17: Queue Management

Mark Handley

Queuing

- The primary purpose of a queue in an IP router is to smooth out bursty arrivals, so that the network utilization can be high.

- But queues add delay and cause jitter.
  - Delay is the enemy of real-time network traffic.
  - Jitter is turned into delay at the receiver’s playout buffer.
  - Understanding and controlling network queues is key to getting good performance from networked multimedia.
TCP and Queues

- TCP needs one delay-bandwidth product of buffer space at the bottleneck link for a TCP flow to fill the link and achieve 100% utilization.

- Thus, when everything is configured correctly, the peak delay is twice the underlying network delay.
  - Links are often overbuffered, because the actual RTT is unknown to the link operator.
  - Real-time applications see the difference between peak and min as jitter, and smooth to peak delay.
Two TCP Flows (Effects of Phase)

Multiple TCP flows and Queues

- If multiple flows all back-off in phase, the router still needs a delay-bandwidth product of buffering.
- If multiple flows back-off out of phase, high utilization can be maintained with smaller queues.
  - How to keep the flows out of phase?
Active Queue Management

Goals of Active Queue Management

- The primary goal: Controlling average queuing delay, while still maintaining high link utilization.
- Secondary goals:
  - Improving fairness (e.g., by reducing biases against bursty low-bandwidth flows).
  - Reducing unnecessary packet drops.
  - Reducing global synchronization (i.e., for environments with small-scale statistical multiplexing).
  - Accommodating transient congestion (lasting less than a round-trip time).
Random Early Detection (RED)

- As queue builds up, randomly drop or mark packets with increasing probability (before queue gets full).
- Advantages:
  - Lower average queuing delay.
  - Avoids penalizing streams with large bursts.
  - Desynchronizes co-existing flows.

**RED Algorithm**

```plaintext
for each packet arrival
    calculate the new average queue size $q_{avg}$
    if $min_{th} < q_{avg} < max_{th}$
        calculate probability $p_a$
        with probability $p_a$:
            mark/drop the arriving packet
    else if $max_{th} < q_{avg}$
        drop the arriving packet
```

**Variables:**
- $q_{avg}$: average queue size
- $p_a$: packet marking or dropping probability

**Parameters:**
- $min_{th}$: minimum threshold for queue
- $max_{th}$: maximum threshold for queue
The argument for using the *average* queue size in AQM

To be robust against transient bursts:

- When there is a transient burst, to drop just enough packets for end-to-end congestion control to come into play.
- To avoid biases against bursty low-bandwidth flows.
- To avoid unnecessary packet drops from the transient burst of a TCP connection slow-starting.
The problem with RED

- Parameter sensitivity
  - How to set $\text{min}_{th}$, $\text{max}_{th}$ and $\text{max}_p$?
- Goal is to maintain mean queue size below the midpoint between $\text{min}_{th}$ and $\text{max}_{th}$ in times of normal congestion.
  - $\text{max}_{th}$ needs to be significantly below the maximum queue size, because short-term transients peak well above the average.
  - $\text{max}_p$ primarily determines the drop rate. Needs to be significantly higher than the drop rate required to keep the flows under control.
- In reality it’s hard to set the parameters robustly, even if you know what you’re doing.

RED Drop Probabilities (Gentle Mode)
Other AQM schemes.

- Adaptive RED (ARED)
- Proportional Integral (PI)
- Virtual Queue (VQ)
- Random Exponential Marking (REM)
- Dynamic-RED (DRED)
- Blue
- Many other variants... (a lot of PhDs in this area!)

Explicit Congestion Notification
Explicit Congestion Notification (ECN)

- Standard TCP:
  - Losses needed to detect congestion
  - Wasteful and unnecessary
- ECN:
  - Routers mark packets instead of dropping them.
  - Receiver returns marks to sender in ACK packets.
  - Sender adjusts its window as it would have done if the packet had been dropped.
- Advantages:
  - Bandwidth up to bottleneck not wasted.
  - No delay imposed by retransmission.

ECN: Backwards Compatibility

- When congestion experienced, a bit in the IP header indicates if both hosts implement ECN.
  - If they do, router marks packet.
  - If they don’t, router drops packet.
Explicit Congestion Notification Codepoints

<table>
<thead>
<tr>
<th>ECT</th>
<th>CE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Not-ECT</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>ECT(1) (used as an ECN nonce)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>ECT(0)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>CE</td>
</tr>
</tbody>
</table>

The ECT and CE bits defined in RFC 2481.

- **ECT**: ECN-Capable Transport
- **CE**: Congestion Experienced.

ECN and AQM

- ECN is only useful if the queue isn’t full.
  - Otherwise the router has to drop the packet whether it wants to or not.
- An active queue management scheme like RED is needed to set the ECN *Congestion Experienced* bit before the queue fills up.
Summary

Multimedia traffic has tight delay constraints.

- Droptail queuing gives unnecessarily large queuing delays if good utilization is needed.
- Packet loss as a signal of congestion hurts real-time traffic much more than it hurts file transfer.
  - No time to retransmit.

AQM combined with ECN can give low loss, low-ish delay, moderate jitter service.

- No admission control or charging needed.
- But no guarantees either - it’s still best-effort.

References

- D. Wetherall, D. Ely, and N. Spring, Robust ECN Signaling with Nonces, RFC 3540, June 2003