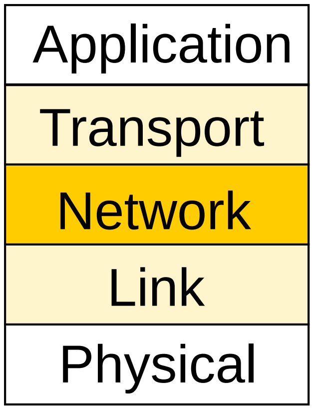
CSEP 561 – Quality of Service

David Wetherall
djw@cs.washington.edu

QOS

- Focus:
  - How to provide “better than best effort”

- Fair queueing
- Application needs
- Traffic shaping
- Guarantees
- IntServ / DiffServ



Network Roadmap – Various Mechanisms

Simple to build,
Weak assurances

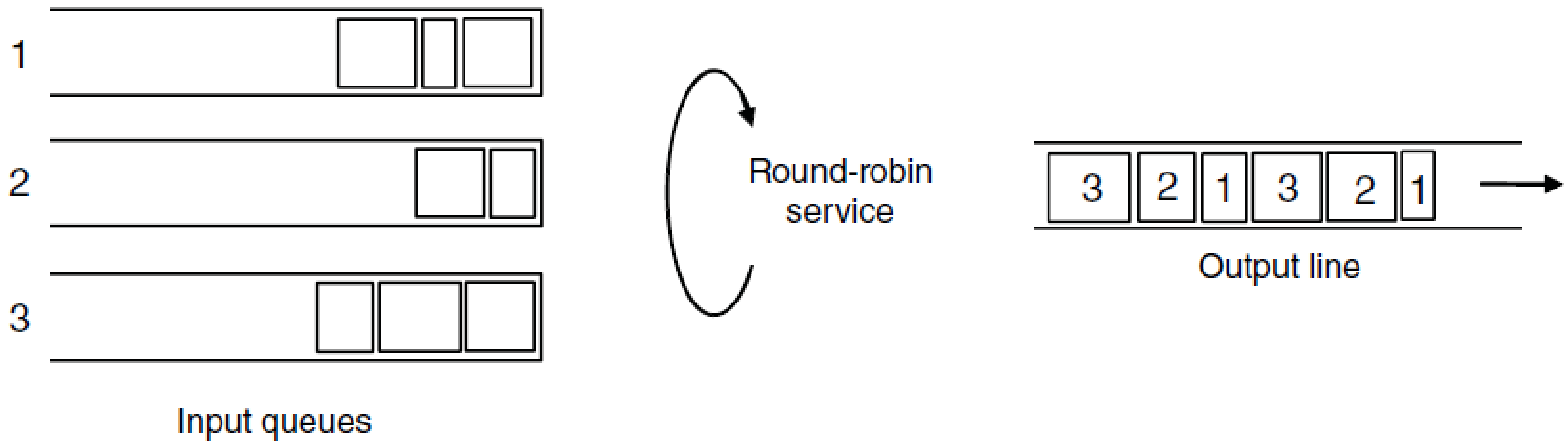
↕

Complex to build,
Strong assurances

FIFO w/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

Fairer Queuing: Round Robin (Nagle)

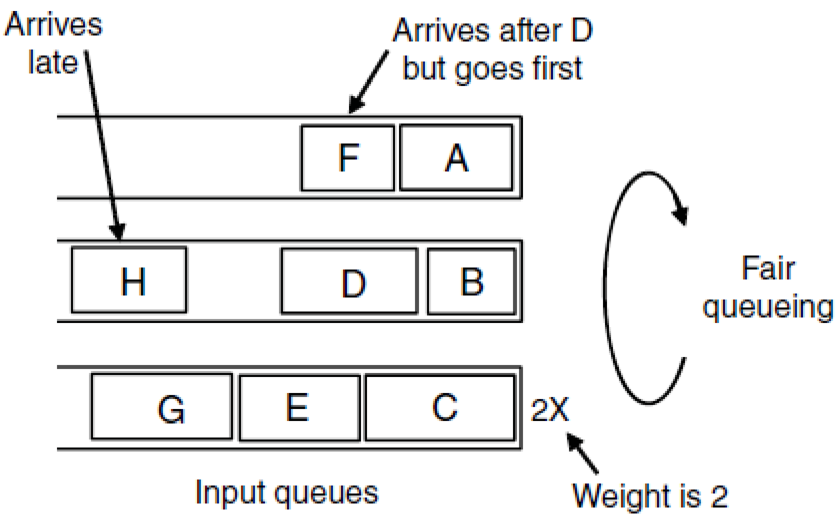
- Take one packet from each input flow in turn



Weighted Fair Queuing (WFQ)

- Want to share bandwidth
  - At the "bit" level, but in reality must send whole packets
- Approximate with <u>finish</u> times for each packet
  - finish (F) = arrive + length*rate; rate depends on # of flows
  - Send in order of finish times,
  - But don't preempt (stop) transmission if a new packet arrives that should go first
- More generally, assign <u>weights</u> to queues
  - This is Weighted FQ (WFQ)

WFQ example



Packet	Arrival time	Length	Finish time	Output order
A	0	8	8	1
B	5	6	11	3
C	5	10	10	2
D	8	9	20	7
E	8	8	14	4
F	10	6	16	5
G	11	10	19	6
H	20	8	28	8

Finish(i) = Max(Arrive(i), Finish(i-1)) + Length/Weight

Deficit Round Robin (Varghese, 95)

- WFQ has complexity O(log N) to pick which packet goes next
  - Disadvantage for high speed implementation
- Deficit Round Robin is a O(1) approximation
  - Fix the number of queues
  - Give them a quantum of service in round robin order
  - Skip queues until they build up enough credit for a large packet

- Gives both efficiency and fairness

QOS Framework

- QOS gives “better than best effort” guarantees. To achieve this we need to:

1. Understand what network services applications need
   → network services
2. Characterize application traffic entering the network
   → flow specifications or SLAs
3. Decide whether to accept offered traffic
   → admission control
4. Differentially process traffic in the network
   → packet scheduling

Application Needs

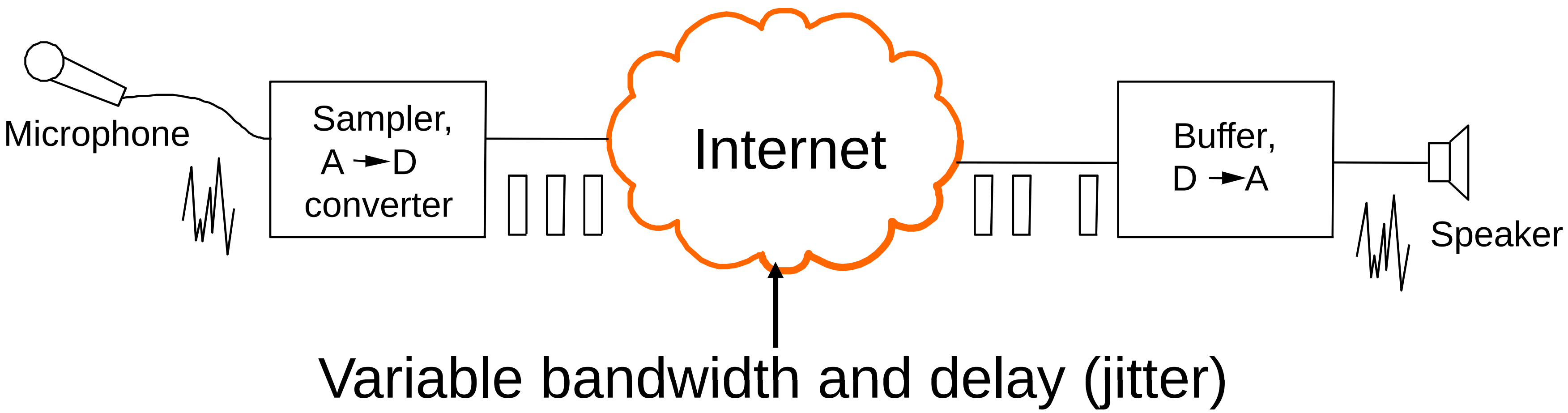
- May vary in terms of bandwidth, delay/jitter, loss

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

- Leads to notion of network services (CBR, VBR, …)

An Audio Example

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

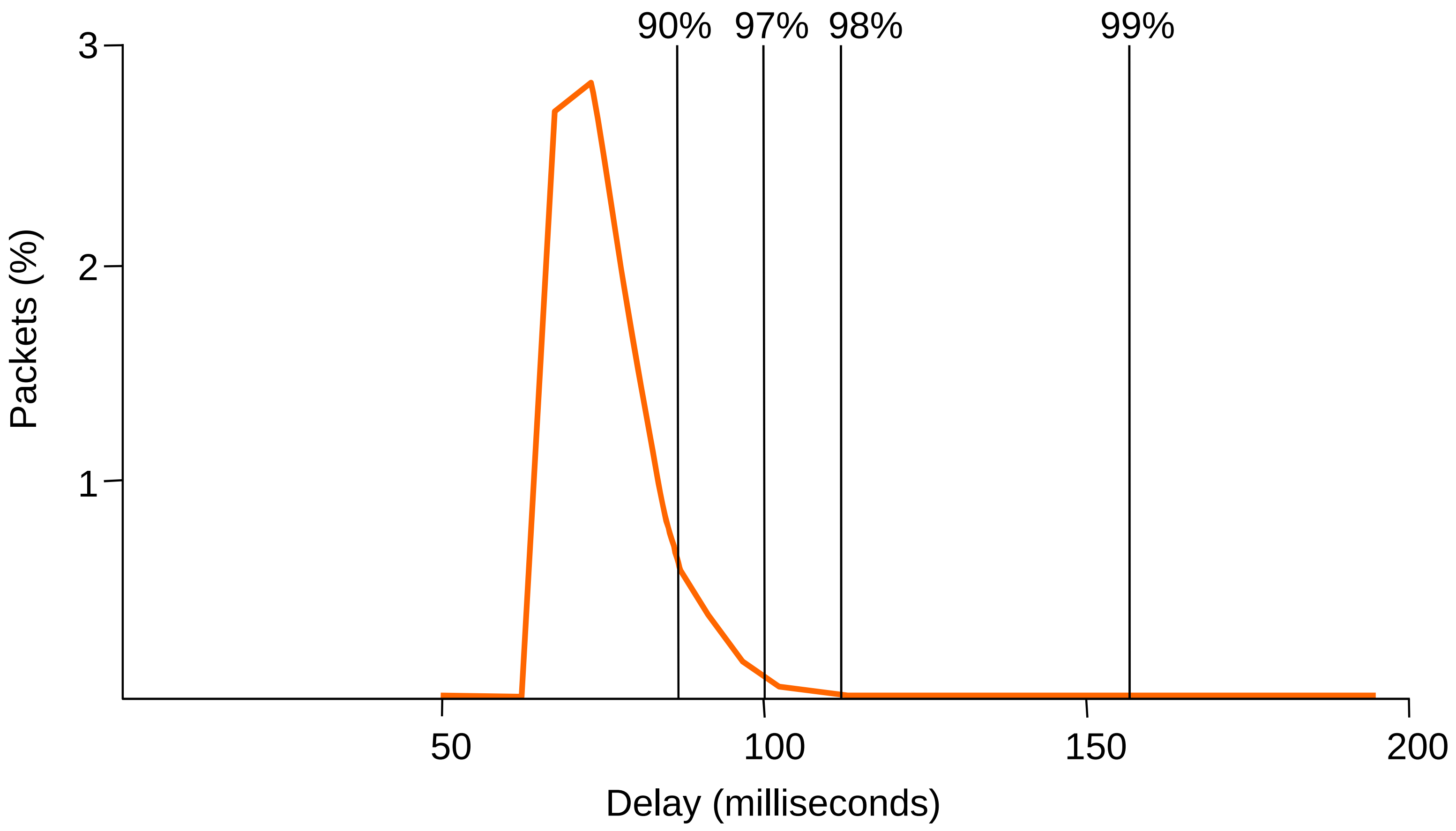


- Real-time apps need assurances from the network
- Q: What assurances does playback require?

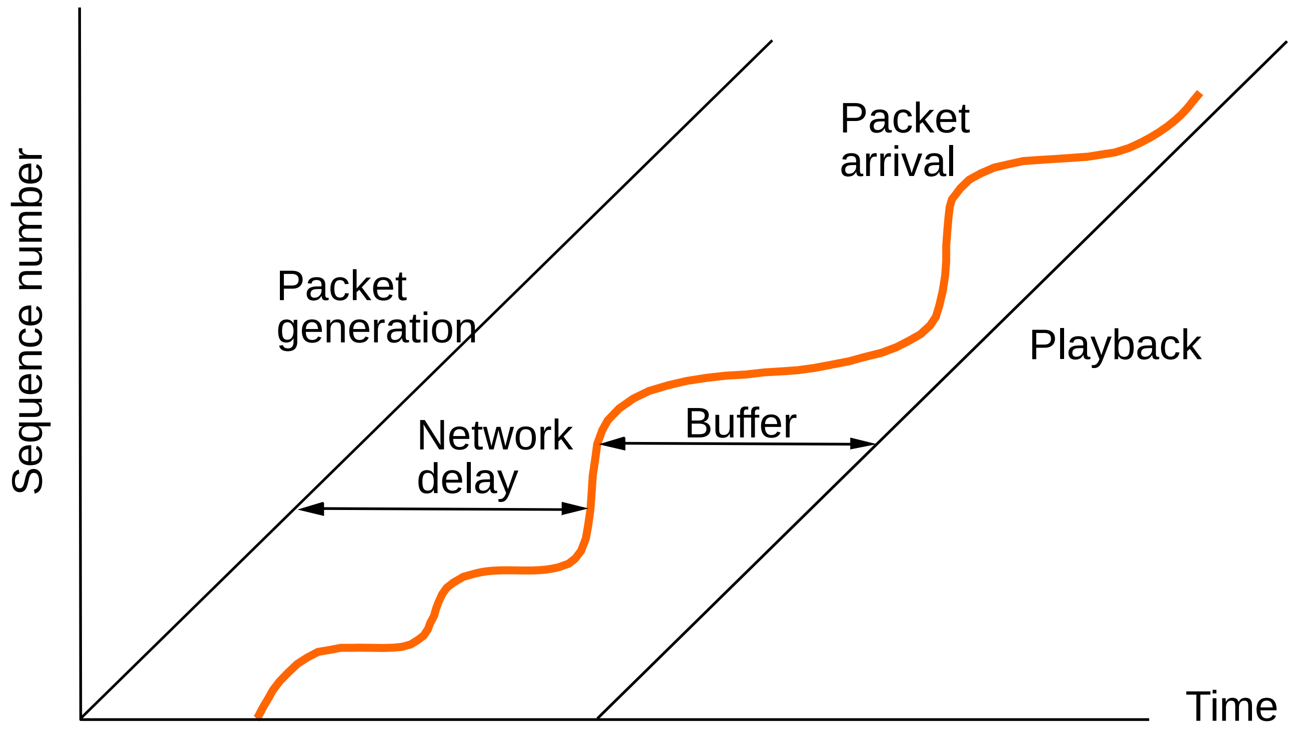
Network Support for Playback

- 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



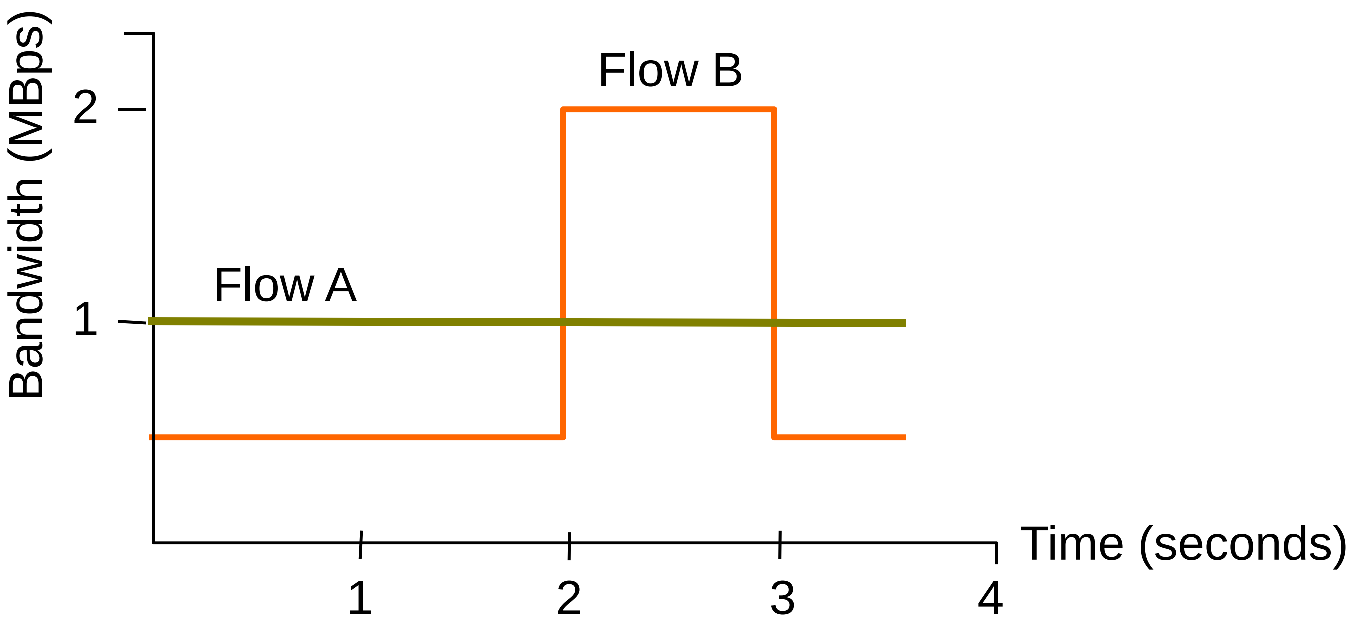
Tolerating Jitter with Buffering



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

Specifying Bandwidth Needs

- Problem: Many applications have variable bandwidth demands



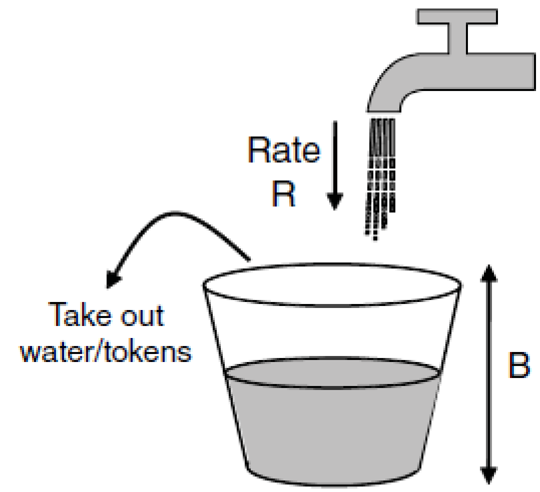
- Same average, but very different needs over time. So how do we describe bandwidth to the network?

Token Buckets

- Common, simple descriptor
- Limits long-term rate and short-term burstiness

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

- Can be used to shape or meter traffic entering network

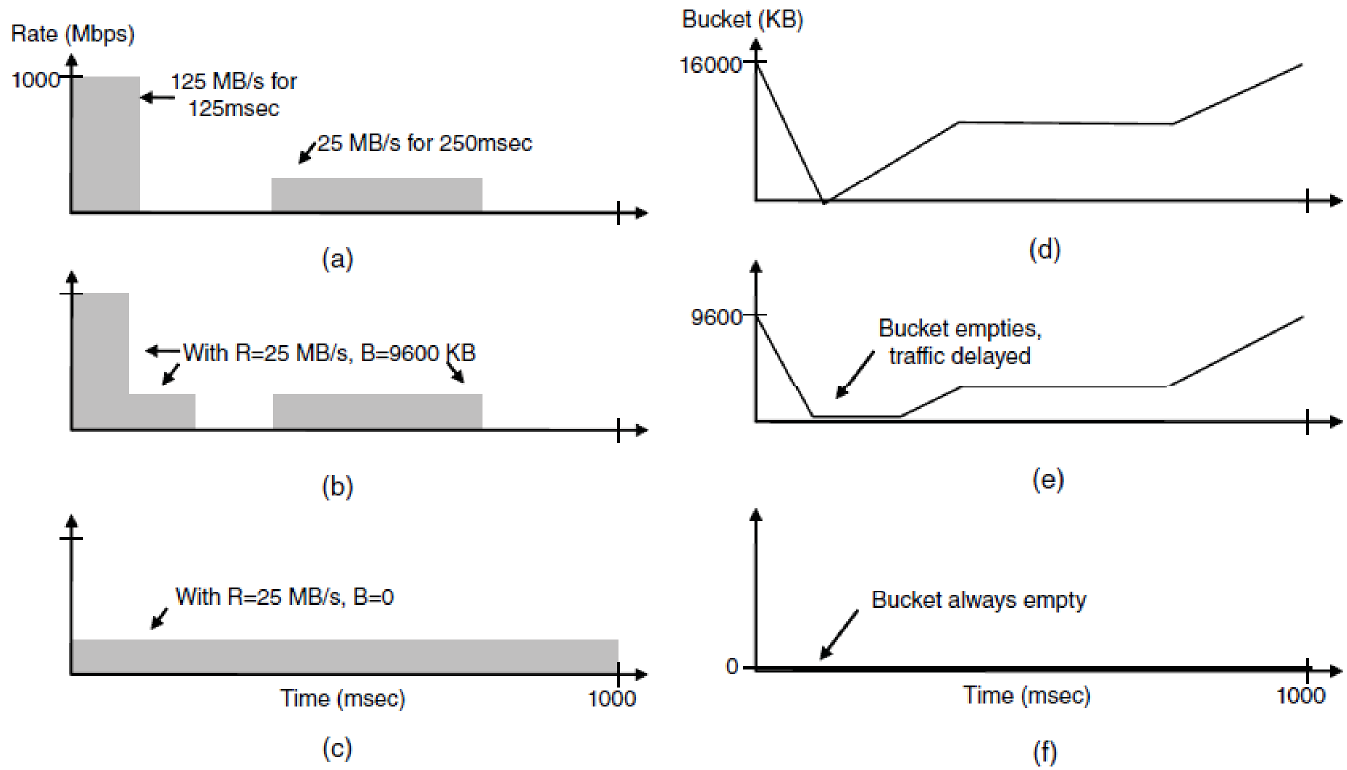


Token Bucket example

R = 200 Mbps
with B=16000KB

R = 200 Mbps
with B=9600KB

R = 200 Mbps
with B=0KB

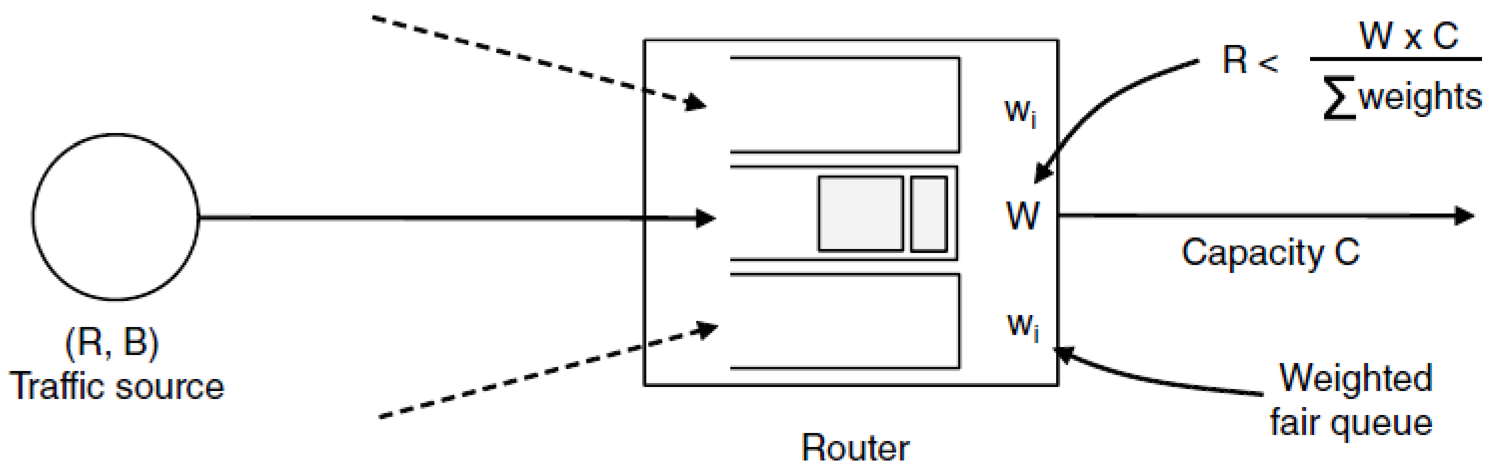


Guarantees

- How do we build the network to provide guaranteed levels of bandwidth and maximum delay/jitter to apps?
- Bearing in mind that traffic is bursty
  - Not viable to reserve resources for apps at burst levels
- Will require that we limit the traffic in the network
  - Admission control

GPS Result (Parekh & Gallagher 92)

- Condition a traffic source with R,B token bucket
- Assign weight for the flow at each WFQ router on the path to be >R/link capacity of the total weight



GPS result cont.

- Result (simplified):
  - Flow is guaranteed a bandwidth of R
  - Flow is guaranteed a delay of B/R plus the path latency

- Holds for any network topology and traffic mix!

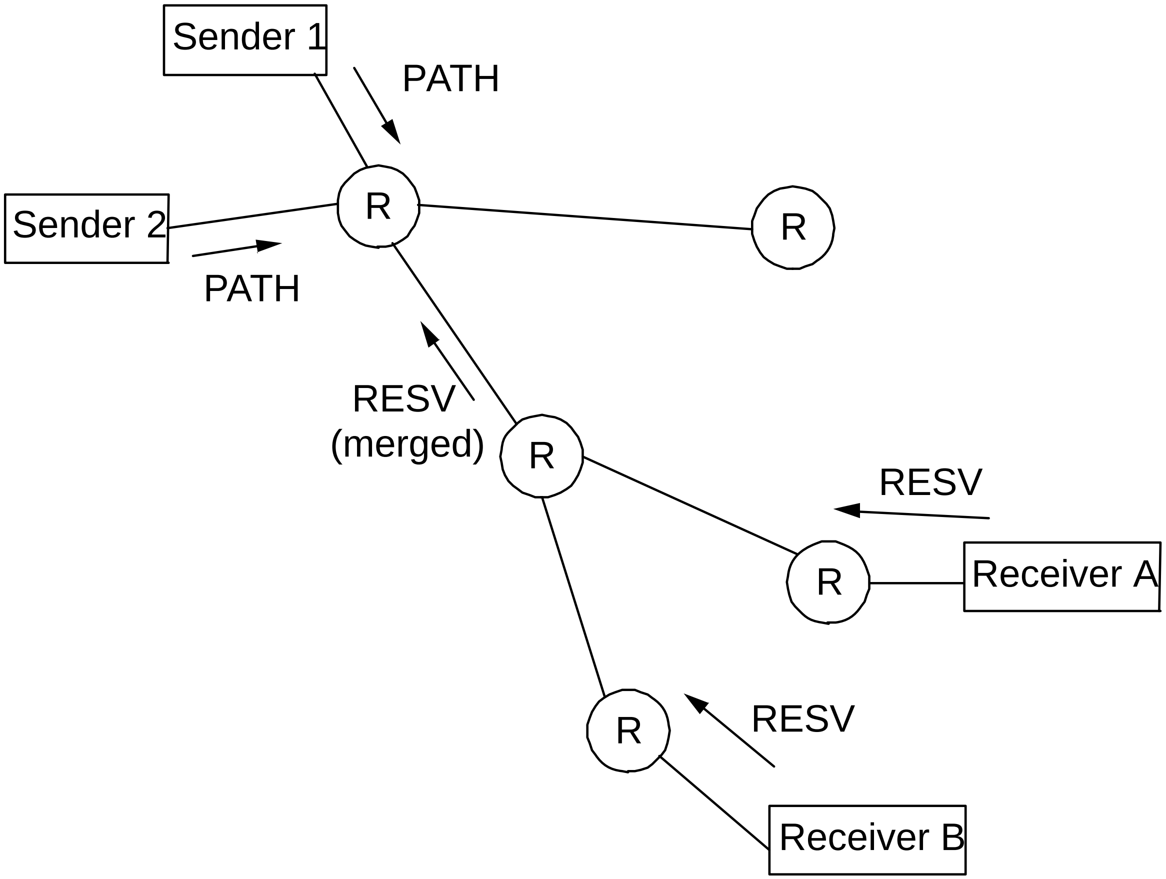
IETF Integrated Services

- 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

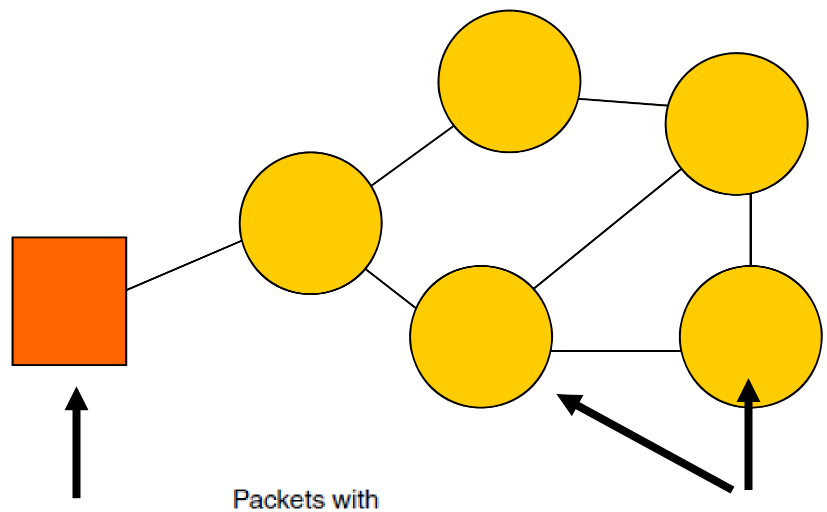
- RSVP is receiver-based to support multicast apps
- Only want to reserve resources at a router if they are sufficient along the entire path
- What if there are link failures and the route changes?
- What if there are sender/receiver failures?

IETF Differentiated Services

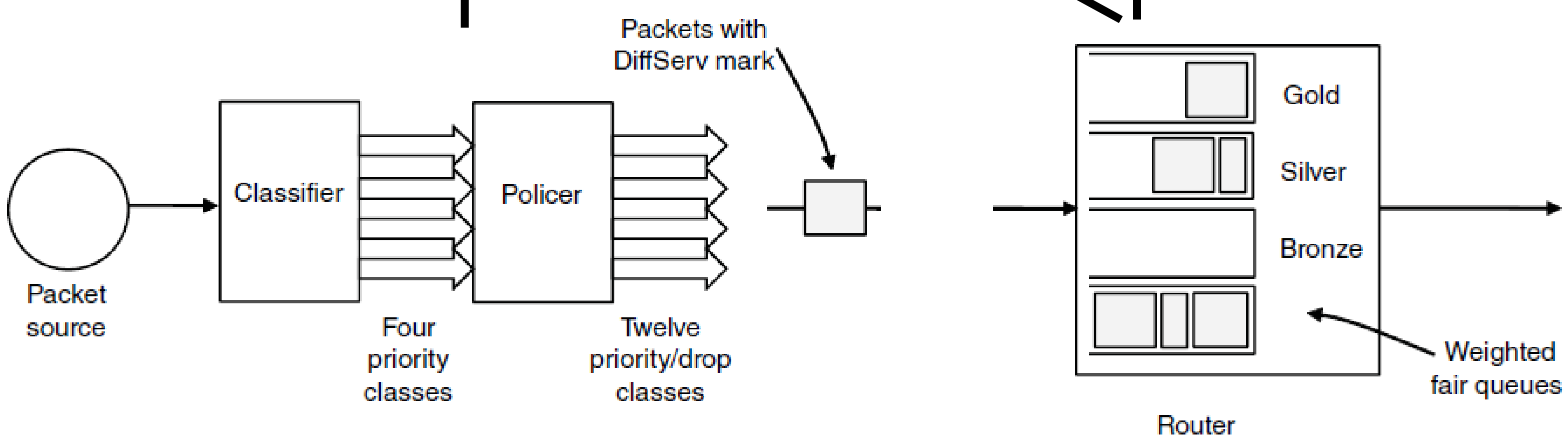
- A more coarse-grained approach to QOS
  - Packets are marked as belonging to a small set of services, e.g, premium or best-effort, using the TOS bits in the IP header
- This marking is policed at administrative boundaries
  - Your ISP marks 10Mbps (say) of your traffic as premium depending on your 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

Mark at Edge routers (per flow state, complex)



Core routers stay simple (no per-flow state, few classes)



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
  - VPNs are a natural service offering
- 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

Shenker paper – utilities

- Don't take them literally, but generally a very useful analytic thought tool (especially in the face of religious views). e.g.:
  - One multi-class network has higher utility than multiple single-class networks
  - Admission control increases utility for real-time services but not for elastic services

Shenker paper – admission control

- Somewhat of a perennial issue …
- Fair to say it has not proved "necessary" in practice
  - Many rate-adaptive applications (e.g., standard vs. high def.), even if at the user level
  - Often sufficient bandwidth in the network (e.g., access limits)
  - Only helps in a narrow regime between "too little" and "too much" bandwidth
  - Multiple classes of traffic is the big win
- Counter is that it may be important as expectations rise
  - Guarantees of non-interference by others
  - SLAs are a coarse form of admission control