Introduction, multimedia networking application, streaming audio and video

In recent years, there has been an explosive growth of new applications on the Internet like streaming video, IP telephony, teleconferencing, interactive games, virtual world, distance learning, and so on. Those multimedia networking applications are referred as continuous-media applications and require services different from those for traditional elastic applications like e-mail, Web, remote login, etc. They are also different from download-and-then-play applications. Especially, the new applications require high quality on the communication latency and the latency variation (delay-sensitive) but may not require high quality on the error rate (loss-tolerant). One key issue for supporting new multimedia networking applications is how to get the high quality for the communication latency on the best-effort Internet which provides no latency guarantee. Another key issue is how to improve the Internet architecture to provide support for the service required by multimedia applications.

- Multimedia-Technology that enables humans to use computers capable of processing textual data, audio and video, still pictures, and animation.
- Today, people not only use the internet to watch movies but also to upload videos (YouTube), make internet calls (Skype and Google talk).
- By the end of the decade and with emerging technologies like 4G and Wi-Fi access, Internet will not only provide phone service for less money, but will also provide numerous value-added services, such as video conferencing, online directory services, and voice messaging

Types or Examples of Multimedia Networking Applications

Multimedia network application is any network application that employs audio or video

1) Streaming stored audio and video.

Applications have the following key features:

– Stored media, the contents has been prerecorded and is stored at the server. So, a user may pause, rewind, or fast-forward the multimedia contents. The response time to the above actions should be in the order of 1-10 seconds.

– Streaming, a user starts playout a few seconds after it begins receiving the file from the server. So, a user plays out the audio/video from one location in the file while it is receiving later parts of the file from the server. This technique is called streaming and avoids having download the entire file before starting playout.

– Continuous playout, once playout begins, it should proceed based on the original timing of the recording. This requires high quality on the end-to-end delay.

2) Streaming live audio and video.

Applications are similar to traditional radio and television, except that audio/video contents are transmitted on the Internet. In these applications, many clients may receive the same program. A key issue here is how to deliver the program efficiently to multiple clients on the Internet. IP multicasting technologies play a key role for this. Similar to streaming stored audio and video applications, applications here require continuous playout and high quality on the end-to-end delay.

3) Real time interactive audio and video.

Applications allow users using audio/video to communicate with each other in real time. Real-time interactive audio on the Internet is known as Internet phone. Applications in this category require very high quality on the end-to-end delay, usually a fraction of one second.

Hurdles for multimedia in today's Internet

The Internet Protocol (IP) used in the Internet provides connectionless best effort service for transmitting datagrams. The IP does not guarantee the end-to-end delay nor the uniform delay for all datagrams in a same packet stream. The variations of packet delays within the same packet stream is called packet jitter. The end-to-end delay and packet jitter in the Internet are major hurdles for multimedia applications on the Internet.

How to overcome hurdles

There are many approaches discussed for overcoming the hurdles mentioned above. At one extreme, it is argued that fundamental changes to the Internet should be made so that the users can explicitly reserve the bandwidth on every link in the path for transmitting the packets.

On the other hand, it is argued that fundamental changes are difficult and incremental improvements over the best-effort IP are more practical. Especially, the improvements include:

– The Internet Service Providers (ISP) should scale/upgrade their networks well to meet the demands. The upgrade includes more bandwidth and caches in networks for heavily accessed data.
– Content distribution networks (CDNs), replicate stored contents and put the contents at edges of the Internet.
– Multicast overlay networks for sending data to a huge number of users simultaneously.
Another approach is differentiated services (Diffserv). In this approach, small changes at the network and transport layers are required and scheduling/policing schemes are introduced at edges of the network. The idea is to introduce traffic classes, assign each datagram to one of the classes, and give datagrams different levels of services based on their class.

Streaming stored audio and video

• Overview

In these applications, clients request audio/video data stored at servers. Upon client's request, servers send the data into a socket connection for transmission. Both TCP and UDP socket connections have been used in practice. The data are segmented and the segments are encapsulated with special headers appropriate for audio/video traffic. The real time protocol (RTP, will be discussed later) is a public-domain standard for encapsulating such segments.
Audio/video streaming applications usually provide user interactivity which requires a protocol for client/server interaction. The real time streaming protocol (RTSP) is a public-domain protocol for this purpose.
Clients often request data through a Web browser. A separate helper application (called media player) is required for playing out the audio/video. Well used helpers include RealPlayer and MediaPlayer.

• Access audio/video through Web server

The stored audio/video files can be delivered by a Web server or by an audio/video streaming server. When an audio file is delivered by a Web server, the file is treated as an ordinary object in the server's file system, like HTML and JEPG files. To get the file, a client establishes a TCP connection with the server and sends an HTTP request for the object. On receiving the request, the Web server encapsulates the audio file in an HTTP response message and sends the message back to the TCP connection. It is more complicated for the video case because usually the sounds (audio) and images are stored in two different files. In this case, a client sends two HTTP requests over two separate TCP connections and the server sends two responses, one for sounds and the other for images, to the client in parallel. It is up to the client to synchronize the two streams.

• Sending multimedia from a streaming server to a helper application

Audio/video files can be delivered by a streaming server to a media player. Streaming servers include those marketed by RealNetworks and Microsoft, and those of public-domain servers. With a streaming server, audio/video files can be transmitted over UDP which has much smaller end-to-end delay than TCP.

• Real-Time Streaming Protocol (RTSP)

RTSP is a protocol which allows a media player to control the transmission of a media stream. The control actions include pause/resume, repositioning of playback, fast-forward, and rewind. RTSP messages use a different port number from that used in the media stream and can be transmitted on UDP or TCP.

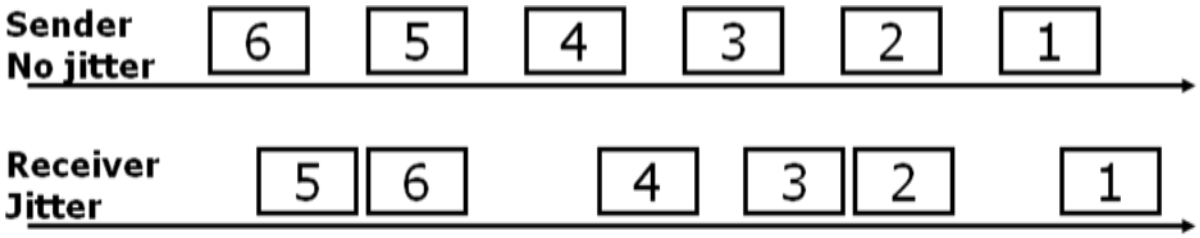
Making the best of the best-effort service

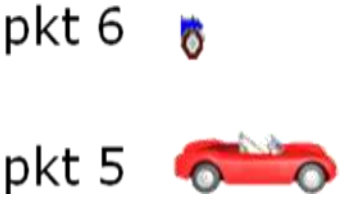
- Limitation of best-effort service
- Removing jitter at the receiver for audio
- Recovering from packet loss

• Limitation of best-effort service

- Packet loss, IP provides the best-effort service but does not guarantee the delivery of packets. Packets may be discarded due to congestions.
- End-to-end delay, IP does not guarantee the end-to-end delay either. The time for transmitting a packet may vary due to the conditions of the network. Also, in order to guarantee the delivery, positive acknowledgement and retransmission are used in TCP. The cost for realizing the reliable transmission in TCP is a longer end-to-end delay.
- Packet jitter, since the end-to-end delay for each packet may depend on the conditions of the network, the delays of packets in the same packet stream may vary. Especially, the packets may arrive to the receiver in a wrong order.

• Removing jitter at the receiver for audio





In applications like Internet phone or audio-on-demand, it is up to the receiver to remove the jitters. Common techniques used include sequence number, timestamp, and delaying playout. The sender can put a sequence number on every packet sent and the receiver can use the sequence number to recover the correct order of the received packets. Timestamp is similar to sequence number, the sender stamps each packet with the time at which the packet is generated. In order to get the correct order from the sequence number and timestamp for a sequence of packets, the receiver need to receive all of the packets in the sequence. Playout delay is used for this purpose. The playout delay should be long enough to receive all packets in a subsequence of packets which can be played. On the other hand, the delay should be short enough so that the user will not notice the delay. The playout delay can be either fixed or adaptive.

Fixed playout delay, the receiver plays out each packet exactly q msecs after the packet is generated. Usually, q is up to a few hundreds msecs.

Adaptive playout delay, the receiver estimate the network delay and the variance of the network delay at the beginning of each talk, and adjusts the playout delay accordingly.

• Recovering from packet loss

A major scheme for handling packet loss for elastic applications is retransmission. However, this scheme does not work well for applications with strict end-to-end delay constraint. Internet phone applications usually use loss anticipation schemes to handle packet loss.

- Forward error correction (FEC) is one of such schemes. The basic idea of this scheme is to include redundant information in the original packet stream. The redundant information can be used to reconstruct the lost packet. One approach for the FEC scheme is to send the exclusive OR of every n packets as a redundant packet. If any one of the n + 1 packet is lost, the receiver can reconstruct it. However the scheme does not work if two or more of the n + 1 packets are lost. Another approach is to send two copies of the same packet, usually one is the original packet and the other is a short version (lower-resolution audio) of the packet. An example is that the short version of packet i is sent together with packet i + 1. FEC uses extra bandwidth of networks.
- Interleaving is another loss anticipation scheme. This scheme re-sequences units of audio data before transmission so that the original adjacent units are separated by some distance in the transmitted stream. The receiver rearranges the received stream into its original order before it is re-sequenced. If a transmitted packet is lost, only a small fraction of each original packet is lost and the quality of the voice may not be damaged much. Interleaving does not use extra bandwidth but introduces extra end-to-end delay.
- Receiver-based repair of damaged audio stream. This scheme reconstructs a lost packet using the other received packets based on the fact that there are large amount of short term self-similar signals in audio data, especially for speech. A simplest approach is packet repetition, using the immediate previous packet to replace the lost one. Another approach is interpolation, using the packets before and after the loss to interpolate a packet to cover the loss.

Protocols for real-time interactive applications

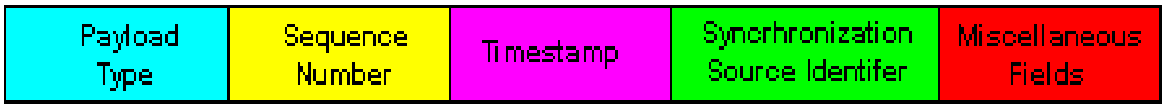
- Real Time Protocol (RTP)
- RTP control protocol (RTCP)
- Session initiation protocol (SIP)
- H.323

• Real Time Protocol (RTP)

In multimedia applications, header fields are appended to audio/video packets for transmission. These header fields include sequence number and timestamps. RTP is a standard for the packet structure which includes the fields for audio/video data, sequence number, timestamp, and other fields.

Usually, the media data is encapsulated in RTP packets which are encapsulated in UDP segments.

RTP packet header fields include



RTP Header

– payload type, 7 bits, used to indicate the type of encoding for audio and video;
– sequence number, 16 bits, incremented by one for each RTP packet sent;
– timestamp, 32 bits, used to give the sampling instant of the first byte in the RTP packet data;
– synchronization source identifier (SSRC), 32 bits, used to identify the source of the RTP stream;
– miscellaneous fields

• RTP control protocol (RTCP)

RTCP is a protocol that a networked multimedia application can use in conjunction with RTP. RTCP packets do not carry audio/video data but contain sender/receiver reports which include the statistics on the number of RTP packets sent, number of packets lost, and interarrival jitters. RTCP packets are sent periodically. There are two types of RTCP packets.

The RTCP packets used by receiver include
– the SSRC of the RTP stream for which the reception report is generated;
– the fraction of the packets lost within the RTP stream;
– the last sequence number received in the stream of RTP packets; and
– the interarrival jitter.

The RTCP packets used by sender include
– the SSRC of the RTP stream;
– the timestamp and real time of the most recently generated RTP packet in the stream;
– the number of packets sent in the stream; and
– the number of bytes sent in the stream.

In an RTP session, if the number of receivers is large, the overhead of the RTCP packets generated by receivers can be large. The period of sending RTCP packets for receivers in a large multicast tree should be carefully designed to limit the overhead.

• Session initiation protocol (SIP)

SIP provides mechanisms for the following.

– It establishes calls between a caller and a callee over an IP network. It allows the caller to notify the callee that it wants to start a call. It allows the participants to agree on media encodings and to end a call.
– It allows the caller to determine the current IP address of the callee. Users may have multiple or dynamic IP addresses.
– For call management like adding new media streams, changing the encoding, inviting new participants, call transfer, and call holding.

Setting up a call to a known IP address. The caller initiates the call by sending an INVITE message which includes an identifier for the callee (e.g., the name of callee and his/her IP address), an identifier for the caller, the encoding scheme used in the audio data, the port number through which the caller wants to receive the RTP packets. The callee, after receiving the INVITE message, sends an SIP response message which includes an OK message, an indication of callee's IP address, desired encoding scheme for reception, and port number for the conversation. After receiving the SIP response message, the caller sends the callee an ACK message. After that, the call connection is set-up. The SIP

messages are transmitted through a well known port number 5060 for SIP. The RTP packets are transmitted on different connections.
SIP addresses can be in the form of IP addresses or that similar to email addresses.
Name translation and user location. SIP proxy and SIP registrar are used to track the users with multiple or dynamic IP addresses.

• H.323

H.323 is popular standard for real-time audio and video conferencing among end systems in the Internet. The standard includes the following:
– A specification for how endpoints negotiate common audio/video encodings.
– H.323 mandates RTP for audio and video data encapsulation and transmission over the network.
– A specification for how endpoints communicate with their respective gatekeepers (a device similar to and SIP registrar).
– A specification for how Internet phones communicate through a gateway with ordinary phones in the public circuit-switched telephone networks.

Beyond best effort

There are number of techniques such as sequence numbers, timestamps, FEC, RTP, and H.323 for improving the performances of the best-effort Internet for multimedia applications which require high quality of service on the end-to-end delay. However, these techniques do not change the best-effort nature of the Internet. Now we discuss some technologies which are used to provide the true quality of service for multimedia applications. The central idea for those technologies is to add new architectural components to the Internet to change its best-effort nature. Those technologies have been under active discussion in the Internet Engineering Task Force (IETF) working groups for Diffserv, Intserv, and RSVP.
To see how to change the best-effort nature of the Internet by adding new architectural components, we start from a simple example. Assume that routers R1 and R2 are gateways for networks N1 and N2, respectively, and are connected by a link of limited bandwidth. There are two traffic streams, one is a multimedia application and the other is an FTP from N1 to N2. By the best-effort IP routing, the datagrams for both streams will be transmitted without any priority by routers. In this case, the busty arrival of FTP packets may delayed the transmission of multimedia packets although the multimedia application has strict requirement on the end-to-end delay while the FTP does not. One way to solve this problem seems to add new functions to IP such that the protocol can treat the two streams differently. There are a few principles for this.

- First, classification of packets is needed to allow a router to distinguish among the packets belonging to different classes of traffic.
- It is desirable to provide a degree of isolation among traffic flows so that one flow is not adversely affected by another misbehavior flow.
- While providing isolation among flows, it is desirable to use resources (like bandwidth of links and buffers) as efficiently as possible.
- If resources are not enough, a call admission process is needed in which flows declare their QoS requirements and are then either admitted to the network (at the required QoS) or blocked from the network (if the required QoS can not be provided by the network).

Scheduling and policing mechanisms are used to provide QoS guarantees.

• Scheduling - QoS

A scheduling mechanism is a scheme for selecting packets for transmission from an output link queue with packets of multiple data streams.
A simple scheduling scheme is first-in-first-out (FIFO). In FIFO scheduling, packets are transmitted in the order of their arrival.
Priority queuing is another scheduling scheme. In this scheme, arrived packets are classified into priority classes at the output queue. The priority of a packet may depend on an explicit marking in its header, its source/destination addresses, port numbers, or other criteria. Usually, each priority class has its own queue. The priority queue scheduling chooses a packet by FIFO scheme for transmission from the highest priority class that has a non-empty queue.
In round robin scheduling, packets of each data stream are put in a distinct queue and the queues are served in a circular manner. Assume that there are n queues. In the simplest form, the scheduling scheme serves queue 1 (selects a packet

by FIFO for transmission from queue 1 if queue 1 is not empty), then serves queue 2. In general, queue i + 1 mod (n + 1) is served after queue i is served.

Weighted fair queuing (WFQ) is a generalization of round robin. In this scheme, packets are classified and each class is given a distinct queue. Each queue i is also assigned a weight w_i. Queues are served in a round robin manner with queue i be guaranteed to receive a fraction of service equal to $w_i/(sum_j (w_j))$, where the sum_j is taken over all classes that have non-empty queues.

• Policing - QoS

Policing schemes are used to specify the data rate at which a data stream is allowed to enter a network. Criteria for policing include:

– Average rate, number of packets per time interval at which packets of a data flow can be sent into the network. A key issue is the interval of time over which the average rate will be policed.

– Peak rate, the maximum number of packets that can be sent into the network over a short period of time.

– Burst size, the maximum number of packets that can be sent into the network over an extremely short interval of time.

Leaky bucket is a mechanism used to specify policing limits shown above. A leaky bucket consists of a bucket of size b. When the bucket is full, it has b tokens. If the bucket has less than b tokens, new tokens are added to the bucket with a constant rate of r tokens per second until the bucket becomes full. In the leaky bucket policing, when a packet is transmitted into a network, it must first remove a token from the bucket. If the bucket is empty, the packet must wait for a token. The burst size defined by the leaky bucket is b, the peak rate over time t is rt + b, and the average rate is r.

The leaky bucket can be combined with the weighted fair queue. Each queue i is associated with a leaky bucket with size b_i and new token generation rate r_i.

Integrated service (Intserv) and differentiated service (Diffserv)

The principles and mechanisms discussed above are used in two architectures, integrated service (Intserv) and differentiated service (Diffserv), proposed to providing QoS in the Internet. Intserv is a framework developed within the IETF to provide individualized QoS guarantees to individual application sessions. Diffserv provides the ability to handle different classes of traffics in different ways within the Internet.

• Intserv

There are two key features in the Intserv architecture.

– Reserved resources, a router is required to know what amounts of resources have been reserved for ongoing sessions.

– Call setup, a session which requires a QoS guarantee must reserve resources at every router on the path for transmission. The session must send the traffic characterization and specification of the desired QoS to routers. Routers determine if the call of the session is admitted or not. Routers reserve the resources to guarantee the QoS required if they decide to admit the call.

The RSVP protocol is used for the call setup.

• Diffserv

Intserv provides the QoS guarantee for each individual session. This has advantages but also introduces problems. Especially, the per-flow resource reservation may give significant workload to routers. Also Intserv does not allow for more qualitative or relative definitions of service distinctions.

The Diffserv architecture is proposed to provide scalable and flexible service. In Diffserv, different classes of traffics can be treated in different ways in the Internet. There are two sets of functional elements in the Differ architecture.

– Edge functions: packet classification and traffic conditioning, packets arriving to the edge router are first classified based on the values of one or more packet header fields.

– Core function: forwarding, a classified packet is forwarded to its next hop router according to the per-hop behavior associated with the packet's class. The per-hop behavior affects how a router's buffer and link bandwidth are shared with other classes of traffics.

Resource Reservation Protocol (RSVP)

The resource reservation protocol (RSVP) is used by application sessions to reserve resources in the Internet. Especially, RSVP is used to reserve bandwidth for multicast trees (unicast is treated as a degenerate case of multicast). RSVP is receiver-oriented. The receiver of a data flow initiates and maintains the resource reservation. RSVP is a protocol used by sessions to send the reservation request and does not specify how the network provides the reserved resources. It is not a routing protocol either and does not determine the links in which the reservations are to be made. RSVP operates in a two-pass manner. A transmitting source advertises its content by sending an RSVP path message through a multicast tree, indicating the bandwidth required for the content, the timeout interval, and information about the upstream path to

the source. Each receiver sends an RSVP reservation message upstream on the multicast tree. The reservation message specifies the rate at which the receiver wants to receive the data. When a router receives a reservation message, it first checks if its downstream links can accommodate the reservation. If yes, it adjusts its packet scheduler to accommodate the reservation and sends a reservation upstream on the multicast tree. Otherwise it rejects the reservation and sends an error message to the corresponding receivers.