CS640: Introduction to Computer Networks

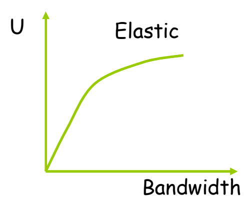
Aditya Akella

Lecture 20 – QoS

Why a New Service Model?

- Best effort clearly insufficient
  - Some applications need more assurances from the network
- 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

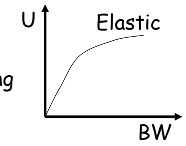
Utility curve – Elastic traffic



Does equal allocation of bandwidth maximize total utility?

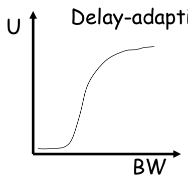
Admission Control

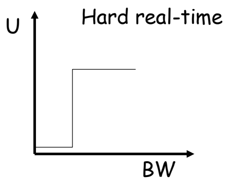
- If U(bandwidth) is concave → elastic applications
  - Incremental utility is decreasing with increasing bandwidth
  - Is always advantageous to have more flows with lower bandwidth
    - No need of admission control and explicit QoS mechanisms



4

Utility Curves – Inelastic traffic



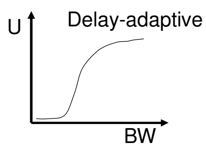


Does equal allocation of bandwidth maximize total utility?

5

QoS and Admission Control

- If U is convex → inelastic applications
  - U(number of flows) is no longer monotonically increasing
- Need admission control and special QoS mechanisms
  - Admission control → deciding when the addition of new people would result in reduction of utility



6

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?

7

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 tolerant (e.g. 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). Ensure that such apps continue to see a lightly loaded network.
- **Datagram/best effort service**

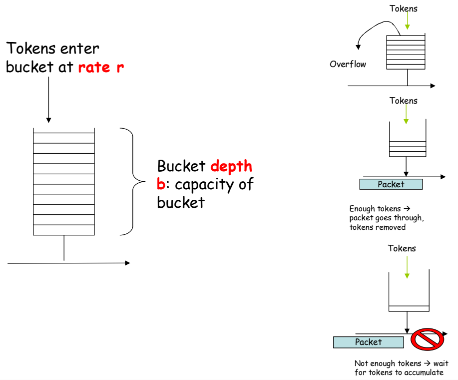
8

Scheduling for Guaranteed Traffic

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

9

Token Bucket Filter



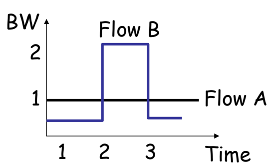
10

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

11

Token Bucket Specs



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

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

12

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**

13

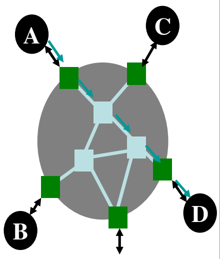
Putting It All Together

- 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 priority queue
  - Predictive traffic classes
    - Worst case delay for classes separated by order of magnitude
    - Strict priority queueing – coupled with admission control into each priority level
    - Higher priority steals scheduling cycles from lower priority - One way isolation
  - Best effort traffic acts as lowest priority class

14

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
- Key properties
  - Receiver driven
  - Soft state
    - Periodically refresh reservations



15

PATH Messages

- PATH messages carry sender's flow properties
- 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

16

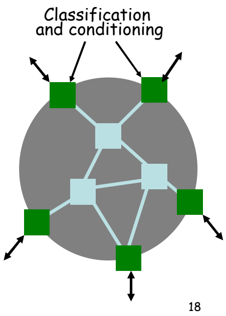
RESV Messages

- 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

17

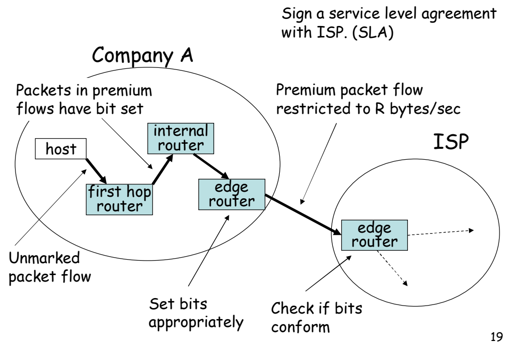
Differentiated Services: Motivation and Design

- Edge routers do coarse grain enforcement
  - Label packets with a type field
    - Uses IP TOS bits
    - E.g. a priority stamp
- Core routers process packets based on packet marking
- More scalable than IntServ
  - No signaling
  - No per-flow state in the core
  - More useful between a pair of neighboring networks, while IntServ was end-to-end
  - Typically used by multi-campus enterprises with all campuses connected to the same ISP



18

DiffServ Example



Expedited Forwarding

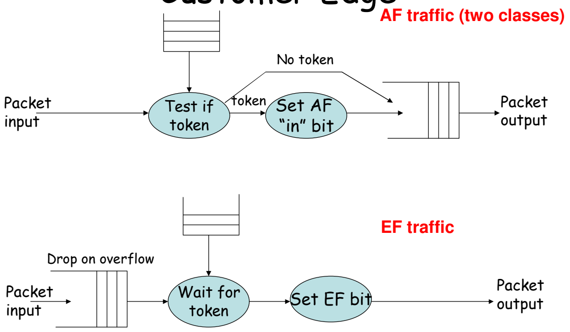
User sends within agreed profile & network commits to delivery with requested profile
- Strong guarantee
- User cannot exceed profile → packets will get dropped

- Core router → Simple forwarding: if packet marked as EF, put in priority queue
  - EF packets are forwarded with minimal delay and loss (up to the capacity of the router)

Assured Forwarding

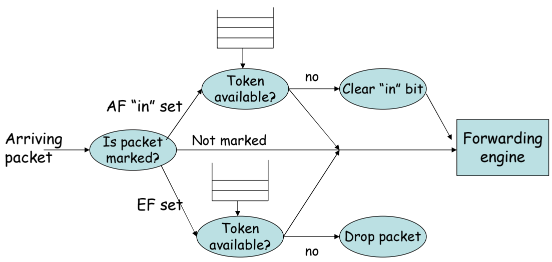
- AF defines 4 classes
  - Strong assurance for traffic within profile & allow source to exceed profile
    - Implement services that differ relative to each other (e.g., gold service, silver service…)
  - Within each class, there are at least two drop priorities
    - Traffic unlikely to be dropped 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 lower "out-of-profile" priority
  - A congested router drops lower priority packets with a lot higher probability
    - Implemented using RED based priority queuing

Traffic Conditioning: At Customer Edge



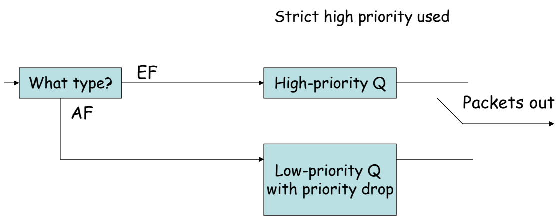
22

Edge Router Policing: At ISP Edge



23

Router Output Processing



24