Congestion Control and Resource Allocation
How does congestion arise?

- Router receives data faster than it can send it
  - Downstream link or router congested or failed
  - Extreme traffic on one output link

- Usually a transient condition, but crisis can occur if not relieved quickly
  - Congestion cascades back into upstream routers
  - Whole sections of the network can be blocked quickly
Congestion Control and Resource Allocation

- Precise allocation is difficult
- Send as much as wanted and then congestion control
- Network model
  - Packet-switched
  - Connectionless
    - Pure connection-less
    - Connect oriented
    - Soft state
  - Service model
    - Best effort
    - QoS
- Effective and fair
Taxonomy

- Router-centric vs. Host-centric
- Reservation-based vs. Feedback-based
- Window-based vs. Rate-based

- Two common strategies
  - Best-effort
  - QoS-based
Congestion indicators

- At routers:
  - Average queue lengths
  - Number of dropped queue items
  - Ping delay to neighbors

- At senders, receivers:
  - Number of timeouts
  - RTT (measured from data send to ACK receive)
  - Ping delay
  - Receive buffer fullness
Some things to keep in mind

• Generating additional traffic to report congestion makes the situation worse – piggybacking on existing traffic is better

• Short timeouts lead to needless re-transmits, aggravating congestion

• Some parts of the network can be fine at the same time that other parts are crashing
Congestion can lead to catastrophe
Queuing (1)

- FIFO
- Tail drop
- Most widely used
  - TCP takes responsibility
- Priority queuing
  - Starving, low priority queues
  - Charge
Queuing (2)

- Problems with FIFO
  - Without routers’ help
  - Regulate source hosts

- Fair queuing
  - Separate queues for flows
  - Packet length
  - Bit-by-bit round-robin
Dealing with congestion
TCP Congestion Control

- Send packet then observe for events
- Congestion -> retransmit packets -> more congestion
- Self clocking using ACKs
Additive Increase / Multiplicative Decrease

- CongestionWindow

- Use MIN(CongestionWindow, AdvertisedWindow) in EffectiveWindow calculation

- Source perceives
  - Packet dropped
  - Multiplicative decrease
  - Additive increase
  - Why not additive increase / additive decrease
Slow Start

- Slow start
  - Exponentially increase
  - Why called “slow start”?
    - Compare to AdvertisedWindow
Fast Retransmit & Fast Recovery

- Fast Retransmit
  - TCP waits 3 duplicate ACKs before retransmission

- Fast Recovery
  - Cut congestion window in half
Congestion Avoidance
“DEC bit” for Digital Network Architecture

- Router monitor load
- When a router is nearing overload, it sets a special “congestion bit” in the packets/segments it processes.
- The congestion bit is returned to senders as part of ACKs
- When a sender sees congestion bit set, it can reduce send rate to that destination
Random Early Deletion (RED)

- Designed to work with TCP Congestion Window approach

- Router implicitly notify source
  - When router nears overload, it starts randomly dropping packets

- This triggers timeouts, which will ultimately cause congestion window size to be reduced.

- Drop probability
  - Early random drop
Source-based control (TCP-Vegas)

• No new requirements on routers, source-based

• Sending TCP determines “no-congestion” RTT (usually determined by ACK for first seg to a destination)

• Dynamic RTT (measured from ACKs) is compared to no-congestion RTT.

• As RTT increases, CW_size is reduced
Quality of Service
An example of why we need QoS: Jitter

Jitter is variation in network delay

Audio sampled at fixed interval + Variable network delay = Uneven playback + Buffering w/ release at sampling interval = Good playback

For streaming media, it is usually acceptable to delay the start of the stream (for buffering) in order to get a jitter-free playback. But we can’t do this for interactive real-time traffic (e.g., telephony).
It would be good if…

- For real-time interactive traffic, it would be good if we could ask the network to guarantee a bound on jitter
  - Example: “99% of packets will be delayed X msec”
Taxonomy

Applications
- Elastic
- Real time
  - Intolerant
  - Tolerant
    - Nonadaptive
    - Adaptive
      - Rate adaptive
      - Delay adaptive
Approaches to QoS
Terminology: Flow

• In QoS discussions, “flow” is used to describe
  – A stream of data from a given source to a given destination
  – Share the common requirement

• Circuit-switching: flow = data flowing over the circuit – all goes over the same path

• Packet-switching: the set of packets going from the source to the destination – may take different paths
Integrated Services (RSVP)

• Service Classes
  – Guaranteed service
    • Intolerant
  – Controlled load
    • Tolerant
    • Adaptive
Service Spec

- Specifies the type of service that the user is requesting

- Examples:
  - Guaranteed max delay (“90% delivered in 100ms”)
  - Jitter limitation (“95% delivered between 14 and 15ms”)
  - Bandwidth guarantee (“100Mbps end-to-end”)
  - Minimization of interference (“controlled load service”)
  - Guaranteed space in receive buffer
Traffic Spec

• The user’s statement of the flow’s parameters

• Example:
  – Token bucket rate + size
  – Peak data rate
  – Minimum packet size
  – Maximum packet size
Resource reservation

- The idea: before a flow starts, network resources are reserved to provide it the service it needs

- This is difficult to implement if you are using “pure” packet switching where packets can take any route

- Most connectionless resource-reservation networks (e.g., ATM) use a Virtual Circuit construct where all packets in a flow follow the same route
Elements of resource reservation-based QoS

1. User specifies level of service desired (“Service Spec”) and type of traffic it will submit (“Traffic Spec”)
2. Network determines if it can support the spec (“Admission Control”)
3. Network must manage resources to satisfy the agreement (“Packet Scheduling”)
4. Network ensures that the user conforms to the agreement (“Policing”)
Admission control

• Not standardized
• A general approach:
  – Request (Flow spec + Traffic spec) is forwarded through the network over the (real or virtual) circuit
  – Each router receiving the request inspects it, considering:
    • the router’s total capacity
    • commitments already in place
    • operating rules (e.g., “no more than 90% of output link bw committed”)
  – The router either:
    • Accepts the request, forwarding it on to the next router, or
    • Declines the request, sending a notice back to the sender node
    • (A variant: instead of declining, the router can reduce the request to what it can support, then forward the reduced request)
  – When the request reaches the destination node, it sends a success notice to the sender (assuming the receiver also accepts the request)
RSVP

- Soft state, 1 minute
- Receiver-oriented
  - Reverse path
- PATH messages and RESV messages are sent very 30 seconds
- Multiple senders and receivers
Packet Classifying & Scheduling

• Associate packets with the appropriate reservation

• The routers manage the resources to satisfy the agreement made during admission control process
  – Weighted Fair Queuing, etc
Not Scalable

- Each flow has states and needs to be stored!
Differentiated Services (1)

• Flow-based methods have some problems:
  – Complex
  – Advance setup required
  – Not stateless – vulnerable to crashes

• A simpler approach: fixed classes of services defined by net admin ("class-based QoS" or "Differentiated Services")
  – No setup per flow
  – Less complexity at routers
  – Stateless
Differentiated Services (2)

- Two classes
  - Premium
  - Best effort

- Who sets the premium bit?

- What does a router do?