Today’s Lecture

- Project 3
  - In-Class Exercise
- Network Issues
  - Quality of Service

Quality of Service

Outline

Realtime Applications
- Require "deliver on time" assurances
  - must come from inside the network
- Example application (audio)
  - sample voice once every 125us
  - each sample has a playback time
  - packets experience variable delay in network
  - add constant factor to playback time: playback point

Example Distribution of Delays

Integrated Services

- Service Classes
  - guaranteed
  - controlled-load
- Mechanisms
  - resource reservation (signalling)
  - admission control
  - policing
  - packet scheduling
Flowspec

• \textit{Rspec}: describes service requested from network
  – controlled-load: none
  – guaranteed: delay target
• \textit{Tspec}: describes flow’s traffic characteristics
  – average bandwidth + burstiness: \textit{token bucket filter}
  – token rate $r$
  – 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 signalling 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

• Associate packet with reservation (classifying):
  – source address, destination address, protocol number, source port, destination port
• Manage packets in queues (scheduling).

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 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

Coding

• Form into small groups
  – Twist: Must sit by people who you did not work with on the project

Write Threaded Code

• UDP Server
  – Step 1
    • Listen on port X (pass in via parameter)
    • Read message / display on screen
  – Step 2
    • Keep track of incoming IP / ports
    • Design
      – Local vs. global
      – struct vs. class

Write Main Function

• main function
  – Step 1
    • Start up the thread
    • Loop until input is QUIT!
  – Step 2
    • Send the typed text to all known other clients
  – Step 3
    • Allow adding of IP via syntax
      – ADDCLIENT 129.74.20.40 8908
  – Step 4
    • Dump current client list via DUMPSTATUS

Build a simple UDP chat server
Moment of Zen

[Comic Panels]

Everyday, back in the Old
Shit, I'm running 20 minutes,
That's going to keep me

Answer: back in the Old
Shit, I'm running 20 minutes,
That's going to keep me

Sorry, Mr. Yang,
I'm going to deep one

Mom, I'm listening
The game with Lincoln.
I'm going to deep one

Mom, I'm listening
The game with Lincoln.
I'm going to deep one