CSE/EE 461 – Lecture 17

David Wetherall

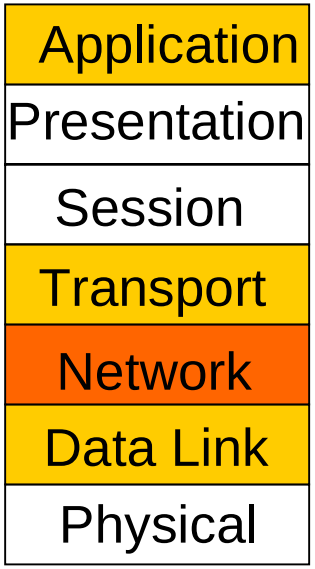
djw@cs.washington.edu

Last Time

- Multicast. See Keshav 11.11.

- Focus
  - How do we communicate efficiently with a group of participants

- Topics
  - Group communication
  - Multicast routing (DVMRP, PIM/CBT)
  - Future: reliable multicast

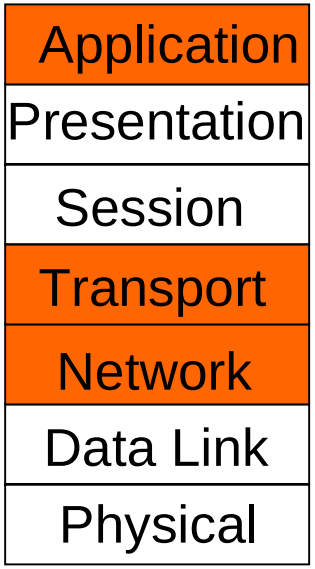


This Time

- Quality of Service. Keshav Ch 9, 13.

- Focus
  - What types of service can the network offer, and what do applications want?

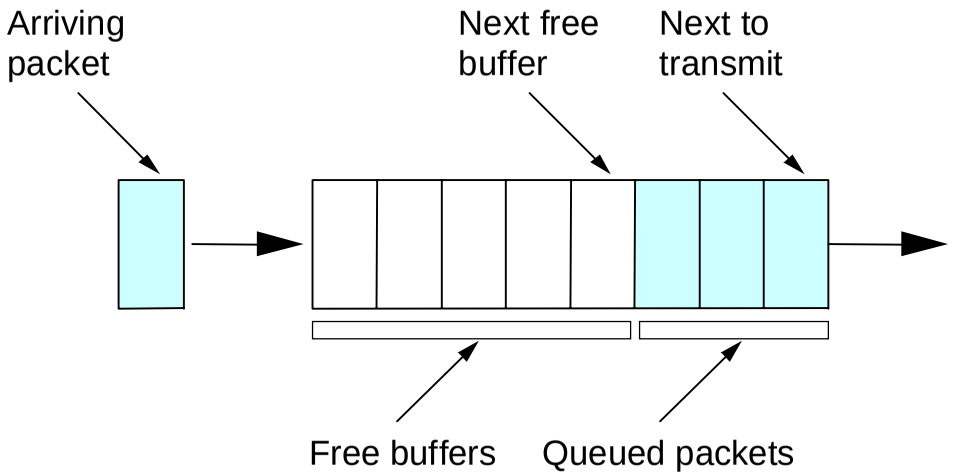
- Topics
  - Queuing and Scheduling (FIFO, WFQ)
  - Congestion avoidance (RED w/ECN)
  - Integrated Services (RSVP)
  - Differentiated Services (Diffserv)

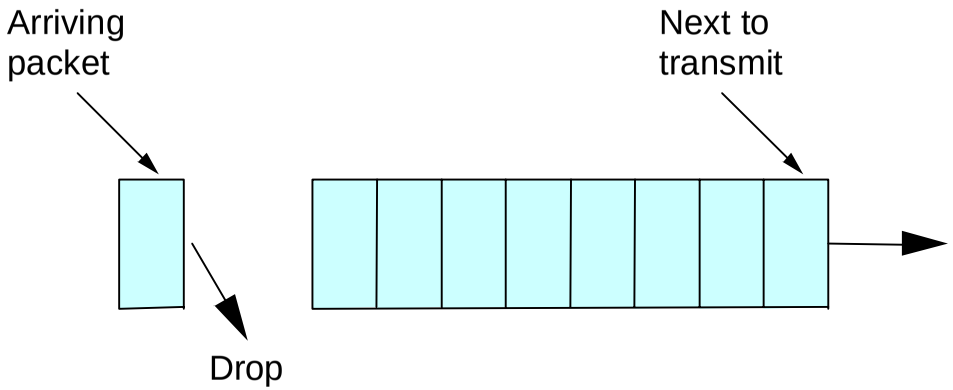


Revisiting "Best Effort" Service

- Our models so far:
  - IP at routers: a first come first serve queue [next slide]
  - TCP at hosts: probes for available bandwidth to causing loss
- The mechanisms at routers and hosts determine the kind of service applications will receive from the network
  - e.g., TCP causes loss and delay as it competes for bandwidth!
  - We want better mechanisms to support demanding applications
- Issues:
  - Scheduling: which packet goes next?
  - Buffer management: which packets get dropped?
  - Congestion: how do hosts use the network yet avoid congestion?

FIFO with Tail Drop



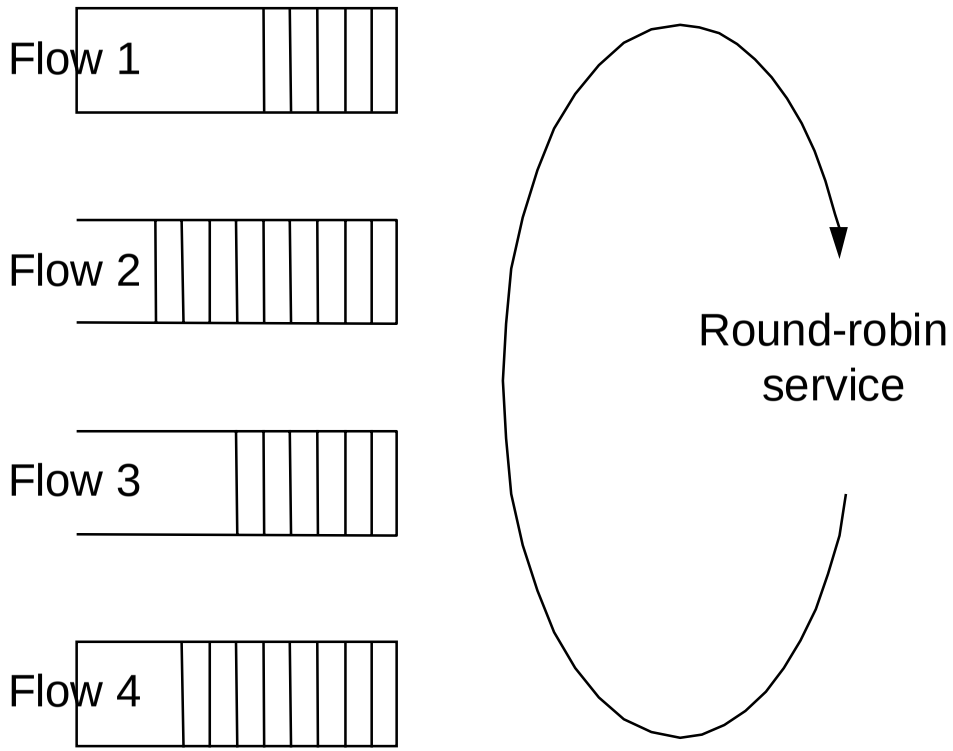


Fair Queuing (FQ)

- FIFO is not guaranteed (or likely) to be fair
  - Flows jostle each other and hosts must play by the rules
  - Routers don't discriminate traffic from different sources

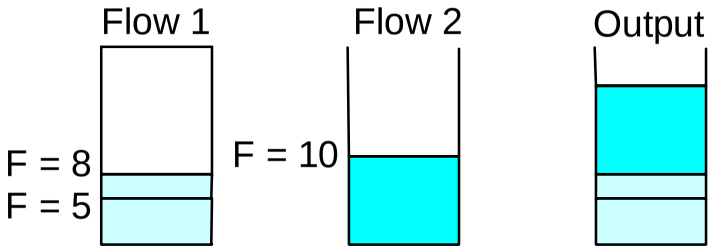
- Fair Queuing is an alternative scheduling algorithm
  - Maintain one queue per traffic source (flow) and send packets from each queue in turn
    - Actually, not quite, since packets are different sizes
  - Provides each flow with its "fair share" of the bandwidth
- Issues:
  - Implementation complexity, definition of flow

Fair Queuing



Fair Queuing

- Want to share bandwidth
  - At the "bit" level, but in reality must send whole packets
- Approximate with finish times for each packet
  - finish (F) = arrive + length*rate; rate depends on # of flows
  - Send in order of finish times, except don't preempt (stop) transmission if a new packet arrives that should go first

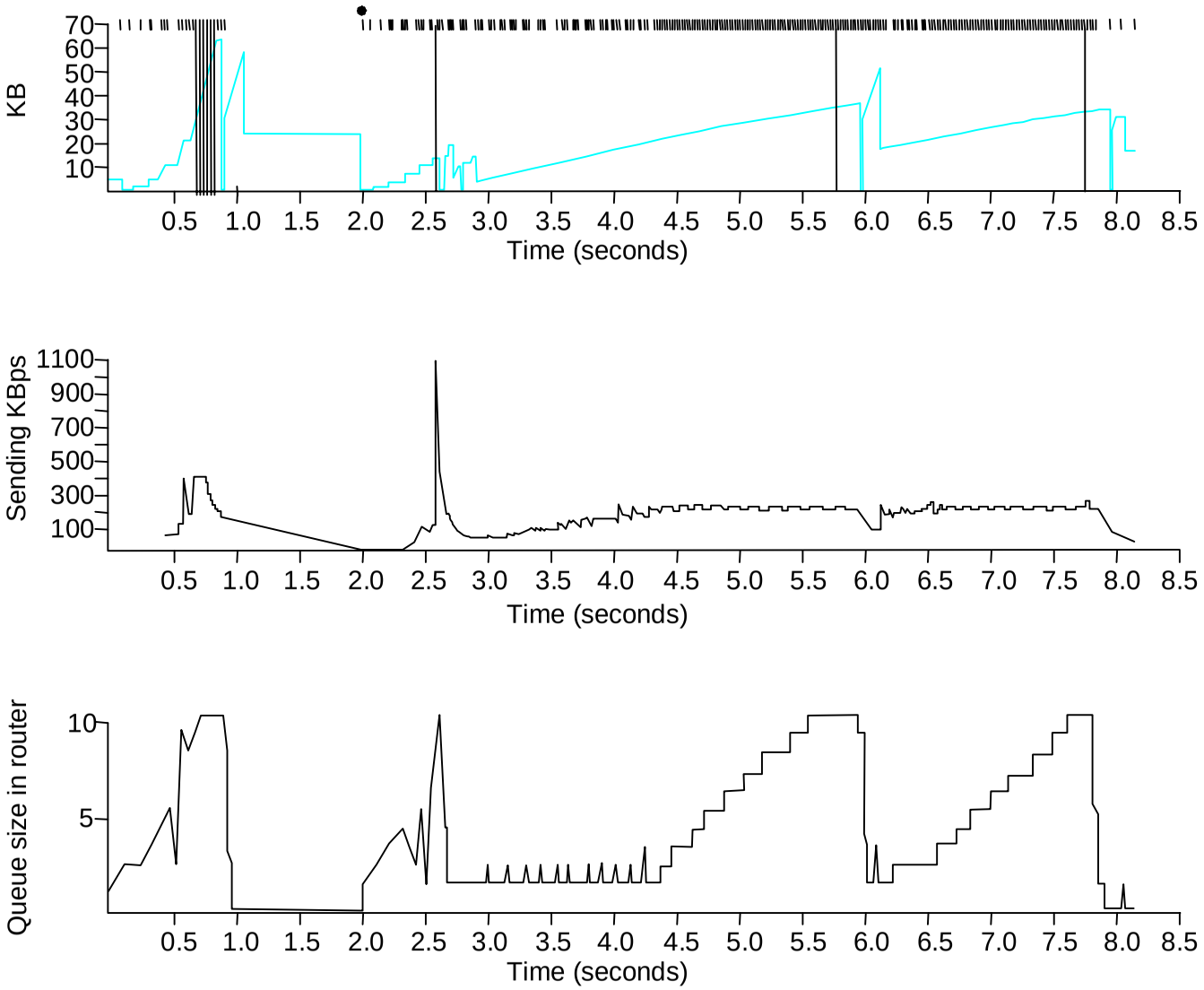


- More generally, assign weights to queues (WFQ)

Congestion Avoidance

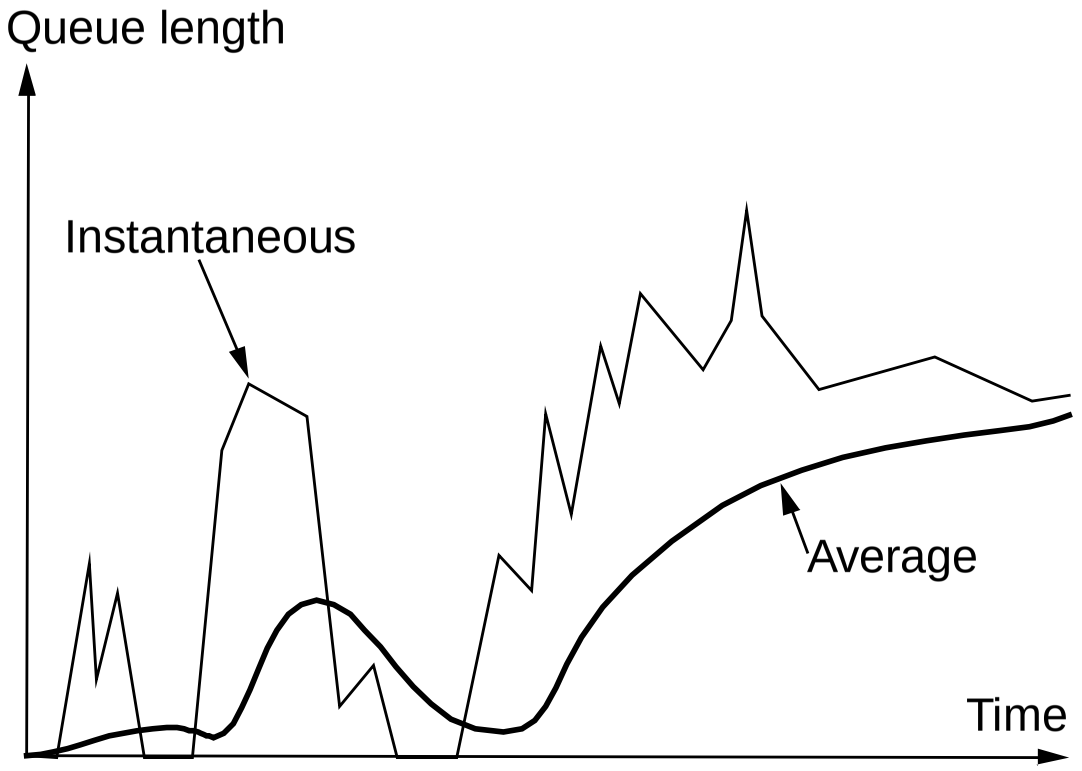
- TCP provides congestion control:
  - It recovers from congestion once it occurs
  - We would like to avoid congestion in the first place. Why?

- Congestion avoidance mechanisms
  - Aim to detect incipient congestion, before loss
  - Common approaches monitor queuing at routers
  - Queue only intended to absorb bursts, not build steadily


KB
70
60
50
40
30
20
10
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5
Time (seconds)
Sending KBps
1100
900
700
500
300
100
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5
Time (seconds)
Queue size in router
10
5
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5
Time (seconds)

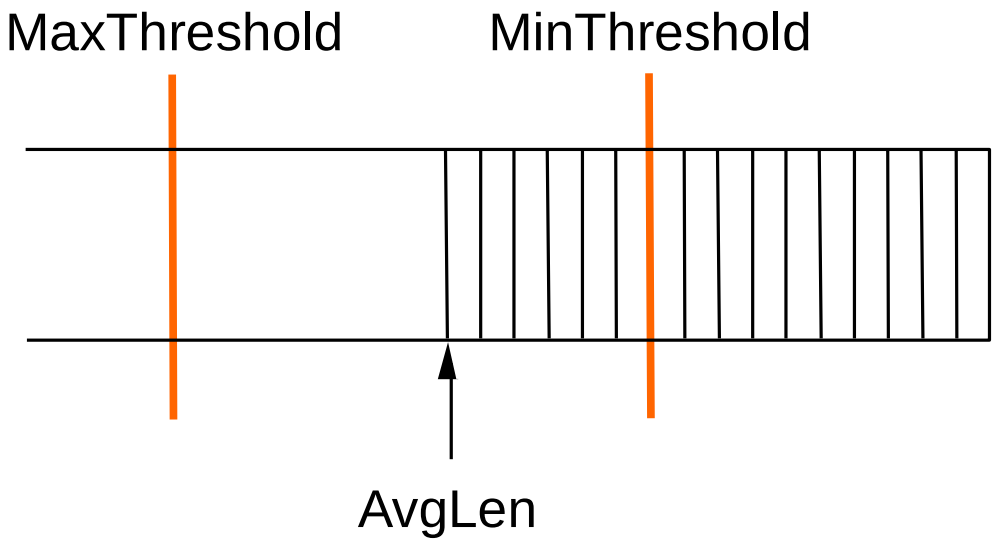
Incipient Congestion at a Router

- Sustained overload causes queue to build and overflow



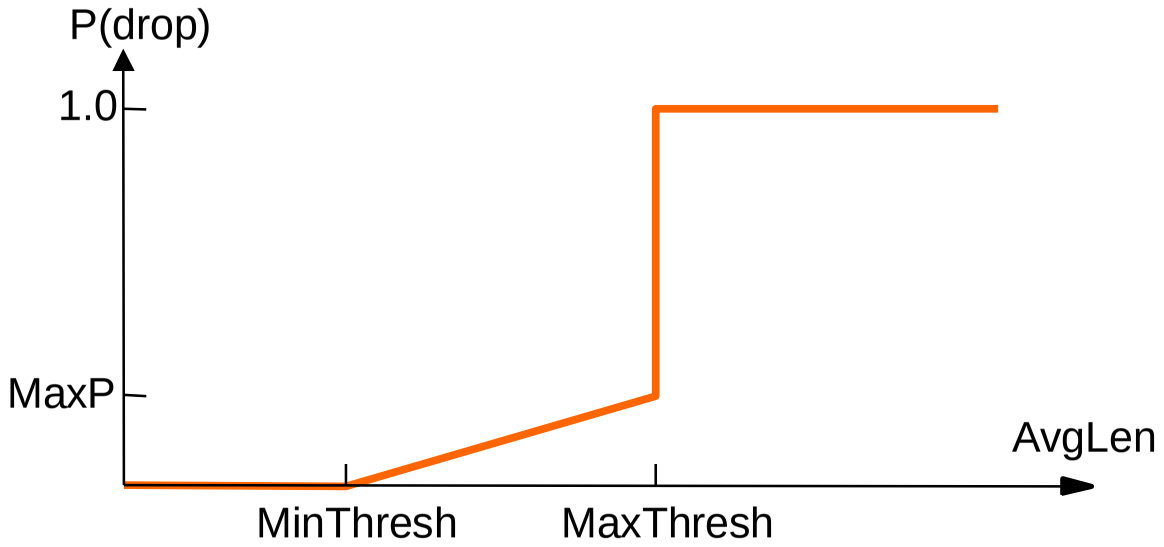
Random Early Detection (RED)

- Send "early" signal by probabilistically dropping a packet, allow source to respond before queue builds



Red Drop Curve

- Start dropping a fraction of the traffic as queue builds
- When queue is too high, revert to drop tail
- Nice theory, difficult to set parameters in practice



Explicit Congestion Notification (ECN)

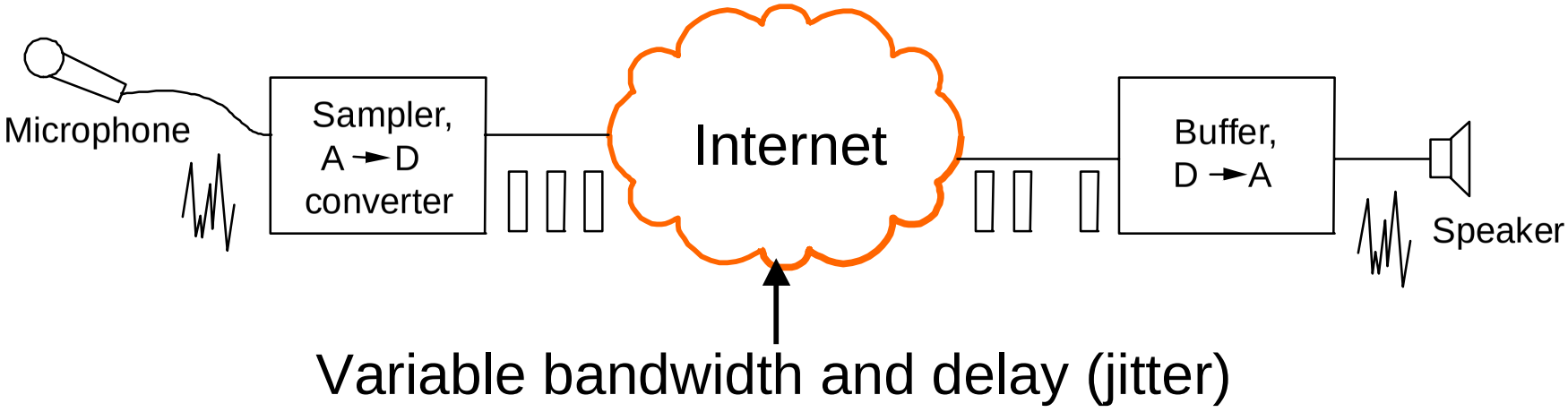
- Why only drop packets to signal congestion?
  - Drops are a robust signal, but there are other means …
  - We need to be careful though: no extra packets

- ECN signals congestion by setting a bit in the IP header
- Receiver returns indication to the sender, who slows

- RED actually works by "marking" packets
  - Mark can be a drop or ECN signal if hosts understand ECN
  - Supports congestion avoidance without loss

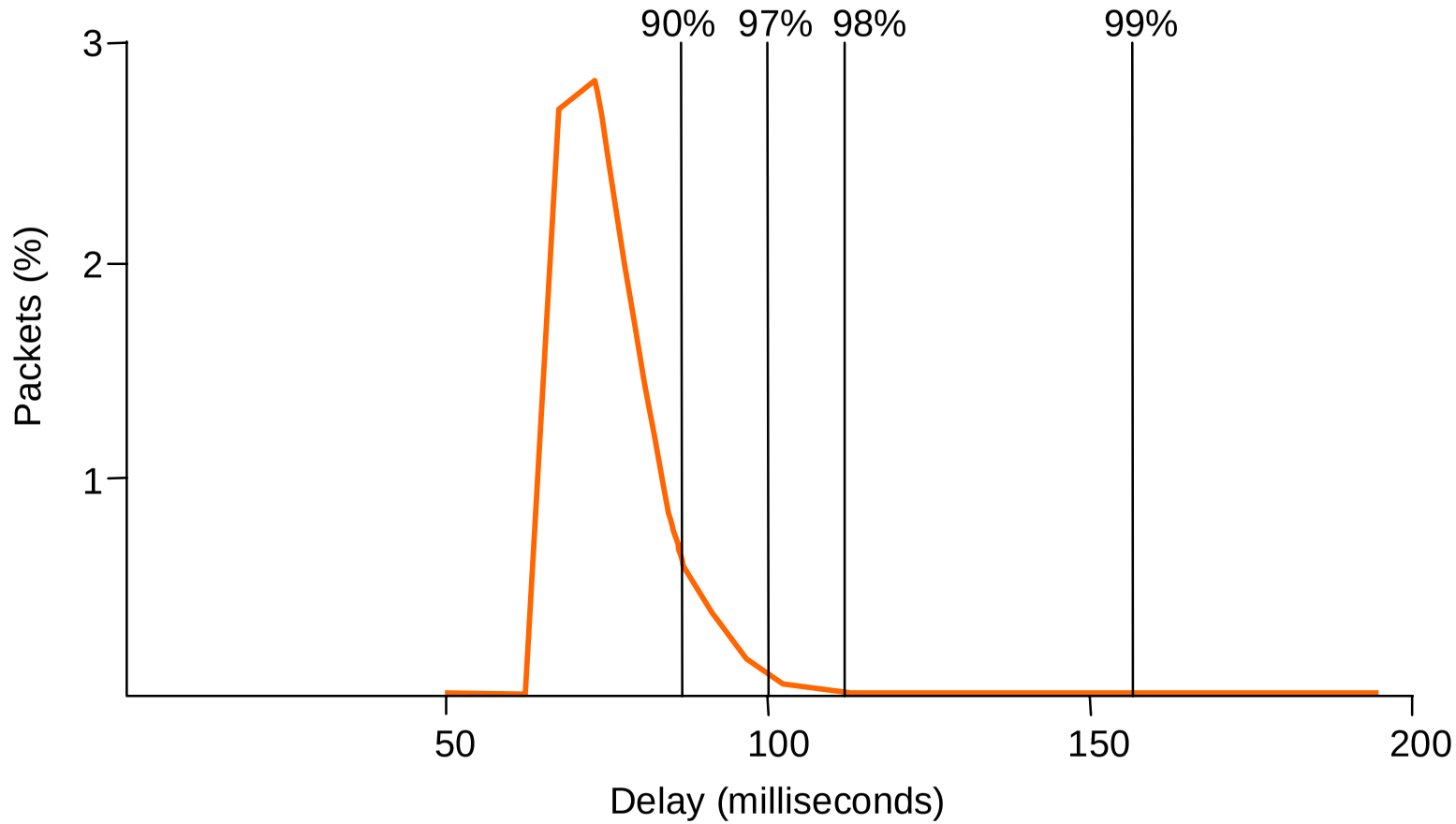
QOS – Application Needs

- Different applications have different network needs
  - Consider voice over IP as a real-time service

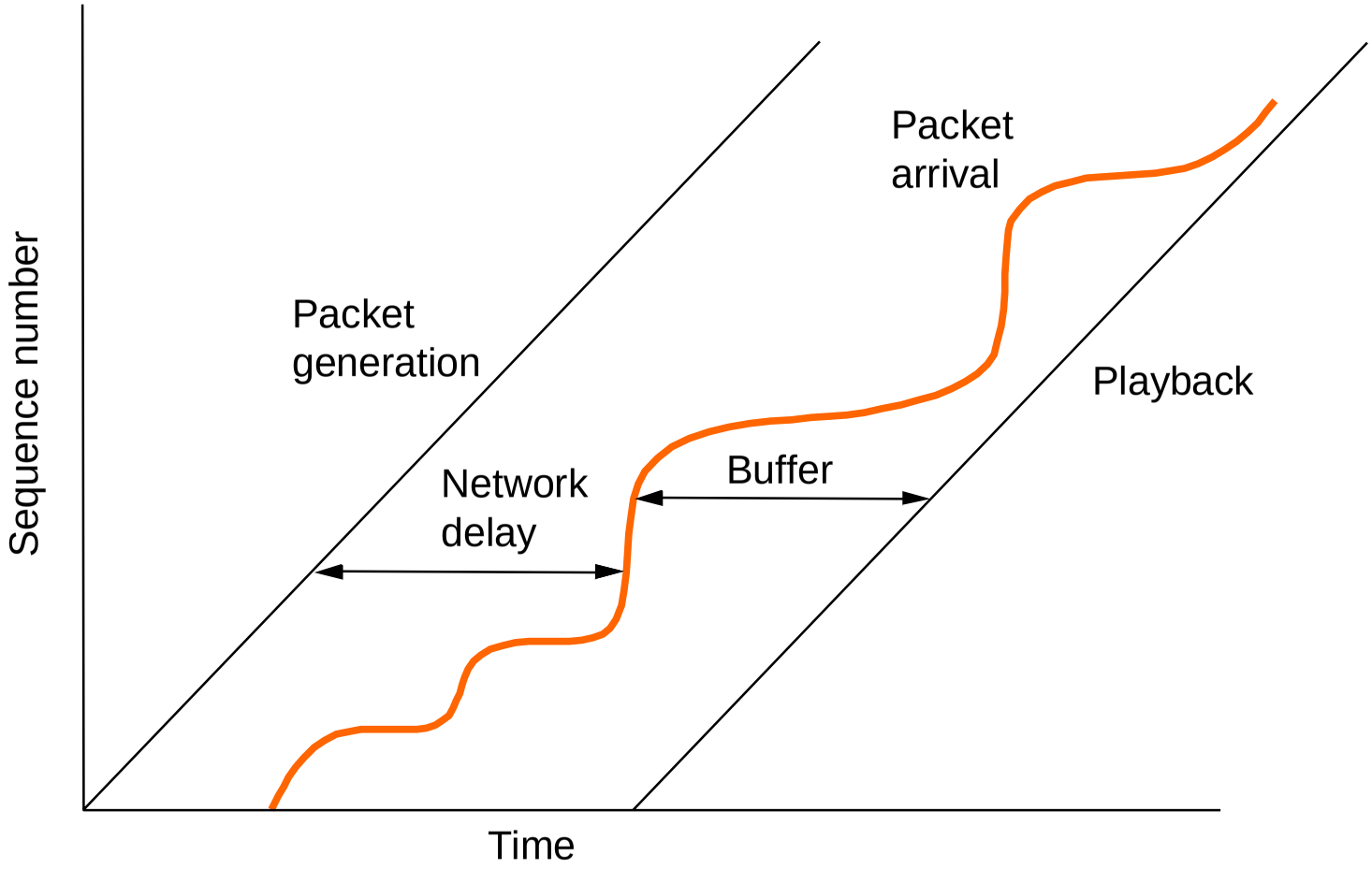


- Real-time apps need assurances from the network

Delay and Jitter



Playback with Buffering

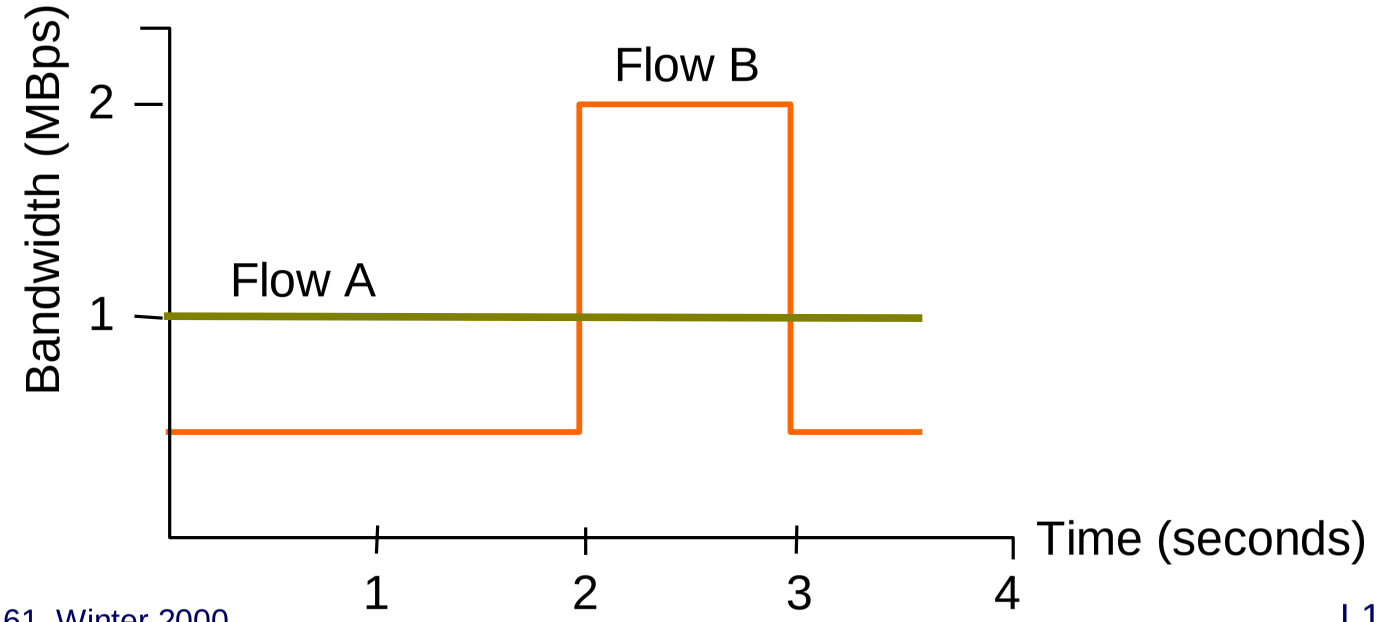


Approaches to QOS

- Fine-grained
  - Each application negotiates individual QOS guarantees
  - IETF Integrated services (Intserv)
- Coarse-grained
  - Several different QOS classes that apps can choose from
  - IETF Differentiated Services (Diffserv)
- To provide assurances we need to limit load
  - Admission control reserves network resources

Intserv

- Application reserves required resources
  - Send messages along a network path, e.g., with RSVP
  - Need to describe flow requirements to routers
  - Routers set aside resources, e.g., separate queues with priority



Diffserv

- Small number of different services: premium, regular
  - Packets marked for kind of service in IP header (TOS redefined)
- Routers understand different services but not flows
  - Might separate classes with WFQ
- Customers buy premium service from ISP ahead of time
  - Much less dynamic than Intserv
  - Marking policed at administrative boundaries

Key Concepts

- Congestion avoidance w/router support
- Different scheduling and buffer management algorithms can provide different kinds of service
- Real-time applications need service assurances (QOS)
- Assurances require admission control
- Two approaches being explored today:
  - Intserv: per flow reservations
  - Diffserv: small number of service classes