A Delay Based Congestion Control Candidate

Stefan Holmer

Details in: draft-alvestrand-rtcweb-congestion
Problem

At what bitrate can a sender transmit data while still keeping the end-to-end delay low and avoiding packet loss?
Model - Queuing

- Packets transmitted at a too high rate will be queued.
- The one-way delay increases with queue length.
- If a queue is full, packets are dropped.
  - Other schemes exist

![Queue diagram]
Modeling

Over-using:
The one-way delay $d_i$ increases as queues are being filled.
\[ \Delta_i = d_i - d_{i-1} > 0 \]

Inversely proportional to the capacity:
\[ \Delta_i = \frac{dL_i}{C_{\text{min}}} + w_i = \frac{dL_i}{C_{\text{min}}} + m_i + v_i \]
\[ E\{v\} = 0 \]
Estimation/Detection

- Estimate the slope and offset.
- Most filters will do the job. We chose the Kalman filter.
  - Adaptive, handles random jitter as noise.
- Measure incoming rate when offset > threshold.
- Adjust target rate to some factor of the incoming rate.
Signaling

- Both estimation and control at the receiver.
- Transmit bandwidth estimates to the sender.
- The sender chooses to transmit at any rate < BW estimate. Employs own simpler algorithm to avoid problems of lost feedback messages.
Results

- Been run through a set of emulated network scenarios with success.
- Used in the wild with Chrome's WebRTC implementation.
- Working on getting some performance numbers.
TODO

- AQM/ECN.
- Packet-loss based detection at the receiver.
- Multiple streams.