

15-744: Computer Networking

L-7 QoS



QoS



- IntServ
- DiffServ
- Assigned reading
 - [She95] Fundamental Design Issues for the Future Internet
 - [CSZ92] Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanisms
- Optional
 - [CF98] Explicit Allocation of Best-Effort Packet Delivery Service

2

Overview



- **Why QOS?**
- Integrated services
- RSVP
- Differentiated services

3

Motivation



- Internet currently provides one single class of “**best-effort**” service
 - No assurances about delivery
- Existing applications are **elastic**
 - Tolerate delays and losses
 - Can adapt to congestion
- Future “real-time” applications may be **inelastic**

4

Inelastic Applications



- Continuous media applications
 - **Lower and upper limit** on acceptable performance.
 - BW below which video and audio are not intelligible
 - Internet telephones, teleconferencing with high delay (200 - 300ms) impair human interaction
- Hard real-time applications
 - Require **hard limits on performance**
 - E.g. control applications

5

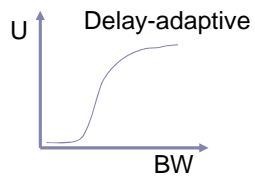
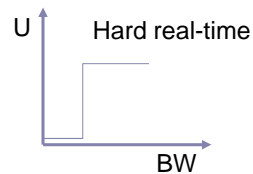
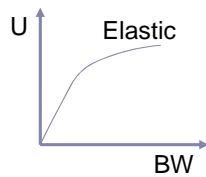
Why a New Service Model?



- What is the **basic objective** of network design?
 - Maximize total bandwidth? Minimize latency?
 - **Maximize user satisfaction** – the total **utility** given to users
- What does utility vs. bandwidth look like?
 - Must be non-decreasing function
 - Shape depends on application

6

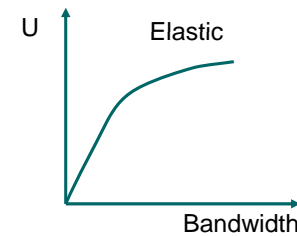
Utility Curve Shapes



Stay to the right and you are fine for all curves

7

Utility curve – Elastic traffic



Does equal allocation of bandwidth maximize total utility?

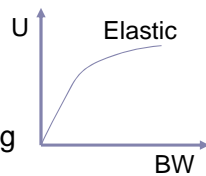
8

Admission Control



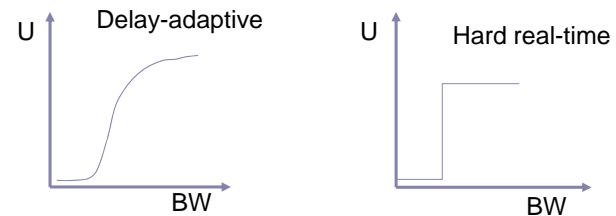
- If $U(\text{bandwidth})$ is concave
→ elastic applications

- Incremental utility is decreasing with increasing bandwidth
 - Is always advantageous to have more flows with lower bandwidth
 - No need of admission control;
- This is why the Internet works!



9

Utility Curves – Inelastic traffic



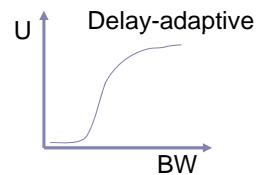
Does equal allocation of bandwidth maximize total utility?

10

Admission Control



- If U is convex → inelastic applications
 - $U(\text{number of flows})$ is no longer monotonically increasing
 - Need admission control to maximize total utility
- **Admission control** → deciding when the addition of new people would result in reduction of utility
 - Basically avoids overload



11

Overview



- Why QOS?
- **Integrated services**
- RSVP
- Differentiated services

12

Components of Integrated Services



1. Type of commitment

What does the network promise?

2. Packet scheduling

How does the network meet promises?

3. Service interface

How does the application describe what it wants?

4. Establishing the guarantee

How is the promise communicated to/from the network

How is admission of new applications controlled?

13

1. Type of commitment



What kind of promises/services should network offer?



Depends on the **characteristics of the applications** that will use the network

14

Playback Applications



• Sample signal → packetize → transmit → buffer → playback

- Fits most multimedia applications

• Performance concern:

- Jitter – variation in end-to-end delay
 - Delay = fixed + variable = (propagation + packetization) + queuing

• Solution:

- Playback point – delay introduced by buffer to hide network jitter

15

Characteristics of Playback Applications



- In general lower delay is preferable.
- Doesn't matter when packet arrives as long as it is before playback point
- Network guarantees (e.g. bound on jitter) would make it easier to set playback point
- Applications can tolerate some loss

16

Applications Variations



- Rigid & adaptive applications
 - Rigid – set fixed playback point
 - Adaptive – adapt playback point
 - Gamble that network conditions will be the same as in the past
 - Are prepared to deal with errors in their estimate
 - Will have an earlier playback point than rigid applications
- Tolerant & intolerant applications
 - Tolerance to brief interruptions in service
- 4 combinations

17

Applications Variations



Really only two classes of applications

- 1) Intolerant and rigid
- 2) Tolerant and adaptive

Other combinations make little sense

- 3) Intolerant and adaptive
 - Cannot adapt without interruption
- 4) Tolerant and rigid
 - Missed opportunity to improve delay

So what service classes should the network offer?

18

Type of Commitments



- **Guaranteed service**
 - For **intolerant and rigid** applications
 - Fixed guarantee, network meets commitment as long as clients send at match traffic agreement
- **Predicted service**
 - For **tolerant and adaptive** applications
 - Two components
 - If conditions do not change, commit to current service
 - If conditions change, take steps to deliver consistent performance (help apps minimize playback delay)
 - Implicit assumption – network does not change much over time
- **Datagram/best effort service**

19

Components of Integrated Services



1. Type of commitment
What does the network promise?
2. **Packet scheduling**
How does the network meet promises?
3. **Service interface**
How does the application describe what it wants?
4. Establishing the guarantee
How is the promise communicated to/from the network
How is admission of new applications controlled?

20

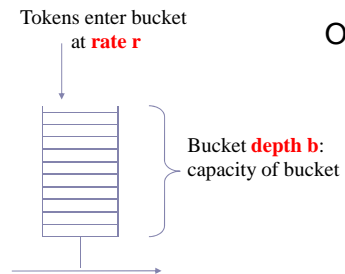
Scheduling for Guaranteed Traffic



- Use **token bucket filter** to characterize traffic
 - Described by rate r and bucket depth b
- Use **WFQ** at the routers
- Parekh's bound for worst case queuing delay = b/r

21

Token Bucket Filter

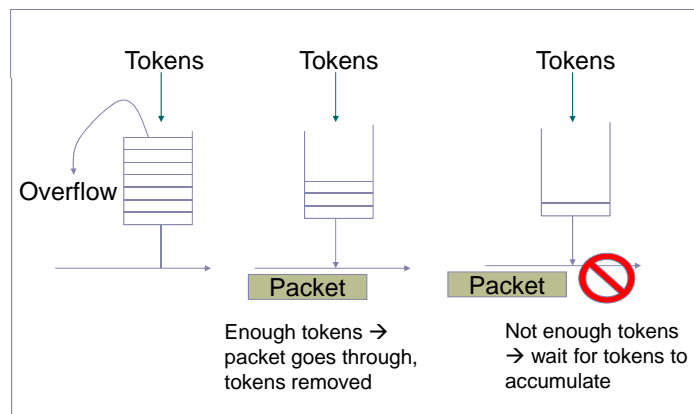


Operation:

- If bucket fills, tokens are discarded
- Sending a packet of size P uses P tokens
- If bucket has P tokens, packet sent at max rate, else must wait for tokens to accumulate

22

Token Bucket Operation



23

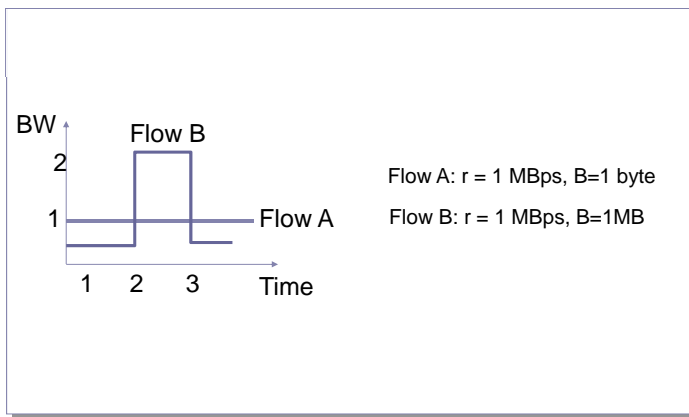
Token Bucket Characteristics



- On the long run, rate is limited to r
- On the short run, a burst of size b can be sent
- Amount of traffic entering at interval T is bounded by:
 - Traffic = $b + r \cdot T$
- Information useful to admission algorithm

24

Token Bucket Specs



25

Predicted Service



Goals:

- Isolation
 - Isolates well-behaved from misbehaving sources
- Sharing
 - Mixing of different sources in a way beneficial to all

Mechanisms:

- WFQ
 - Great isolation but no sharing
- FIFO
 - Great sharing but no isolation

26

Predicted Service



- FIFO jitter increases with the number of hops
 - Use opportunity for sharing across hops
- FIFO+
 - At each hop: measure average delay for class at that router
 - For each packet: compute difference of average delay and delay of that packet in queue
 - Add/subtract difference in packet header
 - Packet inserted into queues expected arrival time instead of actual
 - More complex queue management!
- Slightly decreases mean delay and significantly decreases jitter

27

Unified Scheduling



- Assume 3 types of traffic: guaranteed, predictive, best-effort
- Scheduling: use WFQ in routers
- Each guaranteed flow gets its own queue
- All predicted service flows and best effort aggregates in single separate queue
 - Predictive traffic classes
 - Multiple FIFO+ queues
 - Worst case delay for classes separated by order of magnitude
 - When high priority needs extra bandwidth – steals it from lower class
 - Best effort traffic acts as lowest priority class

28

Service Interfaces



- **Guaranteed Traffic**
 - Host specifies rate to network
 - Why not bucket size b ?
 - If delay not good, ask for higher rate
- **Predicted Traffic**
 - Specifies (r, b) token bucket parameters
 - Specifies delay D and loss rate L
 - Network assigns priority class
 - Policing at edges to drop or tag packets
 - Needed to provide isolation – why is this not done for guaranteed traffic?
 - WFQ provides this for guaranteed traffic

29

Overview



- Why QoS?
- Integrated services
- **RSVP**
- Differentiated services

30

Components of Integrated Services



1. Type of commitment
What does the network promise?
2. Packet scheduling
How does the network meet promises?
3. Service interface
How does the application describe what it wants?
4. **Establishing the guarantee**
 - How is the promise communicated**
 - How is admission of new applications controlled?**

31

Role of RSVP



- Rides on top of unicast/multicast routing protocols
- Carries resource requests all the way through the network
- At each hop consults admission control and sets up reservation. Informs requester if failure

32

RSVP Goals



- Used on connectionless networks
 - Should not replicate routing functionality
 - Should co-exist with route changes
- Support for multicast
 - Different receivers have different capabilities and want different QOS
 - Changes in group membership should not be expensive
 - Reservations should be aggregate – I.e. each receiver in group should not have to reserve
 - Should be able to switch allocated resource to different senders
- Modular design – should be generic “signaling” protocol
- Result
 - Receiver-oriented
 - Soft-state

33

RSVP Service Model



- Make reservations for simplex data streams
- Receiver decides whether to make reservation
- Control msgs in IP datagrams (proto #46)
- PATH/RESV sent periodically to refresh soft state
- One pass:
 - Failed requests return error messages - receiver must try again
 - No e2e ack for success

34

PATH Messages



- PATH messages carry sender's Tspec
 - Token bucket parameters
- Routers note the direction PATH messages arrived and set up *reverse path* to sender
- Receivers send RESV messages that follow reverse path and setup reservations
- If reservation cannot be made, user gets an error

35

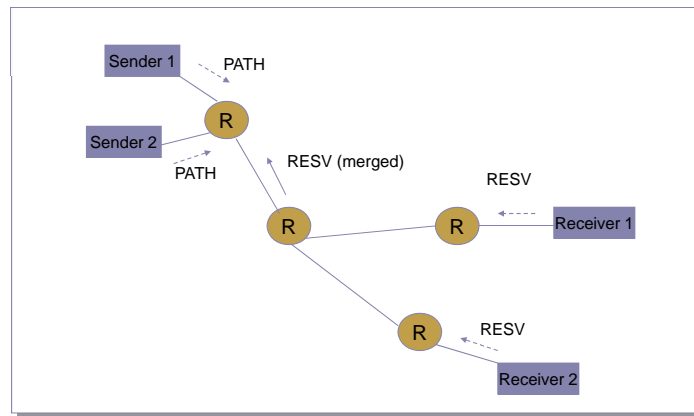
RESV Messages



- Forwarded via reverse path of PATH
- Queuing delay and bandwidth requirements
- Source traffic characteristics (from PATH)
- Filter specification
 - Which transmissions can use the reserved resources
- Router performs admission control and reserves resources
 - If request rejected, send error message

36

PATH and RESV Messages



37

Routing Changes

- Routing protocol makes routing changes
- In absence of route or membership changes, periodic PATH and RESV msgs refresh established reservation state
- When change, new PATH msgs follow new path, new RESV msgs set reservation
- Non-refreshed state times out automatically

38

Overview

- Why QOS?
- Integrated services
- RSVP
- Differentiated services

39

DiffServ

- Analogy:
 - Airline service, first class, coach, various restrictions on coach as a function of payment
- Best-effort expected to make up bulk of traffic, but revenue from first class important to economic base (will pay for more plentiful bandwidth overall)
- Not motivated by real-time! Motivated by economics and assurances

40

Basic Architecture



- Agreements/service provided within a domain
 - Service Level Agreement (SLA) with ISP
- Edge routers do traffic conditioning
 - Perform per aggregate shaping and policing
 - Mark packets with a small number of bits; each bit encoding represents a class or subclass
- Core routers
 - Process packets based on packet marking and defined per hop behavior
- More scalable than IntServ
 - No per flow state or signaling

41

Per-hop Behaviors (PHBs)



- Define behavior of individual routers rather than end-to-end services – there may be many more services than behaviors
- Multiple behaviors – need more than one bit in the header
- Six bits from IP TOS field are taken for Diffserv code points (DSCP)

42

Per-hop Behaviors (PHBs)



- Two PHBs defined so far
- Expedited forwarding aka premium service (type P)
 - Possible service: providing a virtual wire
 - Admitted based on peak rate
 - Unused premium goes to best effort
- Assured forwarding (type A)
 - Possible service: strong assurance for traffic within profile & allow source to exceed profile
 - Based on expected capacity usage profiles
 - Traffic unlikely to be dropped if user maintains profile
 - Out-of-profile traffic marked

43

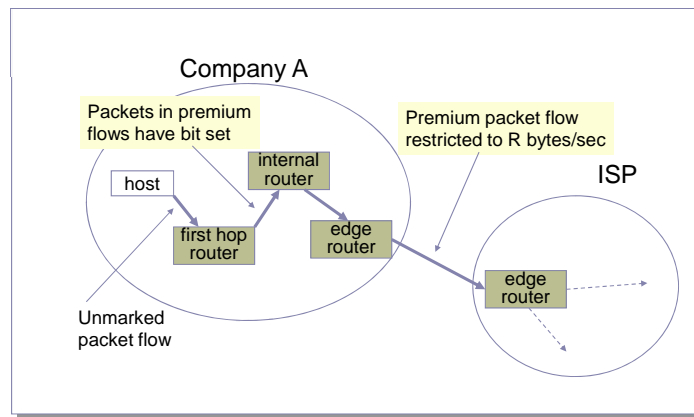
Expedited Forwarding PHB



- User sends within profile & network commits to delivery with requested profile
 - Signaling, admission control may get more elaborate in future
- Rate limiting of EF packets at edges only, using token bucket to shape transmission
- Simple forwarding: classify packet in one of two queues, use priority
 - EF packets are forwarded with minimal delay and loss (up to the capacity of the router)

44

Expedited Forwarding Traffic Flow



45

Assured Forwarding PHB

- User and network agree to some traffic profile
 - Edges mark packets up to allowed rate as "in-profile" or low drop precedence
 - Other packets are marked with one of 2 higher drop precedence values
- A congested DS node tries to protect packets with a lower drop precedence value from being lost by preferably discarding packets with a higher drop precedence value
 - Implemented using RED with In/Out bit

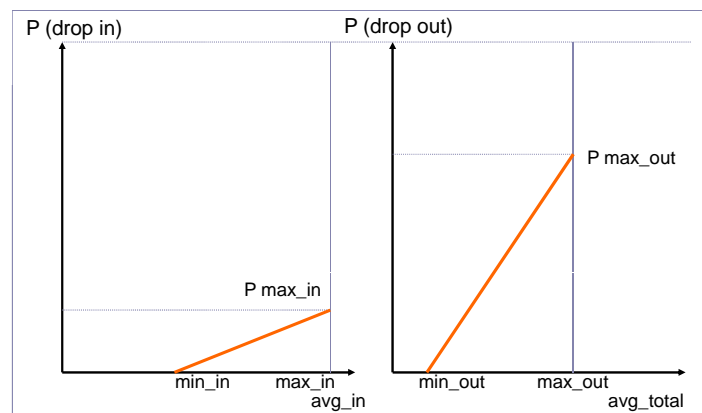
46

Red with In or Out (RIO)

- Similar to RED, but with two separate probability curves
- Has two classes, "In" and "Out" (of profile)
- "Out" class has lower $\text{Min}_{\text{thresh}}$, so packets are dropped from this class first
 - Based on queue length of all packets
- As avg queue length increases, "in" packets are also dropped
 - Based on queue length of only "in" packets

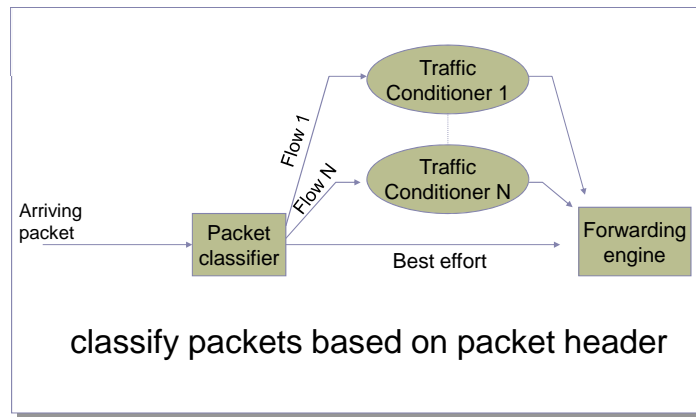
47

RIO Drop Probabilities



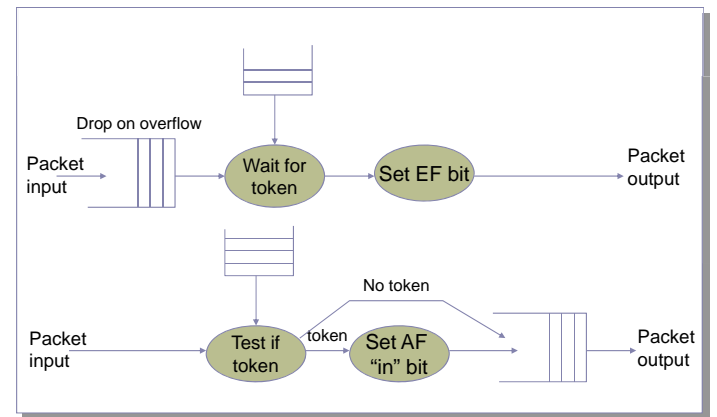
48

Edge Router Input Functionality



49

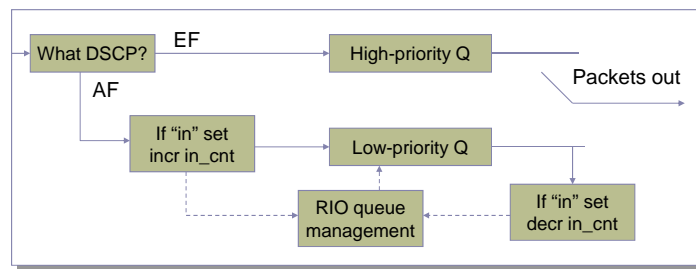
Traffic Conditioning



50

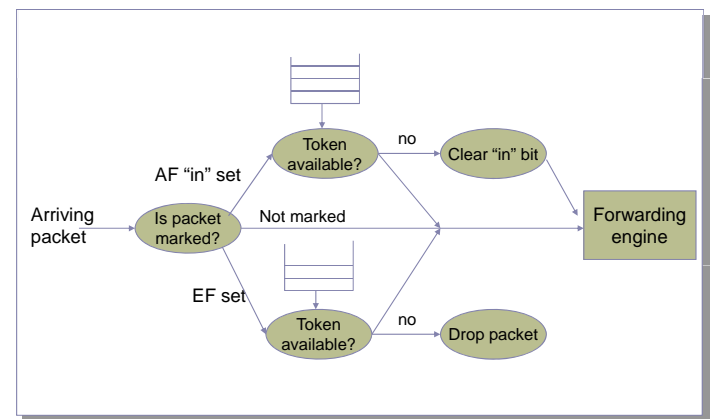
Router Output Processing

- 2 queues: EF packets on higher priority queue
- Lower priority queue implements RED "In or Out" scheme (RIO)



51

Edge Router Policing



52

Comparison



	Best-Effort	Diffserv	Intserv
Service	<ul style="list-style-type: none">• Connectivity• No isolation• No guarantees	<ul style="list-style-type: none">• Per aggregation isolation• Per aggregation guarantee	<ul style="list-style-type: none">• Per flow isolation• Per flow guarantee
Service Scope	<ul style="list-style-type: none">• End-to-end	<ul style="list-style-type: none">• Domain	<ul style="list-style-type: none">• End-to-end
Complexity	<ul style="list-style-type: none">• No set-up	<ul style="list-style-type: none">• Long term setup	<ul style="list-style-type: none">• Per flow setup
Scalability	<ul style="list-style-type: none">• Highly scalable• (nodes maintain only routing state)	<ul style="list-style-type: none">• Scalable (edge routers maintains per aggregate state; core routers per class state)	<ul style="list-style-type: none">• Not scalable (each router maintains per flow state)

53