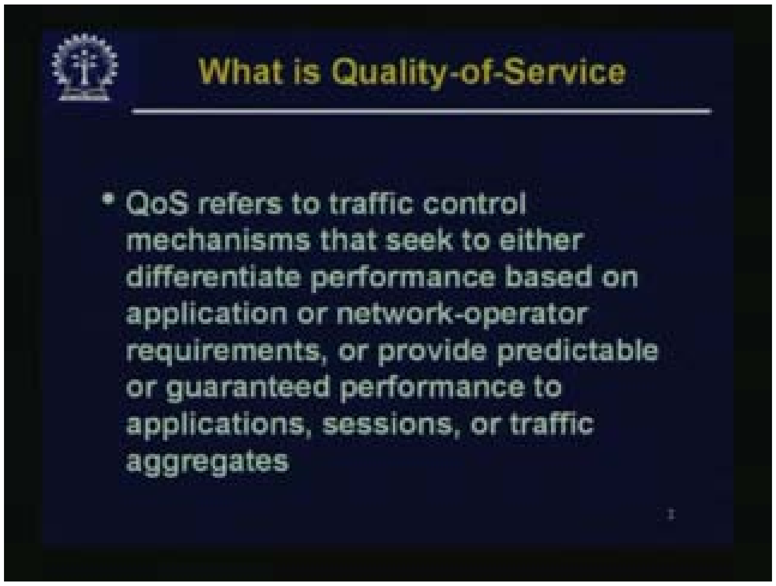
Computer Networks
Prof. S. Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture - 36

Our topic for today is QOSQOS and multimedia, that is quality of service and multimedia. We will just look at these one by one.

(Refer Slide Time: 00:57:00:58)

Quality Of Service

(Refer Slide Time: 01:07 - 01:37)

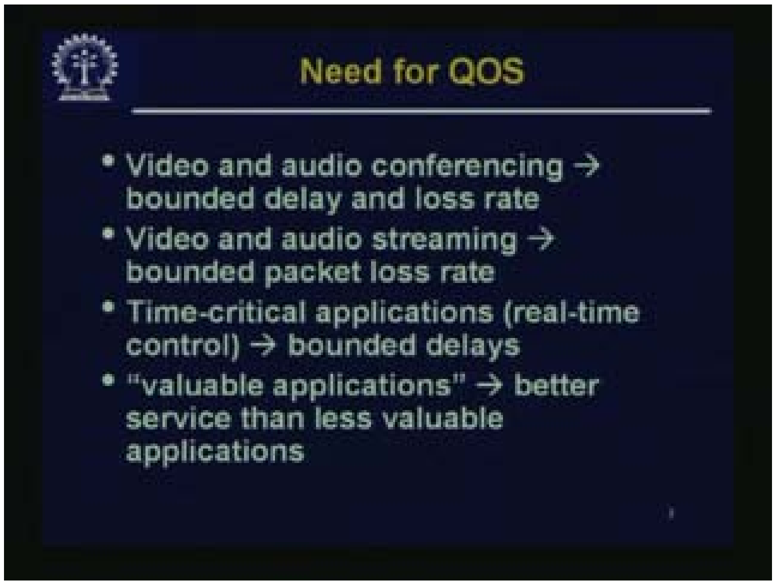


What is Quality-of-Service

- QoS refers to traffic control mechanisms that seek to either differentiate performance based on application or network-operator requirements, or provide predictable or guaranteed performance to applications, sessions, or traffic aggregates

Quality of service: What is quality of service? QOS refers to traffic control mechanisms that seek to either differentiate performance based on application or network operator requirements or provide predictable or guaranteed performance to applications, sessions or traffic aggregates. It talks about lot of things, the basic notion is that there are some applications which require one kind of quality of service. By the way the quality of service may mean so many different things but the most important of them are the network delay and the packet loss. These two parameters may sort of come in various ways, they affect differently when you talk about different applications. And by multimedia we mean audio, video, etc going through the net which has got a different kind of QOS characteristics, a different set of requirements than a file transfer. We will see how we can handle QOS and how multimedia traffic is handled in the network. Sometimes it may so happen that some applications are more important than others, so they need a better guarantee of performance. Such things also come under quality of service.

(Refer Slide Time: 02:46 - 04:24)

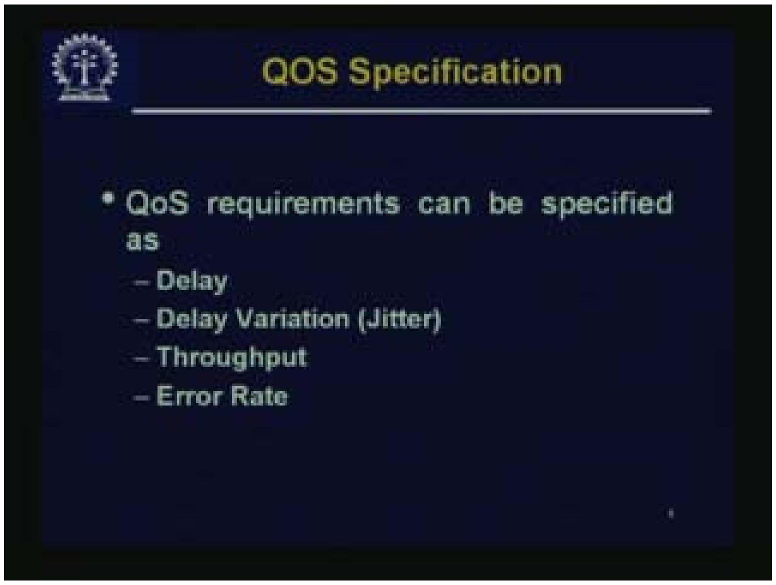


Need for QOS

- Video and audio conferencing → bounded delay and loss rate
- Video and audio streaming → bounded packet loss rate
- Time-critical applications (real-time control) → bounded delays
- "valuable applications" → better service than less valuable applications

Take examples: Video and audio conference requires bounded delay and loss rate. That means the delay has to be bounded and also the variability of the delay should also be within the bounds and the loss rate of packets should again be bounded. As you can see, this is quiet different from a file transfer where the loss of some packets cannot be tolerated at all. It is somewhat tolerant of losses, but more sensitive about delays. When you are downloading a file, they are usually not that sensitive to trace. Video and audio streaming means maybe from video server or some internet radio, etc; audio or video is being streamed to different clients so this again requires a bounded packet loss rate. It may not be so sensitive to delay. There might be some time critical applications like real time control. Here again, bounded delay is important. There are some valuable called premium applications. Maybe those who run these premium applications are ready to pay more for this and they expect better guaranteed service than a low premium or less valuable applications.

(Refer Slide Time: 04:25 - 04:47)



QOS Specification

- QoS requirements can be specified as
  - Delay
  - Delay Variation (Jitter)
  - Throughput
  - Error Rate

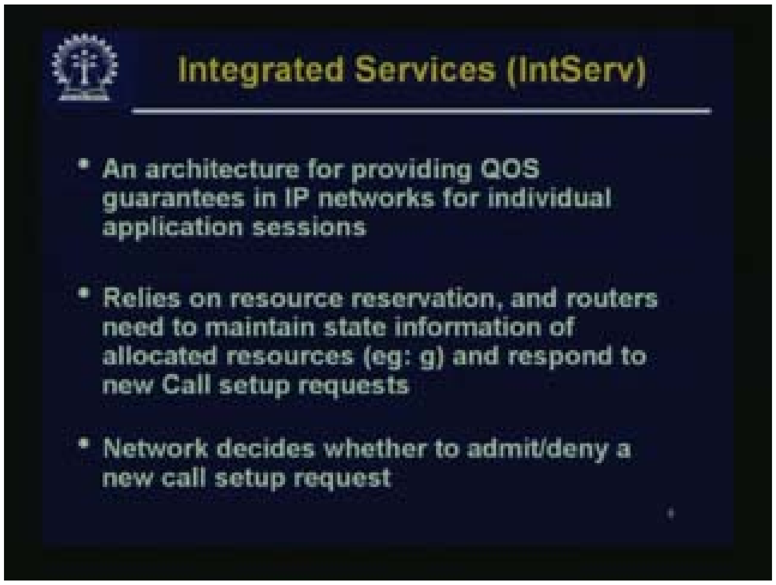
QOS requirements can be specified as delay, delay variation (jitter), throughput which is the rate at which you can send data and error rate. Error rate is something where some packets get lost.

There are two approaches to this, stateless and stateful QOS solutions. Stateless solution: Router maintains no fine grained state about traffic. This has got a positive point that this is very scalable. Actually the kind of thing we have talked about till now is that routers do not maintain any states. It is simply the raw packet forwarding capacity that comes to this. In this way it is very scalable, this is also quite robust but it has weak services because there is no guarantee about the kind of delay or performance a particular application you will have to encounter.

On the other hand, we have stateful solutions. Routers maintain per-flow state. This flow is very important while guaranteeing quality of service. Suppose there is some audio conferencing going on between some people, the audio signal nature is digitized and packetized and then these packets are going on. This is a packet switched network, essentially at the heart of it this is a packet switched network. but at the same time we not only talk about a single packet in entity in itself which is how we have dealt with packets so far but the end users are interested in the flow, that means the flow of packet. All these streams of packets that are flowing which contain these audio signals constitute a flow and the users are actually interested in the flow.

Suppose somebody wants some guarantee on this service, he wants a guarantee on this service of this flow. Merely looking at packets is not enough; you have to go from a purely stateless situation to a stateful situation. Actually you have to make out that this is a part of particular flow. Naturally if you are stateful, if you go into the packet and look at what kind of packet it is and maintain a state about it in the router then you can give powerful services like guaranteed services plus high resource utilization, fine grain differentiation between different flows, protection where here there is a problem that this is much less scalable that means when the number of flows through a router starts increasing, and that is what will happen in any of the core routers, then the core router will not be able to handle this. So it is much less scalable and much less robust.

(Refer Slide Time: 08:11 - 10:25)



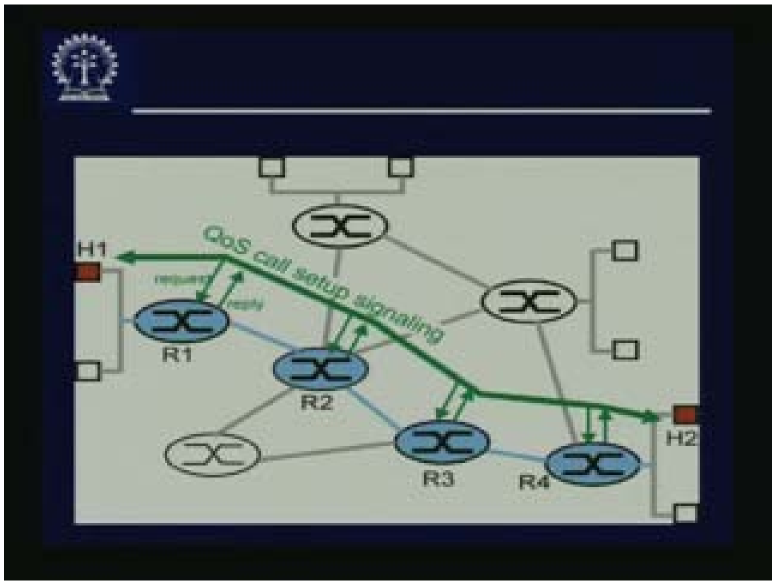
Integrated Services (IntServ)

- An architecture for providing QOS guarantees in IP networks for individual application sessions
- Relies on resource reservation, and routers need to maintain state information of allocated resources (eg: g) and respond to new Call setup requests
- Network decides whether to admit/deny a new call setup request

Let us look at the Integrated Services. This is a fully stateful system and is also known as intserv. This is an intserv architecture or isa. It is the architecture for providing QOS guarantees in IP networks for individual application sessions. We are talking about an entire session and the flow that has gone on during that session of flow consisting of so many packets. It relies on resource reservation and routers need to maintain state information of allocated resources (for example g) and respond to new call setup requests.

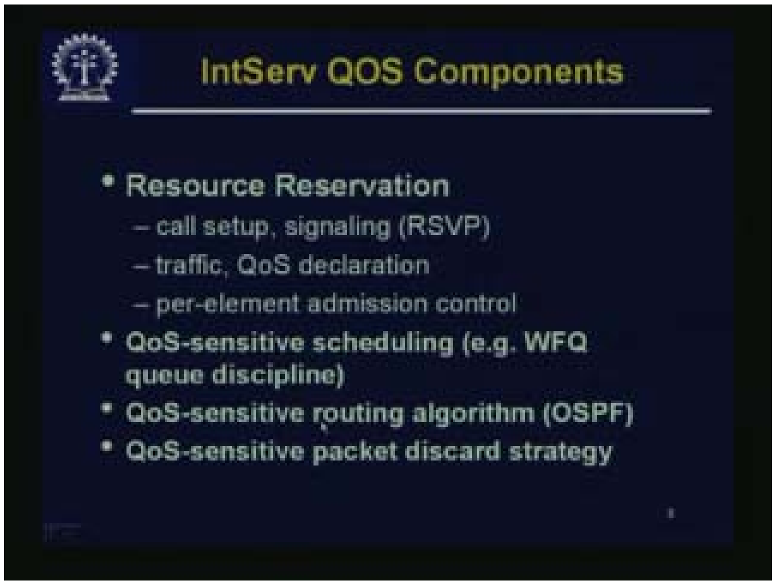
The point is, in this situation how will you guarantee that this particular flow which is starting will get the requested kind of guarantee of service that means its delay would be bounded etc? For this we have to specially reserve resources like buffers, resources like CPU time and so on at the intermediate points. So, you have to reserve these resources for this particular flow so long as this flow continues. That is how you give better service to this particular flow than to any other ordinary packet. It will have to maintain the state in the form of these reservations and respond to new call setup request. For example, as soon as you take out some resources, then naturally it goes out of the available resource pool and if a new call setup request comes for another session some intermediate router may not be able to handle that so it will not be admitted. Network decides whether to admit or deny a new call setup request.

(Refer Slide Time: 10:26 - 10:55)



This is the basic idea of intserv. Therefore in this case since you have to reserve resources there is a call setup phase. This is almost like a virtual circuit which is being setup. There is a call setup request which goes from the source to the destination and back and on the way it reserves resources at each of the intermediate nodes. And if each of the intermediate nodes, that is, each of the intermediate routers are ready to provide these resources then the call will be admitted.

(Refer Slide Time: 10:54 - 12:20)

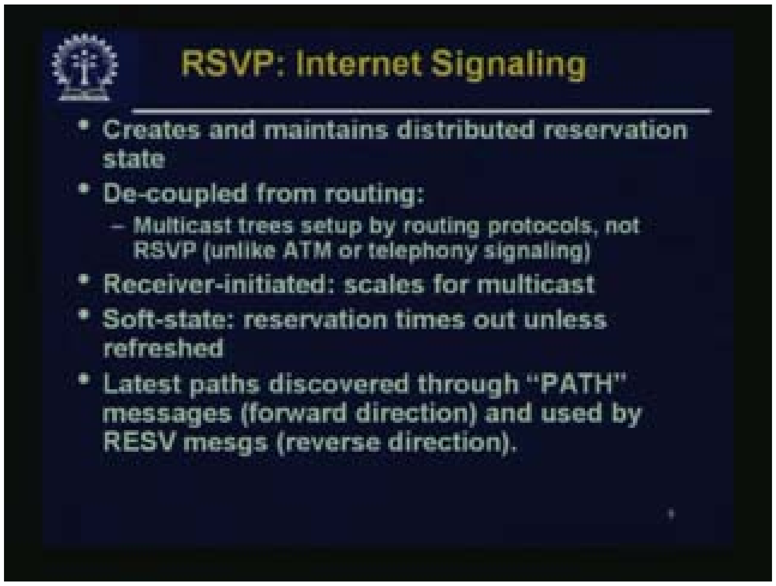


IntServ QOS Components

- Resource Reservation
  - call setup, signaling (RSVP)
  - traffic, QoS declaration
  - per-element admission control
- QoS-sensitive scheduling (e.g. WFQ queue discipline)
- QoS-sensitive routing algorithm (OSPF)
- QoS-sensitive packet discard strategy

For resource reservation there is a particular protocol called reservation protocol or RSVP. RSVP constitutes of call setup and signaling. Then there is a traffic QOS declaration and per element admission control which means whether the call could be admitted or not. Then there will be QOS sensitive scheduling. That means, when a packet arrives first it may go into the buffer of a router and there maybe various buffers. Each of these packets will have to be scheduled for processing and then forwarding. This processing may not be done in a uniform manner but it may be done in a QOS sensitive manner. Similarly routing algorithm may also be QOS sensitive, packet discard strategies etc come under the intserv QOS components. We have talked about scheduling and some of the other things. Now let us discuss about RSVP.

(Refer Slide Time: 12:21 - 13:44)

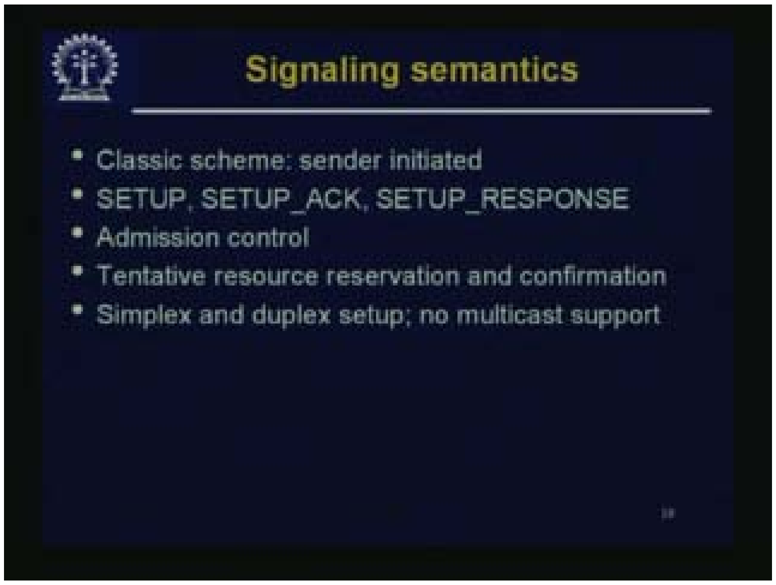


RSVP: Internet Signaling

- Creates and maintains distributed reservation state
- De-coupled from routing:
  - Multicast trees setup by routing protocols, not RSVP (unlike ATM or telephony signaling)
- Receiver-initiated: scales for multicast
- Soft-state: reservation times out unless refreshed
- Latest paths discovered through "PATH" messages (forward direction) and used by RESV mesgs (reverse direction).

RSVP is internet signaling. It creates and maintains distributed reservation state. This is decoupled from routing. One thing is to use routing for setting up paths but then this has to reserved resources therefore it is decoupled from routing. It uses multiple trees setup by routing protocols, not RSVP. Multicast tree is setup by routing protocols. RSVP is for reserving protocols there unlike ATM or telephony signaling. While doing the call setup you actually assign the virtual setup. This is initiated by the receiver, this scales for a multicast and it may use soft state. That means reservation times out unless refreshed. so you have to refresh reservation from time to time, otherwise one particular flow may be reserved and never be used so that part will go away. That is why reservation will be timed-out. And latest paths discovered through path messages (forward direction) and used by reservation messages in the reverse direction.

(Refer Slide Time: 13:45 - 14:08)

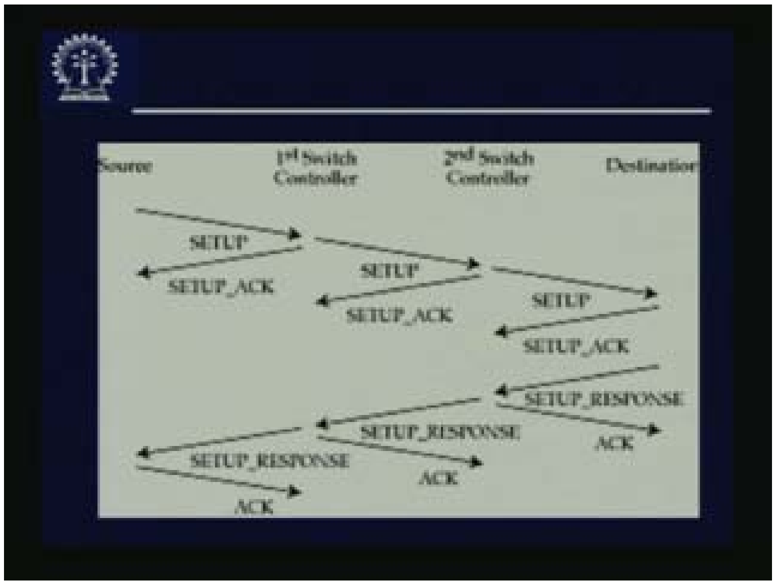


Signaling semantics

- Classic scheme: sender initiated
- SETUP, SETUP_ACK, SETUP_RESPONSE
- Admission control
- Tentative resource reservation and confirmation
- Simplex and duplex setup; no multicast support

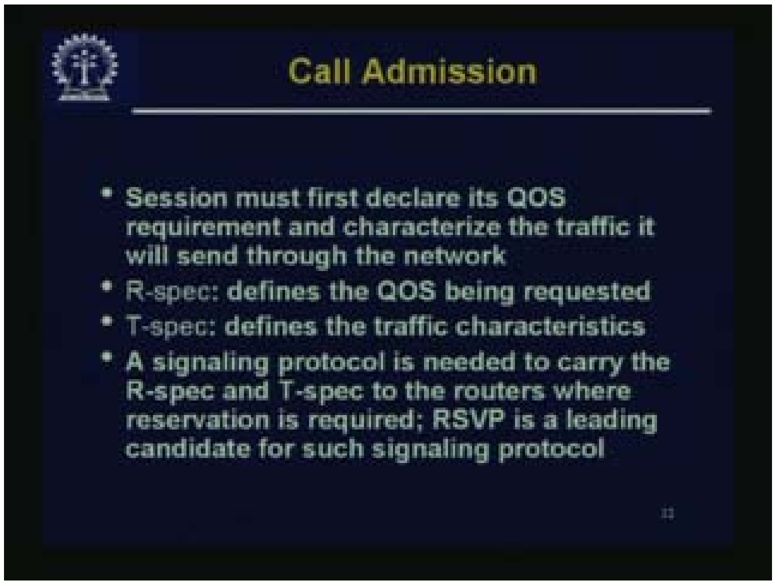
Signaling Semantics: In signaling semantics you have a setup acknowledgement, setup response, admission control, tentative resource reservation and confirmation. This can be done from both directions, simplex and duplex setup and with no multicast support. These are the signaling semantics in brief.

(Refer Slide Time: 14:09 - 15:02)



Suppose this is the source, the source would initiate the setup. It will send a setup call to the first switch controller, router to another one and finally to the destination. Each of the intermediate nodes send a setup ACK so that the source knows that this is the way it is going to go and the path is being setup. And then a setup response will come and the acknowledgement will go to the destination and another setup response and acknowledgement will keep coming in. What we will have to see is that, in this fashion, after this, the resource has to be reserved at each of the node for whatever service that is being asked for.

(Refer Slide Time: 15:03 - 16:19)

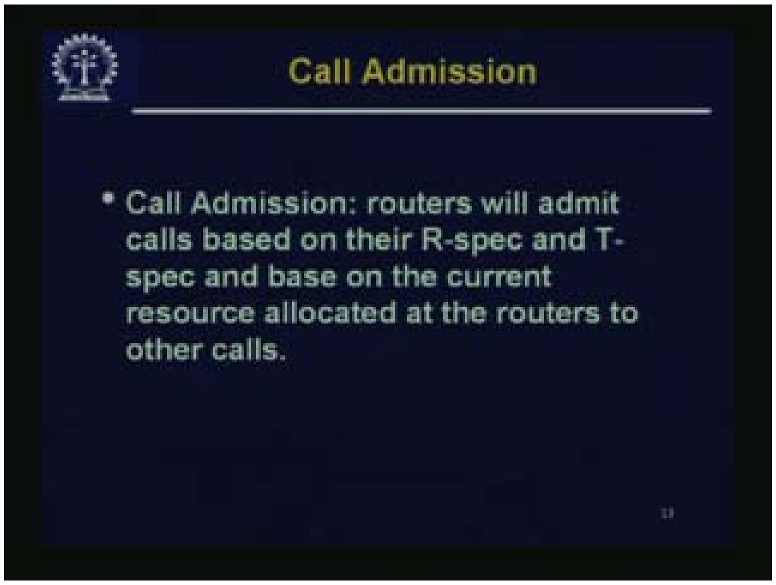


Call Admission

- Session must first declare its QOS requirement and characterize the traffic it will send through the network
- R-spec: defines the QOS being requested
- T-spec: defines the traffic characteristics
- A signaling protocol is needed to carry the R-spec and T-spec to the routers where reservation is required; RSVP is a leading candidate for such signaling protocol

Session must first declare its QOS requirements and characterize the traffic it will send through the network. There are two specifications r spec and t spec: The r spec defines the QOS being requested, what kind of bound we want on the delay, what kind of bound we want on the jitter, what kind of packet loss is acceptable to me and depending on this the resource is going to be reserved. That means maybe a special queue or buffer will be setup just for this flow or something will be done with scheduling. There will be some weight given to it for processor time and so on in the intermediate nodes. And then there is a t spec that is, traffic characteristics like what kind of traffic, what is the burstiness of the traffic etc? Such traffic characteristic will also have to be sent. Therefore depending on the t spec and the r spec the intermediate nodes will compute what kind of resource is visibly required. So signaling protocol is needed to carry the r spec and t spec to the routers where reservation is required. RSVP is a leading candidate for such signaling protocol.

(Refer Slide Time: 16:20 - 17:00)

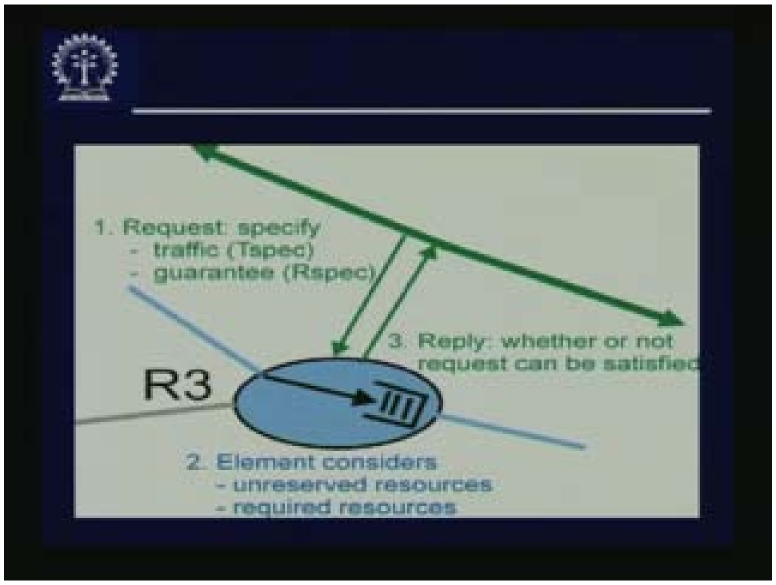


Call Admission

- Call Admission: routers will admit calls based on their R-spec and T-spec and base on the current resource allocated at the routers to other calls.

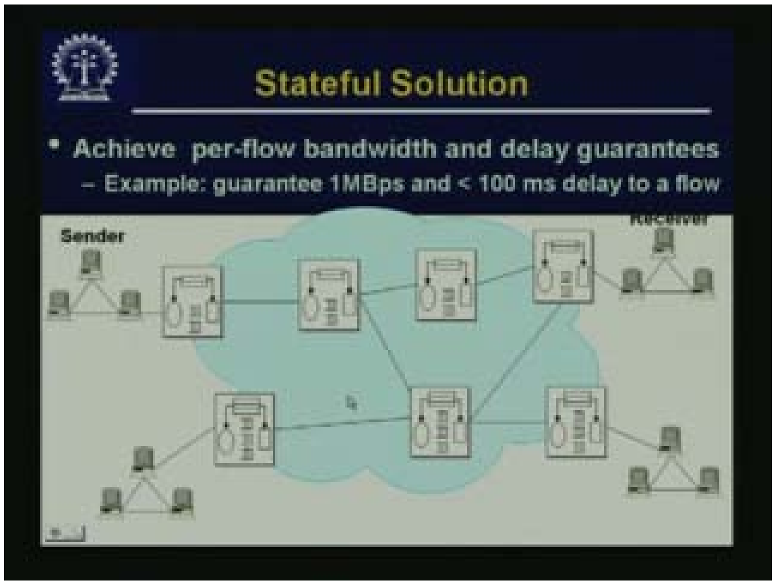
As far as call admission is concerned, routers will admit calls based on their r spec and t spec that whether they can handle or they can manage to give so much resources for this. There is the financial side also, that whoever has initiated whether he has an agreement to pay for such premium service and based on current resources allocated at the routers to other calls. If other senders have already reserved most of the resources in the intermediate routers then a new call may not be admitted.

(Refer Slide Time: 17:01 - 17:30)



This is the router, the request specifies the traffic the t spec and the guarantees of service which is required which is the r spec. Now this reply to the element considers unreserved resources and the required resources for this new request. If it can handle that then it replies whether or not request can be satisfied.

(Refer Slide Time: 17:31 - 17:46)

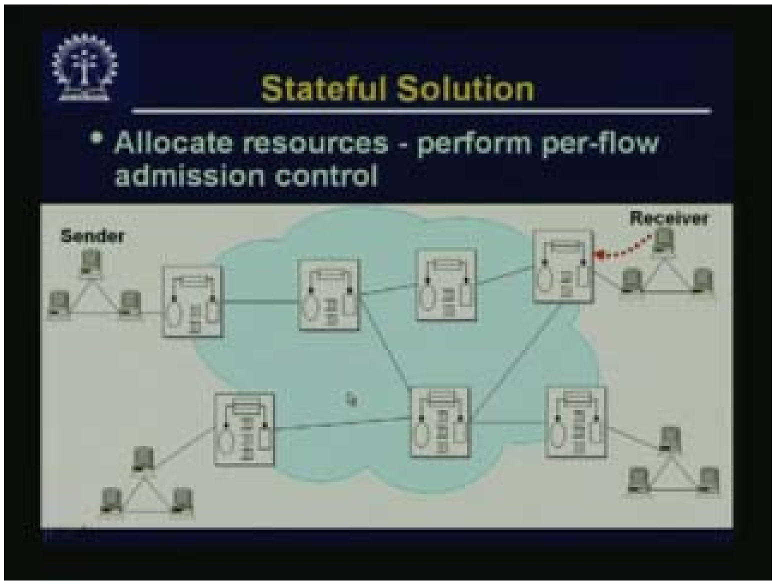


Stateful Solution

- Achieve per-flow bandwidth and delay guarantees
  - Example: guarantee 1MBps and < 100 ms delay to a flow

This achieves per-flow bandwidth and delay guarantees. For example, suppose we want a guarantee 1 MB/sec and < 100 ms delay to a flow and if we have a network like this, first the receiver will just allocate resources which means perform per-flow admission control.

(Refer Slide Time: 17:47 - 18:06)

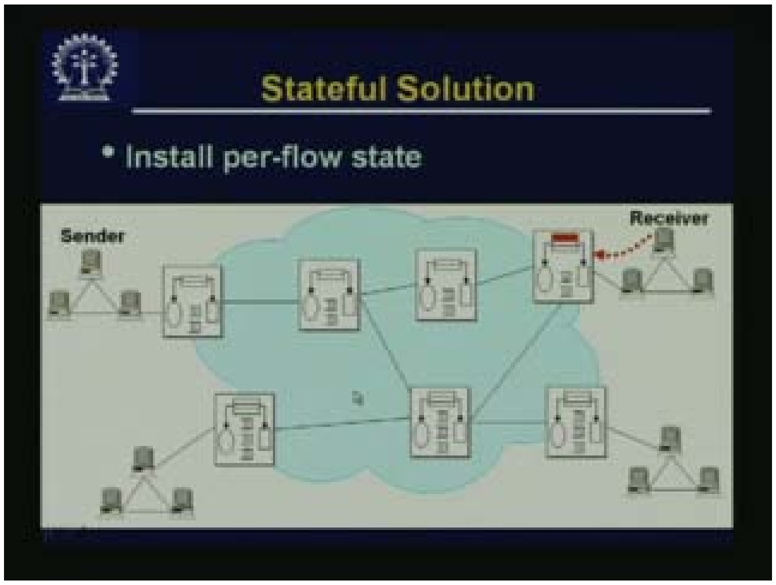


Stateful Solution

- Allocate resources - perform per-flow admission control

First the path has to be setup, that has not been shown and this is in the response path, so perform per-flow admission control.

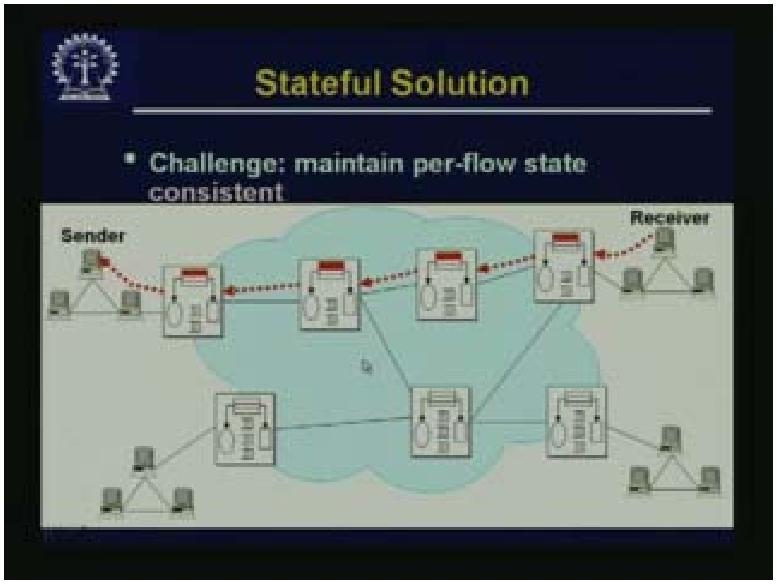
(Refer Slide Time: 18:07 - 18:14)


Stateful Solution

- Install per-flow state

Install per-flow state, so this flow has to be recognized, for that the router has to have a per-flow state.

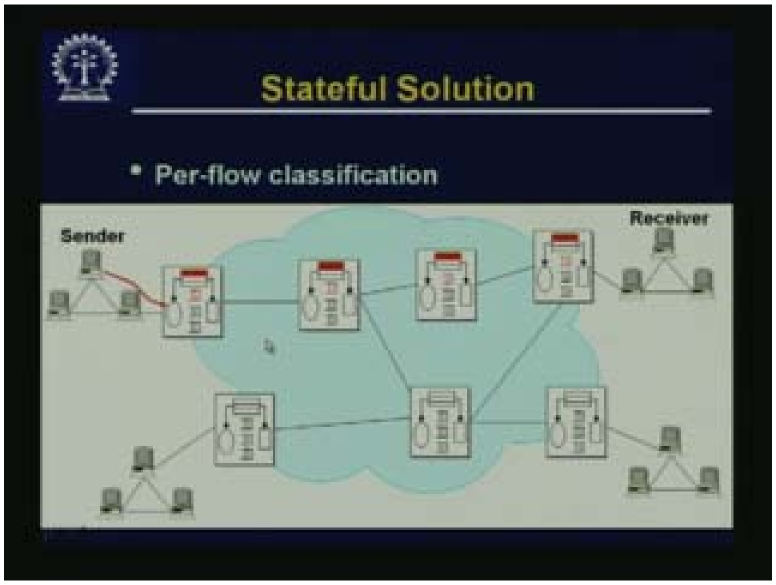
(Refer Slide Time: 18:15 - 18:25)


Stateful Solution

- Challenge: maintain per-flow state consistent

Challenge: maintain per-flow state in a consistent manner right up to the sender. If all the routers agree then only you can give it.

(Refer Slide Time: 18:26 - 18:36)

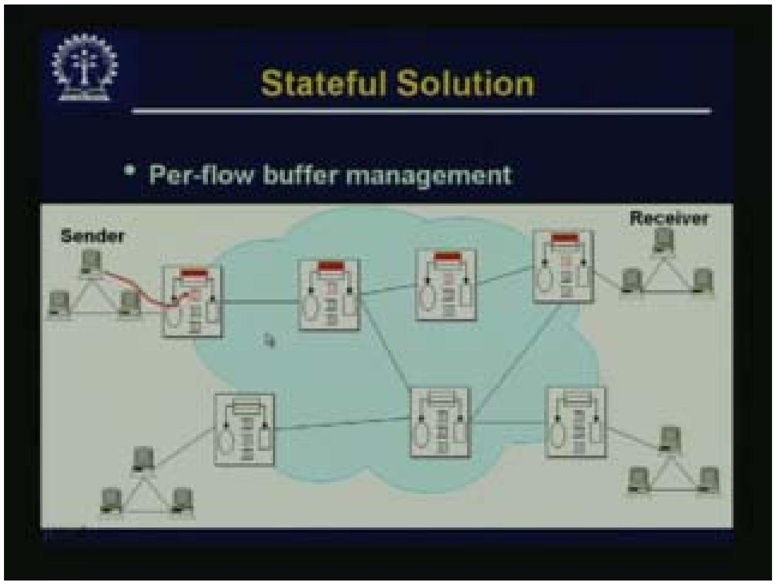


Stateful Solution

- Per-flow classification

When sender starts sending first of all it has to be classified that this is the integral part of this flow, so per-flow classification is necessary.

(Refer Slide Time: 18:37 - 18:48)

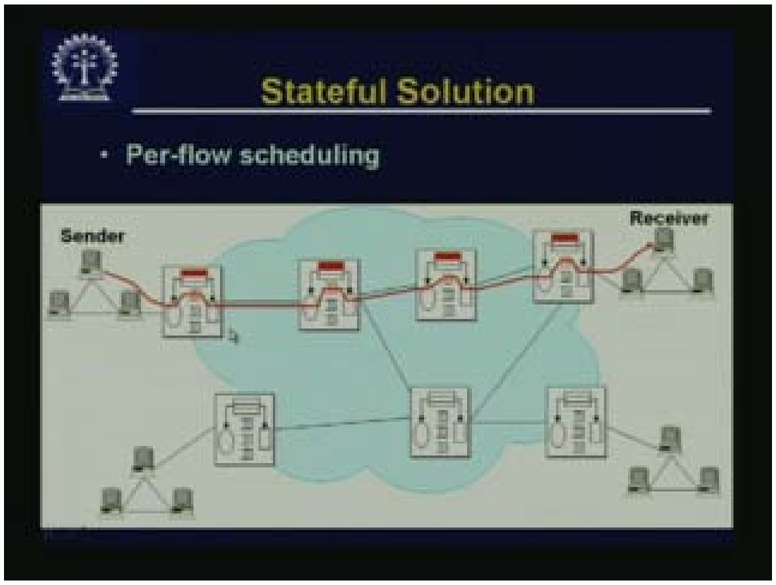


Stateful Solution

- Per-flow buffer management

Depending on how it has been classified, it will be put in the appropriate buffer and the corresponding buffer has to be managed, so per-flow buffer management is also necessary.

(Refer Slide Time: 18:49 - 19:03)

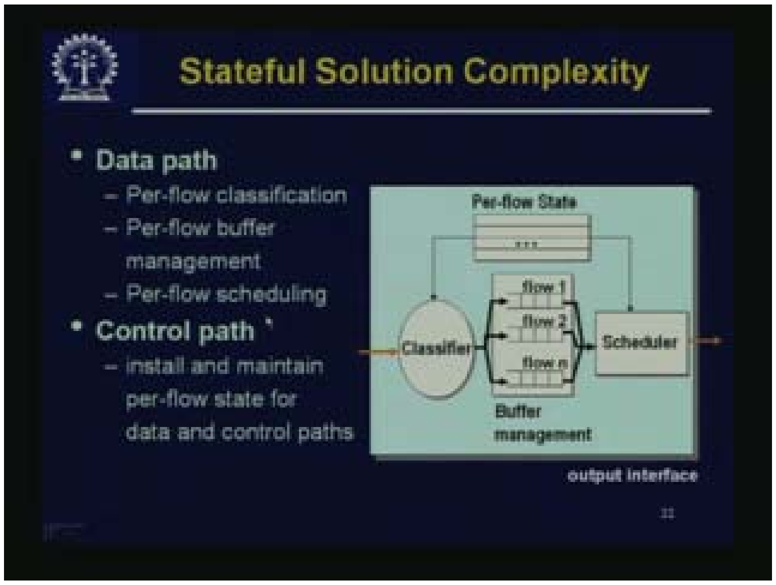


Stateful Solution

- Per-flow scheduling

And this way it will go through all the routers, and per flow scheduling is necessary for CPU resources and the flow will go through.

(Refer Slide Time: 19:04 - 19:56)

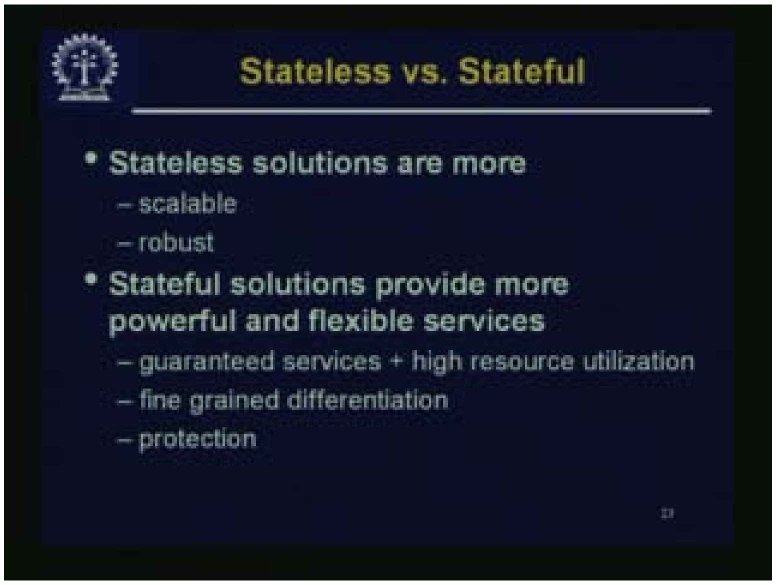


Stateful Solution Complexity

- Data path
  - Per-flow classification
  - Per-flow buffer management
  - Per-flow scheduling
- Control path
  - install and maintain per-flow state for data and control paths

In the data path, we have per-flow classification, per-flow buffer management, per-flow scheduling whereas in the control path we have the installed and maintained per-flow state for data and control paths. For each of the flow the router has to do a lot of work so this does not scale unfortunately. This is very good at giving services but if you are trying to maintain a per-flow kind of classification and management at each of the nodes, this would not scale because if the number of flows goes beyond a certain level then it will be very difficult to maintain. It will be very difficult to properly give it buffers and queues etc in the intermediate nodes so that is why it does not scale so well.

(Refer Slide Time: 19:57 - 21:00)



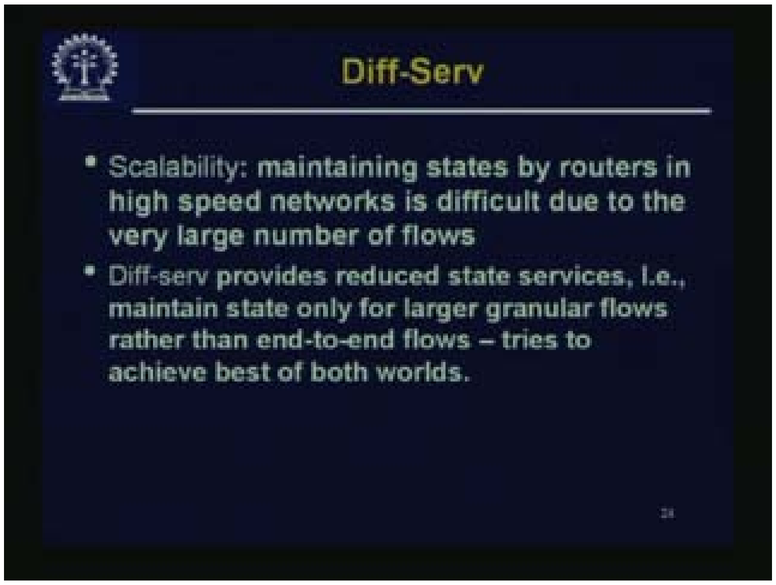
Stateless vs. Stateful

- Stateless solutions are more
  - scalable
  - robust
- Stateful solutions provide more powerful and flexible services
  - guaranteed services + high resource utilization
  - fine grained differentiation
  - protection

23

Stateless solutions are more scalable and they are more robust. Stateful solutions provide more powerful and flexible services. It gives guaranteed services plus high resource utilization. It can make fine grained differentiation between different flows and it also receives protection on the way since we are short of demarcating each of the flows. Demarcating the flow is somewhat similar, either to setting up ATM virtual circuits, when you do that, you also reserve request but that is done while call setup itself, that is not done by a separate reservation protocol. Hence it is similar to that or it maybe similar to MPLS where a particular flow can be differentiated from other flows and it maybe treated specially.

(Refer Slide Time: 21:01 - 22:56)



Diff-Serv

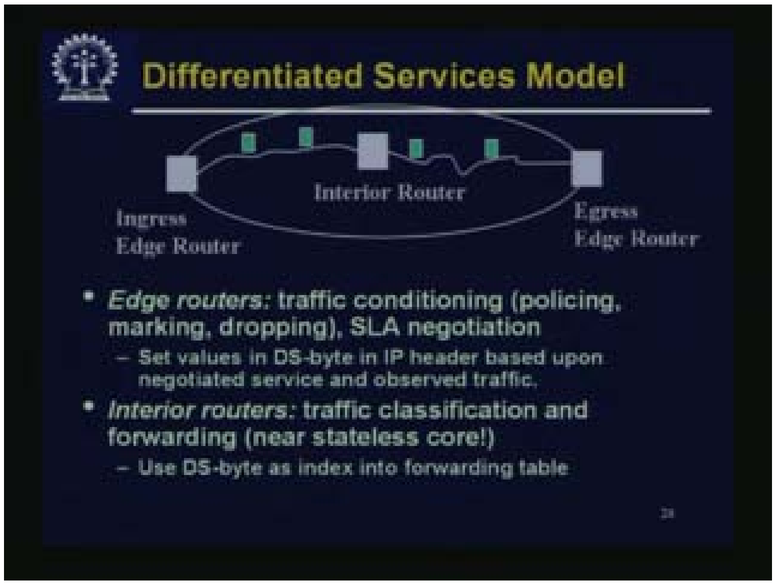
- Scalability: maintaining states by routers in high speed networks is difficult due to the very large number of flows
- Diff-serv provides reduced state services, i.e., maintain state only for larger granular flows rather than end-to-end flows – tries to achieve best of both worlds.

24

Then on one hand you have, the raw internet just takes a packet by itself and tries to route it and if it cannot route, it is the best effort and it goes off. That means it is discarded, that is one end of the spectrum. The other end of the spectrum is that you have a full power flow classification and management for each of the flows where you can have a very good control etc but it does not scale. So there is a question, can we do something in between and that is well the next model comes which is known as a diff-serv, differentiated service. This is also stateful but here each distinct flow does not necessarily mean a distinct state. These are sort of soft states which can accommodate a number of flows together so that the pressure on the routers in the intermediate nodes or in the backbone of the network becomes less.

Scalability: Maintaining states by routers in high speed networks is difficult due to a very large number of flows. So diffserv provides reduced state services, that is maintain state only for large granular flows rather than end to end flows, tries to achieve best of both worlds. So diffserv intends to address the following difficulties with intserv and RSVP. Flexible service models: intserv has only two classes, wants to provide more qualitative service classes, wants to provide relative service distinction like platinum, gold, silver etc and simpler signaling than RSVP. Many applications and users may only want to specify a more qualitative notion of service rather than a quantitative notion of service.

(Refer Slide Time: 22:57 - 24:00)



Differentiated Services Model

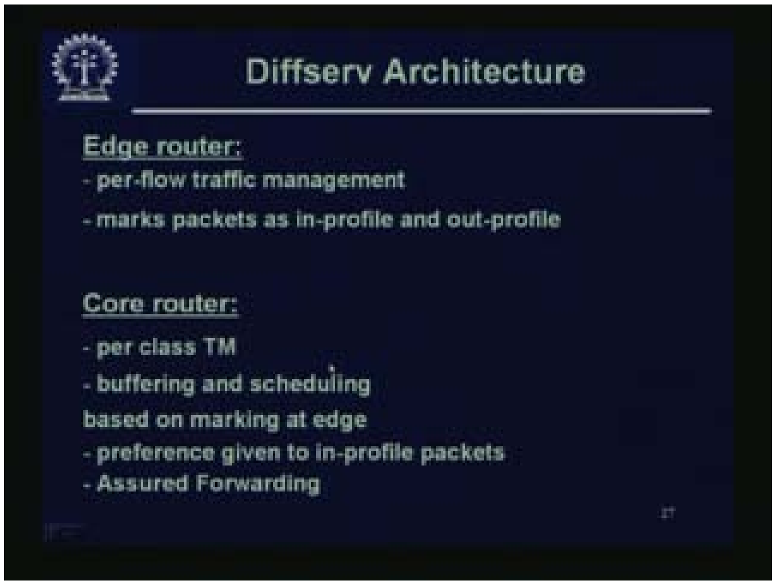
Interior Router
Ingress Edge Router
Egress Edge Router

- *Edge routers:* traffic conditioning (policing, marking, dropping), SLA negotiation
  - Set values in DS-byte in IP header based upon negotiated service and observed traffic.
- *Interior routers:* traffic classification and forwarding (near stateless core!)
  - Use DS-byte as index into forwarding table

28

This is a diffserv model, you have an ingress edge router and you have the interior routers, it goes to the egress edge router.
Edge routers: Traffic conditioning that means, policing, marking, dropping etc are done at the edge. The reason we want to do them at the edge of the network is that, there the number of flows etc is much lower and you can handle it over there and when you come to the core the core routers are very busy so we do not want to do that over there. Therefore set values in DS-byte in IP header based on negotiated services and observed traffic.
Interior routers: Traffic classification looking at this byte and forwarding near stateless core. This is almost like MPLS or ATM. Use DS-byte as index into a forwarding table.

(Refer Slide Time: 24:01 - 24:45)



Diffserv Architecture

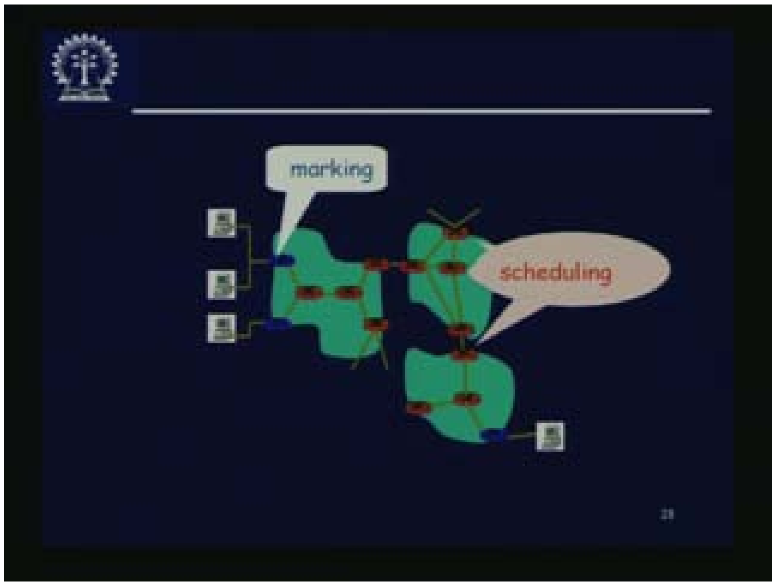
Edge router:
- per-flow traffic management
- marks packets as in-profile and out-profile

Core router:
- per class TM
- buffering and scheduling
based on marking at edge
- preference given to in-profile packets
- Assured Forwarding

27

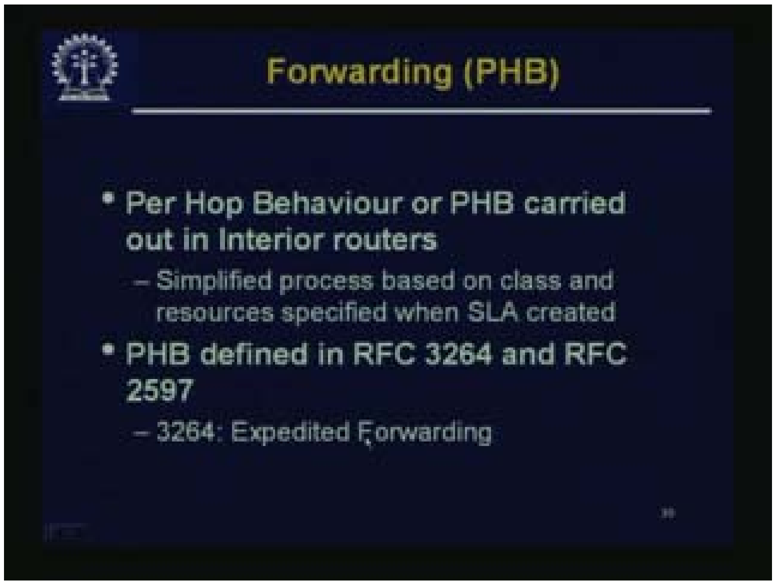
Edge router: It does per-flow traffic management, marks packets as in profile and out profile.
Core router: It does per class TM, buffering and scheduling based on marking at edge. Preference is given to in-profile packets. In-profile means some traffic shape has been negotiated or is expected and then something comes which is out of the shape. That means somebody is sending more packets per unit time than that was negotiated, so those are out of profile packets.
Assured forwarding: This is what the core router does.

(Refer Slide Time: 24:46 - 25:34)



Marking and power flow classification etc is done at the edge but in between the scheduling is done. The packet is marked in the type of service in IPv4 and traffic class in IPv6 renamed as DS. Then 6 bits used for differentiated service code point DSCP and determine the PHB or Per Hop Behavior that the packet will receive, 2 bits are currently unused. And this service code will dictate how this particular packet will be treated.

(Refer Slide Time: 25:35 - 26:47)

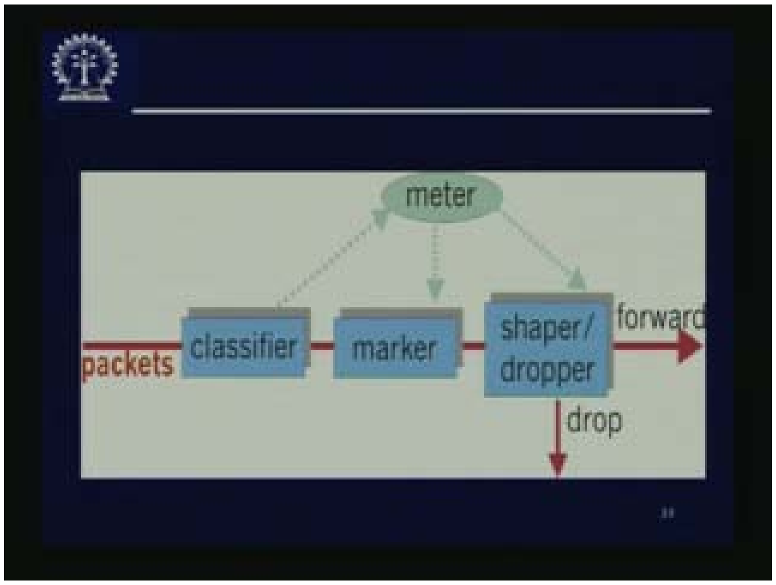


Forwarding (PHB)

- Per Hop Behaviour or PHB carried out in Interior routers
  - Simplified process based on class and resources specified when SLA created
- PHB defined in RFC 3264 and RFC 2597
  - 3264: Expedited Forwarding

For forwarding Per Hop Behavior or PHB carried out in the interior routers is simplified process based on class and resources specified when SLA was created. PHB has been defined in a certain RFC. For example for premium service, it should have low loss, low delay, low jitter, assigned bandwidth is equivalent to point to point leased line etc is the kind of service we want, although this is not leased, this is the plain packet switch network. It is guaranteed through policing and shaping in order to stay within the departure rate of leaky bucket and WFQ. Since all the intermediate nodes guarantee that it will get premium service. Traffic conditioning: It maybe desirable to limit traffic injection rate of some class, users declare traffic profile, example rate and burst size, traffic is metered and shaped if non-conformed. We have seen this already when we discussed congestion control.

(Refer Slide Time: 26:48 - 30:34)



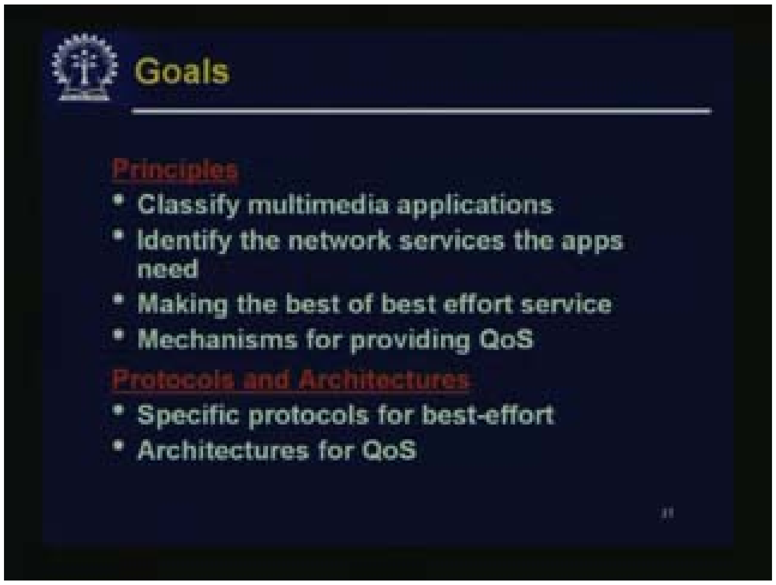
The packets are classified and then they are marked. When they are classified and marked naturally for the session, some kind of t spec that is traffic spec has been already negotiated so it meters to see whether it conforms or not and then the traffic maybe shaped. If by chance some burst comes and if the system can handle it then it will keep it for some time and then forward it only at an appropriate rate. But if it goes too much then it drops it here itself.

This QOI is absolutely important especially for multimedia traffic as was explained. for ordinary data traffic like web traffic or FTP, file download etc, those usually are not very sensitive to delay and variation in the delay that means you may get a lot of packets in one bunch then packets maybe slowing and coming extra and these things really do not matter much if it is within some bounds. But in multimedia, these things are very important.

For example, when we talk, if the delay is beyond a certain level then we feel very odd. For example, few years back if you had made an international call which is routed in a very peculiar way, you would find a lot of delay if you are speaking to somebody in USA, which is quiet annoying. Additionally what is annoying is that if the delay remains constant, some other people can manage it. But if the delay keeps varying a lot, that is, if there is a lot of jitter then that is very irritating for people.

On the other hand, unlike data, maybe a few bits are dropped here and there; it will not matter too much. Maybe if you analyze it using machines you will find some difference in quality but as such to human perception most of the time it does not matter too much. The multimedia traffic is quite different from ordinary data traffic. And as network is spreading and the way it is going, people find it cheaper and most convenient to use this internet or use this computer network for what was traditionally telecoms domain in the sense for audio or video conferencing. We have discussed about inserv, diffserv, RSVP etc, but the trouble is, many of the routers on the way are not going to support this because you need a consistent support throughout the path. So people still try to use the raw internet which is just handling it at the packet level not really taking note of the flows and still try to handle multimedia and for that a set of protocols has evolved. Let us just look at those protocols now.

(Refer Slide Time: 30:35 - 31:17)



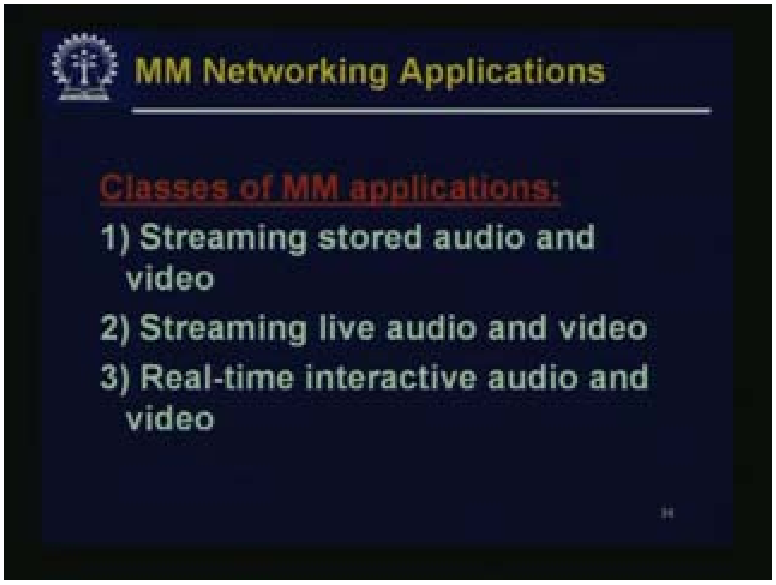
Goals

Principles
- Classify multimedia applications
- Identify the network services the apps need
- Making the best of best effort service
- Mechanisms for providing QoS

Protocols and Architectures
- Specific protocols for best-effort
- Architectures for QoS

Principles: Classify multimedia applications; identify the network services that the applications need. Making the best of best effort service, you already have a best effort service and how can you make the best of it is what we will see in RTP, etc and there are some mechanisms for providing QOS. These are specific protocols for best effort and architecture for QOS. At the network level if you have supported QOS these would have gone through much better.

(Refer Slide Time: 31:18 - 33:24)



MM Networking Applications

Classes of MM applications:

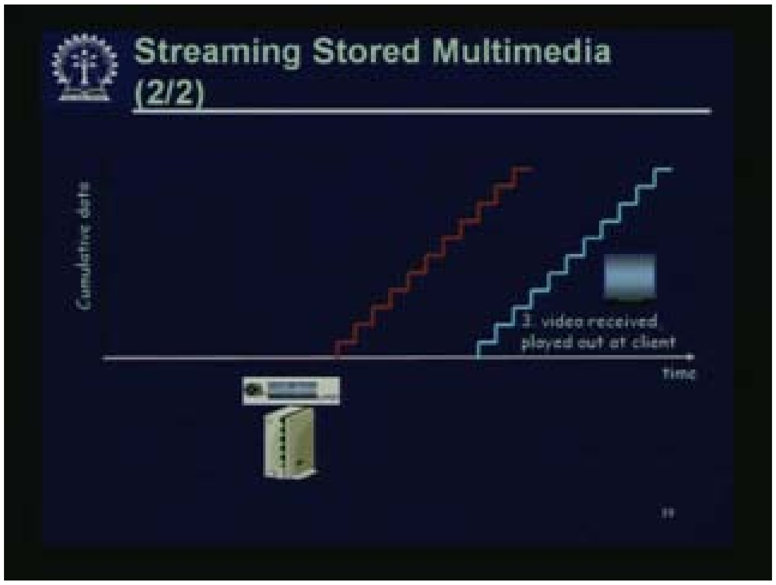
1) Streaming stored audio and video
2) Streaming live audio and video
3) Real-time interactive audio and video

There are various classes of MM applications and not all are the same. There is streaming of stored audio and video, that is one kind of application, for example, you download a song, movie that is a streaming, the stored audio and video is being streamed. Then there is streaming of live audio and video, then there are real time interactive audio and video and by this way you can generally classify. By multimedia application you mostly need audio and video so by this way they can be classified.

Fundamental characteristics: They are typically delay sensitive. They are also sensitive to delay jitter which is a variation in delay.

Loss tolerant: Infrequent losses cause minor glitches, so this is the antithesis of data which are loss intolerant but delay tolerant. Sometimes we like VCR like functionality. suppose we are seeing a video, so we try to equate it with the kind of functionality like in a VCR so client can pause, rewind, fast forward, push slider bar etc. And at the same time when a multimedia session starts maybe an initial delay of 5, 10 seconds is acceptable, 1 to 10 seconds until command effect is also acceptable but we need a separate control protocol because we really have no other way of getting any kind of control, feedback, etc on these streams as we will see. Timing constraint for still–to-be transmitted data: In time for play out.

(Refer Slide Time: 33:25 - 34:42)

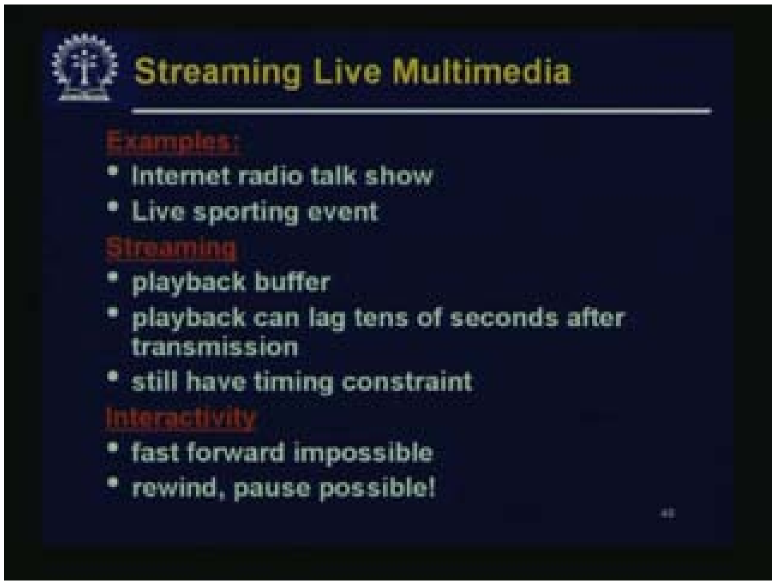


Streaming Stored Multimedia (2/2)

Suppose we have, streaming of stored multimedia. Suppose this is the recorded video which is being sent, this is the cumulative data that has been sent, this is like a staircase because they have been sent as packets so chunks of data is going together so they are being sent. First of all you record the video and the video is sent and in the third position the video is received and played at the client. This is streaming of stored multimedia. There is a network delay in between when the video is sent and the video is received. As soon as you send it, you cannot start seeing it.

Streaming: At this time the client is playing a part of the video while the server is still sending the later part of the video. Maybe in this particular diagram the client has already started seeing something although some of the data has still not been sent.

(Refer Slide Time: 34:43 - 38:05)



Streaming Live Multimedia

Examples:

- Internet radio talk show
- Live sporting event

Streaming

- playback buffer
- playback can lag tens of seconds after transmission
- still have timing constraint

Interactivity

- fast forward impossible
- rewind, pause possible!

Therefore the streaming of stored multimedia is different from streaming of live multimedia. For example, internet radio talk show is a live multimedia, a live sporting event which is being web cast. Streaming has a playback buffer, so playback can lag 10’s of seconds after transmission but it still has some timing constraint. The interactivity is limited, for example, obviously if it is live you cannot fast forward it but rewind and pause maybe possible. It solely depends on how you handle it. So this is unlike a stored multimedia where fast forward would also be possible.

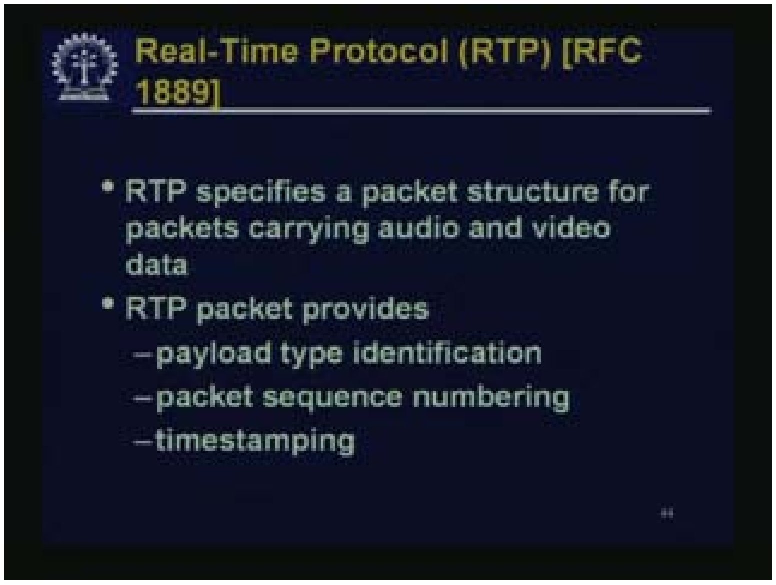
Interactive real time multimedia: Is one of the most important applications like IP telephony, video conference of the web, distributed interactive worlds etc are the applications of interactive real time multimedia which has the most stringent QOS requirements.

End-to-end delay requirements: Different applications have different kinds of requirements. for example if it is audio, if the end-to-end delay is less than 150ms it is good and less than 400ms is acceptable but beyond that people seem to be bothered a lot. It includes application level that is packetization and the application level delay and the network delays. Higher delays are noticeable and they impair the interactivity.

Session initialization: How does callee advertise its IP address port number, encoding algorithms etc because like in an IP telephony call all these issues are there. If you are using a standard TCP or UDP IP this is what you would have to do actually but there are no guarantees on delay or loss. But multimedia application requires some QOS and level of performance to be effective.

Today’s internet multimedia applications use application level techniques to mitigate as best as possible. This is what we mean by best efforts, this means we cannot do anything, maybe at the network layer etc it would implement an inserv or diffserv, but in most of the places that is not practical at the moment so we try to do something at the end points i.e. at the application level. The application level necessarily means the end point because it is the intermediate, it will not go right up to the application layer but maybe it will penetrate a little deeper. This is mostly done from end to end.

(Refer Slide Time: 38:06 - 40:18)



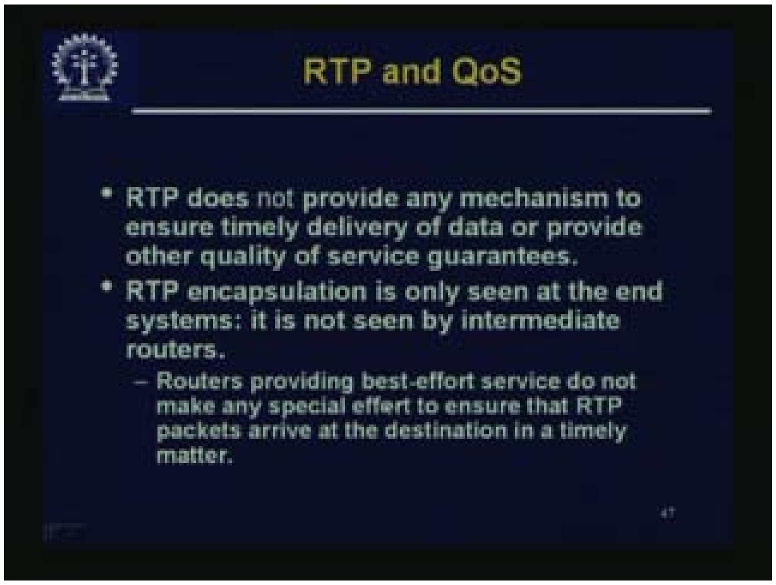
Real-Time Protocol (RTP) [RFC 1889]

- RTP specifies a packet structure for packets carrying audio and video data
- RTP packet provides
  - payload type identification
  - packet sequence numbering
  - timestamping

In this pseudo protocol, the number of protocols, first we look at real time protocols RTP that is RFC 1889. RTP specifies a packet for packets carrying audio and video data. So RTP packet provides payload type identification such as what type of payload it is carrying, packet sequence numbering, time stamping that is important for giving some feedback. RTP runs in the end systems. RTP packets are encapsulated in UDP segments. Interoperability: If two internet phone applications run RTP then they maybe able to work together and may communicate to each other through the network. RTP runs on top of UDP.

We have seen in both TCP and UDP, the UDP is a very lightweight protocol so it is also expected to be much more efficient than TCP. TCP is a heavy weight protocol so it has a lot of overhead and all these ACKs keep on flowing. RTP libraries provide some kind of support over an above the transport layer library, that is how a RTP runs on UDP so this is a provider transport layer interface that extend a UDP. So port numbers, IP addresses are there in UDP itself and then payload type identification, packet sequence numbering and time stamping is what RTP adds to UDP. And then, the application is here, it moves a little bit into the application domain. Maybe just look at the RTP header to do some what better service.

(Refer Slide Time: 40:19 - 41:43)



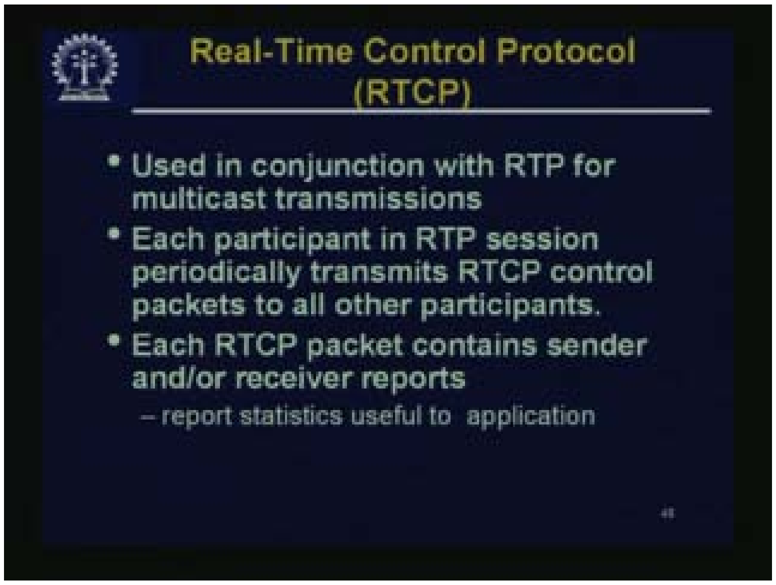
RTP and QoS

- RTP does not provide any mechanism to ensure timely delivery of data or provide other quality of service guarantees.
- RTP encapsulation is only seen at the end systems: it is not seen by intermediate routers.
  - Routers providing best-effort service do not make any special effort to ensure that RTP packets arrive at the destination in a timely matter.

RTP by itself does not provide any mechanism to ensure timely delivery of data or provide other quality of service guarantees, because that really depends upon what is happening on the way. An RTP is running just between the two end points. So RTP encapsulation is only seen at the end system which is not seen by intermediate routers. So routers providing best effort service do not make any special effort to ensure that RTP packets arrive at the destination in a timely manner. The services in between is still a best effort.

Since we look at it at the end points there we can effect on some kind of control. If you have noted, this RTP header goes in front of the payload for the data. For example, it is the voice which is traveling then the voice signal is digitized and packetized before each packet produced by your application layer, the RTP header will come which is available in the data. There is a separate protocol control called Real Time Control Protocol or RTCP.

(Refer Slide Time: 41:44 - 43:25)



Real-Time Control Protocol (RTCP)

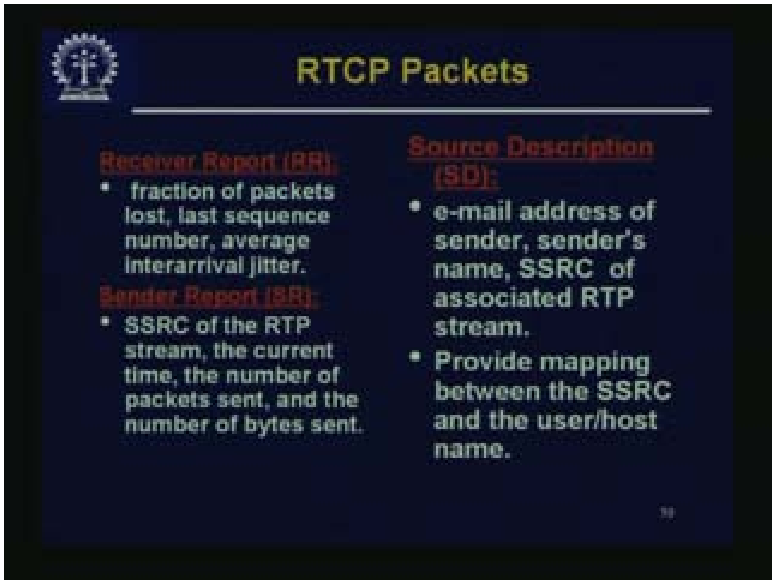
- Used in conjunction with RTP for multicast transmissions
- Each participant in RTP session periodically transmits RTCP control packets to all other participants.
- Each RTCP packet contains sender and/or receiver reports
  - report statistics useful to application

48

This is used in conjunction with RTP for multicast transmissions. Each participants in RTP session periodically transmits RTCP control packets to all the other participants. Each RTCP packet contains sender and receiver reports. So reports statistics etc are useful to applications and this is what you might use to improve the performance of your session. Inside the network, inside the backbone nobody really knows about your session so it is only the end points. But they exchange this kind of a statistics through the RTCP packets and then the application can control the rate at which things can be done, etc.

Therefore, some kind of feedback mechanism is present. The statistics include number of packets sent, number of packets lost, inter arrival jitter etc. Feedback can be used to control performance, sender may modify its transmissions based on feedback. This is the main point, you get some kind of feed back as this is running on UDP. There is no ACKs, packets coming on which you can send these statistics. Therefore what is done is that, this RTCP is actually in band control so a part of the bandwidth is kept for RTCP and through this bandwidth they exchange this kind of information and the time stamp, so you know that how much delay is being experienced at the other end. And after having all these statistics you may do what you can at the application layer to modify the way you are transmitting in order to get better service.

(Refer Slide Time: 43:26 - 44:14)



RTCP Packets

Receiver Report (RR):

- fraction of packets lost, last sequence number, average interarrival jitter.

Sender Report (SR):

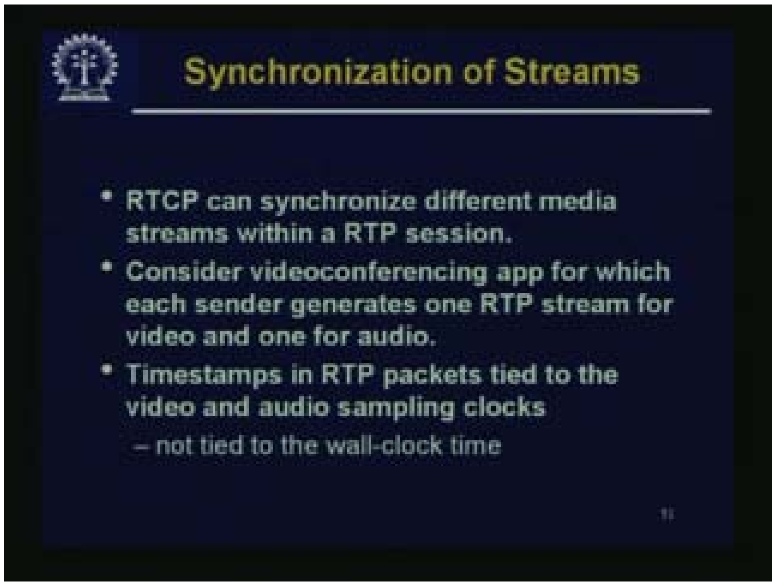
- SSRC of the RTP stream, the current time, the number of packets sent, and the number of bytes sent.

Source Description (SD):

- e-mail address of sender, sender's name, SSRC of associated RTP stream.
- Provide mapping between the SSRC and the user/host name.

There are receiver reports, fraction of packets lost, last sequence number, average inter arrival jitter, etc are a part of the receiver report. Sender report: SSRC of the RTP stream, the current time, the number of packets sent, and the number of bytes sent. Source Description: E-mail address of sender, sender's name, SSRC of associated RTP stream etc. Provide mapping between SSRC and the user or host name.

(Refer Slide Time: 44:15 - 45:46)

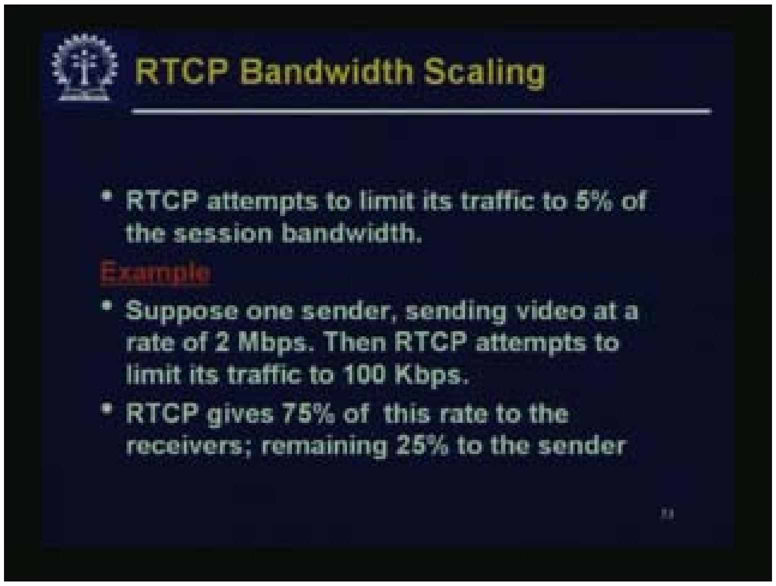


Synchronization of Streams

- RTCP can synchronize different media streams within a RTP session.
- Consider videoconferencing app for which each sender generates one RTP stream for video and one for audio.
- Timestamps in RTP packets tied to the video and audio sampling clocks
  - not tied to the wall-clock time

Another use of RTCP is for synchronization. RTCP can synchronize different media streams within a RTP session. So consider video conferencing applications for which each sender generates one RTP stream for video and one for audio. The point is, you have got two streams of packets for the video streams and then the audio has to be synchronized with it. The lips are moving in some other way and sound is coming in some other way so that will not help in anyway hence they have to be synchronized. In order to synchronize this, RTCP uses the time stamp which I referred to earlier and tries to synchronize the two different streams. This is another important use of RTCP. So time stamps in the RTCP packets are tied to the video and audio sampling clocks. They are not tied to the wall clock time. The wall clock times are also exchanged and this is to get an idea about the delay. Each RTCP sender report packet contains for the most recently generated packet in the associated RTP stream, timestamp of the RTP packet and wall clock time for when packet was created. This gives you an idea about the delay. Receivers can use this association to synchronize the play out of audio and video.

(Refer Slide Time: 45:47 - 48:02)



RTCP Bandwidth Scaling

- RTCP attempts to limit its traffic to 5% of the session bandwidth.

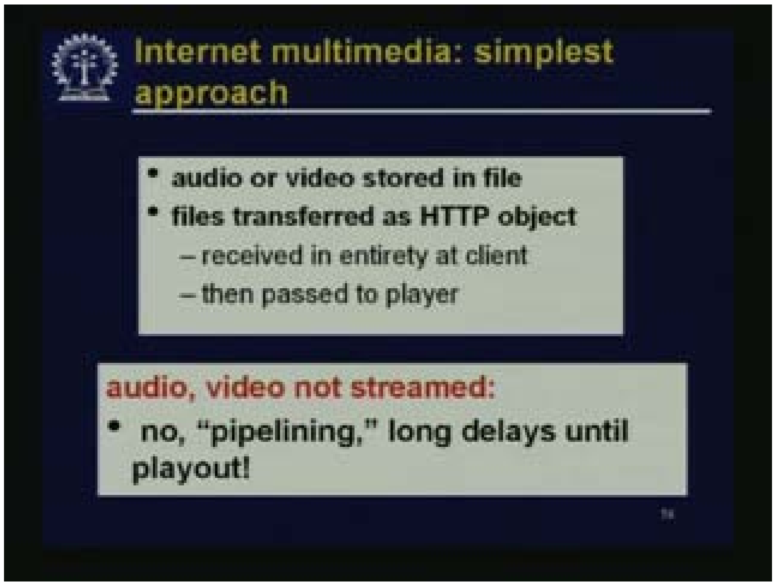
Example

- Suppose one sender, sending video at a rate of 2 Mbps. Then RTCP attempts to limit its traffic to 100 Kbps.
- RTCP gives 75% of this rate to the receivers; remaining 25% to the sender

RTCP Bandwidth Scaling: RTCP attempts to limit its traffic to 5% of the session bandwidth. Remember, that this is in band control that means whatever bandwidth is given for the session it is within that only. So whatever the RTP actual data packets are using, maybe 5% of that, the RTCP tries to use and in this 5% it is kind of divided, like 25% for the sender, and 75% for the receiver because there may be many receivers and this 75% will again be distributed amongst the receivers.

Suppose one sender sending video at a rate of 2 mbps then RTCP attempts to limit its traffic. That means to the RTCP traffic to 100 kbps. RTCP gives 75% percent of this rate to the receivers, remaining 25% to the sender. We come to another protocol called RTSP known as the Real Time Streaming Protocol. Earlier we were talking about interactive live audio or video conferencing, but now let us talk about streaming. User interactive control is provided, for example, the real time streaming protocol where a helper application displays content which is typically requested via a web browser, for example a real player, then we have the typical functions namely decompression if it is compressed in audio or video, jitter removal which means you play them back at a constant rate, error correction: use redundant packets to be used for reconstruction of original stream, some kind of Graphical User Interface for user controls like rewind, fast forward, etc are given by RTSP. What it does not do is, it does not define how audio and video is encapsulated for streaming over network, it does not restrict how streamed media is transported, whether TCP or UDP does not specify how the media player buffers audio and video so these are more at the application layer.

(Refer Slide Time: 48:03 - 49:08)



Internet multimedia: simplest approach

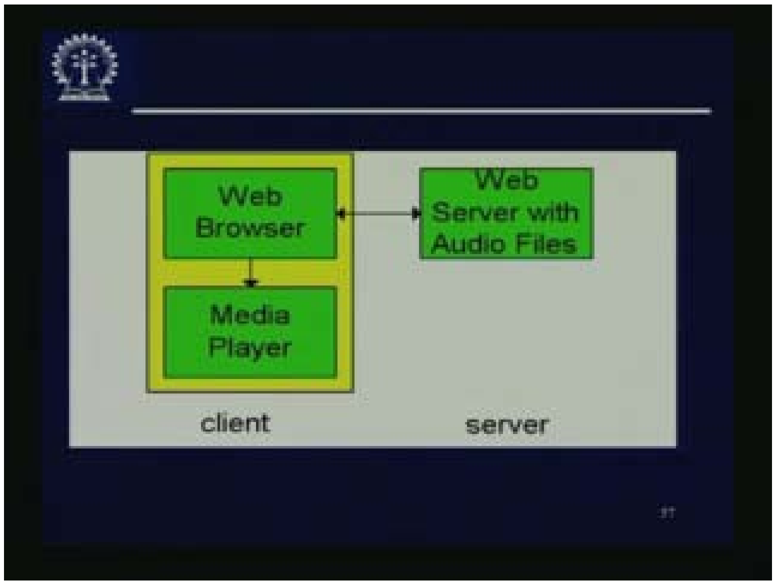
- audio or video stored in file
- files transferred as HTTP object
  - received in entirety at client
  - then passed to player

audio, video not streamed:

- no, "pipelining," long delays until playout!

This is the simple approach for steaming audio or video. Audio or video is stored in file. Files are transferred as http object that is received in entirety at client then pass to player. So audio or video are not streamed and so no pipelining, long delays until play out. Here the problem is that if it is some kind of a movie and if it is downloaded then the entire file is downloaded, so downloading that entire file will take a lot of time, this is a very simple approach and then there is no problem, you can download the entire file and then play it use options like fast forward or reverse or speed up etc. But streaming in the media means, as it starts streaming the audio or video file then after some delay the play back also starts.

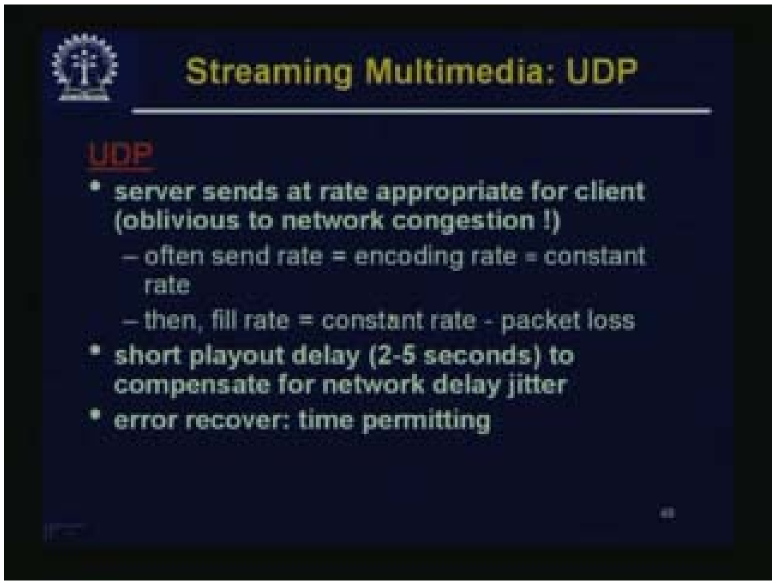
(Refer Slide Time: 49:09 - 50:43)



Simple situation is, we have a web browser, web server with audio files and a media player and that is how we do it. The ACTP request, so browser gets a meta file then the browser launches a player passing the metafile to the player and the player contacts the server and the meta file contain all the necessary information, this is called progressive download. Let us look at the streaming multimedia client buffering.

Suppose as before this is the constant bit rate video that is being transmitted and now because of the network delay and the availability of the network delay this is what the profile looks like of how it is received. But if you do this after some sufficient delay and you buffer it, the main point is, these two lines are never crossing. So now you can play it at a constant rate so all these delay variation etc that will sort of go and you would get a nice streaming audio or video. This is the basic idea of streaming multimedia. In this part, the difference between these two where this is the buffered part, so the client side buffering play out delay compensate for network added delay or delay jitter etc.

(Refer Slide Time: 50:44 - 51:29)

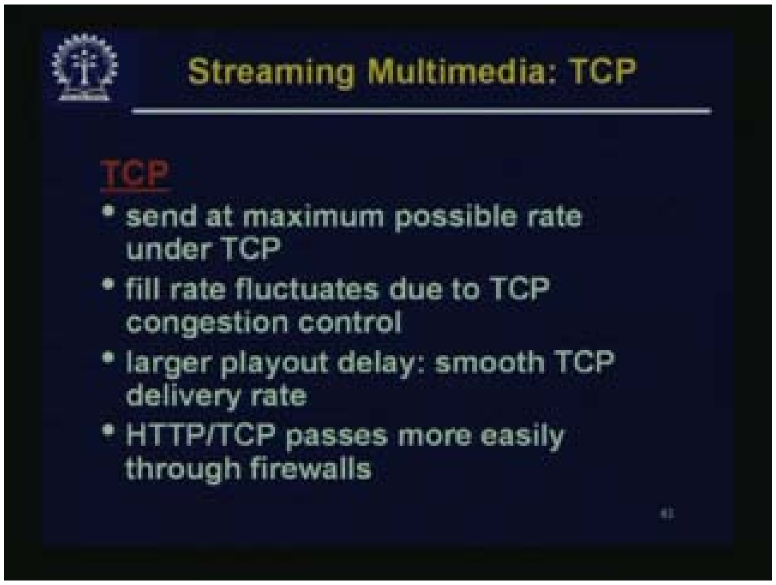


Streaming Multimedia: UDP

UDP
- server sends at rate appropriate for client (oblivious to network congestion !)
  - often send rate = encoding rate = constant rate
  - then, fill rate = constant rate - packet loss
- short playout delay (2-5 seconds) to compensate for network delay jitter
- error recover: time permitting

Streaming multimedia: Is where UDP is usually preferred. We will also look at how TCP is done. The server sends at rate appropriate for client oblivious to network congestion. So, often sent rate is equal to encoding rate that is constant rate, then there is a fill rate for constant rate minus packet loss. Short play out delay 2-5 seconds compensate for the network delay jitter. Somebody does not really mind so much, that if the play back starts after 5 seconds but in this 5 seconds the entire thing will be buffered and that will take care of the jitter. Error recovery takes place if time permits.

(Refer Slide Time: 51:30 - 52:07)

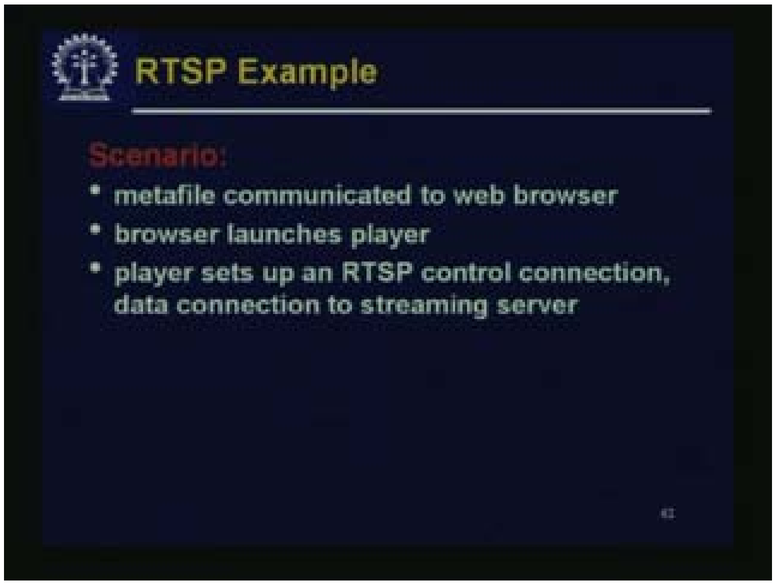


Streaming Multimedia: TCP

TCP
- send at maximum possible rate under TCP
- fill rate fluctuates due to TCP congestion control
- larger playout delay: smooth TCP delivery rate
- HTTP/TCP passes more easily through firewalls

TCP: it sends at maximum possible rate under TCP and fill rate fluctuates due to TCP congestion control because in TCP congestion control the windows size can keep on varying so larger play out delay is smooth TCP delivery rate. If you start playing it after quite some time so that you can handle this TCP variation in rates etc then you can get a smooth rating. Then http TCP passes more easily through firewalls rather than UDP.

(Refer Slide Time: 52:08 - 52:24)



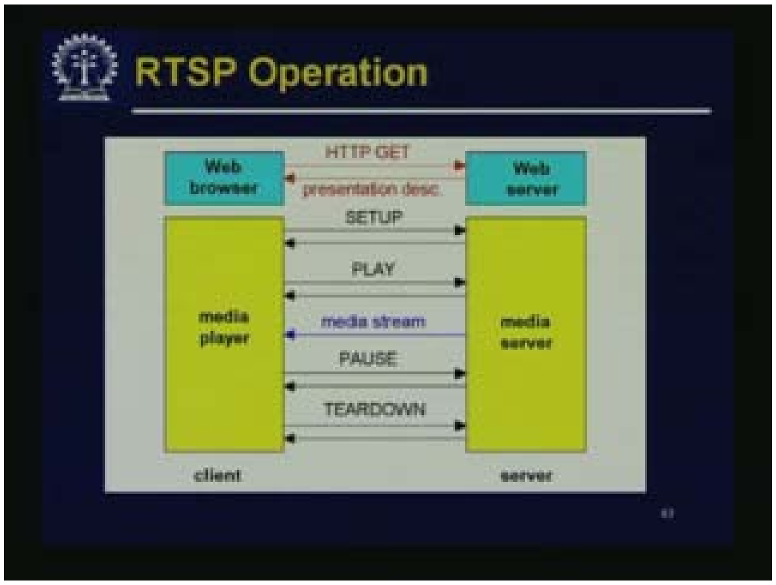
RTSP Example

Scenario:

- metafile communicated to web browser
- browser launches player
- player sets up an RTSP control connection, data connection to streaming server

RTSP example: Meta file is created and communicated to the web browser. Browser launches player as we have seen and player sets up an RTSP control connection data connection to streaming server.

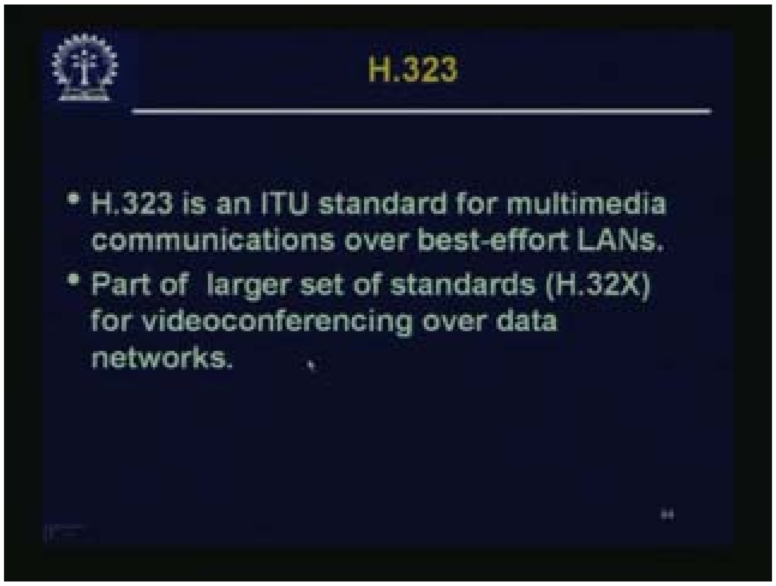
(Refer Slide Time: 52:25 - 52 - 47)



RTSP Operation

Web browser gets an http GET and web server presentation description. Then the media player is setup a connection with the media server and it starts getting it, buffers it and starts playing. The media stream comes, then you may pause, you may teardown, etc.

(Refer Slide Time: 52:48 - 53:09)

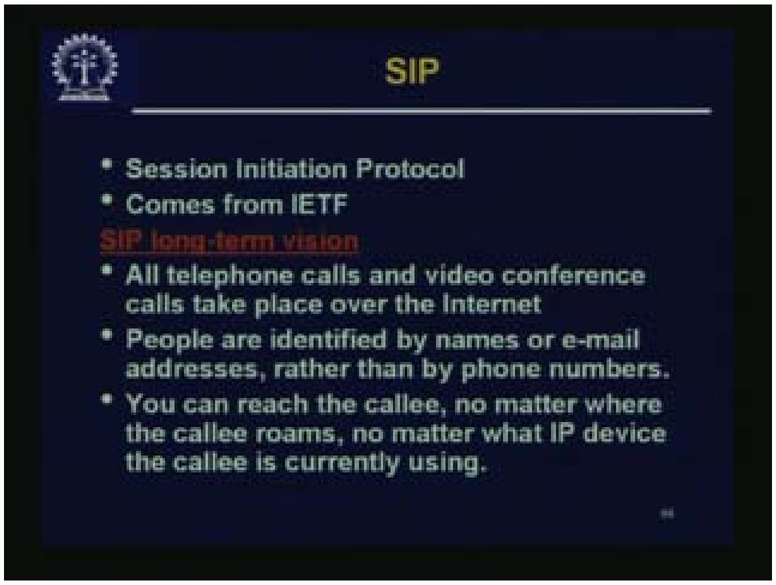


H.323

- H.323 is an ITU standard for multimedia communications over best-effort LANs.
- Part of larger set of standards (H.32X) for videoconferencing over data networks.

Finally I will just mention about two protocols namely: H.323 which is an ITU standard which is the old protocol for setting up telephone calls. H.323 is aging a little bit and there is a newer protocol namely session initiation protocol.

(Refer Slide Time: 53:10 - 53:28)



SIP

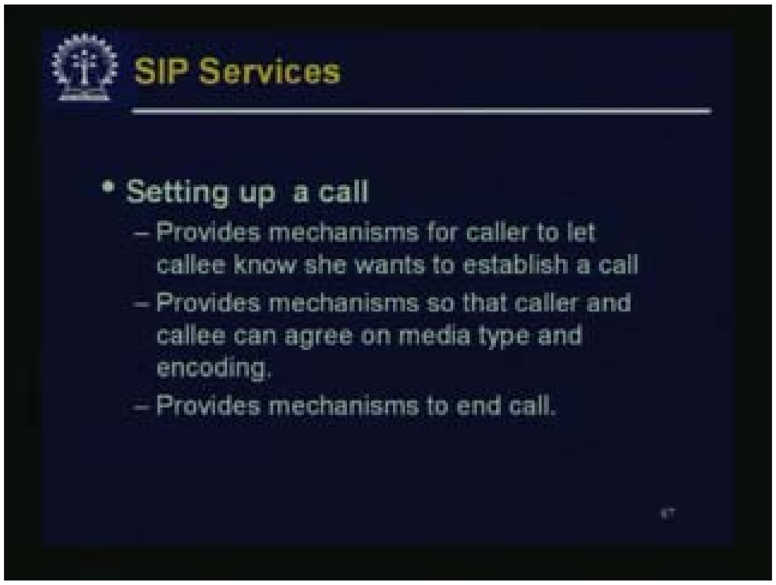
- Session Initiation Protocol
- Comes from IETF

SIP long-term vision

- All telephone calls and video conference calls take place over the Internet
- People are identified by names or e-mail addresses, rather than by phone numbers.
- You can reach the callee, no matter where the callee roams, no matter what IP device the callee is currently using.

All telephone calls and video conferencing calls take place over the internet. That is the vision, you can reach the callee no matter where the callee roams, no matter what IP devices the callee is currently using. Therefore, for setting up calls there is a large protocol.

(Refer Slide Time: 53:29 - 53:58)

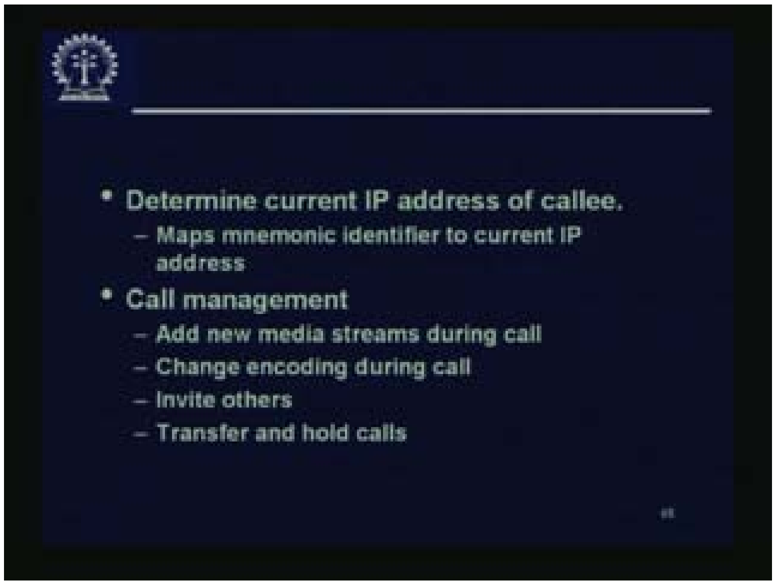


SIP Services

- Setting up a call
  - Provides mechanisms for caller to let callee know she wants to establish a call
  - Provides mechanisms so that caller and callee can agree on media type and encoding.
  - Provides mechanisms to end call.

Setting up a call: Provides mechanisms for caller to let callee know she wants to establish a call. And provides mechanism so that caller and callee can agree on media type and encoding because that has to agree and it provides a mechanism to end the call. The first one is called an invite and there are all kinds of commands for this. It also has the facility to determine the current IP address of callee, maps mnemonic identifier to current IP address.

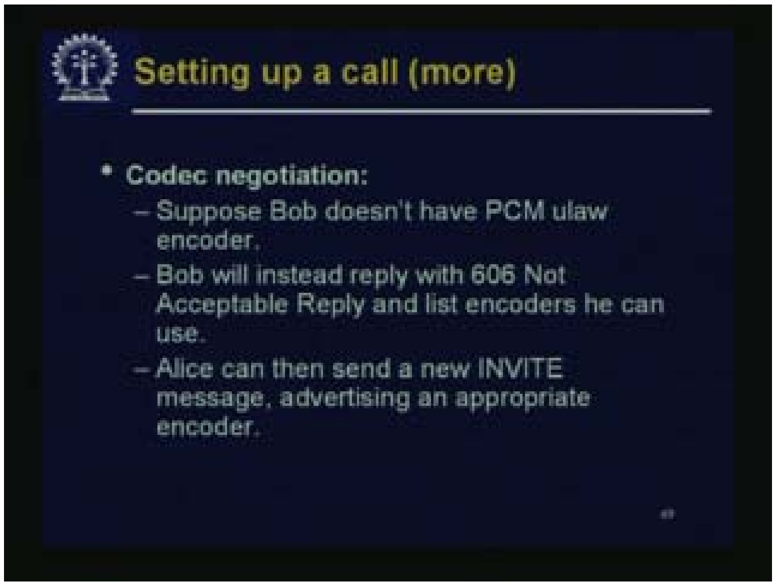
(Refer Slide Time: 53:59 - 54:29)



- Determine current IP address of callee.
  - Maps mnemonic identifier to current IP address
- Call management
  - Add new media streams during call
  - Change encoding during call
  - Invite others
  - Transfer and hold calls

There are other components of chip like, chip register and so on. Then it has something for call management, that means add new media streams during call, change encoding during calls, invite others for a conference, transfer and hold calls etc.

(Refer Slide Time: 54:30 - 54:38)

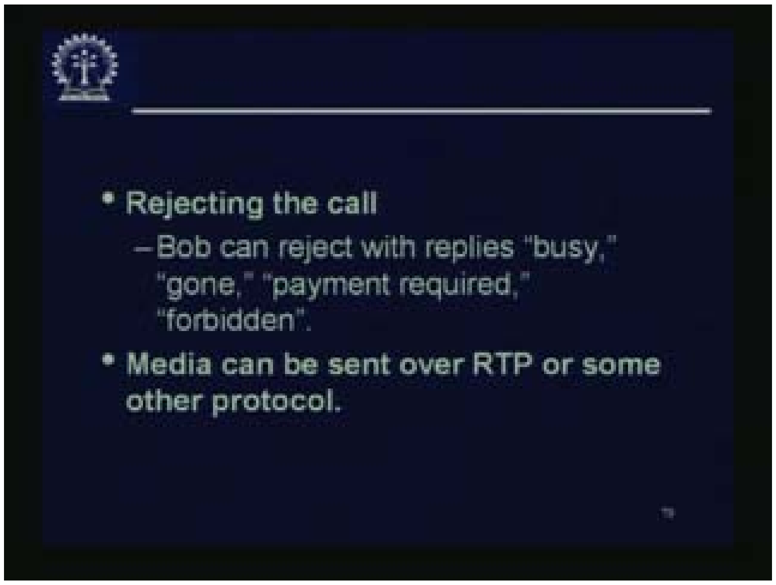


Setting up a call (more)

- Codec negotiation:
  - Suppose Bob doesn't have PCM ulaw encoder.
  - Bob will instead reply with 606 Not Acceptable Reply and list encoders he can use.
  - Alice can then send a new INVITE message, advertising an appropriate encoder.

There are all kinds of details about codec negotiation etc.

(Refer Slide Time: 54:39 - 56:15)



- Rejecting the call
  - Bob can reject with replies "busy," "gone," "payment required," "forbidden".
- Media can be sent over RTP or some other protocol.

And then you may reject a call also. The audio, video and multimedia over the internet is going very strongly. At one point of time network performance was so poor that there was no question of accommodating this. But as the network performance has increased exponentially over the past years now it is becoming much possible to handle lot of multimedia traffic through network. This is getting more popular. We talked about inserv, gifserv but the actual strategy that people actually deploy is when they feel this is not sufficient then they put in more bandwidth so put another fiber and take up your multiplexer weight etc so that you solve the problem because the demand for multimedia is so great.