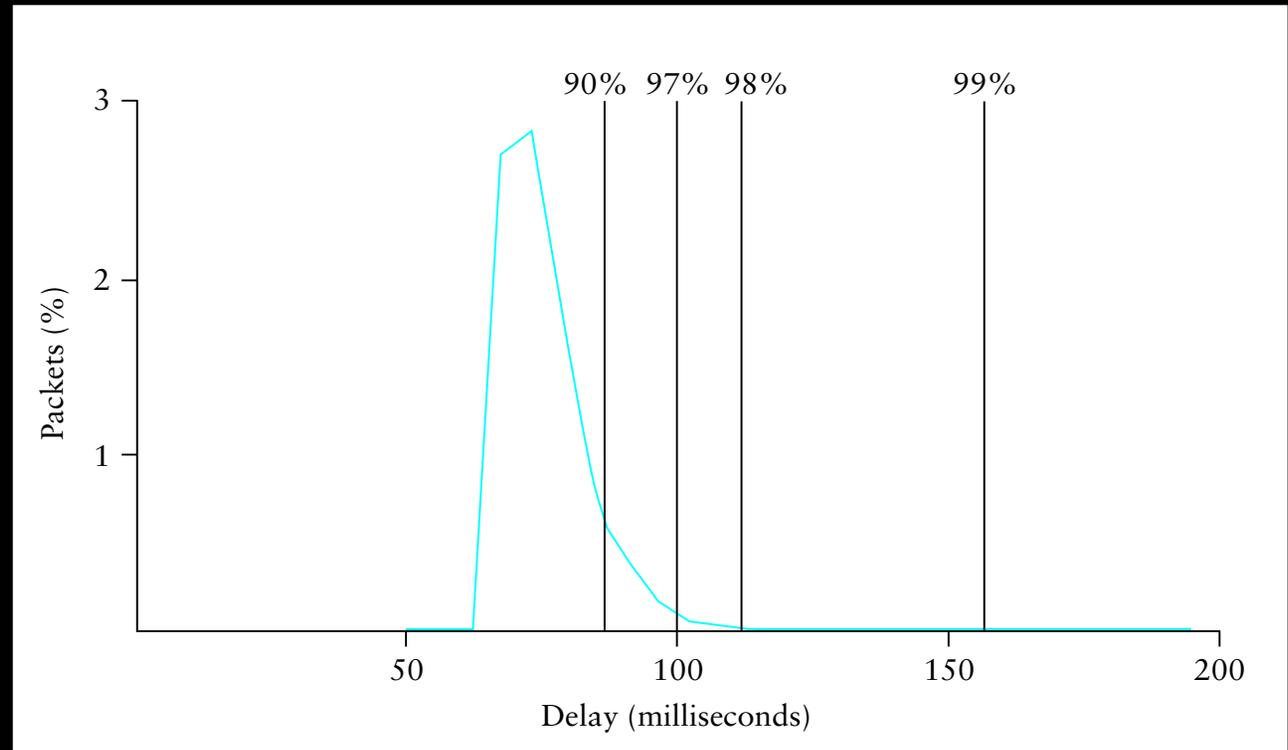


Quality of Service

QoS



- Attempts to offer certain levels of service for real-time apps, e.g., video/audio streaming, VoIP, etc.
- May not have as strong reliability requirements, but has time sensitivity

Types of services needed

- Guaranteed service
 - Intolerant applications
 - Never want late packet
 - Early packets can be buffered
- Controlled load
 - Tolerant, adaptive applications
 - E.g., reduce quality of video stream on packet loss
- Fine-grained QoS (to apps or flows)
- Course-grained QoS (to classes of flows)

Integrated Services (IntServ)

- Flowspec specified by client
- Admission control by network
- Resource reservation
 - clients and routers exchange requests, flowspecs, admission control messages, etc.
 - Calls signalling in ATM world
- Packet scheduling by routers, switch to ensure requirements of flow

Flowspecs

■ RSpec

- Describes service requested from network
- E.g., specify delay target or bound

■ TSpec

- Describes the flow's traffic characteristics
- But traffic changes and jitters, e.g., mpeg 2
- Apps send in bursts, if bursts last too long, then apps overload their average reservations
- Manage queues to prevent excessive queueing or dropping: know how bandwidth changes with time

Token Bucket Filter

- Uses token rate and bucket depth
- To send n bytes, need n tokens
- Get at r / second; can accumulate $\leq B$
- Thus, can send burst of B , but no more than r bytes / second in average

Flow A at average 1MB, always 1MB

Flow B at average 1MB, often 0.5 MB,
sometimes 2 MB

Admission Control

- Looks at RSpec, TSpec and decides if can handle w/o hurting current flows
- For guaranteed service & WFQ, decision easy
- For controlled load, may be heuristic

- Don't confuse with policing, which ensure that flow conforms to TSpec on per-packet basis
- Often closely related to policy

Reservation Protocol (RSVP)

- Uses only soft-state at routers
- Supports multicast flows and unicast
- Thus, receivers keep track of needs
- At each new soft-state request, receiver can change level of reservation

Reservation Protocol (RSVP)

- PATH request (incl. TSpec) from sender to receiver
- Receiver responds (RESV) up path with RSpec
- Each path router does admission control on RESV
- Repeat

- If router/link fails, PATH will take new path when route stabilizes, as does RESV response. Old routers' soft-state will timeout

- In multicast, RESV requests can often be merged (i.e., 100 ms delay handled by 50 ms delay path)
- Merging TSpec more complicated & app-specific

Packet Handling

- Classify each pkt with reservation / flow
 - Use src addr : port, dst addr : port, protocol #
 - Using FIFO not sufficient
- Manage queues so pkts receive QoS
 - For guaranteed service, use WFQ with each flow getting individual queue
 - For controlled load, all can be in same queue, with weight set by total amt of traffic in class
 - Some queue discipline between all different types of traffic → complex

Scalability

- Suppose 64-Kbps audio streams on 2.5 Gbps link \rightarrow 39,000 such flows
- Router needs to maintain state of each, classify, police, queue each flow, etc.
- Led to coarser-grained QoS models

Differentiated Services (DiffServ)

- Premium vs. regular packets, use header
- Premium bit often set by router at admin boundary (e.g., as customer pays for it)

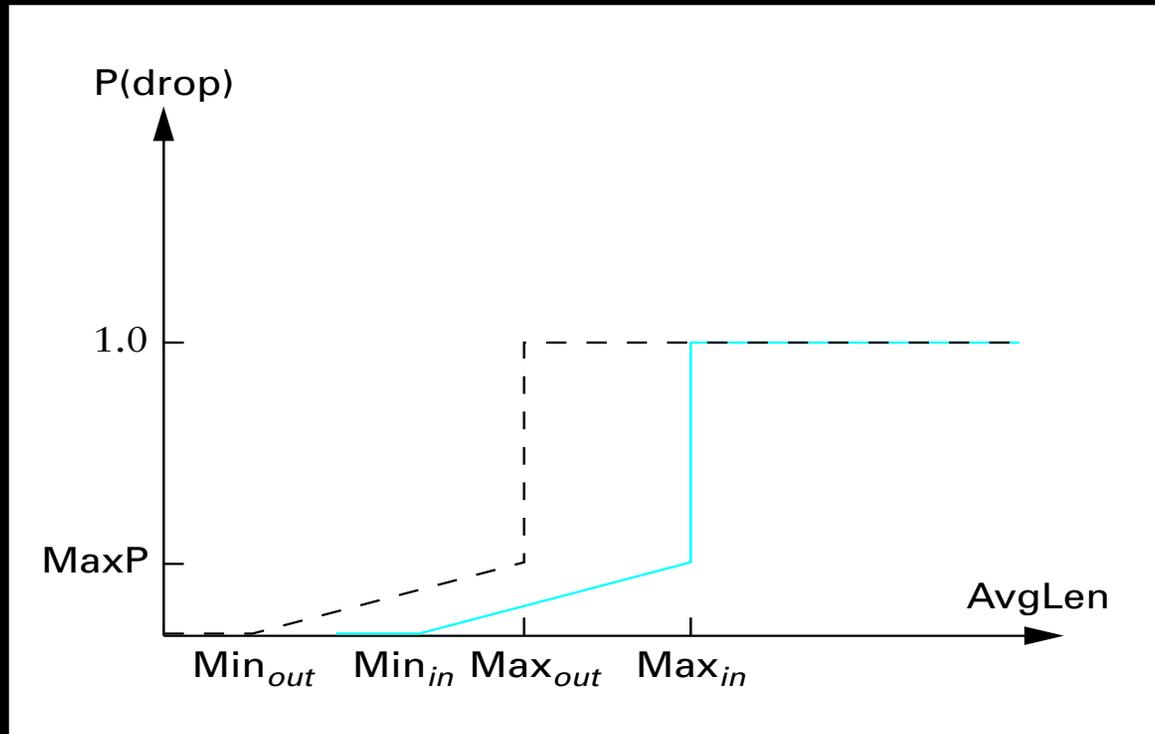
- But how to handle “premium-ness”?
 - Use 6 bits in IP header (TOS field)
 - Define per-hop behaviors (PHBs)

Per-Hop Behaviors

- Expedited Forwarding (EF)
 - Packet should get minimal delay and loss
 - Requires arrival rate of EF pkts $<$ forwarding rate
 - Border gateway routers could ensure all incoming EF pkts $<$ slower link in domain
 - Always prefer EF packets? Use WFQ and assign EF packets higher priority? (Give others some bandwidth, but EF packets may then fail)

Assured Forwarding

Similar to RED
with In and Out
(RIO)



- Customer pays for some assured traffic type. All packets within profile marked IN, all excess packets marked OUT and dropped more readily.
- Interestingly, does not reorder packets, only drops at different rates \rightarrow lack of reordering helps TCP
- Can be generalized to weighted RED (WRED)

Use to determine which queue

- “Best-effort” vs. “Premium” queues

$$B_p = W_p / (W_p + W_{be})$$

- If B_p is 0.2, premium traffic gets good performance so long as premium load only 20% of link bandwidth

RSVP	ATM
Receiver generates reservation	Sender generates connection request
Soft-state (refresh/timeout)	Hard-state (explicit delete)
Separate from route establishment	Concurrent with route establishment
QoS can change dynamically	QoS static for life of connection (mostly)
Receiver heterogeneity	Uniform QoS to all receivers

RSVP starts with connectless and tries to add,
ATM starts with virtual circuits

RSVP also has explicit goal to handle multicast

Equation-based congestion control

- TCP congestion control is end-to-end and doesn't request universal deployment
- Currently, RTPs just use UDP and steal bandwidth when TCP backoffs
- What about UDP + CC for RTPs?
 - Don't want reliability
 - Want smoother flow than TCP's saw-tooth behavior from AIMD
- TCP-friendly congestion control
 - Adapt congestion window over longer periods
 - Ensure rate adheres to model of TCP's behavior

$$\text{Rate} = 1 / (\text{RTT} * \text{sqrt} (\text{loss rate}))$$

Overview: Smart vs. Dumb networks

- RSVP protocols have richness and variety
- But deployment needs to be universal
- End-to-end solutions (TCP) entrenched and can be easily deployed

- DiffServ is middle ground between smart and dumb networks
- Currently, seeing deployment of more RTPs (like VoIP), but will course-grained QoS be sufficient?
- Will network-level QoS be widely used?