to assure that packets receive the desired end-to-end QOS treatment from the network.

These two components are somewhat similar to the IP routing and forwarding concepts. Routing is a task performed jointly by all routers in the network. All routers exchange information among them and reach a consistent agreement in terms of the end-to-end path that packets follow. As for forwarding, each router performs the task individually and independently from the rest of the network using only local information.

Routing is comparatively more complex than forwarding, because it involves cooperation among all the routers in the network. However, routing does not need to work at wire speed. Forwarding is simpler. It is a task performed by a router individually and independently. However, it must operate at wire speed.

An analogy between routing and forwarding, and QOS resource signaling and local operation, can be drawn. QOS resource signaling is somewhat analogous to the routing concept. It involves all routers in the network but has no requirement to work at wire speed. QOS local operation is analogous to the forwarding concept. Like forwarding, QOS local operation is, in concept, simpler, and each router performs it independently and individually. Also, QOS local operation must operate at wire speed.

However, there is a major difference between QOS resource signaling and routing; there are no standardized specifications (such as those which exist for any routing protocol) regarding what is to be signaled, and as a result there is no standard answer for what should be coded on all network routers to achieve the desired end-to-end QOS behavior. The standards in the QOS world do not give us an exact "recipe" as they do for routing protocols.

1.2 Standards and Per-Hop Behavior

The two main standards in the IP realm that are relevant to QOS are the Integrated Services (IntServ) and the Differentiated Services (DiffServ). IntServ is described in RFC1633 [1] and DiffServ in RFC2475 [2].

IntServ was developed as a highly granular flow-based end-to-end resource reservation protocol, but because of its complexity, it was never commonly deployed. However, some of its concepts have transitioned to the MPLS world, namely, to the Resource Reservation Protocol (RSVP).

The DiffServ model was developed based on a class scheme, in which traffic is classified into classes of service rather than into flows as is done with IntServ. Another major difference is the absence of end-to-end signaling, because in the DiffServ model each router effectively works in a standalone fashion.

With DiffServ, a router differentiates between various types of traffic by applying a classification process. Once this differentiation is made, different QOS tools are

applied to each specific traffic type to effect the desired behavior. However, the standalone model used by DiffServ reflects the fact that the classification process rules and their relation to which QOS tools are applied to which type of traffic are defined locally on each router. This fundamental QOS concept is called per-hop behavior (PHB).

With PHB, there is no signaling between neighbors or end to end, and the QOS behavior at each router is effectively defined by the local configuration on the router. This operation raises two obvious concerns. The first is how to achieve coherence in terms of the behavior applied to traffic that crosses multiple routers, and the second is how to propagate information among routers.

Coherence is achieved by assuring that the routers participating in the QOS network act as a team. This means that each one has a consistent configuration deployed which assures that as traffic crosses multiple routers, the classification process on each one produces the same match in terms of which different traffic types and which QOS tools are applied to the traffic.

Unfortunately, the PHB concept has its Achilles' heel. The end-to-end QOS behavior of the entire network can be compromised if a traffic flow crosses a number of routers and just one of them does not apply the same consistent QOS treatment, as illustrated in Figure 1.1.

In Figure 1.1, the desired behavior for the white packet is always to apply the PHB A. However, the middle router applies a PHB different from the desired one, breaking the desired consistency across the network in terms of the QOS treatment applied to the packet.

The word *consistent* has been used frequently throughout this chapter. However, the term should be viewed broadly, not through a microscopic perspective. Consistency does *not* mean that all routers should have identical configurations. Also, as we will see, the tools applied on a specific router vary according to a number of factors, for example, the router's position in the network topology.

The second challenge posed by the PHB concept is how to share information among routers because there is no signaling between neighbors or end to end. Focusing on a single packet that has left an upstream router and is arriving at the downstream router, the first task performed on that packet is

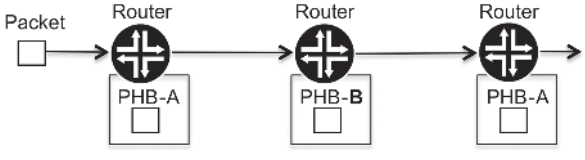

Figure 1.1 End-to-end consistency

classification. The result of this classification is a decision regarding which behavior to apply to that packet. For instance, if the upstream router wants to signal information to its neighbor regarding this specific packet, the only possible way to do so is to change the packet's contents by using the rewrite QOS tool, described in Chapter 2. Rewriting the packet's content causes the classification process on the downstream router to behave differently, as illustrated in Figure 1.2.

However, the classification process on the downstream router can simply ignore the contents of the packet, so the success of such a scheme always depends on the downstream router's consistency in terms of its classifier setup. A somewhat similar concept is the use of the multi-exit discriminator (MED) attribute in an External Border Gateway Protocol (EBGP) session. The success of influencing the return path that traffic takes depends on how the adjacent router deals with the MED attribute.

Although it does pose some challenges, the DiffServ/PHB model has proved to be highly popular. In fact, it is so heavily deployed that it has become the de facto standard in the QOS realm. The reasons for this are its flexibility, ease of implementation, and scalability, all the result of the lack of end-to-end signaling and the fact that traffic is classified into classes and not flows, which means that less state information needs to be maintained among the network routers. The trade-off, however, is the lack of end-to-end signaling, which raises the challenges described previously. But as the reader will see throughout this book, these issues pose no risk if handled correctly.

As an aside, in Multiprotocol Label Switching (MPLS) networks with Traffic Engineering (TE), it is possible to create logical paths called label-switched paths (LSPs) that function like tunnels across the network. Each tunnel has a certain amount of bandwidth reserved solely for it end to end, as illustrated in Figure 1.3.

What MPLS-TE changes in terms of PHB behavior is that traffic that is placed inside an LSP has a bandwidth assurance from the source to the destination. This means, then, that in terms of bandwidth, the resource competition is limited to traffic inside that LSP. Although an MPLS LSP can have a bandwidth reservation,

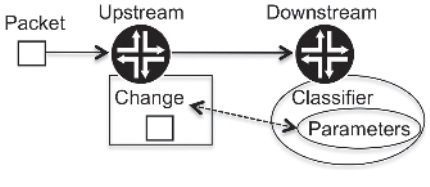

Figure 1.2 Signaling information between neighbors

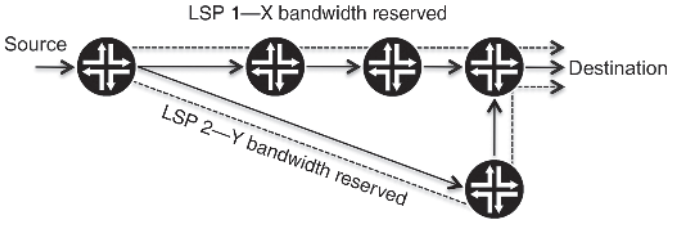

Figure 1.3 MPLS-TE bandwidth reservations

it still requires a gatekeeper mechanism at the ingress node to ensure that the amount of traffic inserted in the LSP does not exceed the reserved bandwidth amount.

Another difference is that MPLS-TE allows explicit specification of the exact path from the source to the destination that the traffic takes instead of having the forwarding decision made at every single hop. All other PHB concepts apply equally to QOS and MPLS.

MPLS is a topic on its own and is not discussed more in this book. For more information, refer to the further reading section at the end of this chapter.

1.3 Traffic Characterization

As we have stated, different traffic types require that the network behave differently toward them. So a key task is characterizing the behavioral requirements, for which there are three commonly used parameters: delay, jitter, and packet loss.

For an explanation of these three parameters, let's assume a very simplistic scenario, as illustrated in Figure 1.4. Figure 1.4 shows a source and an end user connected via a network. The source sends consecutively numbered packets 1 through 3 toward the end user. Packet 1 is transmitted by the source at time t_1 and received by the end user at the time r_1. The same logic applies for packets 2 and 3. A destination application is also present between the end user and the network, but for now its behavior is considered transparent and we will ignore it.

Delay (also commonly called latency) is defined as the time elapsed between the transmission of the packet by the source and the receipt of the same packet by the destination (in this example, the end user). In Figure 1.4, for packet 1, delay is the difference between the values r_1 and t_1, represented by the symbol Δ_1, and is usually measured in milliseconds.

Jitter represents the variation in delay between consecutive packets. Thus, if packet 1 takes Δ_1 to transit the network, while packet 2 takes Δ_2, then the jitter between packets 1 and 2 can be seen as the difference between Δ_1 and Δ_2 (also measured in milliseconds).

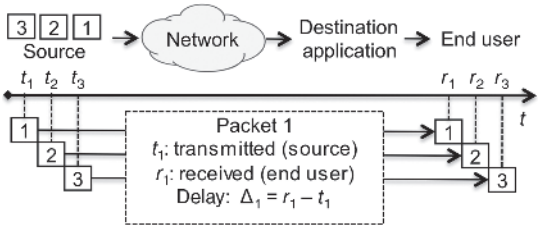

Figure 1.4 Delay, jitter, and packet loss across the network

The other parameter paramount to QOS traffic characterization is packet loss. This parameter represents how many packets are not received compared with the total number of packets transmitted and is usually measured as a percentage.

In terms of the sensitivity that traffic has to these three parameters, it is important to differentiate between real-time and nonreal-time traffic. There is also a third special traffic type, storage, but due to its uniqueness it is detailed in a dedicated section in Chapter 4. For real-time traffic, the main focus sensitivities are generally delay and jitter. So let's start with these two parameters, and we'll focus on packet loss a little later on.

Delay is important because real-time packets are relevant to the destination only if they are received within the time period in which they are expected. If that time period has expired, the packets become useless to the destination. Receiving them not only adds no value but also has a negative impact because although the packet is already useless, receiving it still demands processing cycles at the destination.

Jitter can also be very problematic because it interrupts the consistency of the delay of the packets arriving at destination. This interruption poses serious problems to the application receiving the traffic by forcing it to be constantly adapting to new delay values. Practical experience from voice-over-IP (VoIP) deployments shows that users migrating from a circuit-switched network can easily get used to a delay value even slightly higher than what they previously had as long as it is constant. However, the presence of significant jitter immediately generates user complaints. The bottom line is that when the delay value is always changing, users (and applications) cannot get used to it because it is not constant.

Although the previous descriptions are generally applicable to various types of real-time traffic, they should not all be placed under the same umbrella, because the exact set of requirements depends on the application itself. For example, if the application using real-time traffic is unidirectional, buffering can be used at the destination to reduce the presence of jitter.

Looking again at Figure 1.4, assume that the traffic sent by the source to the end user is a unidirectional video stream. Also assume that the destination application

placed between the network and the end user has a buffer that enables it to store the packets being received, thus allowing the application to decide when those packets should be delivered to the end user.

Assuming a buffer of 1000 ms at the destination application (enough to store all three packets), then by delivering each packet at a separation of 300 ms, which is the average delay, the jitter value experienced by the end user is zero, as illustrated in Figure 1.5.

The drawback to this solution is that it introduces delay, because packets are stored inside the destination application for a certain amount of time and are not immediately delivered to the end user. So there is a trade-off between reducing jitter and introducing delay.

As for the packet loss parameter, a packet of a real-time stream is useful for the destination only if received within a certain time period, a requirement that tends to invalidate any packet retransmission mechanism by the source in case of packet loss. Hence, it is no surprise that the User Datagram Protocol (UDP), a connectionless protocol, is commonly used for the transmission of real-time streams.

Different real-time applications have different levels of sensitivity to packet loss. For example, video applications generally display minor glitches or blocking when low-level loss occurs, but large packet loss can cause total loss of the picture. Similarly, for voice applications, a low-level loss generally causes minor clicks with which the human ear is perfectly capable of dealing. However, large-scale loss can simply cause the call to drop. Finding where to draw the line between what is an acceptable packet loss and what is a catastrophic packet loss scenario is highly dependent on the application.

For nonreal-time traffic, generally speaking, the sensitivity to jitter and delay is obviously much lower, because there is not such a strong correspondence between when the packet is received and the time interval in which the packet is useful for the destination.

As for packet loss, a split can be made regarding whether the application uses a connection-oriented protocol, such as the Transmission Control Protocol (TCP), or a connectionless protocol, such as UDP, for transport at OSI Layer 4. In the first scenario (TCP), the transport layer protocol itself takes care of any

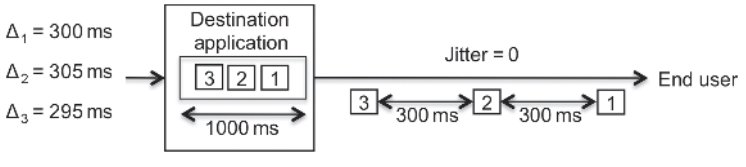

Figure 1.5 Jitter reduction by using a buffer at the destination application

necessary packet retransmissions, while in the second scenario (UDP), the session layer (or a layer higher in the OSI stack) must handle the packet loss.

Another scenario is the network being lossless, meaning that the network itself will assure that there will not be any packet loss, thus removing the need for the transport or higher layers having to worry about that. The concepts of lossless network, UDP, and TCP are detailed further in Chapter 4.

As a teaser for the following chapters, we stated earlier in this chapter that QOS allows implementation of an unfair resource-sharing scheme across different traffic types. In these unfair schemes, offering benefit to some implies impairing others. So, for example, if real-time traffic is more sensitive to delay and jitter, QOS can allow it to have privileged access to network resources in terms of less delay and less jitter. Of course, this is achieved at the expense of possibly introducing more delay and jitter in other traffic types, which can be acceptable if they have higher tolerances to delay and jitter.

1.4 A Router without QOS

A useful starting point is to analyze the effects of the absence of QOS, which acts to provide a perspective on what the end result is that we want to achieve by the change.

In the scenario of a router without QOS enabled, the order of traffic present at the ingress interface is identical to the order of traffic as it leaves the router via the egress interface, assuming that no packet loss occurs, as illustrated in Figure 1.6.

Figure 1.6 shows two types of traffic, white and black, each one with three packets numbered 1 through 3. If QOS is not enabled on the router, the output sequence of packets at the egress interface is the same as it was at the ingress.

One of the many things that QOS allows is for a router to change that output sequence of packets with great exactness. Suppose black packets correspond to sensitive traffic that should be prioritized at the expense of delaying white packets, a behavior illustrated in Figure 1.7.

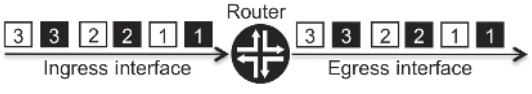

Figure 1.6 Traffic flow across a router without QOS

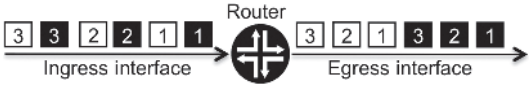

Figure 1.7 Router with QOS enables packet prioritization

To achieve this result requires differentiation: the router must be able to differentiate between white and black packets so it can make a decision regarding the output sequence order. This differentiation is achieved by classifying traffic into different classes of service.

1.5 Conclusion

QOS is all about not being fair when dividing the network resources but rather selecting discriminately who is favored and who gets penalized. QOS does not make the road wider; it just decides who goes first and as a consequence who has to wait.

In terms of standards, the key is the PHB concept, in which each router acts independently from the rest of the network in terms of the behavior it applies to the traffic that crosses it. PHB obligates the routers to consistently apply the desired behavior to each traffic type that crosses it in spite of its independent decision making.

In terms of the parameters used to define the traffic requirements, the more common ones are delay, jitter, and packet loss. The tolerance to these parameters is highly dependent on whether the traffic is real time or not, because for real-time traffic, the time gap in which the packet is considered useful for the destination is typically much narrower. An interesting development detailed in Chapter 4 is storage traffic, since it has zero tolerance regarding packet loss or reordering.

The chapters that follow in this book present the tools and challenges to achieve such traffic differentiation inside the QOS realm.

References

[1] Braden, R., Clark, D. and Shenker, S. (1994) RFC 1633, Integrated Services in the Internet Architecture: An Overview, June 1994. https://tools.ietf.org/html/rfc1633 (accessed August 19, 2015).

[2] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W. (1998) RFC 2475, Architecture for Differentiated Services, December 1998. https://tools.ietf.org/html/rfc2475 (accessed August 19, 2015).

Further Reading

Minei, I. and Lucek, J. (2011) *MPLS-Enabled Applications*. New York: John Wiley & Sons, Inc.