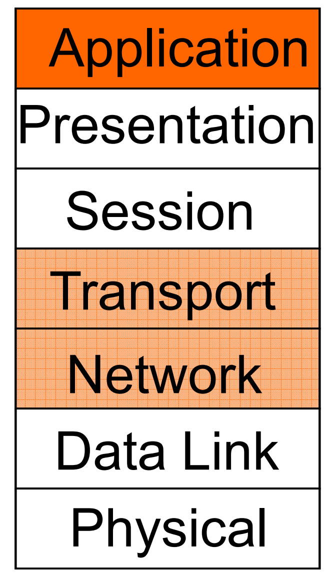
Real-time apps and Quality of Service

- Focus
  - What transports do applications need?
  - What network mechanisms provide which kinds of quality assurances?



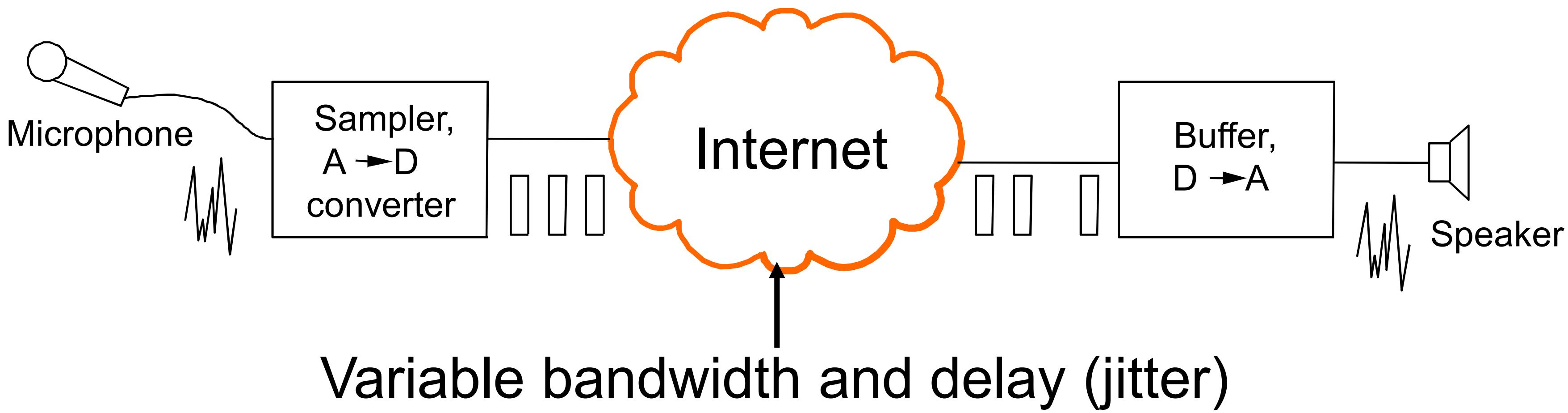
- Topics
  - Real-time versus Elastic applications
  - Adapting to variable delay
  - Token buckets as bandwidth descriptors
  - Scheduling and buffer management
  - Fair Queuing, Intserv / Diffserv

Internet “Best Effort” Service

- Our network model so far:
  - IP at routers: a shared, first come first serve (drop tail) queue
  - TCP at hosts: probes for available bandwidth, causing loss
- The mechanisms at routers and hosts determine the kind of service applications will receive from the network
  - TCP causes loss and variable delay, and Internet bandwidth varies!
- Q: What kinds of service do different applications need?
  - The Web is built on top of just the “best-effort” service
  - Want better mechanisms to support demanding applications

An Audio Example

- Playback is a real-time service in the sense that the audio must be received by a deadline to be useful

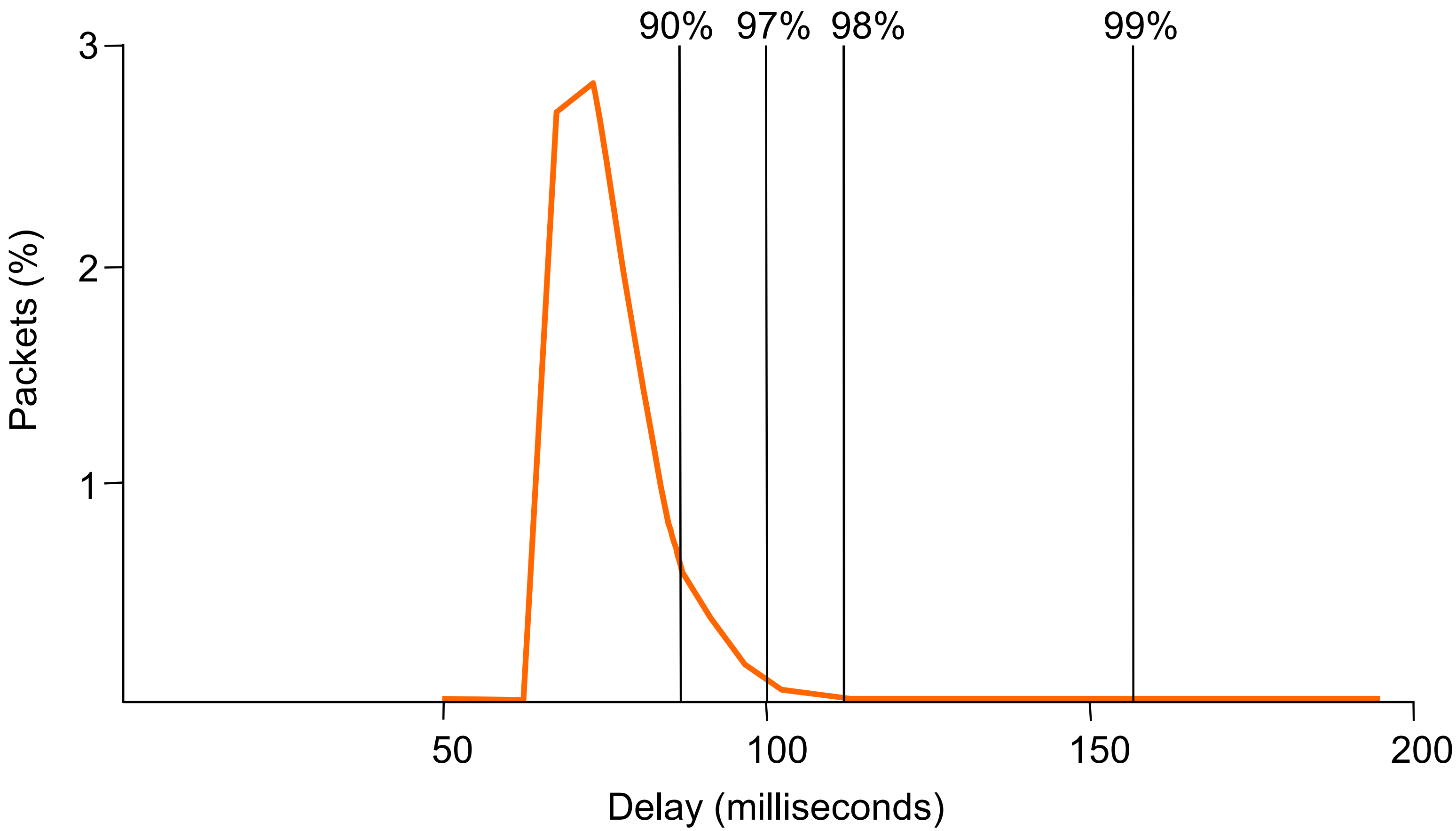


- Real-time apps need assurances from the network
- Q: What assurances does playback require?

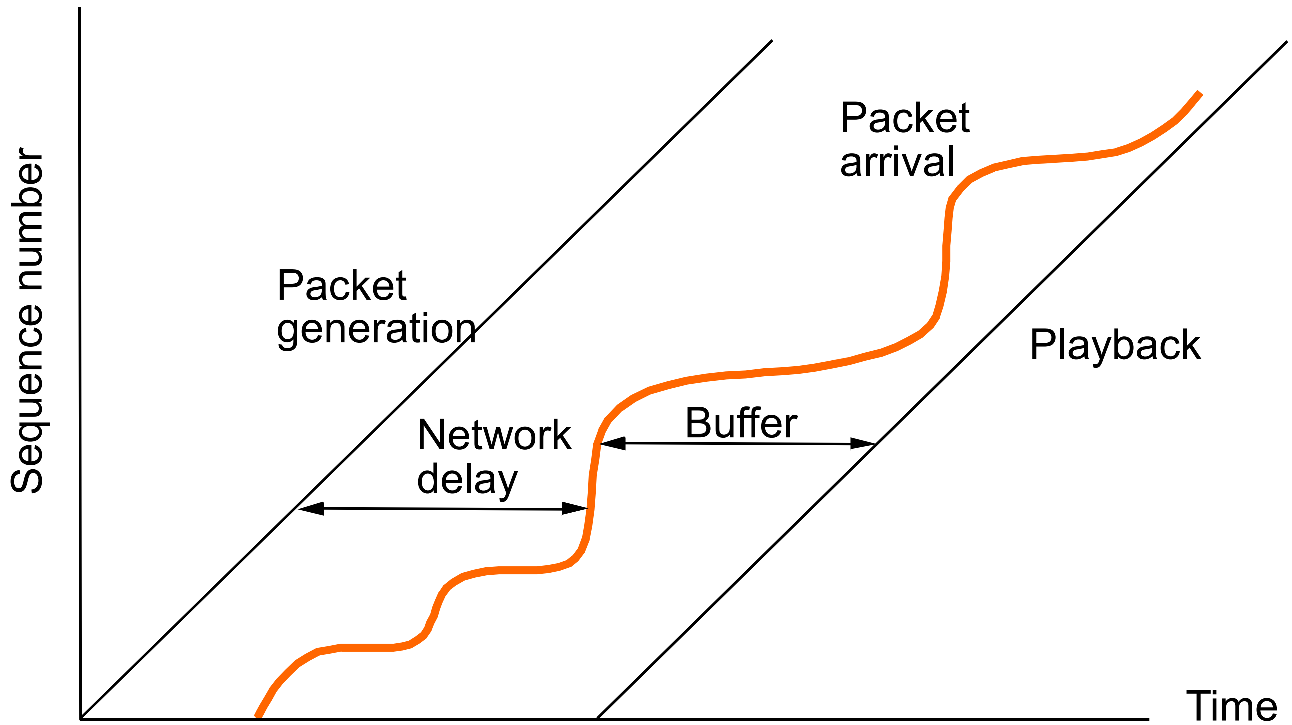
Network Support for Playback

- Bandwidth
  - There must be enough on average
  - But we can tolerate to short term fluctuations
- Delay
  - Ideally it would be fixed
  - But we can tolerate some variation (jitter)
- Loss
  - Ideally there would be none
  - But we can tolerate some losses

Example: Delay and Jitter

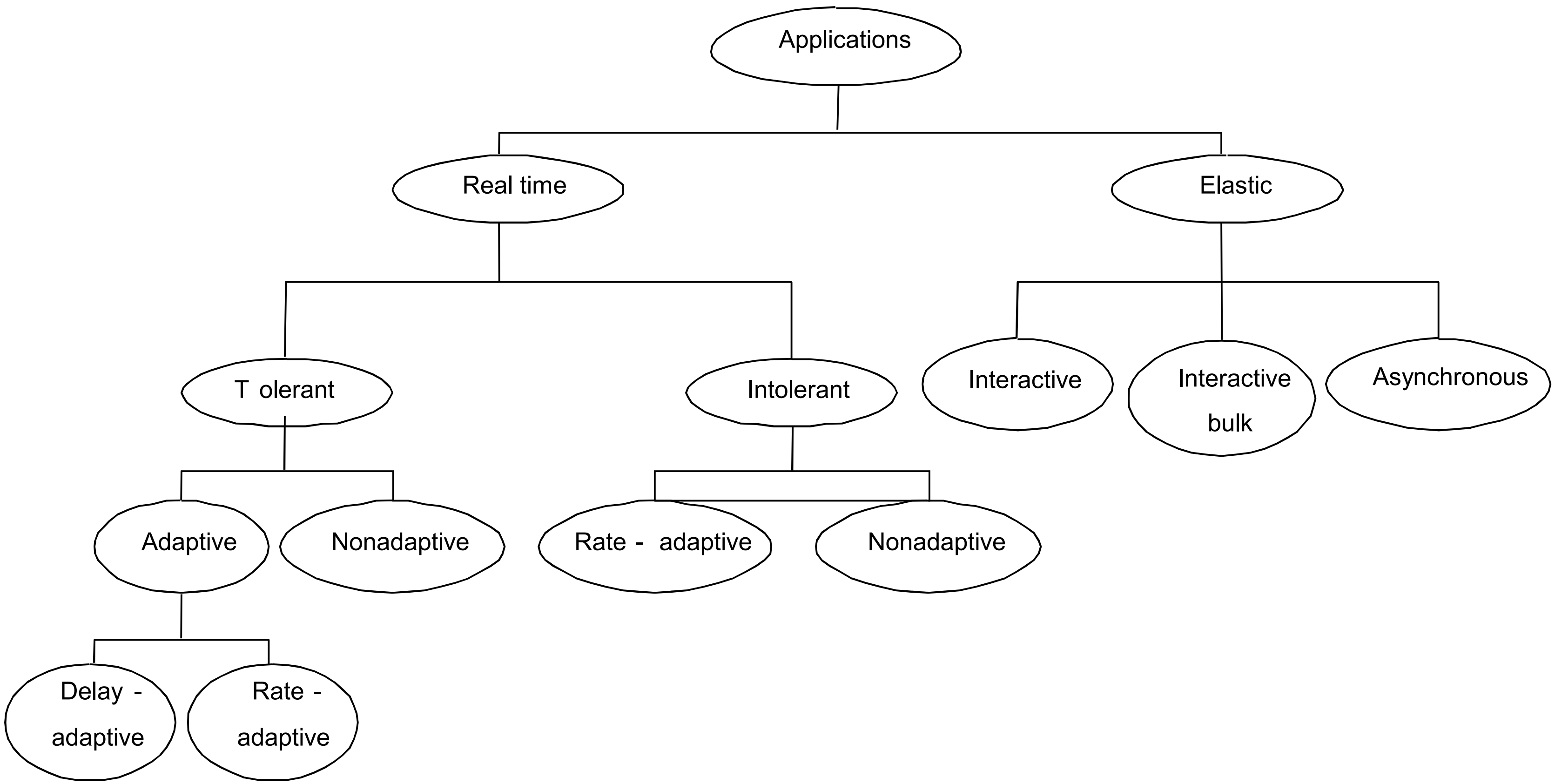


Tolerating Jitter with Buffering



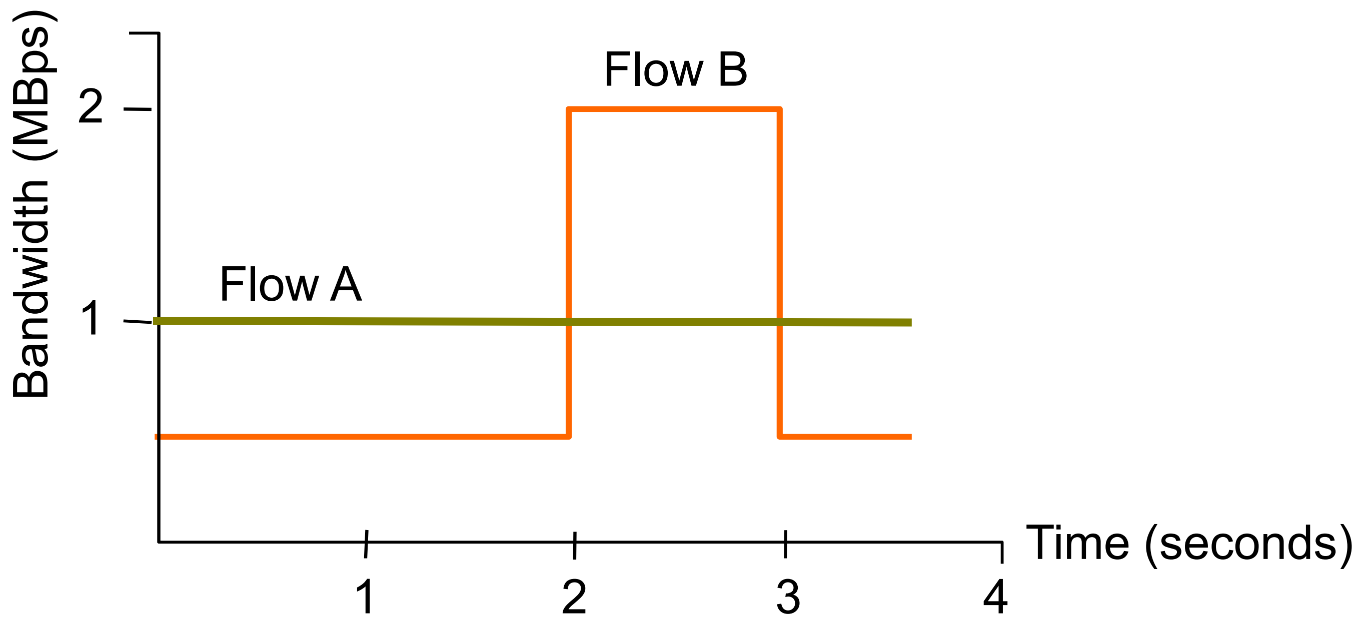
- Buffer before playout so that most late samples will have arrived

Taxonomy of Applications



Specifying Bandwidth Needs

- Problem: Many applications have variable bandwidth demands

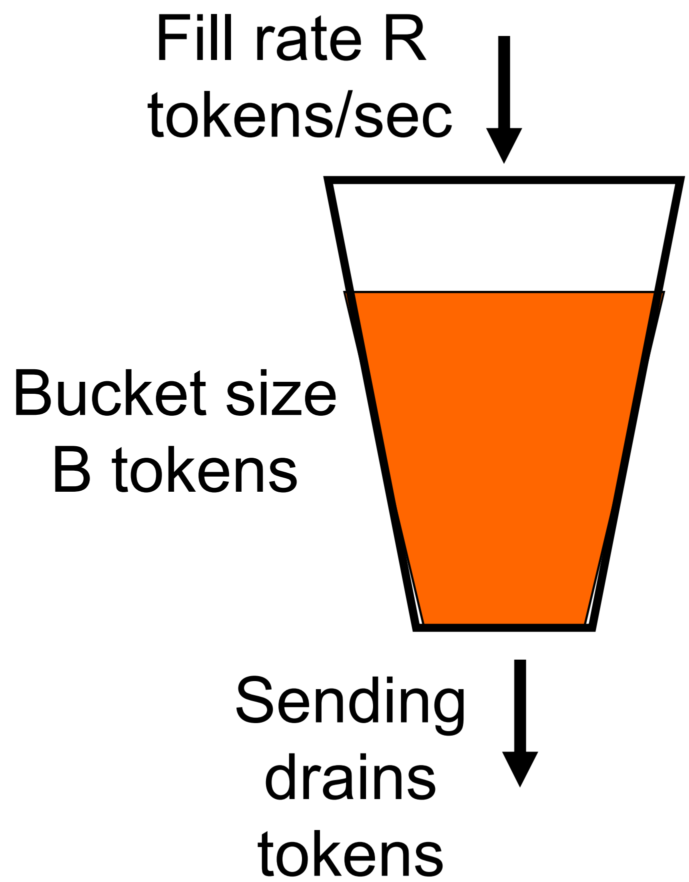


- Same average, but very different needs over time. One number. So how do we describe bandwidth to the network?

Token Buckets

- Common, simple descriptor

- Use tokens to send bits
- Average bandwidth is R bps
- Maximum burst is B bits



Network Roadmap – Various Mechanisms

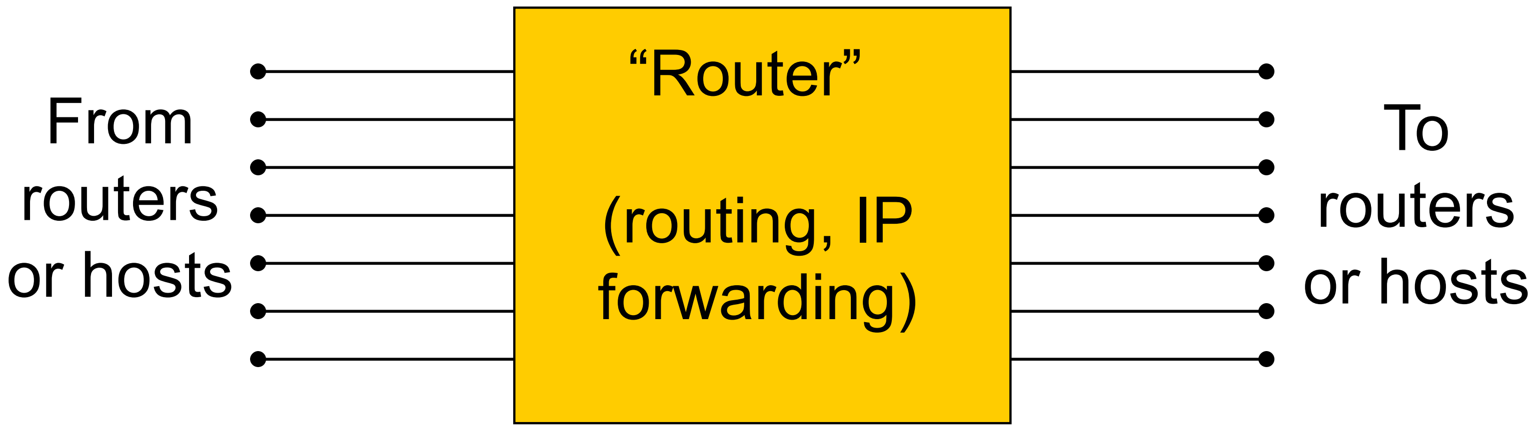
Simple to build,
Weak assurances

↕

Complex to build,
Strong assurances

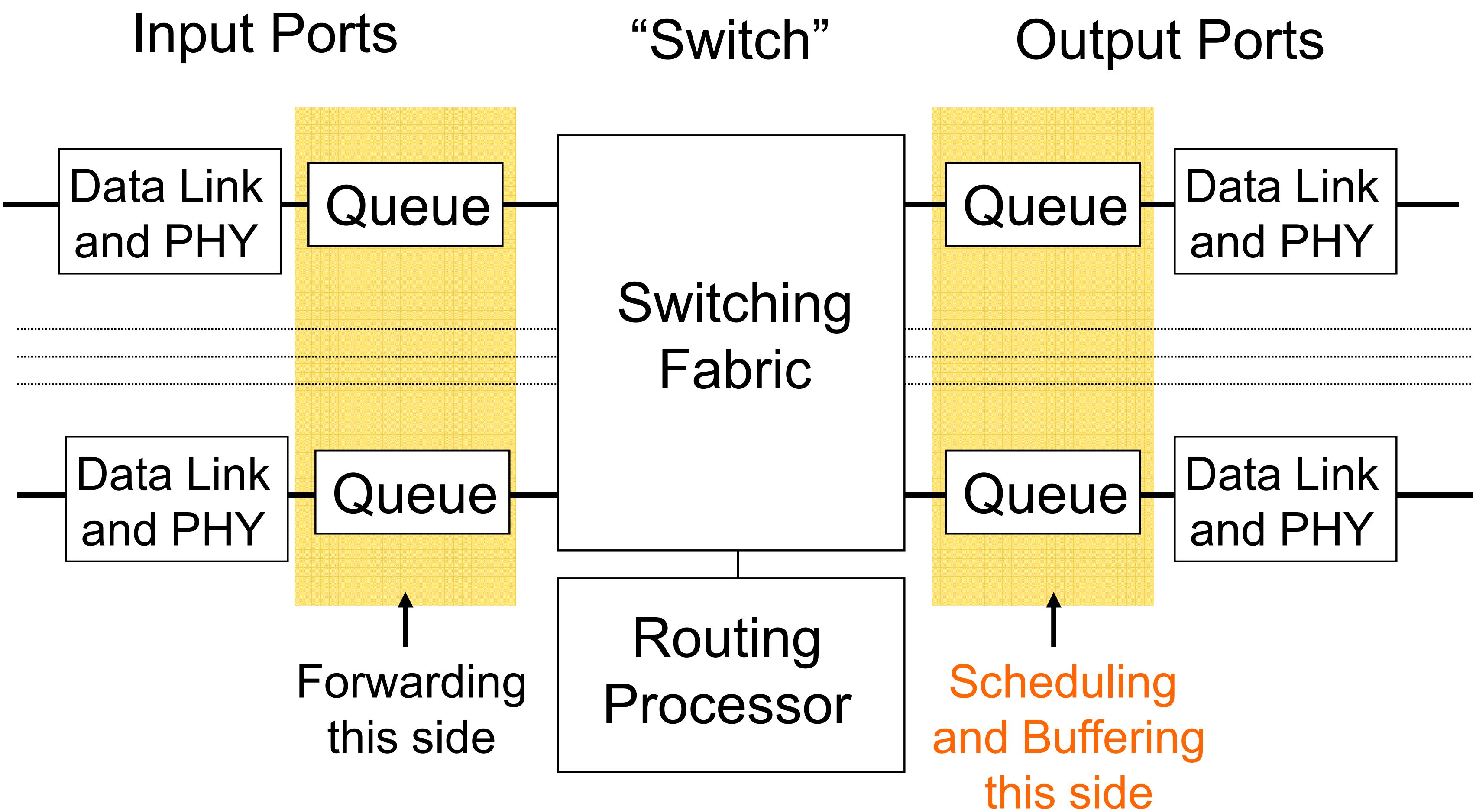
FIFO with Drop Tail	Classic Best Effort
FIFO with RED	Congestion Avoidance
Weighted Fair Queuing	Per Flow Fairness
Differentiated Services	Aggregate Guarantees
Integrated Services	Per Flow Guarantees

What's in a Router?



- By convention, draw input ports on left, output on right. (But in reality a single physical port handles both directions.)

Model of a Router



Scheduling and Buffer Management

- Two different functions implemented at the queue

- A scheduling discipline
  - This is the order in which we send queued packets
  - Examples: FIFO or priority-based

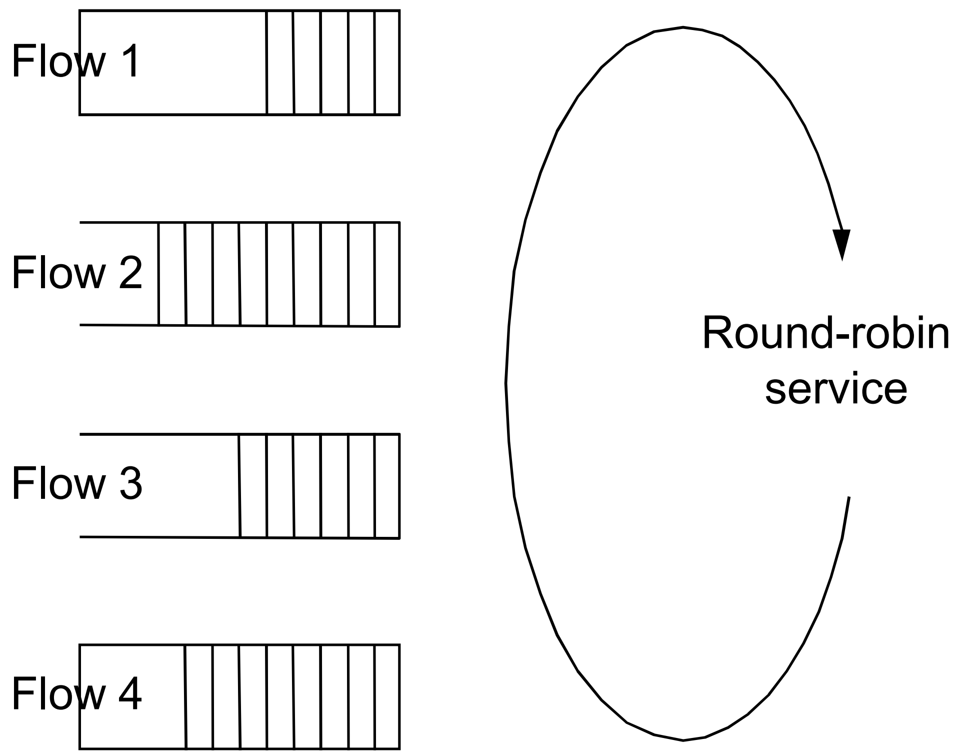
- A buffer management policy
  - This decides which packets get dropped or queued
  - Examples: Drop tail or random drop

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

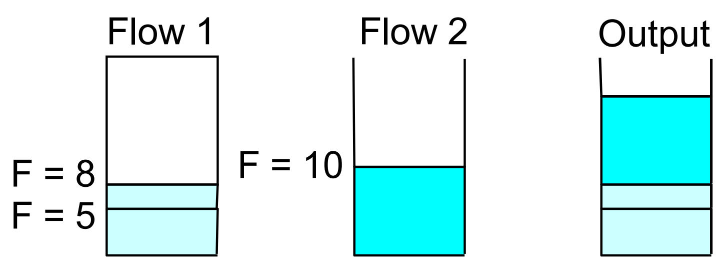
- Fair Queuing is an alternative scheduling algorithm
  - Maintain one queue per traffic source (flow) and send packets from each queue in turn
    - Actually, not quite, since packets are different sizes
  - Provides each flow with its "fair share" of the bandwidth

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 (Weighted FQ, WFQ)

Full support for QOS Guarantees

1. Flowspecs. Formulate application needs
   – Need descriptor, e.g. token bucket, to ask for guarantee
2. Admission Control. Decide whether to support a new guarantee
   – Network must be able to control load to provide guarantees
3. Signaling. Reserve network resources at routers
   – Analogous to connection setup/teardown, but at routers
4. Packet Scheduling. Use different scheduling and drop mechanisms to implement the guarantees
   – e.g., set up a new queue and weight with WFQ at routers

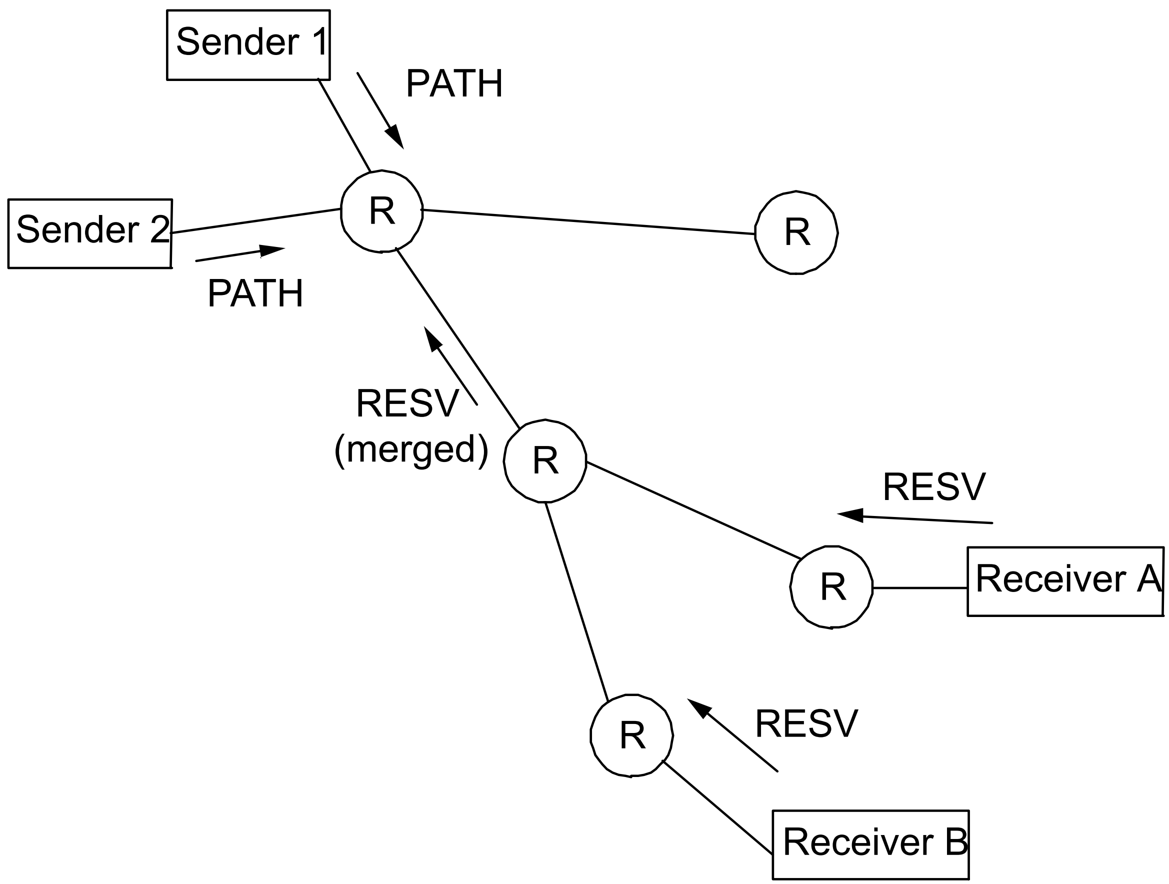
IETF Integrated Services

- Fine-grained (per flow) guarantees
  - Guaranteed service (bandwidth and bounded delay)
  - Controlled load (bandwidth but variable delay)

- RSVP used to reserve resources at routers
  - Receiver-based signaling that handles failures

- WFQ used to implement guarantees
  - Router classifies packets into a flow as they arrive
  - Packets are scheduled using the flow's resources

Resource Reservation Protocol (RSVP)



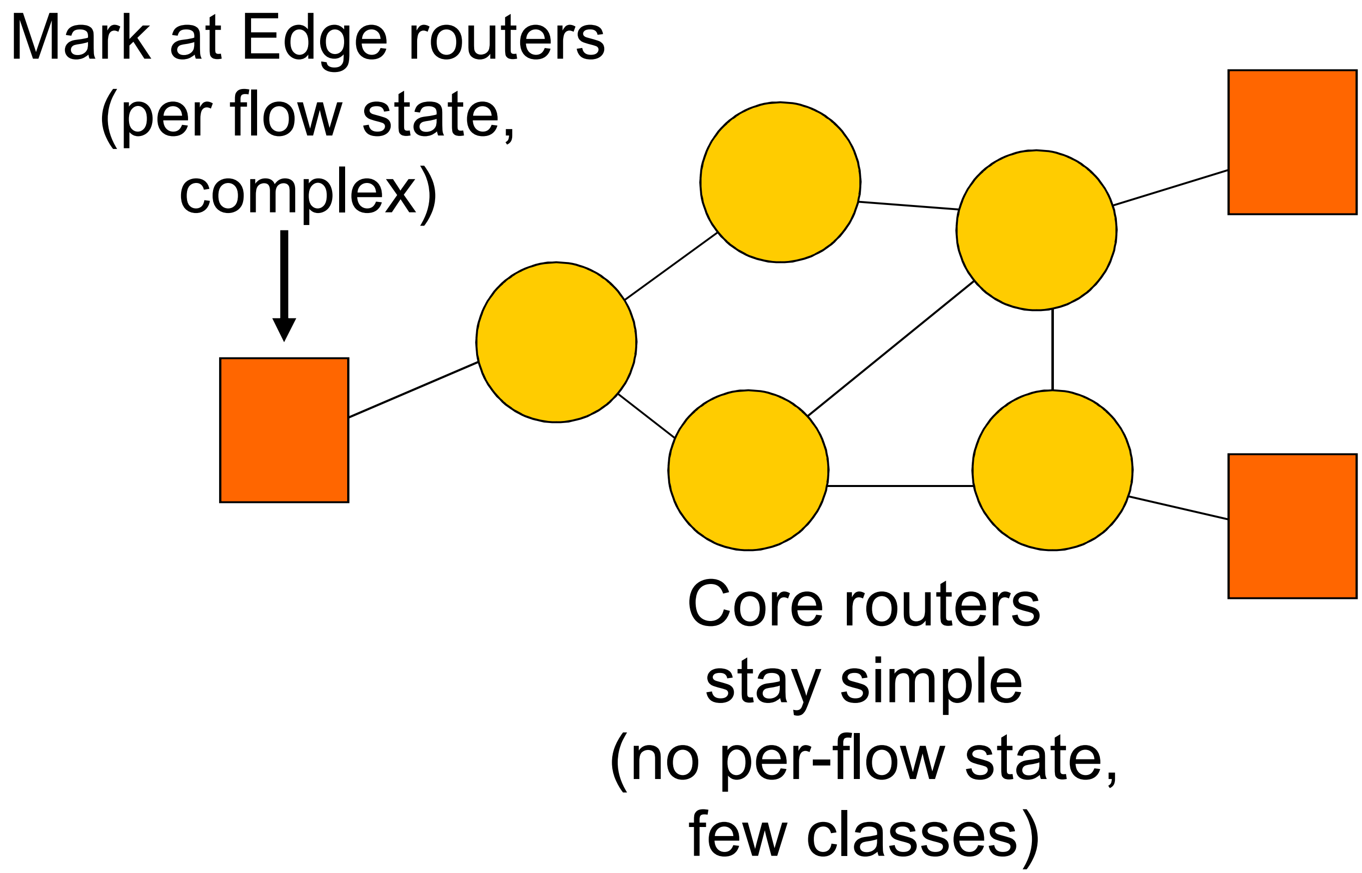
RSVP Issues

- RSVP is receiver-based to support multicast apps
- Only want to reserve resources at a router if they are sufficient along the entire path
- What if there are link failures and the route changes?
- What if there are sender/receiver failures?

IETF Differentiated Services

- A more coarse-grained approach to QOS
  – Packets are marked as belonging to a small set of services, e.g, premium or best-effort, using the TOS bits in the IP header
- This marking is policed at administrative boundaries
  – Your ISP marks 10Mbps (say) of your traffic as premium depending on your service level agreement (SLAs)
  – SLAs change infrequently; much less dynamic than Intserv
- Routers understand only the different service classes
  – Might separate classes with WFQ, but not separate flows

Two-Tiered Architecture



QOS in the Internet today

- Is in its infancy
  - Routers have many knobs (performance issues though)
  - Buy economic incentives stifle innovation/deployment
- Customers may get SLAs, e.g., bandwidth, uptime
  - Mostly a provisioning issue for ISPs
  - For well-provisioned, congestion is at the edges, e.g., DSL
- Network mostly decoupled from hosts
  - Hosts don't mark packets for QOS
  - But network edge devices may classify, e.g., VoIP vs P2P
  - Point solution at edge, or ISP network can then differentiate