CHAPTER 11

IPv6 and Multimedia Traffic

The transportation of multimedia traffic on IP networks is a topical subject because multimedia is becoming cheaper and cheaper and therefore used more and more. All workstations and personal computers available today are equipped with sound boards for recording and reproducing sounds and with video boards for viewing MPEG images[1]. Some of them are now equipped with video input and with small video cameras.

Problems with bearing multimedia flows on IP networks are mainly related to the bandwidth they require and to the strict maximum delay requirements that must be met. This second point is particularly important when multimedia applications have to provide users with real-time interaction.

In the past few years, many experiments have been made to develop a network layered on the Internet for multimedia applications; this network is called Mbone[2]. These experiments have highlighted the intrinsic multicast nature of multimedia traffic (from a source toward many destinations) and therefore the need to improve the routing of multicast packets on IP networks.

Some characteristics of IPv6 will improve the support of multimedia applications (in the following, also called *real-time applications*), such as the availability of the Priority field and of the Flow Label field on the IPv6 header (see Sections 3.1.2 and 3.1.3) and the availability of a large addressing space reserved for multicast addresses (see Section 4.8).

Moreover, other protocols of the stack introduce significant rationalizations in this field. ICMPv6 includes functions for the management of multicast groups (see Section 5.5.3), and OSPFv6 provides the treatment of multicast trees, formerly supported by DVMRP and MOSPF (see Section 7.4.2).

All the innovations cited here aren't enough to solve the problems of using multimedia on networks. IPv6 is part of a more ambitious project called IS (*Integrated Service*) Internet, which is discussed in RFC 1633[3]; it aims to extend the Internet architecture to allow the bearing of either best-effort or real-time traffic, as well as to control the use of transmission links (controlled link sharing).

The *best-effort* traffic is the only type of traffic that has been used on the Internet till now. It is based on the idea that the network's task is to do everything possible to deliver each IP packet, without guaranteeing the packet is delivered or the delivery time.

Multimedia applications frequently generate *real-time* traffic—that is, a type of traffic sensitive to queuing delays and to losses due, for example, to network overloading. Moreover, this type of traffic frequently needs a guaranteed minimum bandwidth.

The possibility of reserving a minimum bandwidth on links for particular classes of users, or protocol stacks, is in general a requirement understood by network administrators, also independently from multimedia applications.

Clearly, typical real-time applications—like the transmission of remote video images, multimedia conferences, and virtual reality—require the extension of IP by introducing the concept of *Quality of Service* (QoS). The extension must in some way allow limited packet delays and must be designed, from the beginning, for IP multicast because most of the multimedia traffic is multicast.

The architectural extension proposed by the IETF includes the following two elements:

- The extended service model, identified by the acronym *IS* (Integrated Services)
- Its possible implementation structure

We should clearly distinguish the model of service, which defines the external behavior, from one of its possible implementations, which can and should change during the life of the model of service itself.

11.1 The Integrated Services Model

The possibility of providing QoS is strictly related to the ability to administer the network resources (for example, the bandwidth). Introducing either resource reservation mechanisms or acceptance/refusal of service request mechanisms (*admission control*) is essential on the basis of the requested QoS and of available resources. A resource reservation accepted by the network guarantees a service whose quality meets the desired requirements and therefore guarantees the application will operate acceptably.

Nevertheless, the introduction of resource reservation mechanisms on the Internet is not accepted by everybody. Some people assert that the resource reservation is only a method to administer resource shortages; to allocate resources to a user means to deprive all other users and therefore to dissatisfy them. Network administrators will soon discover that the real solution consists of the availability of more resources, not in the introduction of reservation or invoice schemes.

Some detractors of this idea also produced the following arguments:

- *In the future, the bandwidth will be infinite.* New transmission techniques—in particular, fiber optics—cause some people to think that in the near future the bandwidth will be so big, widespread, and cheap to be considered infinite. Therefore, reserving network resources wouldn't be necessary.

- *Simple priority schemes are enough.* We have already seen that the IPv6 header has a Priority field used both to distinguish the real-time traffic from the best-effort traffic and to provide different types of real-time traffic with different priorities. The use of this field could only bring adequate real-time service in certain periods and under certain conditions. But the priority is an implementation mechanism, not a model of service!
- *Applications can be adapted to the present traffic of the network.* Techniques can be used to develop real-time applications that can be adapted to the variations of the load on the network. These techniques have been little used until today, but they will be the basis of new multimedia applications.

It is the author's opinion that these items will undoubtedly have a considerable impact on networks in the future, but that they are not sufficient to guarantee real-time services on the entire Internet. In fact, on the one hand, it is true that in the United States the bandwidth will soon be practically infinite; on the other hand, it is true that the situation in Europe, due to the persistence of monopolies, is very different, and in other Eastern Europe or Asian countries, the situation is even worse.

The priority mechanism is not sufficient to guarantee the management of real-time traffic. In fact, if several packets with the same priority compete for resources, with a lack of reservations, the QoS cannot be guaranteed.

The development of adaptive real-time applications doesn't eliminate the need to reduce packet delivery time because the human need to interact and to understand limits, in some way, this capability of adaptation. For example, some voice applications can adapt themselves to delays of many seconds, but they have been shown to make the interaction between users impossible.

The logical conclusion is that routers should be able to reserve resources to provide the QoS and will therefore be modified to identify flows, to maintain state information about flows themselves, and to manage queues of packets separated by different flows. This evolution represents an important and basic change to the Internet model because the Internet architecture has been, till now, based on the concept that the state relevant to various flows should be managed by hosts only[4].

11.2 Coding of Multimedia Information

Before examining a possible implementation of the IS architecture, we need to analyze the adaptive applications mentioned in the preceding sections. The first step for the implementation of these multimedia applications is the elimination of the redundancy in the information, usually obtained through compression algorithms. A disadvantage of this operation, which is essential for reducing the bandwidth necessary for transmission, is that the compression unavoidably introduces delays. Therefore, the choice of the compression algorithm must take into account how much delay it introduces and which is the application typology. For applications such as television broadcasting (which is devoid of interactivity), the introduced delay can be also very high, allowing the use of compression schemes with high compression rates or that favor the quality of images. On the other hand, for videoconference applications (in which a good level of interactivity is necessary), low-delay compression schemes must be favored. Another factor to be considered is whether the compression scheme transmits exactly the same image it received (compression without loss) or an approximation of it (compression with loss). Compression schemes with loss are suitable for videoconference and entertainment applications, but if transmitting X-rays or other medical images is necessary, choosing a compression scheme without loss is advisable, to avoid the risk of wrong diagnoses.

After the redundancy is eliminated, we can reintroduce it in the form of error correction codes. In fact, real-time requirements of many multimedia applications make the retransmission of an erroneous packet impossible because the transmission will then be useless. The only possibility is to increase the redundancy of essential information through codes that allow automatic correction of a certain number of errors during the reception of the erroneous packet.

Until now, we have considered acceptable and unacceptable delays without providing numerical information. The ITU 114 standard "General Delay Recommendation" defines as acceptable delays up to 150 ms, delays between 150 and 400 ms acceptable for some applications, and those delays higher than 400 ms generally unacceptable.

The design of applications must take into account from the beginning that the QoS cannot be guaranteed in particular circumstances; therefore, the coding of the information must be designed to always provide a minimum service, even if a low-quality service.

This service can be implemented through *hierarchical coding*. Let's suppose we want to transmit a numerical flow of CD quality with a 44 KHz sample and samples on 16 bits. Instead of coding the sound as a unique flow of data, subdividing it into the following four subflows to be transmitted with decreasing priorities makes more sense:

- A basis flow coded at 5.5 KHz
- A flow containing differences between 5.5 KHz and 11 KHz
- A flow containing differences between 11 KHz and 22 KHz
- A flow containing differences between 22 KHz and 44 KHz

The network will try to transport all four flows to the destination in time. In case of congestion, however, the network will begin to discard packets belonging to the last flow, then to the next-to-last flow, and so on, guaranteeing the best possible service consistent with the state of congestion of the network.

11.3 Reference Implementation

We have seen that the router is the component that needs more modifications to implement IS Internet. Let's analyze the possible architecture of the router shown in Figure 11-1.

Notice that the router is ideally subdivided into two parts: the *forwarding path* (lower part) and the *background code* (upper part). The additional blocks, in comparison with a common router, are the *packet scheduler*, the *admission control agent*, the *classifier,* and the *reservation setup agent*. These blocks operate on data flows, and this concept is clearly present in IPv6 (see Sections 1.2.8 and 3.1.3).

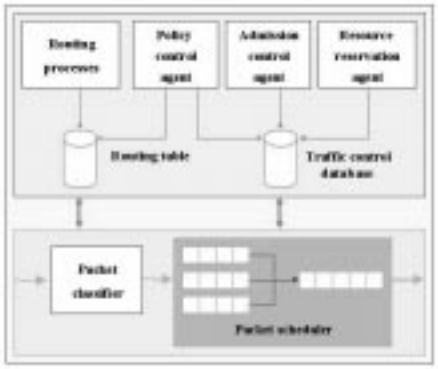

Figure 11-1
Architecture of an IS router

Current routers are designed for best-effort traffic; therefore, they treat packets with a simple FIFO (First In, First Out) queuing for each egress line (see Figure 11-2).

As for integrated services, a router must provide an appropriate QoS for each flow, and it must therefore be equipped with a module for the traffic control. This module consists of the following three submodules:

- The *packet scheduler* guarantees the QoS administering the transmission of packets through a mechanism for the periodical visit of a set of queues.
- The *packet classifier* recognizes which flow a packet belongs to and queues it on the corresponding queue. A queue can be associated with a single flow or to a class of flows.
- The *admission control* decides, in response to a request of resource reservation from the reservation agent, whether this packet can be accepted. The decision is made on the basis of resources reserved by other flows, of the network administration policies set through the control agent, and of globally available resources. In practice, this module checks whether the requested QoS can be provided without colliding with the guarantees of service provided to other flows.

The presence of a classification module and of a packet scheduler requires that each egress line be associated with a set of queues. An example of this association is shown in Figure 11-3.

The presence of a set of queues is a necessary, but not sufficient, characteristic to guarantee the QoS. It is, in fact, necessary that the scheduler guarantees that the frequency with which each queue is served is greater than or equal to that guaranteed during the resource reservation. This forces us to have a separate queue for each real-time flow (in the example, R1, R2, R3 e R4) and a shared queue for the best-effort traffic. The best-effort traffic will clearly be penalized, and it will be served only in the absence of real-time traffic.

The model of admission control is sometimes confused with the so-called *policing*, a control mechanism that checks packet by packet that a

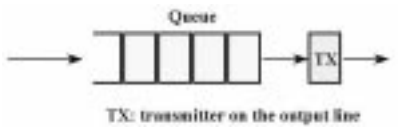

Figure 11-2
A queue for each egress line

host doesn't violate traffic characteristics agreed upon by a previous QoS agreement. In this case, the packet scheduler provides the policing.

The fourth and last component is the resource reservation protocol, which is necessary to create and maintain the state of each flow on the routing path and which allows the interaction between reservation agents. The protocol chosen by the IETF is *RSVP (Resource reSerVation Protocol)*[5, 6].

The implementation for hosts is usually similar to that of routers, but with the addition of applications. Figure 11-4 shows the interconnection between a host and a router. The host's data are received by an application that, if needing QoS for a flow, must request it from the local reservation agent (the RSVP agent).

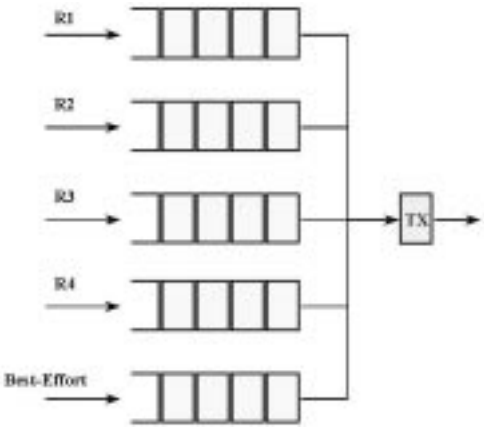

Figure 11-3
Set of queues associated with an exit

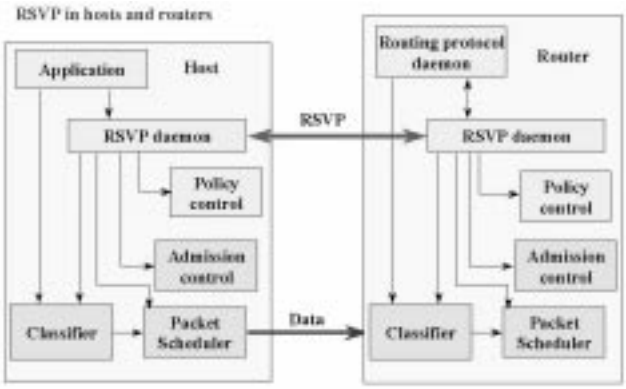

Figure 11-4
Connection between a router and a host in IS Internet

11.4 Traffic Control

Traffic control mechanisms implemented in traditional routers are very simple. But the tasks of the traffic control module of an IS router are unavoidably more complex. In particular, a network can administer its resources in two ways: through the packet scheduler and through buffer management.

11.4.1 The Packet Scheduler

The packet scheduler determines the order in which each packet is served (transmitted). It represents the main control function on how a network serves its users.

The simplest scheduling algorithm consists of ordering packets as a function of their priority. In this way, packets with higher priority are transmitted first. This method of transmission can cause an indefinite waiting period for lower priority packets if the traffic of higher priority data is very heavy.

Currently, the commonly used algorithm for the management of real-time traffic is WFQ (*Weighted Fair Queuing*)[7], which is based on a scheme similar to that shown in Figure 11-3. Each queue is associated with a weight proportional to the frequency it must be served. The packet scheduler uses weights to determine which queue must be served. The WFQ alternates the transmission of packets belonging to several flows, and for each of them, it works like a low-pass filter.

The WFQ algorithm is already available on several routers associated with a classifier; it uses information such as the protocol type or the type of application to which packets belong.

11.4.2 Buffer Management

The presence of buffers (queues) in the network is essential each time packets arrive at a speed higher than at which they can be retransmitted. Nevertheless, this setup can exist only in a transition period because, if packets arrive for a long period at a speed higher than at which they can be retransmitted, some of them must be discarded.

Packets to be discarded must not be chosen randomly, but as a function of the type of application and of services they require. These considerations,

in addition to the meaning of packets discarded, raise the need for implementing specific buffer management mechanisms for different classes of packets.

In fact, for the TCP, the indication of a discarded packet is interpreted like a signal of network congestion; it induces the protocol itself to reduce the load on the network, thus reducing the speed of packet generation at the source. For real-time applications to discard a packet involves the possibility of maintaining the quality of the desired service; that is, it helps in correctly transmitting many other packets. In fact, if an output buffer is full, discarding a packet within the buffer shortens the delay of all other packets that follow the discarded one.

11.4.3 Packet Classification

The preceding discussion on packet scheduling and on buffer management assumes that the traffic has been subdivided into classes, each of which must be treated in a specific way.

The classification must be made by analyzing many fields of the packet. In fact, the only information relevant for the forwarding process to determine the packet routing is the destination address, and this information is not sufficient to correctly classify the packet received.

We have already seen how IPv6, to reduce the elaborate overhead, marks packets with a flow identification, called a *flow label*, inserted in the IP header. This identifier can be cached in routers and used for a quick classification of packets. This technique simplifies the classification when the source station differentiates flows by marking them with different flow labels.

Nevertheless, in the initial phase of the deployment of IPv6, many applications will transmit using the default flow label (flow label = 0); therefore, it is necessary to recognize data flows in routers, by analyzing, for example, the content of several fields in the packets header such as the source address, the protocol number, or the value of the UDP port. In this way, it is possible to recognize a flow of video information through a well-known port in the UDP header, for example, or to recognize an application from the joint analysis of the TCP header's source port and destination port. Moreover, a classification can be made on the basis of information contained in upper layer packets.

In this way, it is also possible to manage the QoS for already-existing applications, without modifying them, but trying to make decisions on the basis of the header content. This second approach presents a disadvantage, which brings about the introduction of the flow label in IPv6. In fact, finding the information on ports and on applications entails processing the whole chain of headers, with a considerable computing burden, and this process can be quite complicated if the payload is encrypted (see Section 8.1.3).

11.4.4 Access Control

The technique traditionally proposed for implementing access control consists of storing all service parameters of all previous requests and making a decision based on the worst characteristics discovered for each service.

This onerous method can be replaced by another one, which allows us to obtain a better use of links. This goal is reached when each router determines the use of links from existing packet flows, and the router uses this information to accept or not accept new flows entering the network. This technique exposes the system to a higher risk of overloading, balanced by a better use of the link.

We should notice that the need for an admission control function is required by the model of service, although its implementation is not specified. For this reason, manufacturers of routers and network devices are encouraged to find better solutions that, in comparison with their competitors' solutions, allow them to find better uses of the network and a lower risk of overload.

11.5 RSVP

A resource reservation protocol must be designed to allow the network to propagate the resources requested by the different applications. The protocol chosen by the IETF is RSVP *(Resource reSerVation Protocol)*[5, 6].

RSVP can operate in a multicast environment, consisting of a set of sources that send *data* to a particular set of receivers through a distribution

tree (see Figure 11-5). The distribution tree is identified by the multicast address of the set of receivers.

RSVP supports resource reservations both for unicast applications and for multicast applications of the type "many to many," dynamically adapting itself both to variations in the composition of groups and to variations in routing paths.

RSVP is a protocol used by a host to request a specific QoS from an application. RSVP is also used by routers both to retransmit QoS requests along the entire data routing tree and to maintain the state information about flows in routers.

RSVP is a protocol for *simplex* data flows (the sender is treated in a different way from the receiver); therefore, the request of resources is unidirectional. RSVP is layered on IP (both version 4 and version 6); it doesn't transport data, but only control messages (*Path* and *Resv* messages in Figure 11-5).

In RSVP, the receiver is responsible for reservation requests (Resv messages). The sender limits itself to inform receivers about the type of transmission made through information messages (Path messages).

Moreover, a reservation setup protocol must provide a flexible control on the way resources allocated along multicast trees are shared among the different applications and manage very large multicast groups. Because these multicast groups are dynamic, being able to add or to eliminate stations to or from a group, as well as to allow the creation and the cancellation of groups, is therefore necessary.

In IPv6, these functions are provided by ICMP and OSPF. ICMP manages the participation of groups at a single link level (for example, a local area network), whereas OSPF maintains distribution trees of multicast groups among several different subnets (for example, for wide area networks).

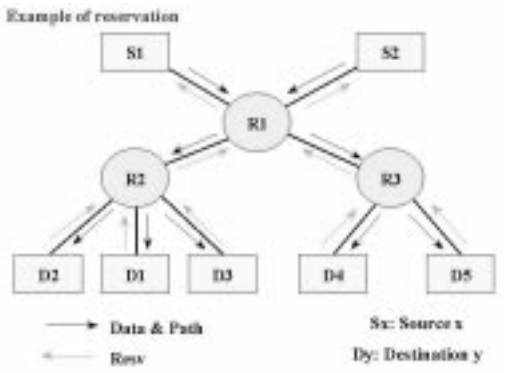

Figure 11-5
Path and Resv Messages

11.5.1 Flowspec and Filterspec

A reservation request must specify both the necessary resources, through a set of parameters called *flowspecs*, and the set of packets to which resources are allocated through a set of parameters called *filterspecs*.

If the admission control procedure gives a positive result, allowing the acceptance of the reservation request, the flowspec parameter is used to define a class of flows in the scheduler and to allocate the relative buffers. On the other hand, the filterspec parameter is used by the classifier to identify, among the packets received, those belonging to the given flow.

RSVP allows the creation and management of the necessary state information in a distributed form along the whole multicast tree. Flowspec and filterspec parameters are transported only by RSVP, leaving their interpretation to admission control functions.

11.5.2 Reservation Styles

RSVP can use different reservation styles. Differences among these styles depend on how the information about resources for a set of receivers is stored in different routers.

At present, the following three styles of reservation have been defined:

- Wildcard reservation
- Fixed filter reservation
- Shared filter reservation

The first method creates a single reservation shared by all senders' flows. We can think of this reservation like a shared channel whose size is equal to the maximum size requested by receivers and independent from the number of senders. In practice, the reservation uses the flowspec that requests the largest number of resources, among all those proposed by receivers.

This technique is particularly suitable for voice applications, such as the transmission of audioconferences, in which a limited number of sources are active at the same time and can share the same resources.

The other two methods use parameters that depend on transmission sources. These techniques are used for applications in which a determined receiver may decide to accept or not accept data flows from determined sources.

In the fixed filter reservation, the receiver requests a dedicated reservation for a particular sender that cannot be shared by other senders, even if belonging to the same multicast group. This reservation style is typically used for video flows.

In the shared filter reservation, the receiver requests a shared reservation for a set of senders that are explicitly identified. This style can be used as an alternative to the first one for voice applications.

11.5.3 Reservation by Receiver

In the RSVP protocol, resource reservation is receiver-initiated, allowing management of heterogeneous receivers in a simple way. In fact, each receiver sends a reservation request suitable to its characteristics and needs (*Resv* messages, in Figure 11-5). To do so, the receiver must have previously acquired source characteristics, in terms of flowspec, through information messages (*Path* messages, in Figure 11-5).

The reservation request is propagated on the network to sources, and each node traversed executes a resource allocation.

11.5.4 The Soft-State Approach

RSVP operates by the use of state information distributed in the routers within the network. This information is stored in special caches on routers, and these caches must be periodically updated by hosts, which must periodically repeat the reservation request.

In this way, useless information is automatically removed in case of errors with a time-out mechanism. In case the routing path has been changed, the suitable information will be automatically learned by new crossed routers by means of the periodical messages generated by RSVP.

This method is used to guarantee the robustness and the simplicity typical of the connectionless protocols used in the Internet.

11.5.5 Routing and Reservations

There is a strict connection between routing and reservation procedures because the latter requires the storage of state information along the path followed by packets. Clearly, in case of a routing change, the state information must be moved on the new path.

In general, RSVP has four main goals:

- To find a path allowing the resource allocation. This process entails the need to use a routing mechanism that differentiates the types of services.
- To find a path with enough resources for a new flow. This goal can be achieved in two different ways. The first requires a modification of routing protocols so that the new path is found on the basis of the most recent average load. The second method requires the redesign of the routing protocols to provide a series of alternative paths on which the reservation can be attempted. In both cases, obtaining dynamic routing based on the load of the network is difficult without creating instability problems. If, however, the dynamic routing is used only during the reservation, the instability doesn't create significant problems.
- To recover errors on the path. In case of failure of a node or of a link, the dynamic routing provides an alternative path. Refresh messages periodically sent by RSVP automatically request a reservation along the new path. This request can clearly fail because of the lack of available resources. This method entails an accurate management of the network configuration, that is due neither to routing protocols nor to reservation protocols used. The time necessary to create the reservation information on the new path shouldn't be too long, in order to avoid problems in the case of real-time applications.
- To implement a change of path not triggered by an error. In some cases, we also need to request a change of the path in the absence of errors. For example, this service can be used to allow the management of mobile stations within the network.

11.6 Integrated Services in an IP over ATM Architecture

Because problems of the use of IP over ATM have already been discussed in Chapter 9, in this section we will focus on aspects relevant to the QoS and in particular on analyzing how resource reservation mechanisms based on RSVP can work successfully with ATM's QoS, in a way similar to the one proposed in Figure 11-6. This description, which is based on

RFC 1821[8], analyzes only present problems without proposing organic solutions.

At a first glance, we can clearly see how the use of RSVP (and therefore of IP-QoS) by applications is much more general than the use of the ATM-QoS because it allows operation with a heterogeneous network.

The most significant issue, from the point of view of the reservation management, is that of the communication between two hosts, not directly connected to an ATM network, but using one or more ATM networks in some parts of the routing path. In this case, the entities connected to the ATM network are IP routers whose aim is to exploit different types of ATM-QoS, to guarantee the desired IP-QoS to the path between the two hosts.

IP routers, according to the description of IP-QoS, must determine whether an existing ATM connection can be used or whether a new one, with the desired characteristics, must be created.

From this example, we can deduce that the main aspects to be analyzed are the following:

- How the IP service model and the ATM service model are related
- How to translate RSVP reservation requests into ATM signaling packets
- How to execute the IP on ATM routing when QoS parameters are present

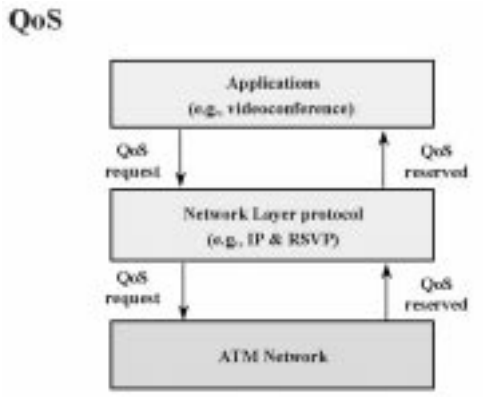

Figure 11-6
RSVP and ATM QoS

11.6.1 The Service Model

The main problem resides in the relationship between IP's QoS and ATM's QoS.

ATM provides five different classes of service:

- *CBR* (*Constant Bit Rate*)*:* For applications requiring a fixed bandwidth and delays
- *VBR-real-time* (*Variable Bit Rate*)*:* For real-time applications with variable bandwidth and with tightly constrained delays
- *VBR-non-real-time:* For variable bandwidth applications without tight delay constraints
- *UBR* (*Unspecified Bit Rate*)*:* Class of service that approximates the best-effort service of IP
- *ABR* (*Available Bit Rate*)*:* An evolved version of UBR able to control the loss rate by a flow control mechanism

The preceding classes oppose those provided by the IP model:

- *guaranteed:* Provides a guaranteed maximum delay bound
- *predictive:* Provides a probabilistic delay bound
- *controlled delay:* Provides several levels of delay from which applications can choose

When we decide the type of connection to be used to transport an IP flow, the QoS requests must be carefully evaluated. For example, we can decide to use a CBR class, or we can open a VBR connection to obtain a better use of the network resources because the IP traffic is usually burst traffic.

Another important element of the service model concerns the resource reservation. In fact, ATM uses only one signaling protocol (UNI 3.1 also called Q.2931) to request the connection and to allocate network resources at the same time. This protocol uses a sender-oriented approach—that is, requests are sent by sources. Moreover, it is based on a hard-state model, in which a connection's characteristics cannot be modified during the connection itself.

The main differences between the reservation protocol adopted by the IS Internet (RSVP) and that adopted by ATM (UNI) are as follow:

- In RSVP, the reservation request is sent by the receiver; whereas in ATM, it is sent by the sender.

- RSVP uses a soft-state approach that provides the possibility to dynamically modify the reservation. In ATM this approach is impossible.
- RSVP adopts a unidirectional allocation, whereas ATM uses a bidirectional allocation in the unicast case and a unidirectional allocation in the multicast case.
- RSVP allows the management of many senders in a unique multicast group. ATM cannot manage these operations.

In ATM, the routing and the reservation are implemented at the same time, unlike RSVP. The comparison will help us analyze the main problems to be solved:

- How to create ATM connections. Because these connections are bidirectional, the receiver could set up point-to-point connections. This solution is potentially wasteful of network resources because resources would be allocated for bidirectional transmission. The receiver must somehow request the sender to create a unidirectional point-to-multipoint connection. Because the QoS is associated with the connection, if different receivers request different QoSs, creating many point-to-multipoint connections with only one receiver is necessary. This approach, in the case of a very large multicast group, makes setting up a large number of connections necessary.
- ATM adopts a hard-state model. This means to take into account the possibility of opening and closing an ATM connection when the IP reservation is modified or released. Moreover, to optimize the use of the ATM network resources, the connection can be left open for use by other flows, or it can be closed. Frequently, the connection is left open for a subsequent use. If this connection is not sufficient to receive the new flow, a new connection can be opened to accommodate the extra traffic.
- RSVP uses control messages (Path) to convey information about sources to receivers before any data is transferred. In ATM, this solution requires a mechanism for setting up a connection whose QoS characteristics will be necessarily different from those that will probably be requested by the Resv message of RSVP.
- Finally, we need to develop security aspects to avoid a situation in which the differences between IP and ATM can allow nonauthorized users to reserve resources.

The main difficulty of implementing the IP routing on ATM in the presence of QoS parameters is that most routing protocols don't use the information about resources available on the network to determine the routing path. Some protocols, like OSPF, allow the determination of the routing depending on the ToS (Type of Service) value of the IPv4 header and on other metrics, but no protocol can manage the huge number of parameters provided by ATM.

The preceding items help us to understand the complexity of mapping the RSVP protocol on ATM.

A possible alternative consists of adapting a different protocol, called ST2[9], to ATM. It presents fewer problems than RSVP because it is based on a hard-state operation in which connections are set up by the sender, and the reservation is made during the connection setup.

The following problems must be solved to adapt ST2 on ATM:

- Managing changes to active stream reservations, which are allowed in ST2
- Avoiding the use of bidirectional connections for the management of point-to-point connections because ST2 uses unidirectional flows that would determine a waste of resources

REFERENCES

[1]ISO/IEC 13818-1, ITU H.220.0*, Information Technology—Generic Coding of Moving Pictures and Associated Audio*.

[2]S. Deering, *RFC 1112: Host Extensions for IP Multicasting*, August 1989.

[3]R. Braden, D. Clark, S. Shenker*, RFC 1633: Integrated Services in the Internet Architecture: an Overview*, June 1994.

[4]D. Clark, *The Design Philosophy of the DARPA Internet Protocols*, ACM SIGCOMM '88, August 1988.

[5]L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala*, RSVP: A New Resource ReSerVation Protocol*, IEEE Network, September 1993.

[6]R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin*, Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification*, Internet Draft, November 1996.

[7]A. Demers, S. Keshav, S. Shenker, *Analysis and Simulation of a Fair Queuing Algorithm, Journal of Internetworking: Research and Experience*, 1, pp. 3-26, 1990, also in Proc. ACM SIGCOMM '89, pp. 3-12.

[8]M. Borden, E. Crawley, *RFC 1821: Integration of Real-time Services in an IP-ATM Network Architecture*, August 1995.

[9]L. Delgrossi, L. Berger, *RFC 1819 Internet Stream Protocol Version 2 (ST2) Protocol Specification—Version ST2+*, August 1995.