Washington University in St. Louis | Engineering

21. Real-Time Multimedia Networking

- New issues for real-time applications
- Protocols for real-time apps (RTP, RTCP)
- Voice-over IP (VoIP)
- Quality-of-service in data networks

Jon Turner – based on slides from Kurose & Ross

Real-Time Multimedia Networking

- Conversational apps (e.g. VoIP) require low delay
  - » best to keep *total delay* under 150 ms
  - » avoid coding methods with large coding delay
  - » use adaptive playout control to minimize playout delay
- No time for retransmissions, so prefer UDP over TCP
  - » usually with Real-time Transport Protocol (RTP) on top
- Coping with congestion
  - » can adapt sending rate by reducing audio/video quality
  - » conceal effects of packets that are lost or late
    - e.g. for video repeat data from a previous frame
- Quality-of-Service (QoS)
  - » improve performance of real-time apps by giving them special treatment – flow reservation or packet priority
  - » technical solutions exist, but hard to implement in Internet

2

Adaptive Playout Delay

- *Goal:* low playout delay, low late loss rate
  - » good for apps like conversational voice that send intermittently
- *Approach:* adaptive playout delay adjustment:
  - » estimate average network delay
    - adjust playout delay in response to changes in network delay
    - for conversational voice, set playout delay at start of "talk spurt"
  - » data played out at steady rate during active periods
- Adaptively estimate packet delay
  - » exponentially weighted moving average (EWMA)

$$d_i = (1-\alpha)d_{i-1} + \alpha(r_i - t_i)$$

- d_i: delay estimate after i-th packet
- α: small constant, e.g. 0.1
- r_i: time received
- t_i: time sent (timestamp)
- $r_i - t_i$: measured delay of i-th packet

Adaptive Playout Delay

- Also useful to estimate average deviation of delay, v_i:

$$v_i = (1-\beta)v_{i-1} + \beta|(r_i - t_i) - d_i|$$

- Estimates d_i, v_i calculated for every received packet, and *occasionally* used to adjust playout delay

$$\text{playout-time}_i = t_i + d_i + Kv_i \qquad (K \text{ typically 4 or 5})$$

  subsequent packets played out at at steady rate
- Naive implementation can produce poor results
  - » if delay measurements stop during "silent periods", measured delay may be highly inaccurate
    - keep sending measurement packets during silent periods
  - » may also need to adjust playout delay during long active periods

4

Adaptive Playout Delay

- *Goal:* low playout delay, low late loss rate
- *Approach:* adaptive playout delay adjustment:
  - » estimate average network delay, adjust playout delay at beginning of each talk spurt
  - » silent periods compressed and elongated
  - » chunks still played out every 20 msec during talk spurt
- Adaptively estimate packet delay
  - » exponentially weighted moving average (EWMA)

$$d_i = (1-\alpha)d_{i-1} + \alpha(r_i - t_i)$$

- d_i: delay estimate after i-th packet
- α: small constant, e.g. 0.1
- r_i: time received
- t_i: time sent (timestamp)
- $r_i - t_i$: measured delay of i-th packet

Adaptive Playout Delay

- Also useful to estimate average deviation of delay, v_i:

$$v_i = (1-\beta)v_{i-1} + \beta|(r_i - t_i) - d_i|$$

- Estimates d_i, v_i calculated for every received packet, but used only at start of talk spurt
- For first packet in talk spurt, playout time is:

$$\text{playout-time}_i = t_i + d_i + Kv_i$$

  remaining packets in talk-spurt are played out periodically

  K typically 4 or 5
- Naive implementation can produce poor results
  - » if delay measurements stop during "silent periods", measured delay may be highly inaccurate
    - keep sending measurement packets during idle periods
  - » may also need to track delay changes *during* talkspurt

Packet Loss Concealment

- Methods for video loss concealment
  - » replace missing "video line" with interpolated data
  - » use data from previous frame when packets are lost
- Methods for audio loss concealment
  - » send redundant bits (Forward Error Correction)
    - for every group of *n* chunks, create redundant chunk by exclusive OR-ing *n* original chunks
    - send *n*+1 chunks, increasing bandwidth by factor 1/*n*
    - reconstruct original *n* chunks if at most one lost – but adds delay
  - » "lossy" recovery exploits redundancy in encoded voice
    - separate "even samples" from "odd samples" and send in separate packets – if packet lost, interpolate missing samples
  - » multi-rate encoding
    - send lower quality version of voice signal along with original
    - if high-quality chunk is lost, substitute low quality version

Exercises

1. Consider an internet telephony application. The sender transmits a new voice packet every 20 ms which is timestamped with the local time at the sender when the first voice sample in the packet was produced. Assume that the one-way propagation delay in the network is 50 ms and that the queueing delay varies in a sawtooth pattern with a minimum of 0, a maximum of 100 ms and a period of 500 milliseconds. More precisely, starting at time 0, it increases linearly to 100 ms at time 500 milliseconds, then drops back to zero and repeats. Plot the number of bytes in the playout buffer, if the receiver used a fixed delay of 200 ms to control the playout. Assume that the sender's and receivers' clocks are exactly in sync.
2. Repeat the last problem using an adaptive playout method. To simplify the calculations, let the delay estimate be simply the average of the last 10 packet delays. Ignore the variance term.

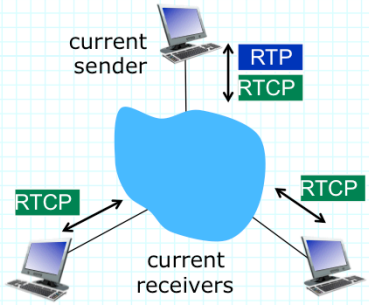
RTP (RFC 3550)

- Provides common information useful for real-time apps
  - » does not specify how they are to be used by apps
- Primary packet fields
  - » payload type (7 bits)
    - type of data and encoding; so, encoding may change within call
    - sample audio types: 64 kb/s PCM, 13 kb/s GSM, 2.4 Kb/s LPC
    - sample video types: motion JPEG, H.261, MPEG2
  - » sequence number (16 bits)
    - incremented by one for each RTP packet sent
    - used to detect packet loss, restore packet sequence
  - » timestamp field (32 bits)
    - sampling instant of first byte in this RTP data packet
    - for audio, increments by one for each sampling period
  - » SSRC field (32 bits)
    - identifies source of RTP stream within multi-party session

Real-Time Control Protocol (RTCP)

- Used in conjunction with RTP
  - » each participant in RTP session periodically sends RTCP control packets to all other participants
- Sender reports
  - » sent by participants that are currently sending audio/video
  - » SSRC, time (both real-time and sampling clock time), number of packets/bytes sent
- Receiver reports
  - » sent by all participants that are receiving audio/video (may also be senders)
  - » fraction of packets lost, most recent seq #, inter-arrival jitter
- Source description packets
  - » sender's name, email address, SSRC

RTP/RTCP Session Example (audio)



- Typical session uses single multicast address
  - » RTP, RTCP packets distinguished from each other via distinct port numbers
- Usually, one active audio sender at a time
- All participants send RTCP reports
  - » to limit control traffic, each participant reduces RTCP sending rate as number of participants increases

Some RTCP Details

- Bandwidth scaling
  - » each RTP session has an assigned "session bandwidth"
  - » RTCP limits control traffic to 5% of session bandwidth
  - » also, divides its 5% between senders (25%) and receivers
  - » participants adjust their RTCP sending rate to stay within limits
    - based on their knowledge of session bandwidth and number of senders/receivers
- Steam synchronization
  - » RTP timestamps are based on sender sample clock
    - but need real-time for playout buffer synchronization
    - also, useful for synchronizing audio and video stream (lip-sync)
  - » RTCP sender reports relate real-time to sample clock
    - include real-time and latest timestamp of media stream
    - receivers can compute real sending time of each media packet

Exercises

1. Consider a voice/video teleconferencing session with 10 participants, operating over a local-area network that uses RTP and RTCP. Audio is encoded at 64 Kb/s and video is sent using motion JPEG at a data rate of 4 Mb/s. Audio packets are sent whenever someone is speaking and each endpoint monitors the received audio packets to determine who the current speaker is. The current speaker's video is transmitted over the network and as different people talk, the active video source switches. In addition, the "previous speaker's" video is sent over the network, allowing the current speaker to see the video from the previous speaker (this allows more natural interaction among participants). What is an appropriate session bandwidth for this application, assuming we have at most two active video senders at a time and at most five active audio senders? Ignore the protocol headers when doing your calculations. Assume 20 ms of audio data per audio packet and 1200 bytes of video data per video packet. Also include an appropriate allocation for RTCP. What is the bandwidth available to RTCP? How many RTCP packets can be sent per second, assuming 100 bytes per report? If there are two senders in the session, how many sender reports can each each sender transmit in a second? How many receiver reports can each participant transmit in a second?

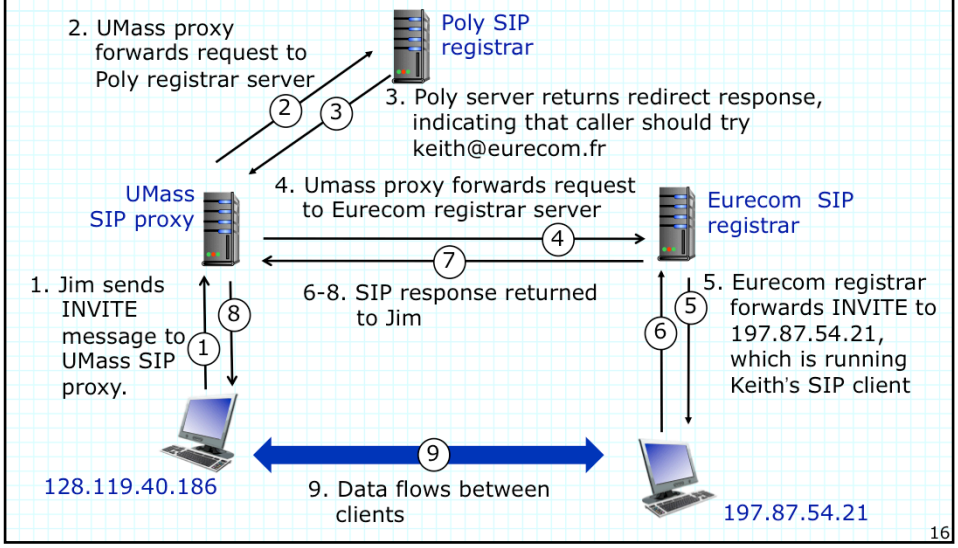
Voice-Over-IP (VoIP)

- Widely used in enterprise phone systems
  - » using VoIP phones and/or PC-software
- Used by some voice service providers
  - » Skype uses peer-to-peer approach with proprietary protocols
  - » Vonage (among others) uses standard internet protocols
- Session Initiation Protocol (SIP – RFC 3261)
  - » typically used in conjunction with RTP/RTCP
  - » defines mechanisms to initiate voice/video teleconferencing sessions in the internet
    - learning current IP address of other SIP users
    - initiating and configuring audio/video session, once IP address of called party is known
    - adding others to on-going call
  - » strictly an application-level protocol

SIP Concepts/Components

- Uniform Resource Identifiers (URI)
  - » used to identify SIP user; email address may be used as URI
- Address-of-record
  - » URI used to identify a user and enable SIP components to learn about a user's current status
- User agents
  - » used to initiate calls, alert user to an incoming call
  - » register user's current location/status
- Registrar
  - » used by UAs to register user's current location/status
  - » info provided to registrar forms basis of user location service
- Proxy
  - » forwards call requests to called party, using information provided by location service

Example: jim@umass.edu calls keith@poly.edu



The Transition to IP Voice Services

- IP has become attractive option for voice services
  - » packet switching equipment has become far less expensive than telephone switching equipment of similar capacity
  - » unregulated nature of Internet leads to lower costs in US
  - » data traffic volume now substantially larger than voice
- Complicated transition
  - » new systems must interoperate with old; no clean slate
    - expensive to convert access part of telephone network
  - » must preserve traditional calling features that customers rely on
  - » still important to ensure good voice quality
    - may require separate switches/links for voice, per-call route selection and bandwidth reservation
    - alternatively, separate high priority queues for voice and over-provisioned links

17

Exercises

1. Refer to RFC 3261. What is a provisional response? Give an example illustrating the use of a provisional response.
2. In SIP, if an invite message reaches a remote user's machine, but the remote user is already engaged in another call, how does the remote machine return the equivalent of a "busy signal" to the original caller?

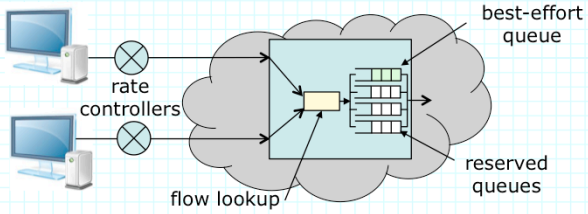
The Great QoS Debate

- Telecom viewpoint
  - » multimedia applications require consistent network performance to provide high quality user experience
  - » to guarantee consistent performance, must reserve network capacity for user sessions
    - new sessions blocked when network resources all used up
- Internet viewpoint
  - » must operate over simple datagram subnets
    - limit application-specific requirements to hosts
    - over-provision subnets to avoid need for reservations
  - » providing Quality-of-Service widely perceived as too complex
    - various approaches tried, no clear success as yet
- Resource reservation based on "paying" for resources
  - » works for telephone networks with usage-based charging
  - » harder in internet, where usage-based charging not the norm

Quality of Service in Internet

- IETF defined IntServ architecture in 1990s
  - » included reservation protocol (RSVP) and collection of related router mechanisms (flow tables, per-flow queues, etc)
  - » subject to much controversy and debate
    - widely viewed as excessively complex and "not scalable"
  - » very little deployment in the public internet
    - even though many router products now support essential elements
- DiffServ architecture developed as a compromise
  - » dropped notion of per-flow reservations and queueing
  - » packets carry a Type of Service (ToS) field with multimedia traffic being assigned to a "low delay" service class
    - issue: if ToS field set by hosts, low delay class may be over-used
    - but no general way for network providers to set it correctly
  - » without reservations, no guarantees are possible
  - » limited deployment to date – mostly for private networks

Implementing Reserved Flows



- Signalling protocol used to request network bandwidth
  - » if resources are available, reservation is accepted
- Rates monitored at access points
  - » rate controllers discard or mark excess packets
- Routers do flow lookup to identify reserved flows
  - » (src IP, dst IP, protocol, src port, dst port)➔queue
- Each reserved flow gets its own queue
  - » weighted-fair queueing allows each to get its assigned share
  - » excess traffic (marked packets) discarded during congestion

21

Making Most of Best Effort

- Reasons to think that best-effort may be good enough
  - » backbone link rates ≥10 Gb/s have become common-place, while residential access links relatively slow (1-10 Mb/s)
    - so, aggregate traffic on backbone links fairly smooth and predictable, leading to minimal queueing, most of the time
  - » over-provisioning becoming a reasonable alternative
    - high bandwidth links have become relatively inexpensive
    - applications have modest bandwidth needs
- Traffic bottlenecks do occur on access links
  - » residential network with 2 Mb/s download speed
    - easy to exhaust with just a single file transfer, or a few high quality video streams
  - » per-flow queueing on access link could protect moderate rate real-time flows from "greedy" apps
  - » with faster access links real-time flows unlikely to fill link