VoIP Protocols and QoS

Internet Protocols

CSC / ECE 573

Fall, 2005

N. C. State University

Announcements

I. Times have been posted for demo slots

II. HW5 and HW6 solutions have been posted

- *HW6 being graded*

III. Final Exam questions?

Today's Lecture

I. VoIP Protocol Overview

II. Motivation and Overview of QoS

III. Mechanisms to Implement QoS

IV. Some QoS protocols: RSVP and DiffServ

V. QoS Assessment

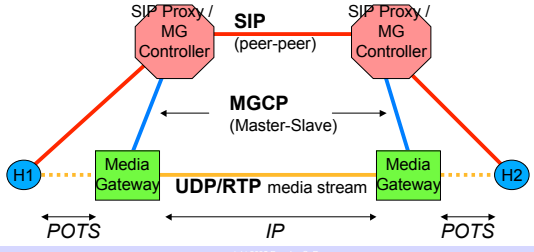
VOICE-OVER-IP PROTOCOLS

Motivation for VoIP?

- Price?
- New services (integration of computer + phone)?
- Freedom from phone companies?

Voice-over-IP Protocols

- Master-slave (M-S) relationship is easiest to manage, easiest to recover from faults
- Peer-to-peer (P2P) is most flexible, most scalable



Session Initiation Protocol (RFC 3261)

- Used to initiate, control, and terminate telephone calls and other services
  - end-to-end call signaling, possibly involving media gateway controllers (MGC) between the end points
- Properties
  - fully distributed (peer to peer)
  - text-based (human readable)
  - modeled on HTTP
  - "SIP is a much simpler protocol than H.323, but is at least as functional" – SIP wins ☺

SIP Key Features

- Functionality
  - SIP is complete for setting up point-to-point or multiparty multimedia calls
  - extensive call handling capabilities
  - media capabilities are negotiated by endpoints
- Flexibility
  - URLs are used for addresses (i.e., to locate the callee)
  - users can move to new locations and access their full telephony features from anywhere
  - users can define what response they want to give when contacted (availability, etc.)

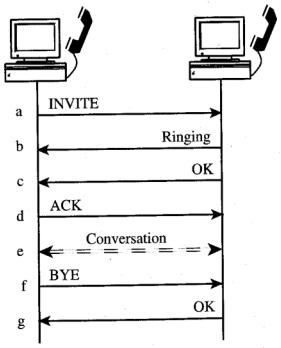
SIP Proxies (i.e., Call Servers)

- Users may require the use of *SIP proxies*, or call servers, to set up and maintain the call
  - find the other party
  - configure the media gateways
  - do user authentication, authorization, and billing
- SIP servers are stateless
  - means any information about the current state of the call is stored in the endpoints (gateways, terminals), rather than in the server
  - reason: to simplify the design of the servers, improve their scalability

Example: "Basic" SIP Call (No Proxy)

Figure 5-5
Example of an SIP call

Source: Carrier Grade Voice Over IP, D. Collins, McGraw-Hill, 2001



Call Routing Example (with *Relaying*)

1. Jon initiates a call to eve@isi.edu, sends INVITE message
2. DNS looks up a SRV record for the SIP server (proxy) at isi.edu; proxy is sip.isi.edu
3. The caller invitation is sent to sip.isi.edu
4. sip.isi.edu indicates Eve is served by SIP server sipgw.cs.isi.edu
5. sip.isi.edu relays the request to sipgw.cs.isi.edu
6. sipgw.cs.isi.edu has database indicating how to reach Eve
   - DB is updated whenever Eve registers a new location
7. sipgw.cs.isi.edu relays the invitation to Eve at her current location

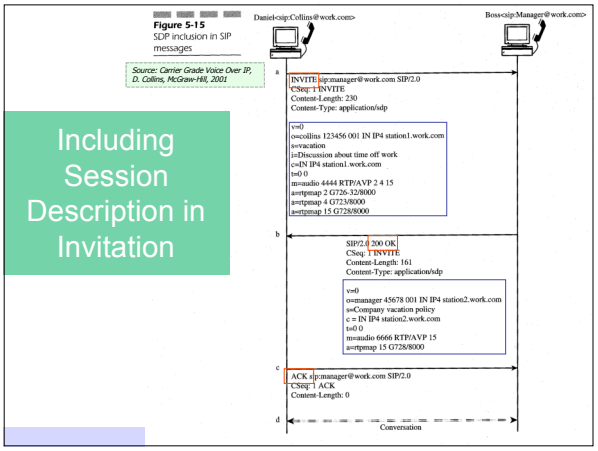
REGISTER Request Method

- Purpose: register the current location of a user
- Who register with?
  - multicast to the well-known "all SIP servers" multicast address "sip.mcast.net" (224.0.1.75)
- What happens if user is registered as being at multiple locations? Several choices:
  1. contact them all at once, wait for the first to respond
  2. contact them one at a time until get successful response

Media Negotiation

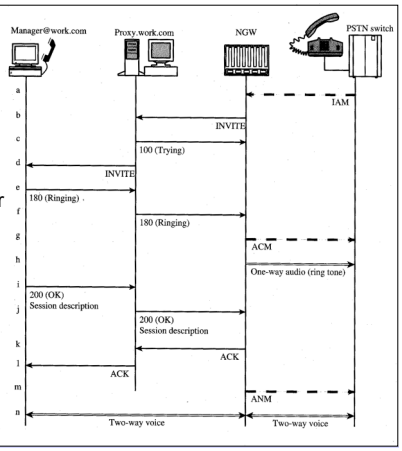
- The invitation (from caller to callee) suggests the type of media sessions to establish
  - using SDP description
- The response (from callee to caller) indicates type of media sessions that are possible / acceptable
- Caller "re-invites" with the commonly-agreed set of media functions

Including Session Description in Invitation



PSTN to SIP Call

- Signaling used for "conventional" (PSTN) phone calls: SS7/ISUP
- Requires translation between SIP and ISUP



REMARKS ON QOS

What QoS Means

- Predictable / Controllable network behavior and performance
  - timescale?
  - granularity?
- Performance metrics
  1. packet drop or "loss" rate
  2. end-to-end latency (delivery time)
  3. throughput (bandwidth)

Applications Requiring QoS

- Voice
- Video
- Other
  - multi-user gaming
  - real-time simulation
  - online auctions
  - …

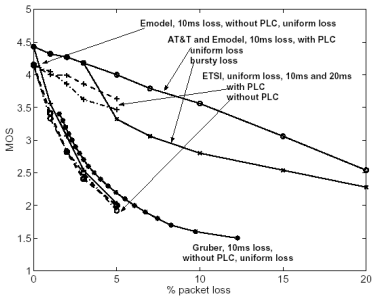
Applications That Don't Need QoS

- *Elastic* applications tolerate "best-effort" Internet service
  - web, ftp, mail
  - "low" loss rates, variable throughput, and small average delays are good enough
- Non-interactive (streaming) audio and video are semi-elastic

How Good Does QoS Need to Be?

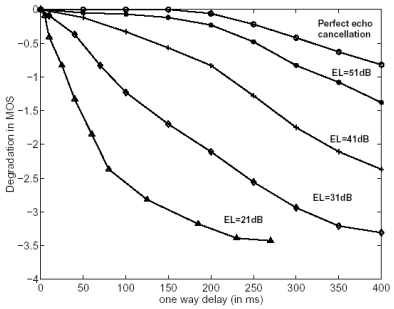
	Voice	Video
Bandwidth (kb/s)	5—64	64—5000
Maximum tolerable packet drop rate	1-3%	1%
Maximum Latency (one-way)	< 150ms (if interactive)	same

Packet Loss Effect on QoS



source: http://www.sprintlabs.com/People/amarko/PAPERS/TON/paper-00149-2002.pdf

One-Way Latency Effect on QoS

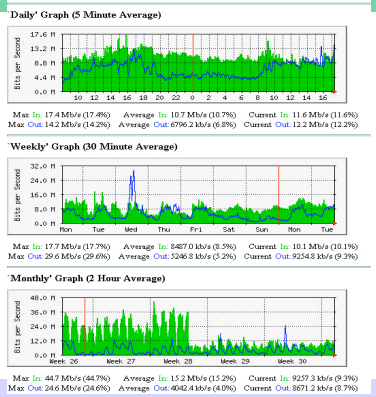


source: http://www.sprintlabs.com/People/amarko/PAPERS/TON/paper-00149-2002.pdf

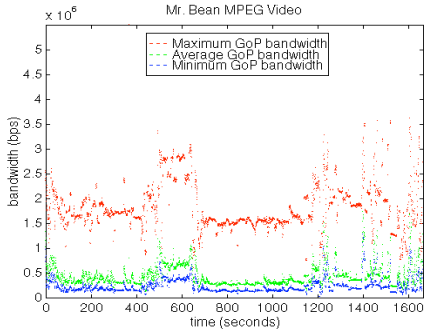
Challenges for Internet QoS

- Social / Political / Economic challenges
  - user resistance to complex pricing models
  - user notions of "fairness"
  - incompatibility with "legacy" protocols and incremental-deployment issues
  - increasing abundance of cheap bandwidth
  - desire to keep routers simple, cheap, and fast

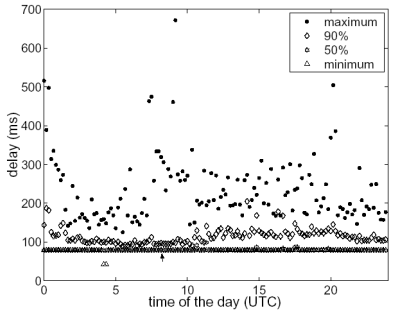
Network Loading Variations



Application Bandwidth Variations



Network Delay Variations



source: http://www.sprintlabs.com/People/amarko/PAPERS/TON/paper-00149-2002.pdf

Sources of Delay

- Application
  - framing delay (time to accumulate one unit of speech or video for processing): 10-30ms
  - video/audio compression and decompression: 30-100 ms
  - jitter removal/smoothing at receiver: 50-200 ms
- Network
  - Media access and serialization delay: 5ms
  - propagation delay:45 ms (cross-continent)
  - queuing and switching delays: 5-200ms

A "Framework" for QoS

- Sender contacts receiver, negotiates application characteristics (bit-rate, etc.)
- Sender notifies network of QoS requirements and application characteristics
- Network determines if QoS can be met, reserves bandwidth and scheduling priority on media path
- QoS is managed by network
- Receiver manages playback buffer

QoS "MECHANISMS"

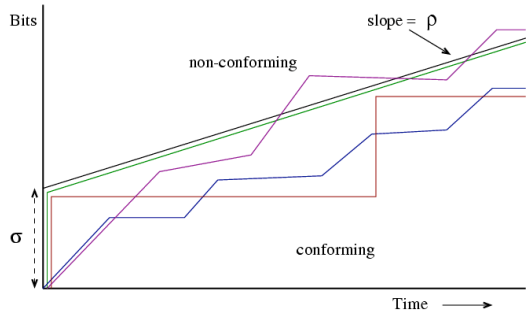
Compression / Decompression Choices

- Lots of standards: take CSC557 ☺
- Tradeoff of…
  - processing overhead and hardware needed
  - perceptual quality desired
- Factors that can be adapted
  - frame rate, size of image, color depth
  - lossiness / quality

Application Traffic Descriptors

- A popular descriptor: *Linear Bounded Arrival Process* (*LBAP*)
- # of bits transmitted in any interval t = $B(t) \leq \rho^* t + \sigma$
  - ρ = the long-term average rate
  - σ = longest "burst" that application may send
- Example (ρ = 5000 bps, σ = 3000 bits, t = .5)
  - $B(t) \leq$ 5500 bits

LBAP Illustration

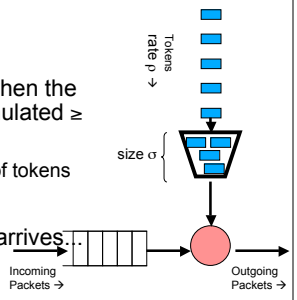


Traffic Shapers or Policers

- *Shapers* or *regulators* enforce the rate at which applications can inject traffic into the network
- *Policers* discard traffic which does not conform to a traffic descriptor
- Mechanisms
  - *leaky bucket*
  - *token bucket*

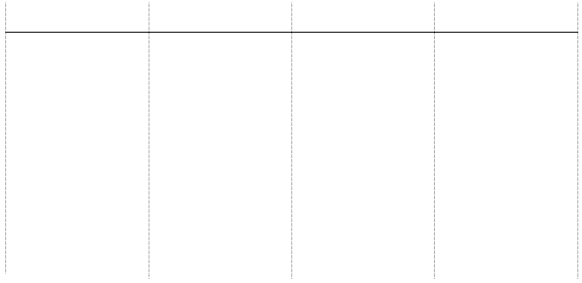
Token Bucket Regulator

- Up to σ tokens accumulate in the "bucket", at rate ρ
  - token bucket size σ
  - token generation rate ρ
- Transmit a packet only when the sum of the tokens accumulated ≥ packet size
  - and remove that amount of tokens from the bucket
- If no token when packet arrives…
  - *policer*: drop packet
  - *shaper*: buffer packet



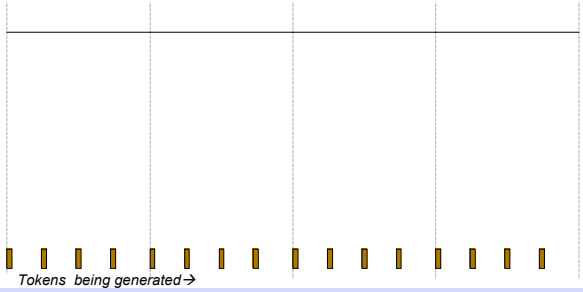
Token Bucket Shaping Example

- Bucket size σ = 4 tokens
- Token generation rate ρ = 4 tokens/Sec



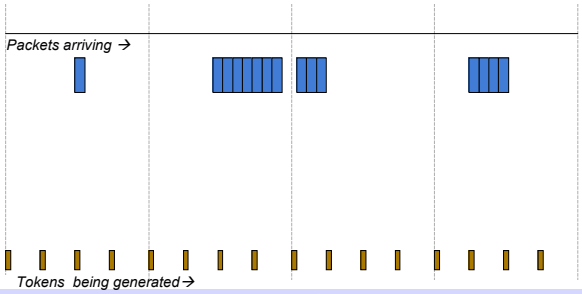
Token Bucket Shaping Example

- Bucket size σ = 4 tokens
- Token generation rate ρ = 4 tokens/Sec



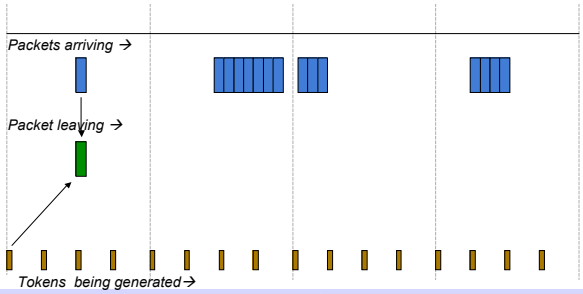
Token Bucket Shaping Example

- Bucket size σ = 4 tokens
- Token generation rate ρ = 4 tokens/Sec



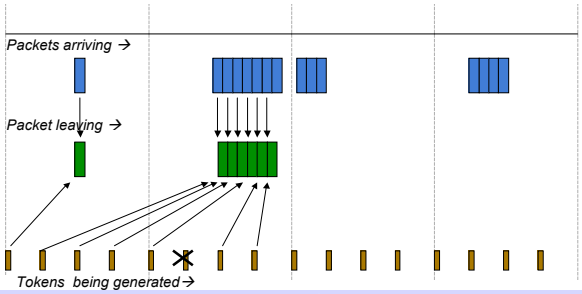
Token Bucket Shaping Example

- Bucket size σ = 4 tokens
- Token generation rate ρ = 4 tokens/Sec



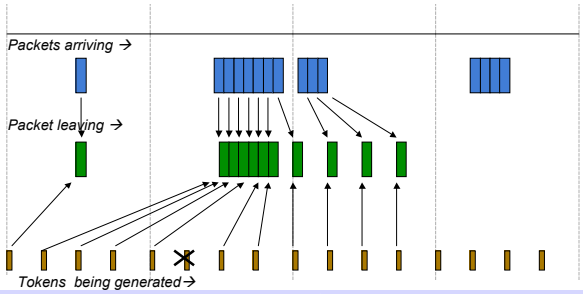
Token Bucket Shaping Example

- Bucket size σ = 4 tokens
- Token generation rate ρ = 4 tokens/Sec



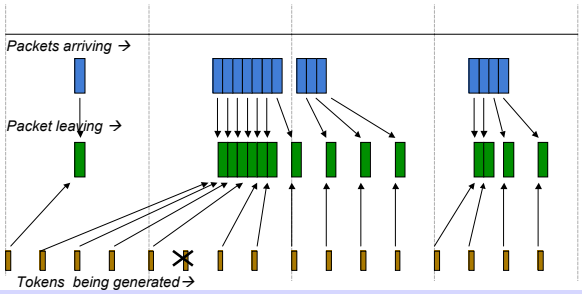
Token Bucket Shaping Example

- Bucket size σ = 4 tokens
- Token generation rate ρ = 4 tokens/Sec



Token Bucket Shaping Example

- Bucket size σ = 4 tokens
- Token generation rate ρ = 4 tokens/Sec



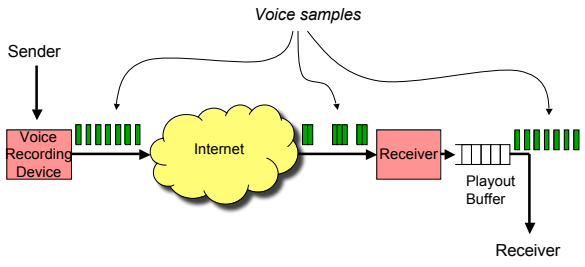
Packet Schedulers

- Determine the order in which incoming packets are output, and at what times
  - affects the queuing time of the packets
- The simplest scheduler: First-in-First-Out
  - *limitation*: all flows treated the same
  - *variation*: remember RED and AQM (active queue management)?
- Many scheduling algorithms (take CSC557 ☺)
  - one example: Weighted Round-Robin (WRR)

Playout Buffer Management

- Voice and video are captured at constant sampling intervals. Examples…
  - voice: 8000 samples / sec = 1 sample every .125 ms
  - video: 30 frames / sec = 1 sample every 333 ms
- Network "jitter" means the packets will not arrive at the receiver at a steady rate
  - receiver uses a *playout buffer* to restore smooth playback

Example



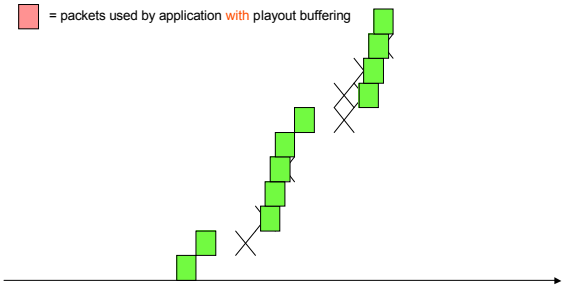
Playout Buffer Illustration

= packet arrivals
= packets used by application without buffering
= packets used by application with playout buffering

Time →

Playout Buffer Illustration

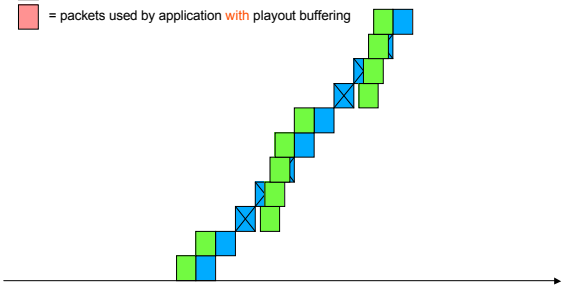
= packet arrivals
= packets used by application without buffering
= packets used by application with playout buffering



Time →

Playout Buffer Illustration

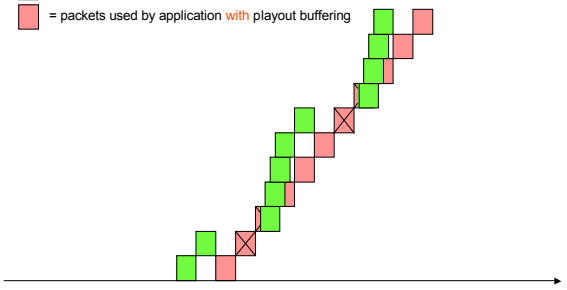
= packet arrivals
= packets used by application without buffering
= packets used by application with playout buffering



Time →

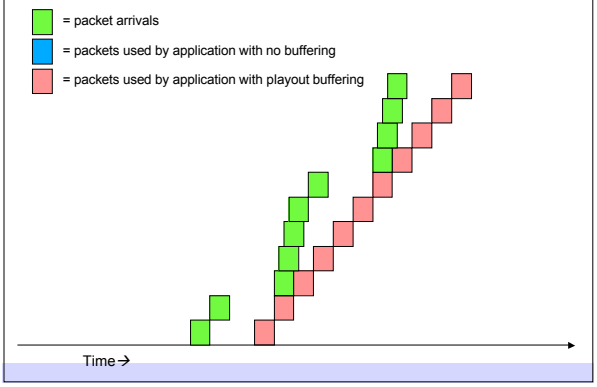
Playout Buffer Illustration

= packet arrivals
= packets used by application without playout buffering
= packets used by application with playout buffering

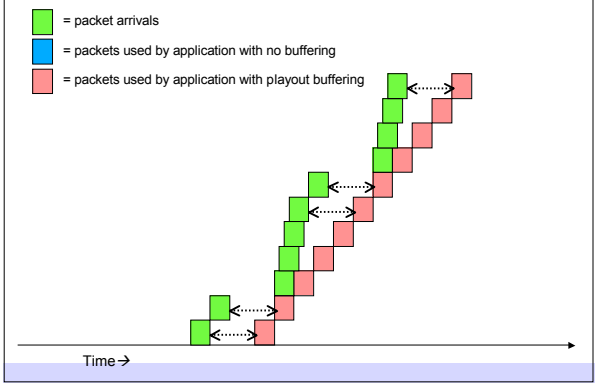


Time →

Playout Buffer Illustration



Playout Buffer Illustration



RSVP AND INTEGRATED SERVICES

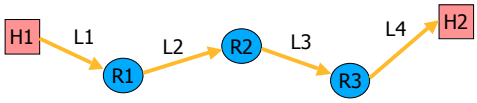
QoS Guarantees (RFC1633)

- Deterministic (100%) guarantees
  - based on peak traffic rate
  - simple, predictable, conservative
  - *Guaranteed Service* (RFC 2212)
- Statistical (< 100%) guarantees
  - *Controlled Load Service*
- No guarantees
  - *Best Effort Service*

The RSVP Protocol (RFC2205)

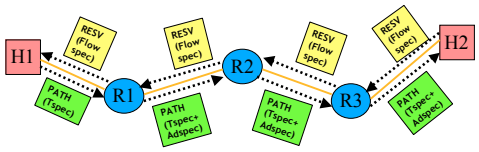
- *Purpose*: announce or *signal*...
  - the sending application requirements to receivers: PATH messages
  - the receivers' resource requirements to the network: RESV messages
- Properties
  - unidirectional (bidirectional QoS requires reservations in both directions)
  - runs directly over IP (unreliable)
  - hop-by-hop (not end-to-end): routers have to process the messages and possibly modify their contents

Point-to-Point Reservations



- *Goal*: establish a virtual circuit from H1 to H2
  - i.e., reserve "resources" in routers R1, R2, and R3
- Most controversial aspect: routers have to maintain per-flow *state*
  - e.g., flow identity, traffic specification, resource reservation, timer, next and previous hops, …

Route Pinning and Message Propagation



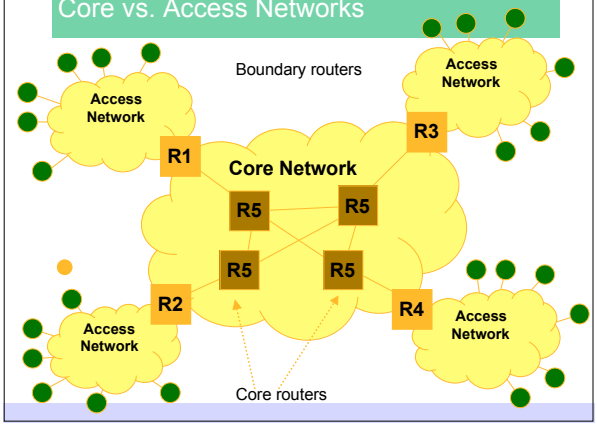
- Route *pinning*
  - make sure the RESV message retraces the same path taken by the PATH message

Guaranteed Service Delay Bounds

- *Goal*: receiver must calculate *R*, rate to be assigned to flow
- Application requirements determine maximum total end-to-end delay bound D_{req}
- Computed from information in **`Tspec`** and **`Adspec`**
  - (see RFC 2212)

DIFFERENTIATED SERVICES (DIFFSERV)

Core vs. Access Networks

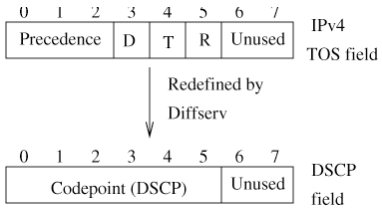


DiffServ

- QoS will be managed on aggregation (bundle) of flows sharing a path
  - trades off efficiency for accuracy, guaranteed QoS
  - simplifies processing in core routers, pushes complexity to the network edge
  - implemented only in core network, not in access networks
  - proper network provisioning for DiffServ is key to acceptable performance

Diffserv Codepoint (DSCP)

- Field in the IP header specifying the class of service the packet is to receive
  - replaces the previous (8-bit) TOS field



Expedited Forwarding (EF) PHB (RFC 2598)

- Qualitative guarantee: “low” loss, delay, and jitter
  - minimal queuing at routers
- Policing and shaping needed only at...
  - access routers, on a per-flow basis
  - border routers, on aggregated flows
- Policing based on simple token bucket
  - packets exceeding the subscribed rate must be dropped

Assured Forwarding (AF) PHB (RFC 2597)

- Traffic will be forwarded with "high probability" as long as within subscribed rate
  - excess traffic more likely to be discarded
- Network must not reorder packets in the same “flow” whether they are in or out of profile
- Four AF classes, with varying priorities for dropping

QoS ASSESSMENT

What Has Been Adopted So Far?

- Playout buffer management
- Streaming
- Overprovisioning
- RTP / RTCP
- SIP
- VoIP

What Has Not Been Adopted So Far?

1. IntServ (but RSVP in use) and Admission control
2. ~DiffServ~
3. WFQ, other complex schedulers
4. Killer issue: incremental deployment
   - is QoS in 95% of the network good enough?

Summary

- Voice and video transmission impose new demands on the Internet
- Providing QoS requires new packet handling mechanisms
- QoS also requires new protocols

Next Lecture

- Nothing!