Quality of Service

Quality of Service

- How “good” are late data and low-throughput channels?
- It depends on the application. Do you care if...
  - Your e-mail takes 1/2 hour to reach your friend?
  - You have to spend 1/2 hour to make a cheaper plane reservation on the Web?
  - Your call to 911 takes 1/2 hour to go through your nifty new IP phone service?

Application Requirements

- Internet currently provides one single class of “best-effort” service
  - No assurances about delivery
- High speed networks have enabled new applications
  - Require “deliver on time” assurances from the network
  - Real-time applications
    - Sensitive to the timeliness of their data
    - Voice
    - Video
    - Industrial control

Timely Delivery

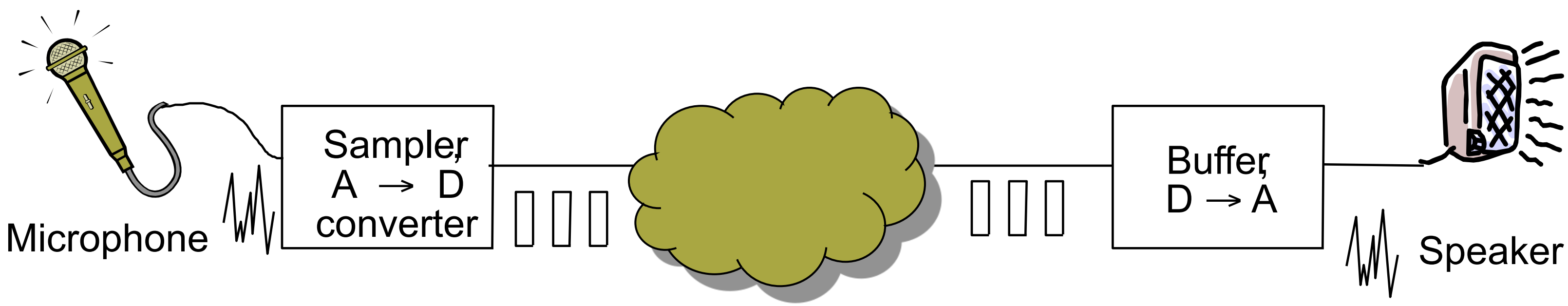
- How to achieve timely delivery
  - When actual RTT small (< 2/3) relative to acceptable delay
    - Retransmit
  - When base RTT (no queuing delay) large (> 2) relative to acceptable delay
    - Impossible
  - Otherwise possible, but not through retransmission

Timely Delivery

- Within the 48-state U.S
  - Base RTT (no queueing delay) peaks around 75 msec
  - Actual RTT is often 10-100 msec
- Humans notice about 50 msec delay for voice
  - Use erasure codes across packets, or
  - Support delay preferences in the network; called quality of service, or QoS

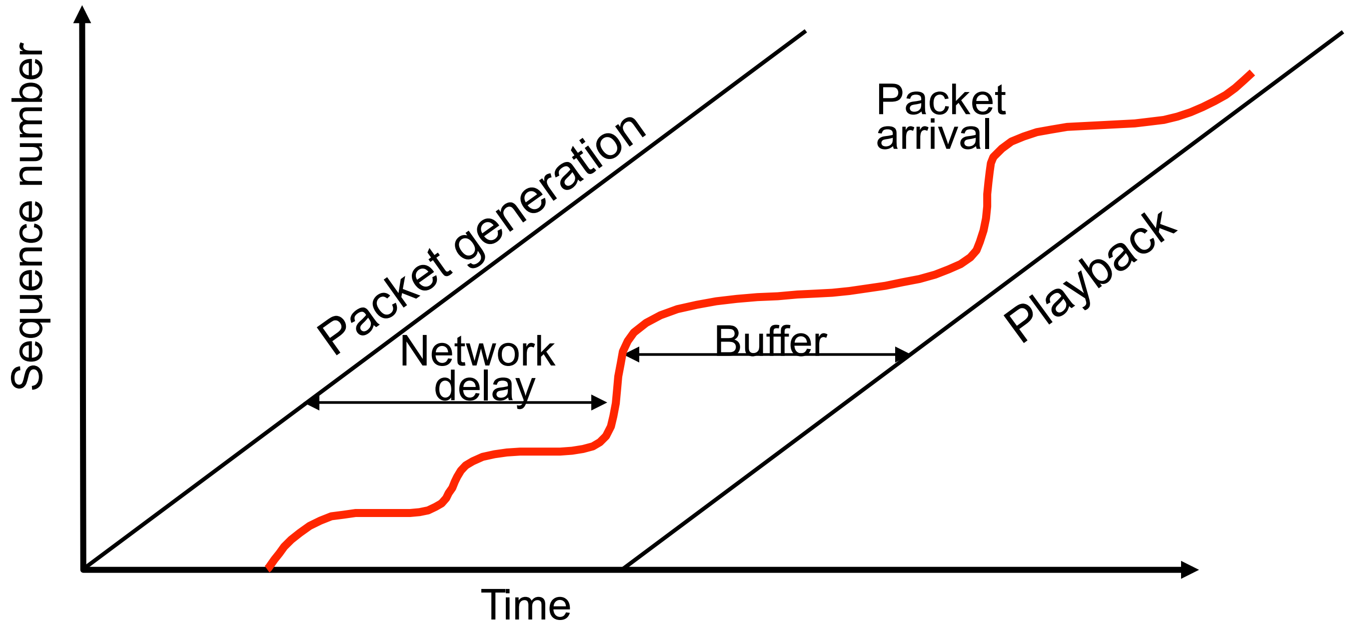
Real-time Applications

- Two types or applications
  - Hard real-time
  - Elastic (soft real-time)
- Example real-time application requirements - audio
  - Sample voice once every 125μs
  - Each packet has a playback time
  - Packets experience variable delay in network
  - Add constant factor to playback time – playback point



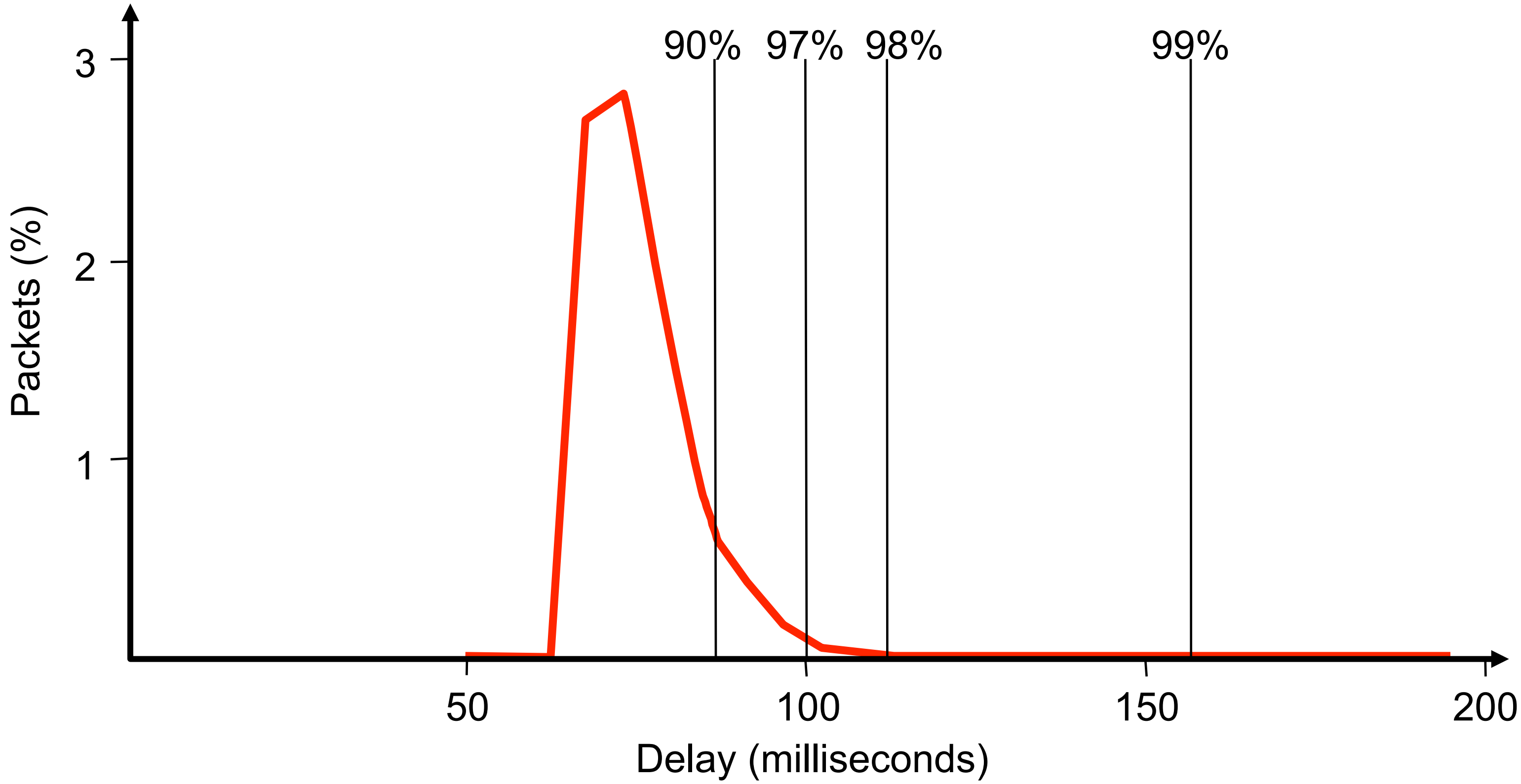
Real-Time Applications

- Playback Buffer



Delay Distribution

What is a good delay?



Quality of Service Approaches

- Approach : Admission control
  - Flow tells the network what it wants
  - Network decides if flow can be admitted
- Fine-grained
  - Provide QoS to individual applications or flows
  - Example: Resource Reservation Protocol (RSVP)
- Coarse-grained
  - Provide QoS to large classes of data or aggregated flows
    - Example: Differentiated Services (DIFFSERV)

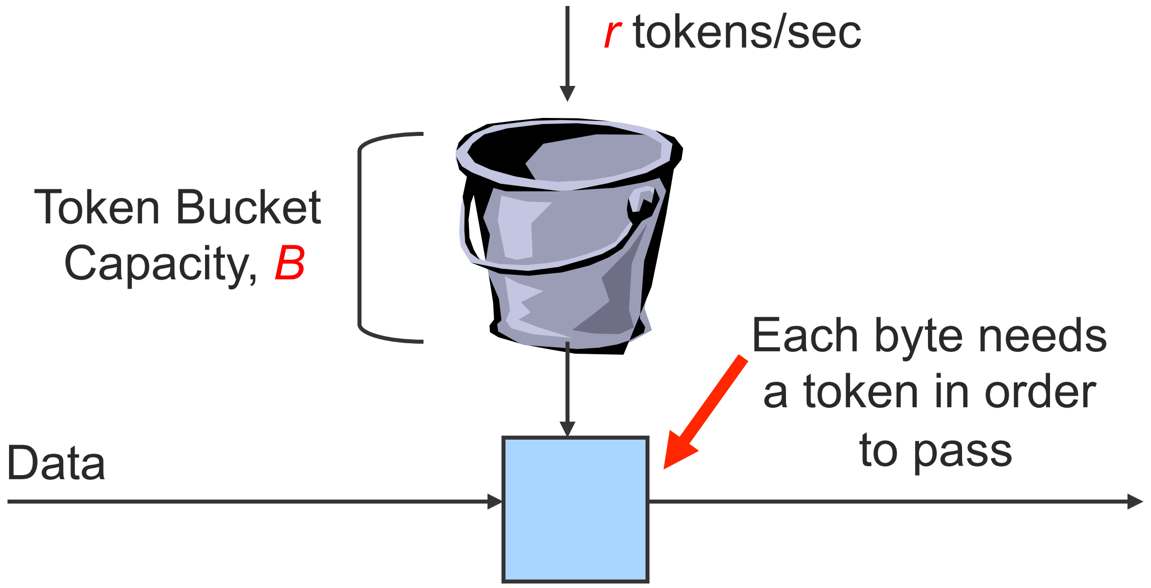
Mechanisms

- Flow specification
  - Tell the network what the flow wants
- Admission control
  - Network decides if it can handle flow
- Reservation
  - Enable admission control
- Packet classification
  - Map packets to flows
- Scheduling
  - Forwarding policy

Characterizing a Flow

- Describe flow's traffic characterization
  - Average bandwidth + burstiness: token bucket filter
  - Token rate: *r*
  - Bucket depth: *B*
- Use
  - 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

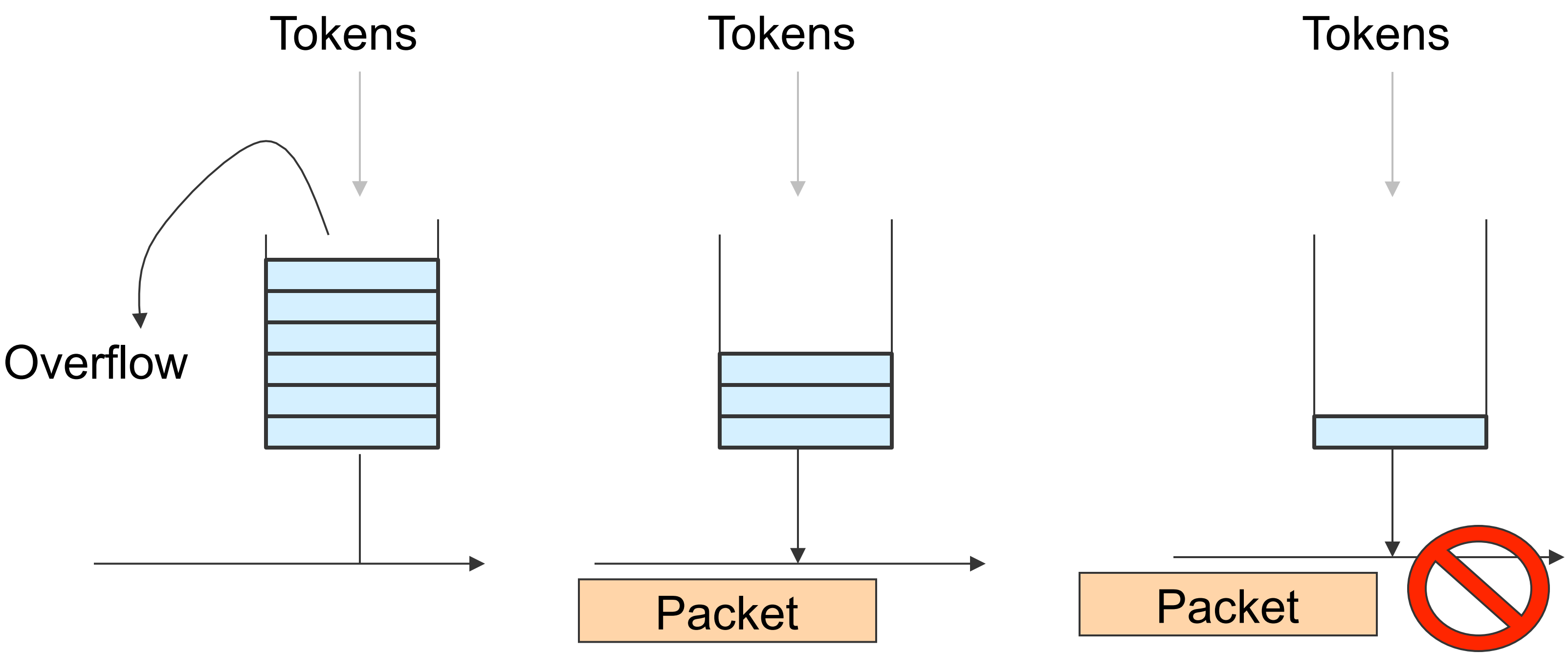
Token Bucket Filters



Dropping Filter: drops packets if token is not available

Buffered Filter: buffers data until tokens become available

Token Bucket Operation



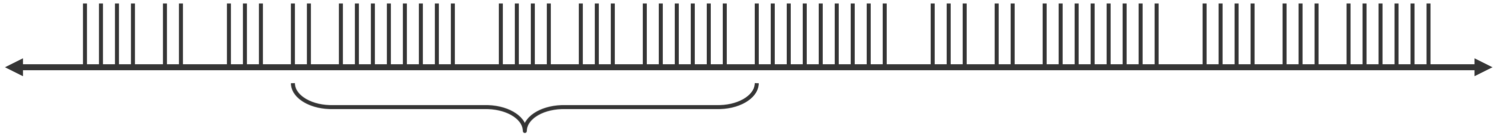
Buffer tokens up to capacity of bucket

Enough tokens → packet goes through, tokens removed

Not enough tokens → wait for tokens to accumulate

Token Bucket Filters

- Question
  - Given a finite length data stream, will it be affected by a token bucket filter?

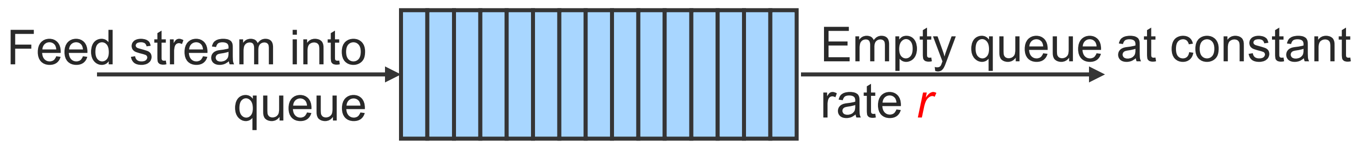


Not if during every time interval, the number of bytes is less than or equal to $B + rt$, where t is the length of the interval

- Given a token rate r and a finite data trace, how can the minimum token bucket size B be found such that the filter has no effect?

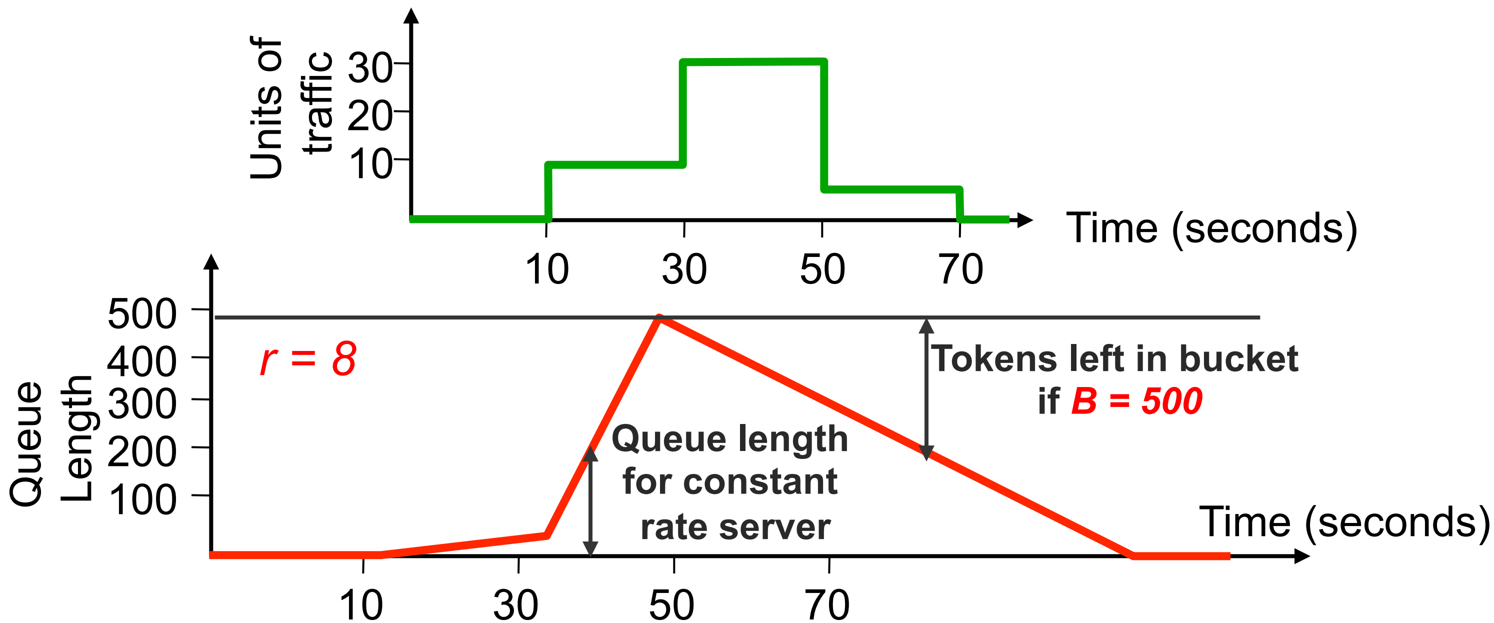
Token Bucket Filters

- Given a token rate *r* and a finite data trace, how can the minimum token bucket size *B* be found such that the filter has no effect?



- Simply observe the maximum buffer size
  - Why?
  - If the buffer is truncated to size *B*, then the number of empty buffer positions is equivalent to the number of tokens in an *(r,B)* token bucket filter

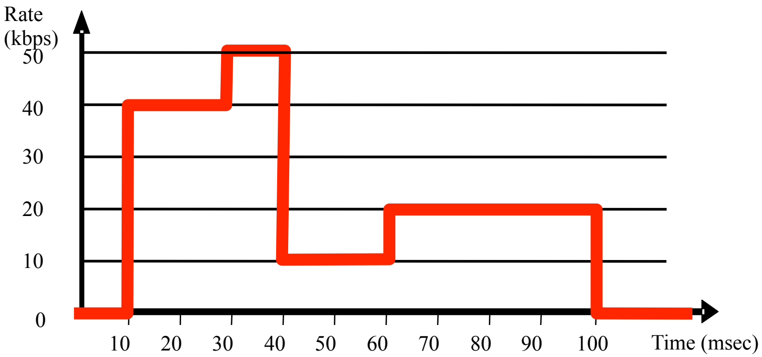
Token Bucket Filters



- The number of empty buffer positions for buffer size *B* and a constant rate server is equivalent to the number of tokens in an *(r,B)* token bucket filter

Token Bucket Filters

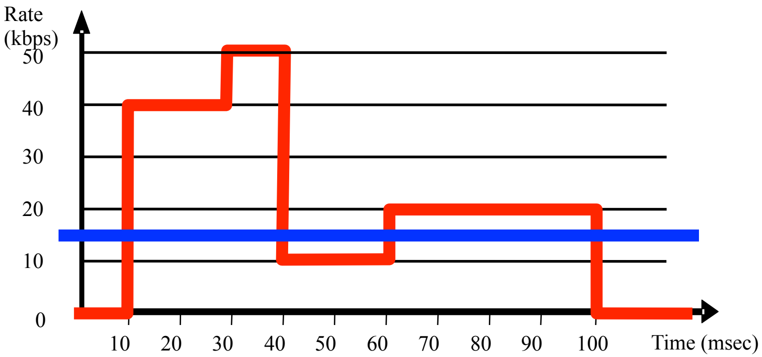
- *r = 15* kbps



- What is the minimum size of *B* required so that the filter lets the stream pass with no loss or delay?

Token Bucket Filters

- *r = 15* kbps

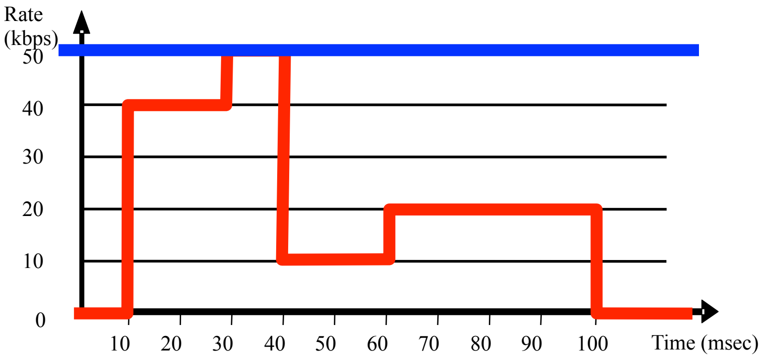


- Min *B* =

  (40 – 15) * 20 +
  (50 – 15) * 10 –
  (15 – 10) * 20 +
  (20 – 15) * 40
  = 950 bits

Token Bucket Filters

- What is the minimum B needed for arbitrary r > 0



- If r ≥ 50
  B = 0
- If 50 > r ≥ 40
  Min B = (50 – r) * 10
- If 40 > r ≥ 20
  Min B = (40 – r) * 20 + (50 – r) * 10
- If 20 > r ≥ 10
  Min B = (40 – r) * 20 + (50 – r) * 10 – (r – 10) * 20 + (20 – r) * 40
- If 10 > r ≥ 0
  Min B = (40 – r) * 20 + (50 – r) * 10 + (10 – r) * 20 + (20 – r) * 40

Differentiated Services

- Goal
  - Scalability through the use of only a small number of service classes
    - Two classes
      - Regular and premium (i.e. first class and bulk mail)
    - Diffserv
      - Proposes 6 bits of IP ToS field (26 = 64 classes)
- Questions
  - Who is allowed to set the premium bit?
    - Typically an ISP
    - Should we allow an individual customer or application?
  - How do routers react to such a classification?
    - IETF has specified per-hop behavior

Differentiated Services

- Expedited forwarding
  - Per-hop behavior
  - Need to strictly limit the load of traffic receiving expedited forwarding
    - Give strict priority
    - Use weighted fair queueing (WFQ) and assign sufficiently large weights for traffic receiving expedited forwarding

Differentiated Services

- Assured forwarding
  - Per-hop behavior
  - Like RED but with “in” and “out” packets (RIO)
  - Does not reorder packets
  - For more than two classes of traffic, use weighted RED
- Profile meters at the edges of ISP networks could mark packets as “in” or “out”