CS640: Introduction to Computer Networks

Aditya Akella

Lecture 21 –
QoS

The Road Ahead

- Admission Control
- Integrated services
- RSVP
- Differentiated Services

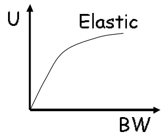
2

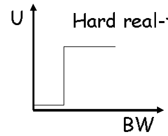
Why a New Service Model?

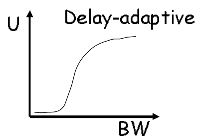
- Best-effort is clearly insufficient
- What is the **basic objective** of network design?
  - Maximize total bandwidth? Minimize latency?
  - **Maximize user satisfaction** – the total **utility** given to users
- What does utility vs. bandwidth look like?
  - Must be non-decreasing function
  - Shape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

3

Utility Curve Shapes



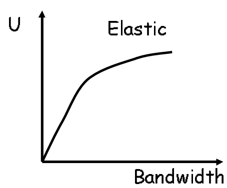




Stay to the right and you are fine for all curves

4

Utility curve – Elastic traffic



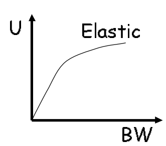
Does equal allocation of bandwidth maximize total utility?

5

Elastic Traffic

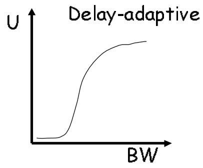
- If U(bandwidth) is concave
  → elastic applications
  - Incremental utility is decreasing with increasing bandwidth
  - Is always advantageous to have more flows with lower bandwidth
    - No need of admission control;

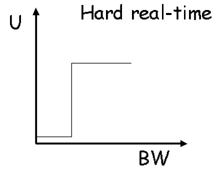
  This is why the Internet works!



6

Utility Curves – Inelastic traffic



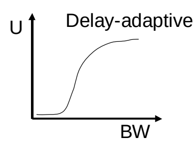


Does equal allocation of bandwidth maximize total utility?

7

Admission Control

- If U is convex → inelastic applications
  - U(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
- We will see how these issues play out in real QoS implementations



8

QoS Instantiation #1: Integrated Services

Key components:

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 (gory details)
   How is the promise communicated to/from the network
   How is admission of new applications controlled?

9

Type of Commitments

- **Guaranteed** service
  - For **hard real-time** applications
  - Fixed guarantee, network meets commitment as long as rates clients send at match traffic agreement
- **Predicted** service
  - For **delay-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**

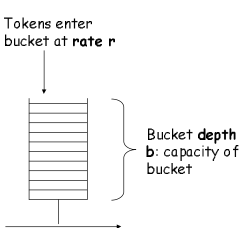
10

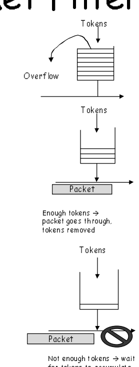
Scheduling for Guaranteed Traffic

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

11

Token Bucket Filter





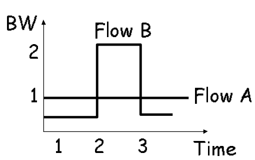
12

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*T
- Information useful to admission algorithm

13

Token Bucket Specs



Flow A: r = 1 MBps, B=1 byte

Flow B: r = 1 MBps, B=1MB

14

Guarantee Proven by Parekh

- Given:
  - Flow *i* shaped with token bucket and leaky bucket rate control (depth *b* and rate *r*)
  - Network nodes do WFQ
- Cumulative queuing delay D_i suffered by flow *i* has upper bound
  - **D_i < b/r**, (where r may be much larger than average rate)
  - Assumes that Σr < link speed at any router
  - All sources limiting themselves to *r* will result in no network queuing

15

Sharing versus Isolation

- Isolation
  - Isolates well-behaved from misbehaving sources
- Sharing
  - Mixing of different sources in a way beneficial to all
- FIFO: sharing
  - each traffic source impacts other connections directly
    - e.g. malicious user can grab extra bandwidth
  - the simplest and most common queueing discipline
  - averages out the delay across all flows
- Priority queues: one-way sharing
  - high-priority traffic sources have impact on lower priority traffic only
  - has to be combined with admission control and traffic enforcement to avoid starvation of low-priority traffic
- WFQ: two-way isolation
  - provides a guaranteed minimum throughput (and maximum delay)

16

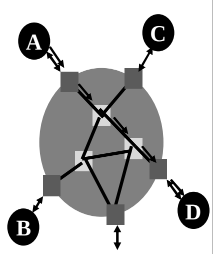
Putting It All Together

- 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
    - 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

17

Resource Reservation Protocol (RSVP)

- Carries resource requests all the way through the network
- Main goal: establish "state" in each of the routers so they "know" how they should treat flows.
  - State = packet classifier parameters, bandwidth reservation, ..
- At each hop consults admission control and sets up reservation. Informs requester if failure



18

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

19

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

20

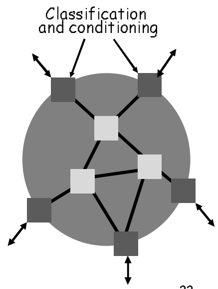
Soft State

- Periodic PATH and RESV msgs refresh established reservation state
  - Path messages may follow new routes
  - Old information times out
- Properties
  - Adapts to changes in routes and sources
  - Recovers from failures
  - Cleans up state after receivers drop out

21

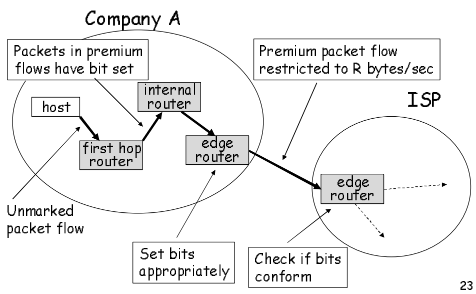
Differentiated Services: Motivation and Design

- Edge routers do fine grain enforcement
  - Typically slower links at edge
  - E.g. mail sorting in post offices
  - Label packets with a type field
    - Uses IP TOS bits
    - E.g. a priority stamp
- Core routers process packets based on packet marking and defined per hop behavior
- More scalable than IntServ
  - No signaling
  - No per-flow state in the core



22

DiffServ Example



23

Assured Forwarding PHB

- AF defines 4 classes
  - Strong assurance for traffic within profile & allow source to exceed profile
    - Implement services that differ relative to each other (e.g., gold service, silver service…)
  - Admission based on expected capacity usage profiles
  - Within each class, there are three drop priorities
    - Traffic unlikely to be dropped if user maintains profile
- User and network agree to some traffic profile
  - Edges mark packets up to allowed rate as “in-profile” or high priority
  - Other packets are marked with one of 2 lower “out-of-profile” priorities
  - A congested router drops lower priority packets first
    - Implemented using clever queue management (RED with In/Out bit)

24

Expedited Forwarding PHB

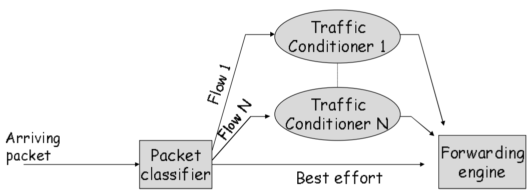
User sends within profile & network commits to delivery with requested profile
- Strong guarantee
- Possible service: providing a virtual wire
- Admitted based on peak rate

• 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)

25

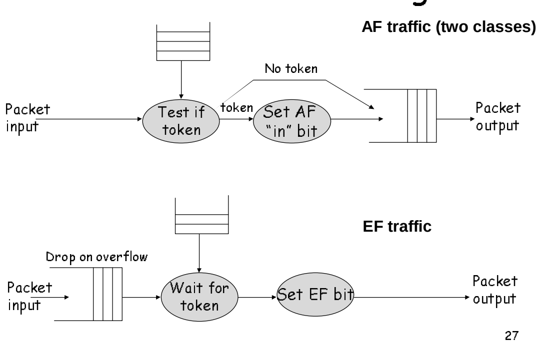
Edge Router Input Functionality



classify packets based on packet header

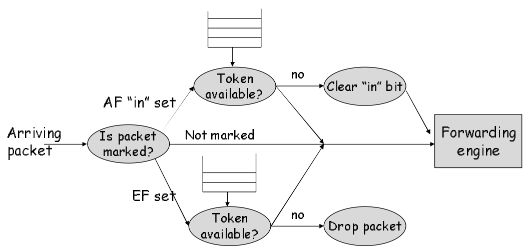
26

Traffic Conditioning



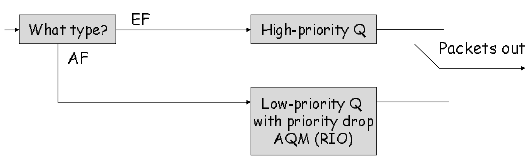
27

Edge Router Policing



28

Router Output Processing



29

Comparison

	Best-Effort	Diffserv	Intserv
Service	• Connectivity • No isolation • No guarantees	• Per aggregation isolation • Per aggregation guarantee	• Per flow isolation • Per flow guarantee
Service Scope	• End-to-end	• Domain	• End-to-end
Complexity	• No set-up	• Long term setup	• Per flow setup
Scalability	• Highly scalable • (nodes maintain only routing state)	• Scalable (edge routers maintains per aggregate state; core routers per class state)	• Not scalable (each router maintains per flow state)

30