Recap

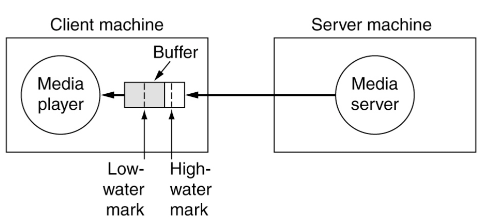
- UDP: IP with port abstraction
- TCP: Reliable, in order, at most once semantics
  - Sliding Windows
  - Flow control: ensure client is not overwhelmed
    - Advertised window from receiver end
  - Congestion control: ensure network is not overwhelmed
    - Congestion window from sender end
    - TCP friendly flows
  - TCP has no timing requirements

Quality of Service

- Outline
  - Realtime Applications
    - Networking with specified delay components
  - Integrated Services
    - Per flow QoS
  - Differentiated Services
    - QoS for aggregated traffic

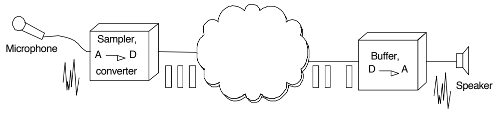
Streaming Audio

The media player buffers input from the media server and plays from the buffer rather than directly from the network.



Realtime Applications

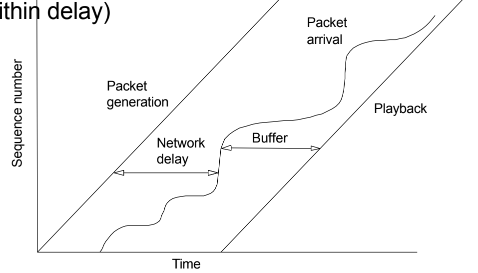
- Require “deliver on time” assurances
  - must come from inside the network



- Example application (audio)
  - sample voice once every 125μs
  - each sample has a playback time
  - packets experience variable delay in network
  - add constant factor to playback time: playback point
    - Similar to skip protection in portable CD players

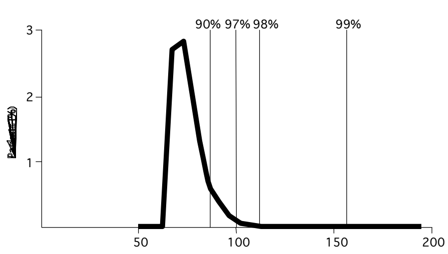
Playback Buffer

- Playback point as insurance against Internet delays
- Multimedia care about delay and jitter (variability within delay)

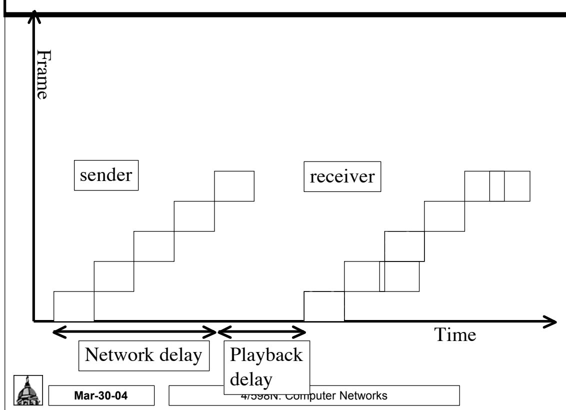


Example Distribution of Delays

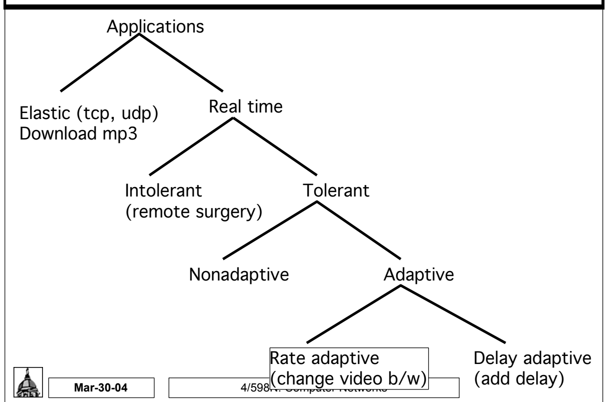
- What is a good delay? 200 msec
- Not acceptable for chat application



Video transmission



Taxonomy of real time applications



QoS Approaches

- Fine grained - individual application or flows
  - Intserv
  - E.g. for my video chat application
- Coarse grained - aggregated traffic
  - Diffserv
  - E.g. All traffic from CSE (costs $$)

Integrated Services

- IETF - 1995-97 time frame
- Service Classes
  - guaranteed
  - controlled-load (tolerant, adaptive applications)
    - Simulates lightly loaded link
- Mechanisms
  - signaling protocol: signals required service
  - admission control: rejects traffic that cannot be serviced
  - Policing: make sure that senders stick to agreement
  - packet scheduling: manage how packets are queued

Flowspec

- Rspec: describes service requested from network
  - controlled-load: none
  - guaranteed: delay target
- Tspec: describes flow's traffic characteristics
  - average bandwidth + burstiness: token bucket filter
    - token rate r and bucket depth B
  - must have a token to send a byte
  - must have n tokens to send n bytes
  - start with no tokens
  - accumulate tokens at rate of r per second
  - can accumulate no more than B tokens

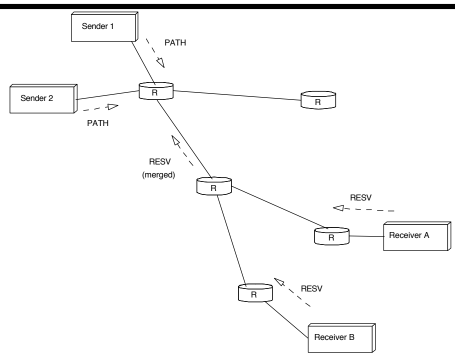
Per-Router Mechanisms

- Admission Control
  - decide if a new flow can be supported
  - answer depends on service class
  - not the same as policing
- Packet Processing
  - classification: associate each packet with the appropriate reservation
  - scheduling: manage queues so each packet receives the requested service

Reservation Protocol

- Called signaling in ATM
- Proposed Internet standard: RSVP
- Consistent with robustness of today's connectionless model
- Uses soft state (refresh periodically)
- Designed to support multicast
- Receiver-oriented
- Two messages: PATH and RESV
- Source transmits PATH messages every 30 seconds
- Destination responds with RESV message
- Merge requirements in case of multicast
- Can specify number of speakers

RSVP Example (multicast)



RSVP versus ATM (Q.2931)

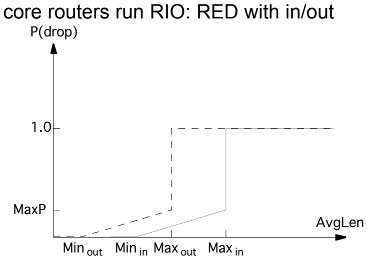
- RSVP
  - receiver generates reservation
  - soft state (refresh/timeout)
  - separate from route establishment
  - QoS can change dynamically
  - receiver heterogeneity
- ATM
  - sender generates connection request
  - hard state (explicit delete)
  - concurrent with route establishment
  - QoS is static for life of connection
  - uniform QoS to all receivers

Differentiated Services

- Problem with IntServ: scalability
- Idea: segregate packets into a small number of classes
  - e.g., premium vs best-effort
- Packets marked according to class at edge of network
- Core routers implement some per-hop-behavior (PHB)
- Example: Expedited Forwarding (EF)
  - rate-limit EF packets at the edges
  - PHB implemented with class-based priority queues or Weighted Fair Queue (WFQ)

DiffServ (cont)

- Assured Forwarding (AF)
  - customers sign service agreements with ISPs
  - edge routers mark packets as being "in" or "out" of profile
  - core routers run RIO: RED with in/out

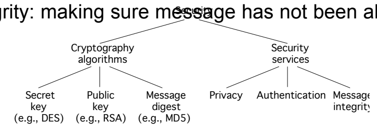


Chapter 8: Security

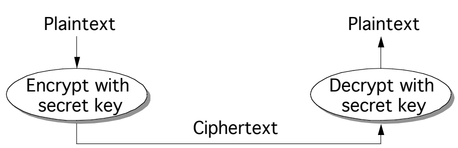
- Outline
  - Encryption Algorithms
  - Authentication Protocols
  - Message Integrity Protocols
  - Key Distribution
  - Firewalls

Overview

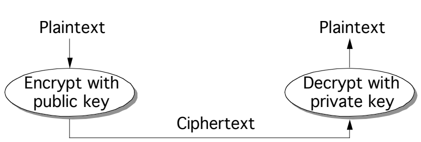
- Cryptography functions
  - Secret key (e.g., DES)
  - Public key (e.g., RSA)
  - Message digest (e.g., MD5)
- Security services
  - Privacy: preventing unauthorized release of information
  - Authentication: verifying identity of the remote participant
  - Integrity: making sure message has not been altered



Secret Key (DES)



Public Key (RSA)



- Encryption & Decryption

$c = m^e \bmod n$

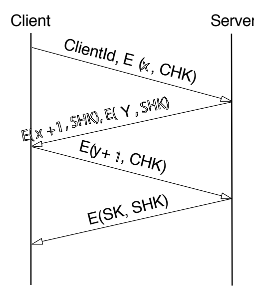
$m = c^d \bmod n$

Message Digest

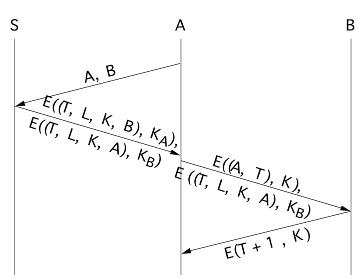
- Cryptographic checksum
  - just as a regular checksum protects the receiver from accidental changes to the message, a cryptographic checksum protects the receiver from malicious changes to the message.
- One-way function
  - given a cryptographic checksum for a message, it is virtually impossible to figure out what message produced that checksum; it is not computationally feasible to find two messages that hash to the same cryptographic checksum.
- Relevance
  - if you are given a checksum for a message and you are able to compute exactly the same checksum for that message, then it is highly likely this message produced the checksum you were given.

Authentication Protocols

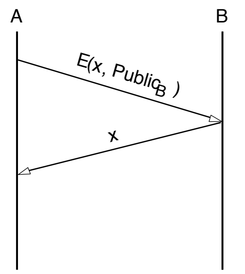
- Three-way handshake



- Trusted third party (Kerberos)



- Public key authentication



Message Integrity Protocols

- Digital signature using RSA
  - special case of a message integrity where the code can only have been generated by one participant
  - compute signature with private key and verify with public key
- Keyed MD5
  - sender: m + MD5(m + k) + E(k, private)
  - receiver
    - recovers random key using the sender's public key
    - applies MD5 to the concatenation of this random key message
- MD5 with RSA signature
  - sender: m + E(MD5(m), private)
  - receiver
    - decrypts signature with sender's public key
    - compares result with MD5 checksum sent with message

Message Integrity Protocols

- Digital signature using RSA
  - special case of a message integrity where the code can only have been generated by one participant
  - compute signature with private key and verify with public key
- Keyed MD5
  - sender: m + MD5(m + k) + E(E(k, rcv-pub), private)
  - receiver
    - recovers random key using the sender's public key
    - applies MD5 to the concatenation of this random key message
- MD5 with RSA signature
  - sender: m + E(MD5(m), private)
  - receiver
    - decrypts signature with sender's public key
    - compares result with MD5 checksum sent with message

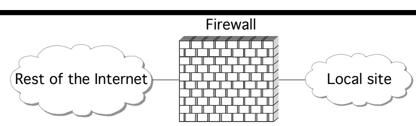
Key Distribution

- Certificate
  - special type of digitally signed document:
    - "I certify that the public key in this document belongs to the entity named in this document, signed X."
  - the name of the entity being certified
  - the public key of the entity
  - the name of the certified authority
  - a digital signature
- Certified Authority (CA)
  - administrative entity that issues certificates
  - useful only to someone that already holds the CA's public key.

Key Distribution (cont)

- Chain of Trust
  - if X certifies that a certain public key belongs to Y, and Y certifies that another public key belongs to Z, then there exists a chain of certificates from X to Z
  - someone that wants to verify Z's public key has to know X's public key and follow the chain
- Certificate Revocation List

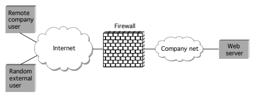
Firewalls



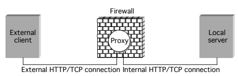
- Filter-Based Solution
  - example
    (192.12.13.14, 1234, 128.7.6.5, 80)
    (*,*, 128.7.6.5, 80)
  - default: forward or not forward?
  - how dynamic?
  - stateful

Proxy-Based Firewalls

- Problem: complex policy
- Example: web server



- Solution: proxy



- Design: transparent vs. classical
- Limitations: attacks from within

Denial of Service

- Attacks on end hosts
  - SYN attack
- Attacks on routers
  - Christmas tree packets
  - pollute route cache
- Authentication attacks
- Distributed DoS attacks