VOIP/SIP
Mixer/Translator

- RTP Mixer, translator
- A mixer combines several media streams into a new stream (with possible new encoding)
- Reduced bandwidth networks (video or telephone conference)
- Appears as a new source, with its own identifier
- Translator:
  - A translator changes encoding from one format to another
  - Single media stream
  - May convert encoding
SSRC= Synchronizing Source Identifier
CSRC= Contributing Source Identifier

SSRC=5
CSRC=17, CSRC=19
PCM=64Kbps

SSRC=17
PCM=64Kbps

SSRC=19
PCM=64Kbps

SSRC= Synchronizing Source Identifier
CSRC= Contributing Source Identifier
Translator

End system source

SSRC=17

PCM=64Kbps

SSRC=7

CSRC=17

MPEG Audio=32 Kbps to 320 Kbps

SSRC=7
Mixer and translator in tandem

SSRC= Synchronizing Source Identifier
CSRC= Contributing Source Identifier

SSRC=17
SSRC=19

PCM=64Kbps

MP3

Translator

Mixer

SSRC=7
CSRC=17, CSRC=19

SSRC= Synchronizing Source Identifier
CSRC= Contributing Source Identifier
Internet telephony

- Phone service over internet
- Voice converted into packets
- Use RTP to transport audio to end-points
- VOIP: implement all features available in the traditional phone network (Public Switched Telephone Network-PSTN)

Main Problems: Should interoperate with existing PSTN
Need to identify end-points
VOIP modes

I) Caller has phone,
II) Callee has Phone
III) Both have IP Phones-PCs
Session initiation protocol

- SIP core protocol for establishing sessions
- Handles setup, modification, and teardown of VoIP session
- Transports session information among end-points
- Signaling involves sending messages
SIP addressing

- SIP uses naming similar to e-mail
- SIP URLs
  - SIP:badri@cs.rutgers.edu
  - SIP:17324452082@rutgers.edu
  - SIP:17324452082@rutgers.edu;user=phone
SIP allows mobility of endpoints

SIP:jlo@afleck.com
afleck.com
SIP:jlo@eden.afleck.com
Tel: 17325452222
Tel: 17324452083
SIP network entities

- SIP user agent
  - IP phone, runs Voip software
- SIP redirect server
  - Redirects connection requests
- SIP proxy server
  - Handles SIP requests on behalf of user
- SIP registrar
  - Maintains mapping from names to addresses
SIP redirect server

- Accept SIP requests and maps destination address to zero or more new addresses
- Allows callee to contact new location directly
- Redirect server does not initiate any SIP requests of its own

SIP:badri@cs.rutgers.edu
Moved: badri@home
SIP:badri@home
200: OK
SIP proxy server

- Accepts SIP requests on behalf of user
- Proxy server initiates SIP requests at new location if user has changed end-points
- Callee sees proxy server as end-point
- Proxy server hides end-point mobility from callee
SIP Registration

Register
SIP: badr@cs.rutgers.edu
Contact: 128.64.8.2

DNS/location server

Invite
SIP: badri@cs.rutgers.edu

Invite
SIP: 128.64.8.2
Locating users

- Find Domain of proxy server using SIP URL
- From DNS lookup find proxy servers (registrar as well)
- From proxy server learn IP address of end-point
- May have to go through multiple redirections (URL remappings)
SIP messages

- Similar to HTTP
- Request, responses are text messages
- Messages can contain body which relate to session descriptions
SIP methods

- INVITE – initiate call
- ACK – confirm response
- BYE – terminate call
- CANCEL – cancel searches
- OPTIONS - features supported on the other side
- REGISTER – register with location service
SIP responses

- Can convey a lot of status information with responses
- 200: OK
- 100: TRYING
- 180: Ringing
- 181: call forwarded
- 300: multiple choices
SIP call Establishment

INVITE

Ringing

OK

ACK

Conversation

BYE

OK
PSTN internetworking

- A gateway is required to convert SIP messages to circuit switched signaling and vice-versa
- Proxy server will contact PSTN gateway
- PSTN gateway initiates call to the PSTN callee
- Two-way audio conversation occurs through the gateway
Quality of service (QoS) Ch 6.5

**Best effort internet**
- Does not offer any guarantees on delay, bandwidth, and loss
- Many applications (IP telephony, video, etc) require delay bounds
- QoS in a network principles: network, application-level support for multimedia
  - making the best of best effort service
  - mechanisms for providing QoS
- specific protocols, architectures for QoS
Protocols for QoS

- Support protocols for multimedia applications and requirements
- making the best of today’s best effort service
- policing mechanisms
- Resource reservation protocols
  - Intserv
  - RSVP
  - Diffserv
Multimedia, Quality of Service: What is it?

Multimedia applications: network audio and video

QoS

network provides application with *level of performance needed for application to function*. 
Multimedia Performance Requirements

Requirement: deliver data in “timely” manner

- **interactive multimedia**: short end-end delay
  - e.g., IP telephony, teleconf., virtual worlds, DIS
  - excessive delay impairs human interaction

- **streaming (non-interactive) multimedia**:
  - data must arrive in time for “smooth” playout
  - late arriving data introduces gaps in rendered audio/video

- reliability: 100% reliability not always required
Policing mechanisms= Leaky Bucket

- Used in conjunction with resource reservation to police the host’s reservation
- At the host-network interface, allow packets into the network at a constant rate
- Packets may be generated in a bursty manner, but after they pass through the leaky bucket, they enter the network evenly spaced
Leaky Bucket: Analogy
Leaky Bucket (cont’d)

- The leaky bucket is a “traffic shaper”: It changes the characteristics of packet stream.
- Traffic shaping makes more manageable and more predictable.
- Usually the network tells the leaky bucket the rate at which it may send packets when the connection begins.
Leaky Bucket:
Doesn’t allow bursty transmissions

- In some cases, we may want to allow short bursts of packets to enter the network without smoothing them out
- For this purpose we use a token bucket, which is a modified leaky bucket
Token Bucket

- The bucket holds tokens instead of packets
- Tokens are generated and placed into the token bucket at a constant rate
- When a packet arrives at the token bucket, it is transmitted if there is a token available. Otherwise it is buffered until a token becomes available.
- The token bucket has a fixed size, so when it becomes full, subsequently generated tokens are discarded
Token Bucket

Packets from host

Token Generator
(Generates a token once every T seconds)

Network
Token Bucket vs. Leaky Bucket

Case 1: Short burst arrivals

Arrival time at bucket

Departure time from a leaky bucket
Leaky bucket rate = 1 packet / 2 time units
Leaky bucket size = 4 packets

Departure time from a token bucket
Token bucket rate = 1 tokens / 2 time units
Token bucket size = 2 tokens
Token Bucket vs. Leaky Bucket

Case 2: Large burst arrivals

- **Arrival time at bucket**
  - Time points: 0, 1, 2, 3, 4, 5, 6
  - Arrivals at: 1, 2, 3, 4

- **Departure time from a leaky bucket**
  - Leaky bucket rate: 1 packet / 2 time units
  - Leaky bucket size: 2 packets
  - Departures at: 1, 3

- **Departure time from a token bucket**
  - Token bucket rate: 1 token / 2 time units
  - Token bucket size: 2 tokens
  - Departures at: 1, 4
IETF Integrated Services

- architecture for providing QoS guarantees in IP networks for individual application sessions
- resource reservation: routers maintain state info (a la VC) of allocated resources, QoS req’s
- admit/deny new call setup requests:

**Question:** can newly arriving flow be admitted with performance guarantees while not violated QoS guarantees made to already admitted flows?
Intserv: QoS guarantee scenario

- Resource reservation
  - call setup, signaling (RSVP)
  - traffic, QoS declaration
  - per-element admission control

- QoS-sensitive scheduling (e.g., WFQ)
Arriving session must:

- declare its QOS requirement
  - **R-spec**: defines the QOS being requested
- characterize traffic it will send into network
  - **T-spec**: defines traffic characteristics
- signaling protocol: needed to carry R-spec and T-spec to routers (where reservation is required)
  - **RSVP**
Intserv QoS: Service models

Guaranteed service:

- worst case traffic arrival: leaky-bucket-policed source

\[ D_{\text{max}} = \frac{b}{R} \]
IETF Differentiated Services

Problems with Intserv:

- **Scalability**: signaling, maintaining per-flow router state difficult with large number of flows
- **Flexible Service Models**: Intserv has only two classes. Also want “qualitative” service classes
  - “behaves like a wire”
  - relative service distinction: Platinum, Gold, Silver

Diffserv approach:

- simple functions in network core, relatively complex functions at edge routers (or hosts)
- Don’t define service classes, provide functional components to build service classes
Diffserv Architecture

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

**Core router:**
- per class traffic management
- buffering and scheduling based on marking at edge
- preference given to in-profile packets
- Assured Forwarding
Edge-router Packet Marking

- profile: pre-negotiated rate A, bucket size B
- packet marking at edge based on per-flow profile

possible usage of marking:
- class-based marking: packets of different classes marked differently
- intra-class marking: conforming portion of flow marked differently than non-conforming one
Classification and Conditioning

- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6
- 6 bits used for Differentiated Service Code Point (DSCP) and determine PHB that the packet will receive
- 2 bits are currently unused
Classification and Conditioning

may be desirable to limit traffic injection rate of some class:
- user declares traffic profile (eg, rate, burst size)
- traffic metered, shaped if non-conforming
Per hop behavior

- PHB result in a different observable (measurable) forwarding performance behavior
- PHB does not specify what mechanisms to use to ensure required PHB performance behavior
- Examples:
  - Class A gets $x\%$ of outgoing link bandwidth over time intervals of a specified length
  - Class A packets leave first before packets from class B
  - Class B Packets get dropped over Class A packets
Implementing PHB

- **Bandwidth**
  - Can use multiple queues
  - Schedule packets in one queue more often than packets in the other queue: Round robin with different rates

- **Delay**
  - Use priority queue; schedule lower level queue after higher level queue

- **Loss**
  - When need to drop packets prefer one class over another