Class of Service Design for "Triple Play" Networks

APRICOT 2005

24 February, 2005

Jeff Doyle

Senior Network Architect

Juniper Networks Professional Services

Juniper NETWORKS™

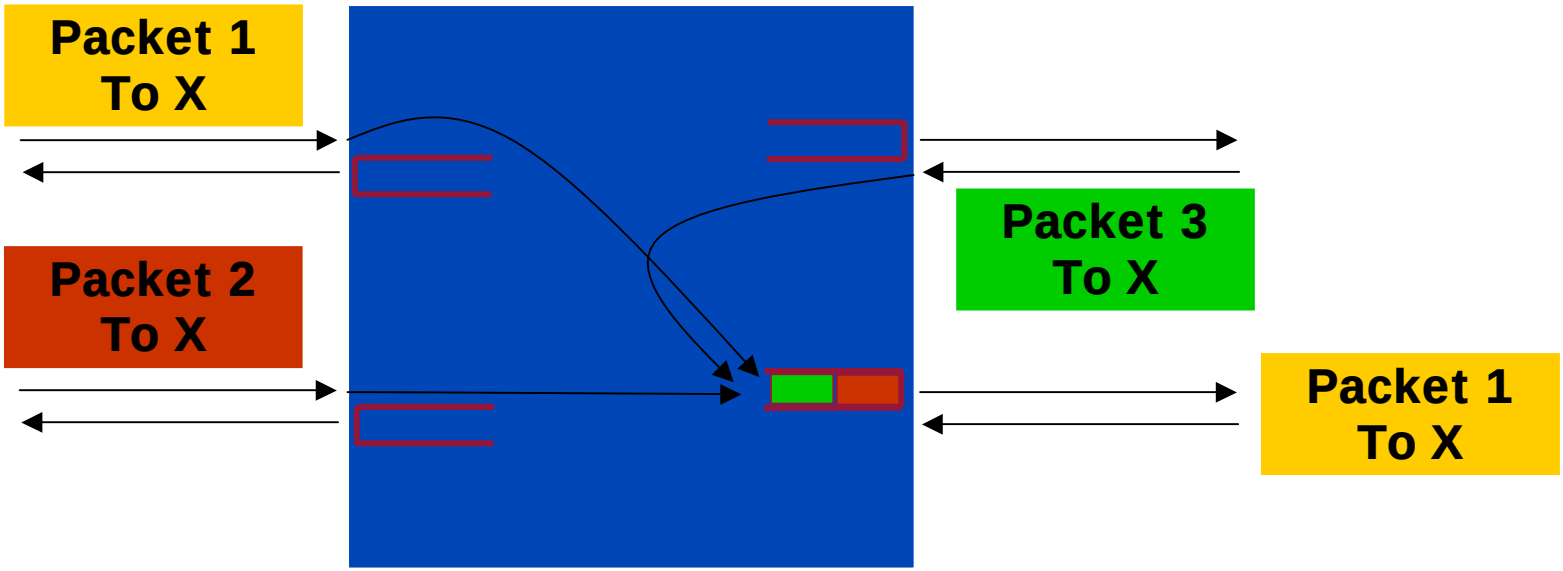
QOS Fundamentals

- QOS is still widely misunderstood
  - …or at least understood superficially
- So we'll first review some important networking fundamentals

Statistical Multiplexing

- Non-statistical multiplexing systems (e.g., TDM or WDM) – bits, bytes, frames
  - Each unit of input bandwidth paired up with unit of output bandwidth – done at provisioning time
    - Therefore, no congestion-related buffering is required
- Statistical multiplexing (statmux) – packets, frames, cells
  - Packets arrive on one port and can go out any other port – depends on the result of a lookup
    - E.g., MAC/IP address, VPI/VCI, DLCI, etc. – not on something decided at provisioning time
  - More packets might want to exit a port than the port has bandwidth
    - "Over-subscription" – offers huge economic advantages
    - Therefore statmux devices need buffers to handle congestion, though a properly provision network will usually have empty buffers

Buffers:
What Structure? How Big?



- Simplest buffer is a single FIFO (first-in-first-out) queue per output port
- Buffer size depends on link speed and higher-layer protocol
  - Tradition for TCP is that the buffer size should be the bandwidth of the port times the longest round-trip-time flow (“bandwidth-delay product”)
  - E.g., ~32MB required for a 100ms RTT flow over an STM-16

Buffers: What Structure?

- A single FIFO queue results in best effort
  - All packets treated with the same priority
  - If a buffer is full and another packet arrives, that new packet is silently dropped
  - Hosts have to detect and react to loss
- “Best effort” is not an insult
  - It allowed the Internet to get to critical mass
- But there are technical and commercial requirements that require IP to move past only “best effort”
  - Internet versus “Triple Play” or Multiservice

QOS

- QOS means not treating all data the same
  - i.e., not just best effort
  - “Managed Unfairness”
- Goal is to offer different traffic classes different
  - Bandwidth/throughput
  - Delay
  - Jitter
  - Loss

Throughput

- Throughput is easy to measure for TDM
  - It's the bandwidth of the channel (e.g., a DS-3 is 45Mbps)
- Throughput is harder to measure for IP since it's any-to-any and statmux
  - Throughput of what? Rate across an access circuit? an application flow? host-to-host aggregate flows? network-to-network aggregate flows?
  - Routers aren't the only part of the system
    - What if a host is a bottleneck?
    - How good are the two TCP implementations?
    - Is routing stable?

Throughput (cont.)

- Routers affect throughput by allocating different bandwidth to different traffic classes
- In best-effort routers don't actively do anything
  - Assume that TCP detects/reacts to loss in a way that results in fairness
- Several ways of allocating bandwidth to traffic classes
  - E.g., strict priority of one class over others
  - E.g., prioritize one class, but cap it to prevent starvation
  - E.g., equal prioritization but different bandwidths
  - Hybrids of the above
  - All of the above require that routers have multiple queues

Delay: Contributors

- Latency within the equipment along the physical path
  - E.g., time in a SONET multiplexer
  - This is usually measured in 1s or low 10s of microseconds
- Propagation delay along physical links
  - This depends on distance
  - One-way delay between US east and west coast is 30-35 milliseconds
- Queuing delay in statistical multiplexing devices
  - This depends on queue occupancy in those devices
  - Best case is close to 0
  - Worst case is the sum of maximum queuing delays in every statmux device in the path

Delay

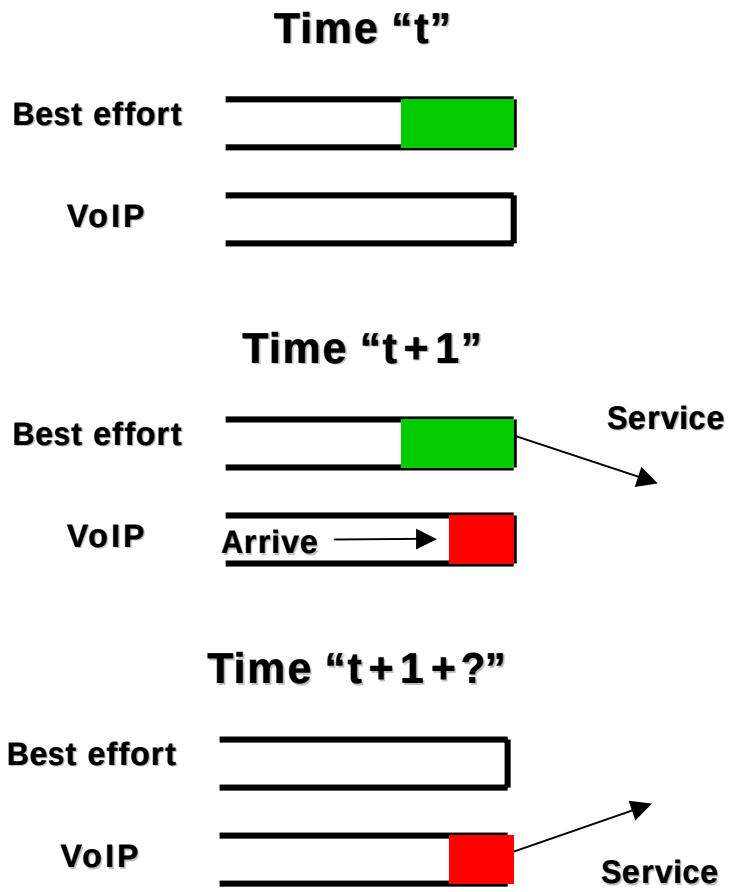
- In a well functioning packet network, propagation delay is the major source of delay, by several orders of magnitude
- This means that delay is usually something that a piece of network equipment can’t change

Jitter

- Jitter is the variation in delay over time
- TDM devices can contribute to jitter, but the amount is so small that it can be ignored
- The primary contributor to jitter is the variability of queuing delay over time
  - Variability in intra-equipment latency is a second-order contributor

Jitter Example

- Best-effort queue starts being serviced right before a VoIP packet arrives
  - VoIP packet has to wait for best-effort packet
  - Wait time depends on size of green packet
  - Hence ATM's small cell size
- This happens hop-by-hop



Serialization Delays by Link Speed and Packet Size

	DS-1	DS-3	OC-3	OC-12	OC-48	OC-192
40	0.2073	0.0072	0.0021	0.0005	0.0001	0.0000
256	1.3264	0.0458	0.0132	0.0033	0.0008	0.0002
320	1.6580	0.0572	0.0165	0.0041	0.0010	0.0003
512	2.6528	0.0916	0.0264	0.0066	0.0016	0.0004
1500	7.7720	0.2682	0.0774	0.0193	0.0048	0.0012
4470	23.1606	0.7994	0.2307	0.0575	0.0144	0.0036
9180	47.5648	1.6416	0.4738	0.1181	0.0295	0.0074

- Conclusion: Jitter matters more on slower links, and bigger packets hurt most

Jitter: When Does It Matter?

- Traditional Internet applications don't really care about jitter
  - TCP treats round-trip-time as a dynamically changing value
- Some applications have a problem with jitter
  - For interactive voice, jitter can result in "jerky" playback
    - Jitter can be smoothed with a play-out buffer
    - But too much jitter requires a long play-out buffer which can frustrate humans and make the service unusable
  - For circuit emulation, excess jitter can cause hard circuit failures in the TDM domain
    - Therefore, supporting such services requires a properly provisioning network of routers with adequate QOS mechanisms

Loss

- Packets can be lost in two primary ways
  - Congestion – a packet wants to go out a certain port but the associated transmit queue is 100% full
  - Errors – a packet gets corrupted such that some hop in the path needs to drop the packet
- In practice, packet loss almost always means congestion
  - TCP explicitly makes this assumption
  - Note: This assumption isn't so good for wireless

Loss (cont.)

- Question: Is loss bad?
- Answer: Not always
  - TCP works by finding the maximum bandwidth it can use while trying not to cause sustained congestion
  - Start by transmitting slowly then increasing the transmission rate until a drop is detected
    - Since drops mean congestion, TCP will react by slowing down “some”
    - Over time, TCP will reach an equilibrium of maximum bandwidth without congestion; multiple TCPs doing this in parallel results in fair allocation of bottleneck bandwidth
- TCP needs to see loss to do its job
  - But sustained congestion causes TCP problems

Loss (cont.)

- How is this related to QOS?
  - Throughput commitments between ingress/egress port pairs is way easier to offer than from an ingress port to “anywhere”
    - Specifically, ensure the “committed” traffic has adequate allocated bandwidth along the path
    - So throughput commitments for, e.g., VPN services make more sense than for an Internet service
  - What to do with traffic sent along that path above the agreed-upon rate is a policy question
    - Drop it on ingress
    - Pass it on with increased drop probability
    - Buffer and “shape” it on ingress

Loss (cont.)

- So far we've only talked about routers doing passive queue management and implicit congestion notification
  - If a queue fills up then start dropping and assume hosts will notice the drop
- This passive approach ("tail drop") interacts negatively with TCP
- Alternative: active queue management
  - RED (Random Early Detection) detects incipient congestion and starts dropping "a little bit" in an attempt to prevent filling
- Alternative: explicit congestion notification
  - Mark a packet rather than drop it, hence indicating congestion without requiring retransmission and without having to wait for a retransmission timer to time-out
- Note that RED and ECN are useful independent of QOS

How is QoS achieved?

- Classification
- Policing/Marking
- Shaping
- Buffer management
- RED – Random Early Discard

Where to apply QoS

1. Where you want to limit loss or latency
   - Cores and high bandwidth links are not the issue
   - “End-to-end QoS” not the right mind set
   - More like, “Find the weakest link”
2. Where you want to incur loss :^)
   - Imposing per subscriber bandwidth limits
   - Important to the provider business model

CoS Requirements

- How to classify and queue voice, video, data, and network control traffic
- Remember, queuing only helps with occasional congestion
- Only remedy to consistent congestion is more bandwidth

CoS Requirements: Voice

- Low jitter is the primary requirement
  - 30ms max
- Low latency is also important
  - 150ms max
- Reasonably resilient to packet loss
- Typical packet size < 100B

CoS Requirements: Real-Time Video

- Low packet loss is primary requirement
- Jitter is less of a concern
  - Due to playback buffers in receiving video system
  - Up to 150ms acceptable
- 150ms max. one-way latency
- Typical video packets are large (>500B)

CoS Requirements: Network Control

- Routing protocol packets
- Maybe VoIP signaling, etc.
- Should normally be <1% of bandwidth on any link
- Very resilient to packet loss
  - But, should receive highest priority in times of congestion
  - NC queue should never risk “starvation”

CoS Requirements: Data

- “Everything else”
- Highly resilient to packet loss, latency, jitter
- Highly variable packet sizes

Example Queuing Design

Class	Buffer Size	Transmit Rate	Priority
Network Control	5%	5%	high
Video	40%	25%	high
Voice	15%	30%	strict-high
Data	40%	remainder	low

RED Requirements: Voice

- Most voice traffic is UDP, not TCP
  - Therefore RED is of limited use
  - Nevertheless, it is common to configure RED for voice
- Typical jitter budget in voice media gateways ~5ms
  - Packets received beyond this period, as determined by timestamp in RTP header, are dropped
  - Space left by dropped out-of-budget packets filled with comfort noise
  - Therefore a very aggressive drop profile drops packets that are likely to be dropped anyway
- Juniper Networks routers have max. latency of 200ms under severe congestion
  - 100% drop probability at 25% queue fill reduces this max. latency to 50ms.

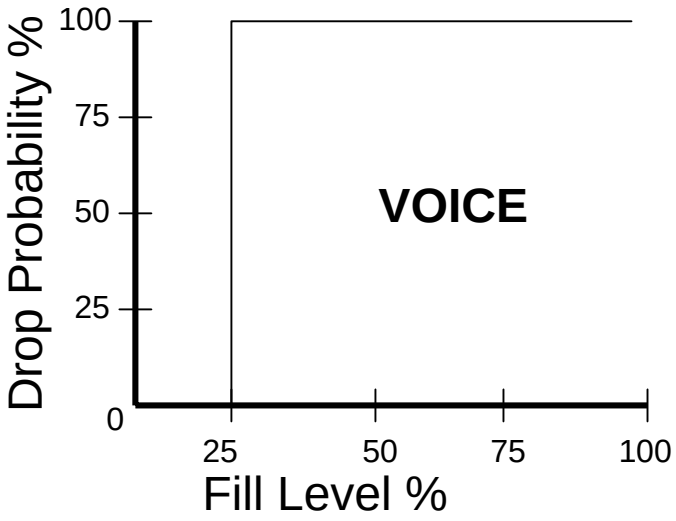
RED Requirements: Video

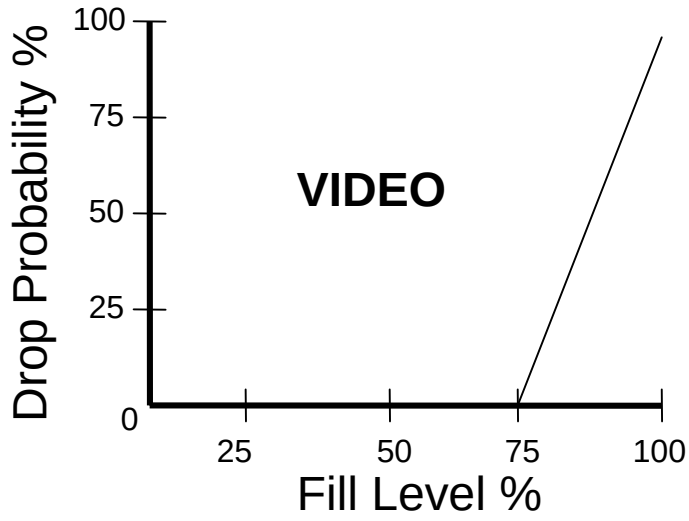
- Sensitive to packet loss
- Jitter, latency matter less
- Therefore, lenient drop profile is required

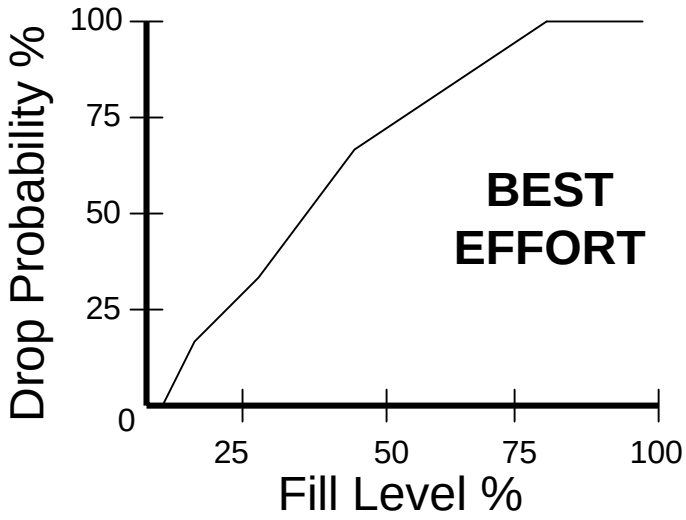
RED Requirements: BE and NC

- NC:
  - RED is undesirable
  - Therefore no drop profile configured
- Data (Best Effort):
  - Moderate drop profile desired here
  - Interpolate option smooths effect of RED, preventing abrupt changes of drop behavior at finite fill points

Example Drop Profiles*







* Using “interpolate” option

Thank You

http://www.juniper.net