Networks and Distributed Systems

Lecture 19/20 – Quality of Service (QoS)

Quality of Service

- For many years, packet-switched networks have offered the promise of supporting multimedia applications
    - Those that combine audio, video, and data.
- After all, once digitized, audio and video information become like any other form of data
    - A stream of bits to be transmitted.
- Recently, however, improvements in coding have reduced the bandwidth needs of audio and video applications, while at the same time link speeds have increased.

Quality of Service

- There is more to transmitting audio and video over a network than just providing sufficient bandwidth, however.

- Participants in a telephone conversation, for example, expect to be able to converse in such a way that one person can respond to something said by the other and be heard almost immediately.

- Thus, the timeliness of delivery can be very important. We refer to applications that are sensitive to the timeliness of data as *real-time applications.*

Quality of Service

- Voice and video applications tend to be the canonical examples, but there are others such as industrial control
  - You would like a command sent to a robot arm to reach it before the arm crashes into something.
- Even file transfer applications can have timeliness constraints
  - A database update complete overnight before the business that needs the data resumes on the next day.

Quality of Service

- The distinguishing characteristic of real-time applications is that they need some sort of assurance *from the network that data is likely to arrive on time (for some definition* of “on time”).

- Whereas a non-real-time application can use an end-to-end retransmission strategy to make sure that data arrives *correctly, such a strategy cannot provide* timeliness.

Quality of Service

- This implies that the network will treat some packets differently from others
  - something that is not done in the best-effort model.
- A network that can provide these different levels of service is often said to support quality of service (QoS).

Real-Time Applications

- Data is generated by collecting samples from a microphone and digitizing them using an A →D converter
- The digital samples are placed in packets which are transmitted across the network and received at the other end
- At the receiving host the data must be played back at some appropriate rate

Real-Time Applications

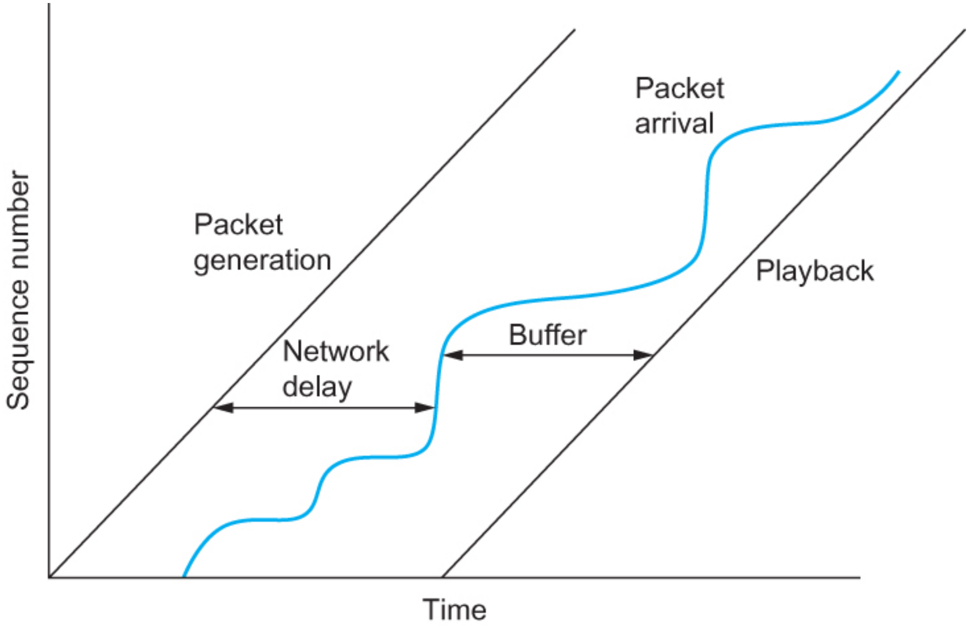
- For example, if voice samples were collected at a rate of one per 125 μs, they should be played back at the same rate
- We can think of each sample as having a particular playback time
- The point in time at which it is needed at the receiving host
- In this example, each sample has a playback time that is 125 μs later than the preceding sample
- If data arrives after its appropriate playback time, it is useless

Real-Time Applications

- For some audio applications, there are limits to how far we can delay playing back data
- It is hard to carry on a conversation if the time between when you speak and when your listener hears you is more than 300 ms
- We want from the network a guarantee that all our data will arrive within 300 ms
- If data arrives early, we buffer it until playback time

Quality of Service

- Real-Time Applications



A playback buffer

Taxonomy of Real-Time Applications

- The first characteristic by which we can categorize applications is their tolerance of loss of data
  - “loss” might occur because a packet arrived too late to be played back as well as arising from the usual causes in the network.

- On the one hand, one lost audio sample can be interpolated from the surrounding samples with relatively little effect on the perceived audio quality.
  - It is only as more and more samples are lost that quality declines to the point that the speech becomes incomprehensible.

Taxonomy of Real-Time Applications

- On the other hand, a robot control program is likely to be an example of a real-time application that cannot tolerate loss
  - losing the packet that contains the command instructing the robot arm to stop is unacceptable.
- Thus, we can categorize real-time applications as *tolerant or intolerant* depending on whether they can tolerate occasional loss

Taxonomy of Real-Time Applications

- A second way to characterize real-time applications is by their adaptability.
  - For example, an audio application might be able to adapt to the amount of delay that packets experience as they traverse the network.
    - If we notice that packets are almost always arriving within 300 ms of being sent
    - We can set our playback point accordingly, buffering any packets that arrive in less than 300 ms.

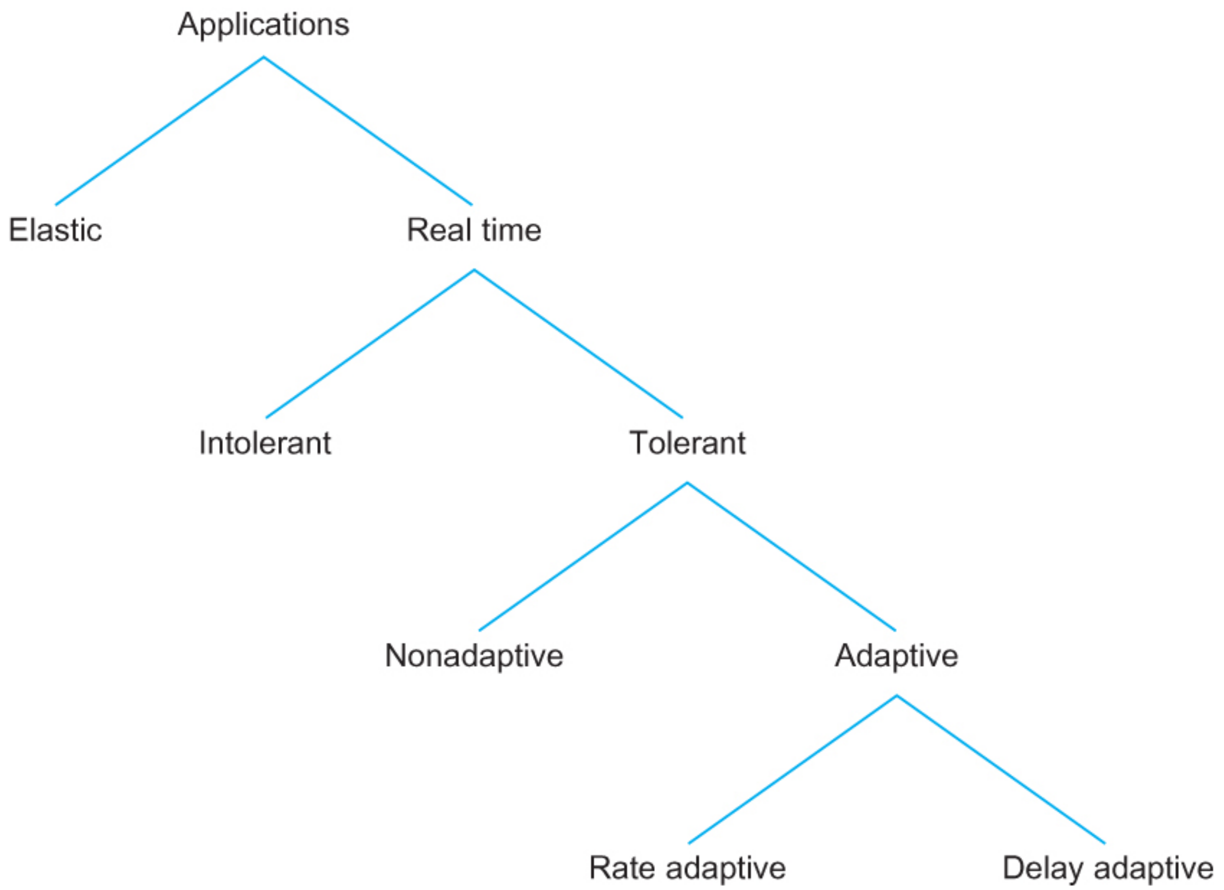
    - Suppose that we subsequently observe that all packets are arriving within 100 ms of being sent.
    - If we moved up our playback point to 100 ms, then the users of the application would probably perceive an improvement.

Taxonomy of Real-Time Applications

- We call applications that can adjust their playback point *delay-adaptive applications.*
- Another class of adaptive applications are *rate adaptive. For example,* video coding algorithms can trade off bit rate versus quality.
  - If the network can support a certain bandwidth, we can set our coding parameters accordingly.
- If more bandwidth becomes available later, we can change parameters to increase the quality.

Quality of Service

- Taxonomy of Real-Time Applications



Approaches to QoS Support

- *fine-grained approaches, which provide QoS to individual applications or flows*
- *coarse-grained approaches, which provide QoS to large classes of data or aggregated* traffic

- In the first category we find "Integrated Services," a QoS architecture developed in the IETF and often associated with RSVP (Resource Reservation Protocol).

- In the second category lies "Differentiated Services," which is probably the most widely deployed QoS mechanism.

Integrated Services (RSVP)

- The term "Integrated Services" (often called IntServ for short) refers to a body of work that was produced by the IETF around 1995–97.
- The IntServ working group developed specifications of a number of *service classes designed to meet the needs of some of the* application types described above.
- It also defined how RSVP could be used to make reservations using these service classes.

Integrated Services (RSVP)

- Service Classes
  - Guaranteed Service
    - The network should guarantee that the maximum delay that any packet will experience has some specified value
  - Controlled Load Service
    - The aim of the controlled load service is to emulate a lightly loaded network for those applications that request the service, even though the network as a whole may in fact be heavily loaded

Integrated Services (RSVP)

- Overview of Mechanisms
  - Flowspec
    - In best-effort service we can just tell the network where we want our packets to go and leave it at that,
    - A real-time service involves telling the network something more about the type of service we require
    - The set of information that we provide to the network is referred to as a *flowspec.*
  - Admission Control
    - When we ask the network to provide us with a particular service, the network needs to decide if it can in fact provide that service. The process of deciding when to say no is called *admission control*.
  - Resource Reservation
    - We need a mechanism by which the users of the network and the components of the network itself exchange information such as requests for service, flowspecs, and admission control decisions. We refer to this process as *resource reservation*

Integrated Services (RSVP)

- Overview of Mechanisms
  - Packet Scheduling
    - Finally, when flows and their requirements have been described, and admission control decisions have been made, the network switches and routers need to meet the requirements of the flows.
    - A key part of meeting these requirements is managing the way packets are queued and scheduled for transmission in the switches and routers.
    - This last mechanism is *packet scheduling.*

Integrated Services (RSVP)

- Flowspec

There are two separable parts to the flowspec:

  - The part that describes the flow’s traffic characteristics (called the *TSpec)* and
  - The part that describes the service requested from the network (the *RSpec).*

  - The RSpec is very service specific and relatively easy to describe.
  - For example, with a controlled load service, the RSpec is trivial: The application just requests controlled load service with no additional parameters.
  - With a guaranteed service, you could specify a delay target or bound.

Integrated Services (RSVP)

- Flowspec
  - Tspec
    - We need to give the network enough information about the bandwidth used by the flow to allow intelligent admission control decisions to be made
    - For most applications, the bandwidth is not a single number
      - It varies constantly
    - A video application will generate more bits per second when the scene is changing rapidly than when it is still
      - Just knowing the long term average bandwidth is not enough

Integrated Services (RSVP)

- Flowspec
  - Suppose 10 flows arrive at a switch on separate ports and they all leave on the same 10 Mbps link
  - If each flow is expected to send no more than 1 Mbps
    - No problem
  - If these are variable bit applications such as compressed video
    - They will occasionally send more than the average rate
  - If enough sources send more than average rates, then the total rate at which data arrives at the switch will be more than 10 Mbps
  - This excess data will be queued
  - The longer the condition persists, the longer the queue will get

Integrated Services (RSVP)

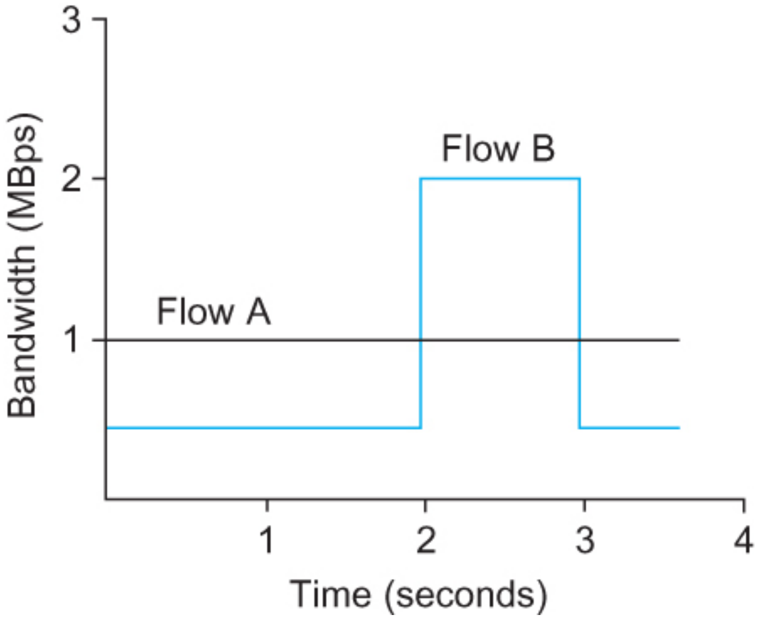
- Flowspec
  - One way to describe the Bandwidth characteristics of sources is called a Token Bucket Filter
  - The filter is described by two parameters
    - A token rate *r*
    - A bucket depth *B*
  - To be able to send a byte, a token is needed
  - To send a packet of length *n*, *n* tokens are needed
  - Initially there are no tokens
  - Tokens are accumulated at a rate of *r* per second
  - No more than *B* tokens can be accumulated

Integrated Services (RSVP)

- Flowspec
  - We can send a burst of as many as *B* bytes into the network as fast as we want, but over significant long interval we cannot send more than *r* bytes per second
  - This information is important for admission control algorithm when it tries to find out whether it can accommodate new request for service

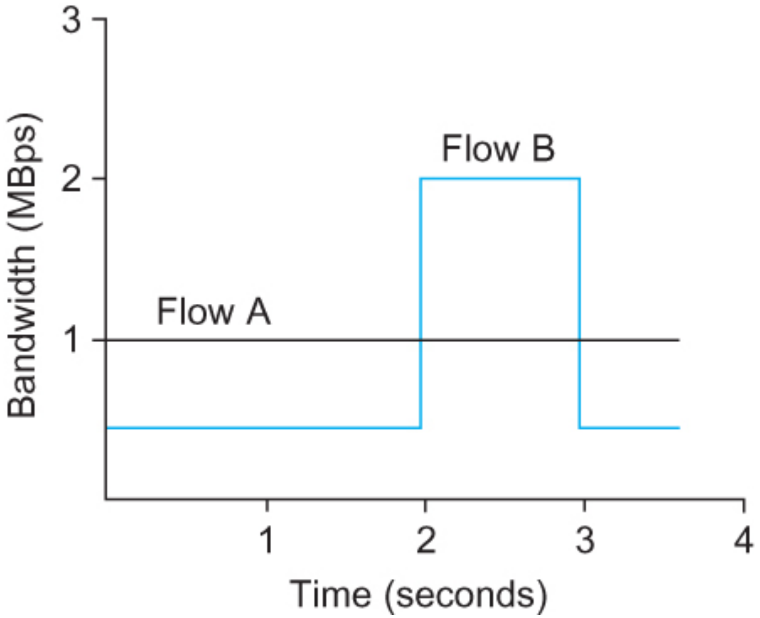
Quality of Service

- Flowspec
  - The figure illustrates how a token bucket can be used to characterize a flow's Bandwidth requirement
  - For simplicity, we assume each flow can send data as individual bytes rather than as packets
  - Flow A generates data at a steady rate of 1 MBps
    - So it can be described by a token bucket filter with a rate *r* = 1 MBps and a bucket depth of 1 byte
    - This means that it receives tokens at a rate of 1 MBps but it cannot store more than 1 token, it spends them immediately



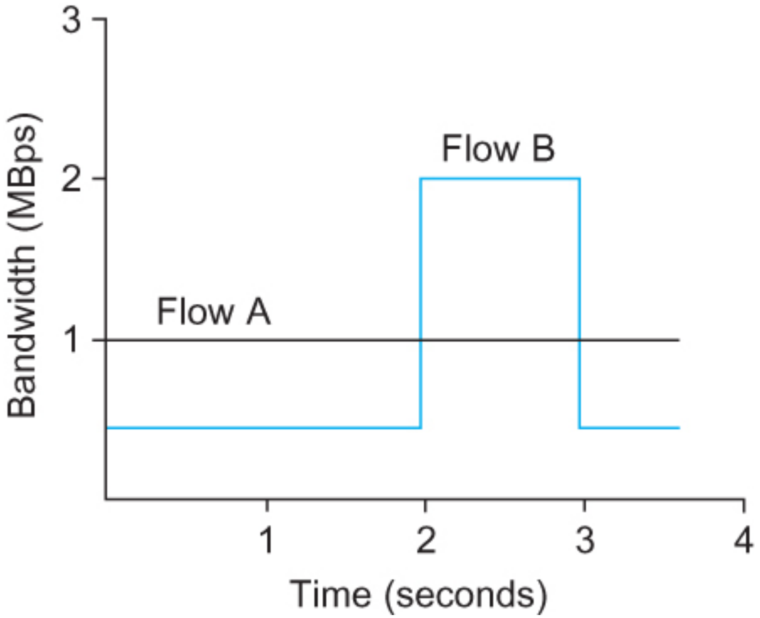
Quality of Service

- Flowspec
  - Flow B sends at a rate that averages out to 1 MBps over the long term, but does so by sending at 0.5 MBps for 2 seconds and then at 2 MBps for 1 second
  - Since the token bucket rate *r* is a long term average rate, flow B can be described by a token bucket with a rate of 1 MBps
  - Unlike flow A, however flow B needs a bucket depth *B* of at least 1 MB, so that it can store up tokens while it sends at less than 1 MBps to be used when it sends at 2 MBps



Quality of Service

- Flowspec
- For the first 2 seconds, it receives tokens at a rate of 1 MBps but spends them at only 0.5 MBps,
    - So it can save up 2 × 0.5 = 1 MB of tokens which it spends at the 3rd second



Integrated Services (RSVP)

- Admission Control
  - The idea behind admission control is simple: When some new flow wants to receive a particular level of service, admission control looks at the TSpec and RSpec of the flow and tries to decide if the desired service can be provided to that amount of traffic, given the currently available resources, without causing any previously admitted flow to receive worse service than it had requested. If it can provide the service, the flow is admitted; if not, then it is denied.

Integrated Services (RSVP)

- Reservation Protocol
  - While connection-oriented networks have always needed some sort of setup protocol to establish the necessary virtual circuit state in the switches, connectionless networks like the Internet have had no such protocols.
  - However we need to provide a lot more information to our network when we want a real-time service from it.
  - While there have been a number of setup protocols proposed for the Internet, the one on which most current attention is focused is called Resource Reservation Protocol (RSVP).

Integrated Services (RSVP)

- Reservation Protocol
  - One of the key assumptions underlying RSVP is that it should not detract from the robustness that we find in today's connectionless networks.
  - Because connectionless networks rely on little or no state being stored in the network itself, it is possible for routers to crash and reboot and for links to go up and down while end-to-end connectivity is still maintained.
  - RSVP tries to maintain this robustness by using the idea of *soft state in the routers.*

Quality of Service

- Integrated Services (RSVP)
    - Reservation Protocol
        - Another important characteristic of RSVP is that it aims to support multicast flows just as effectively as unicast flows
        - Initially, consider the case of one sender and one receiver trying to get a reservation for traffic flowing between them.
        - There are two things that need to happen before a receiver can make the reservation.

Quality of Service

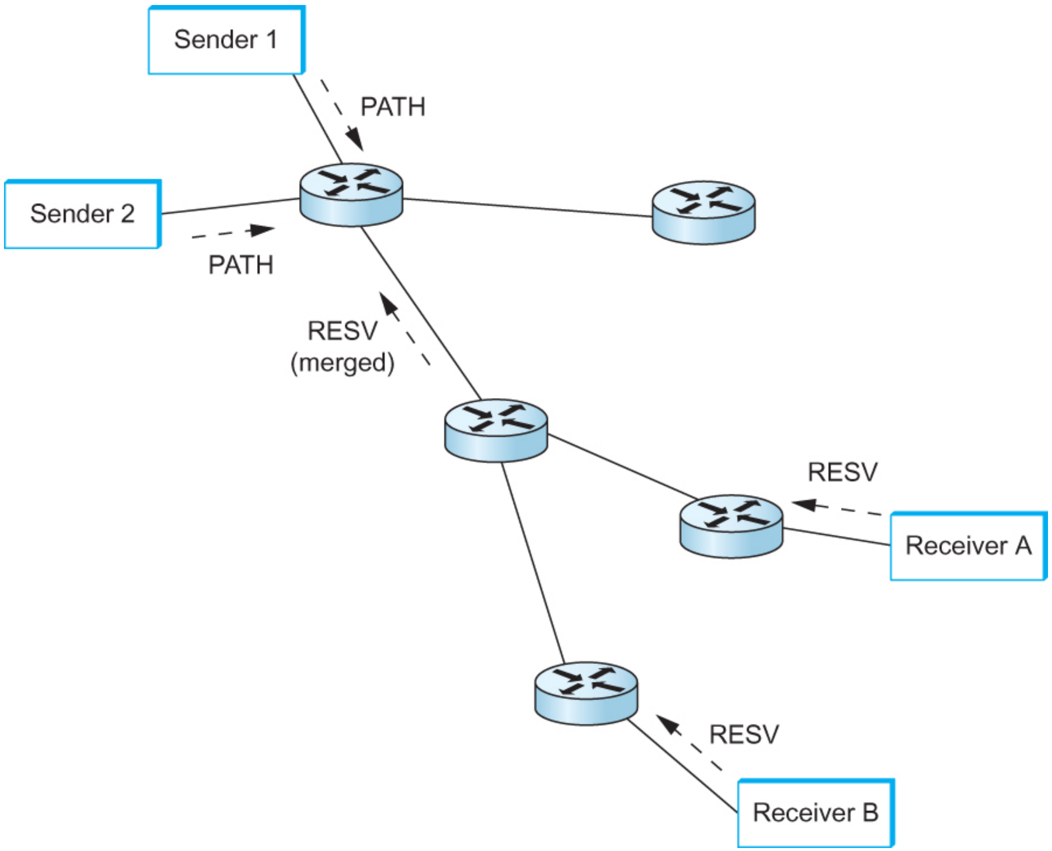
- Integrated Services (RSVP)
  - Reservation Protocol
    - First, the receiver needs to know what traffic the sender is likely to send so that it can make an appropriate reservation. That is, it needs to know the sender’s TSpec.
    - Second, it needs to know what path the packets will follow from sender to receiver, so that it can establish a resource reservation at each router on the path. Both of these requirements can be met by sending a message from the sender to the receiver that contains the TSpec.
    - Obviously, this gets the TSpec to the receiver. The other thing that happens is that each router looks at this message (called a PATH message) as it goes past, and it figures out the *reverse path that will be used to send* reservations from the receiver back to the sender in an effort to get the reservation to each router on the path.

Quality of Service

- Integrated Services (RSVP)
  - Reservation Protocol
    - Having received a PATH message, the receiver sends a reservation back “up” the multicast tree in a RESV message.
    - This message contains the sender’ s TSpec and an RSpec describing the requirements of this receiver.
    - Each router on the path looks at the reservation request and tries to allocate the necessary resources to satisfy it. If the reservation can be made, the RESV request is passed on to the next router.
    - If not, an error message is returned to the receiver who made the request. If all goes well, the correct reservation is installed at every router between the sender and the receiver.
    - As long as the receiver wants to retain the reservation, it sends the same RESV message about once every 30 seconds.

Quality of Service

- Integrated Services (RSVP)
  - Reservation Protocol



Making reservations on a multicast tree

Packet Classifying and Scheduling

- Once we have described our traffic and our desired network service and have installed a suitable reservation at all the routers on the path
  - The only thing that remains is for the routers to actually deliver the requested service to the data packets

Packet Classifying and Scheduling

- There are two things that need to be done:
  - Associate each packet with the appropriate reservation so that it can be handled correctly, a process known as *classifying packets*
  - Manage the packets in the queues so that they receive the service that has been requested, a process known as packet *scheduling*

Classifying Packets

- This is usually done by examining the five fields in the packet:
  - Source address
  - Destination address
  - Protocol number
  - Source port number
  - Destination port number
- The flow will be mapped based on these information into a service class

Classifying Packets

- The details of classification are closely related to the details of queue management.

- A simple as a FIFO queue in a router will be inadequate to provide many different services and to provide different levels of delay within each service

- A more sophisticated buffer management like Fair Queuing or Weighted Fair Queuing is more likely to be used

Scheduling Packets

- packet scheduling ideally should not be specified in the service model
- This is an area where implementors can try to do creative things to realize the service model efficiently

Scalability

- Integrated Services architecture and RSVP has a significant enhancement over the best-effort service model of IP
    - However, many Internet service providers felt that it was not the right model for them to deploy.
- The reason for this relates to one of the fundamental design goals of IP: scalability

Scalability

- In best effort service model:
  - Routers don’t need to store a state for each individual flows passing through them
- As Internet grew, the only things routers had to do to keep up with this growth was:
  - Move more data
  - Deal with larger routing tables

Scalability

- Now, in RSVP you can imagine the situation where every flow passing through a router might have a reservation

- But, why should this be a problem?
  - Think of the magnitude of the flows passing through a high data rate link such as OC-48 (2.5 Gbps)

Scalability

- Assume we have router with an OC-48 (2.5 Gbps) link and that only audio stream flows (64 kbps) are passing through that router

- How many flows we will have:

2.5 x 10^9 / 64 x 10^3 = 39,000 flows

Scalability

- For each of the 39,000 flows the router needs to:
  - Save some state for each one (stored in memory), and also refresh these states periodically
  - Classify, police and queue each of those flows
  - Admission control decisions to be made every time such a flow request a reservation
  - Some mechanisms are needed to “push back” on users so that they don’t make arbitrarily large reservations for long periods of time

Differentiated Services

- While the Integrated Services architecture allocates resources to individual flows, the Differentiated Services model (DiffServ) allocates resources to a small number of classes of traffic.

- In fact, some proposed approaches to DiffServ simply divide traffic into two classes.

Differentiated Services

- Suppose that we have decided to enhance the best-effort service model by adding just one new class, which we'll call "premium."
  - Clearly we will need some way to figure out which packets are premium and which are regular old best effort.
- Rather than using a protocol like RSVP to tell this to the routers
  - it would be much easier if the packets could just identify themselves to the router when they arrive.
- This could obviously be done by using a bit in the packet header
  - if that bit is a 1, the packet is a premium packet
  - if it's a 0, the packet is best effort

Differentiated Services

- With this in mind, there are two questions we need to address:
  - Who sets the premium bit, and under what circumstances?
  - What does a router do differently when it sees a packet with the bit set?

Differentiated Services

- There are many ways to the first question
  - But a common approach is to set the bit at an administrative boundary.
- The router at the edge of an ISP’s network might set the bit for packets arriving on an interface that connects to a particular company’s network.
- The ISP might do this because that company has paid for a higher level of service than best effort.

Differentiated Services

- Assuming that packets have been marked in some way, what do the routers that encounter marked packets do with them?

- Here again there are many answers.

- The IETF standardized a set of router behaviors to be applied to marked packets.
  - These are called "per-hop behaviors" (PHBs), a term that indicates that they define the behavior of individual routers rather than end-to-end services

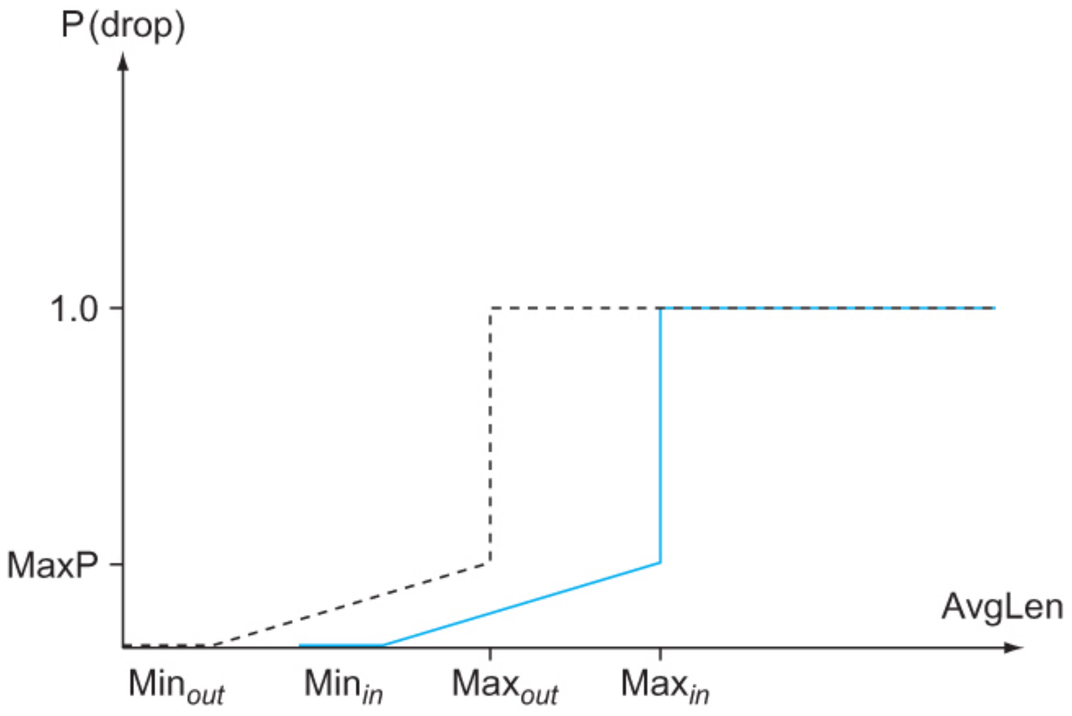
Differentiated Services

- The Expedited Forwarding (EF) PHB
  - One of the simplest PHBs to explain is known as “expedited forwarding” (EF).
  - Packets marked for EF should be forwarded by the router with minimal delay and loss.
  - The only way that a router can guarantee this is if the arrival rate of EF packets at the router is strictly limited to be less than the rate at which the router can forward EF packets.

Differentiated Services

- The Assured Forwarding (AF) PHB
  - Uses an approach known as “RED with In and Out” (RIO) or “Weighted RED,” both of which are enhancements to the basic RED algorithm.
  - For our two classes of traffic, we have two separate drop probability curves. RIO calls the two classes “in” and “out” for reasons that will become clear shortly.
  - Because the “out” curve has a lower MinThreshold than the “in” curve, it is clear that, under low levels of congestion, only packets marked “out” will be discarded by the RED algorithm.
  - If the congestion becomes more serious, a higher percentage of “out” packets are dropped, and then if the average queue length exceeds Min_{in}, RED starts to drop “in” packets as well.

The Assured Forwarding (AF) PHB



RED with In and Out drop probabilities