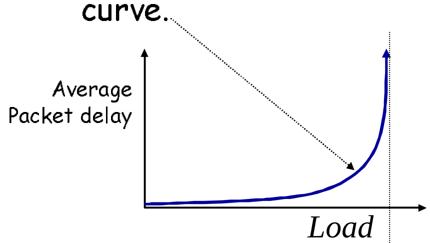
CS244a: An Introduction to Computer Networks

Handout 8: Congestion Avoidance and Active Queue Management

Nick McKeown
Professor of Electrical Engineering and Computer Science, Stanford University
nickm@stanford.edu
http://www.stanford.edu/~nickm

Congestion Avoidance

- TCP reacts to congestion *after* it takes place. The data rate changes rapidly and the system is barely stable (or is even unstable).
- Can we *predict* when congestion is about to happen and avoid it? E.g. by detecting the knee of the curve.

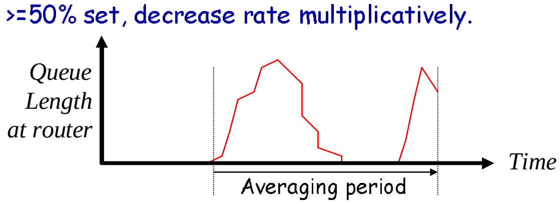


Congestion Avoidance Schemes

- Router-based Congestion Avoidance:
  - DECbit:
    - Routers explicitly notify sources about congestion.
  - Random Early Detection (RED):
    - Routers implicitly notify sources by dropping packets.
    - RED drops packets at random, and as a function of the level of congestion.
- Host-based Congestion Avoidance
  - Source monitors changes in RTT to detect onset of congestion.
  - A variety of algorithms are described in Section 6.4.3.

DECbit

- Each packet has a "Congestion Notification" bit called the DECbit in its header.
- If any router on the path is congested, it sets the DECbit.
  - Set if average queue length >= 1 packet, averaged since the start of the previous busy cycle.
- To notify the source, the destination copies DECbit into ACK packets.
- Source adjusts rate to avoid congestion.
  - Counts fraction of DECbits set in each window.
  - If <50% set, increase rate additively.
  - If >=50% set, decrease rate multiplicatively.



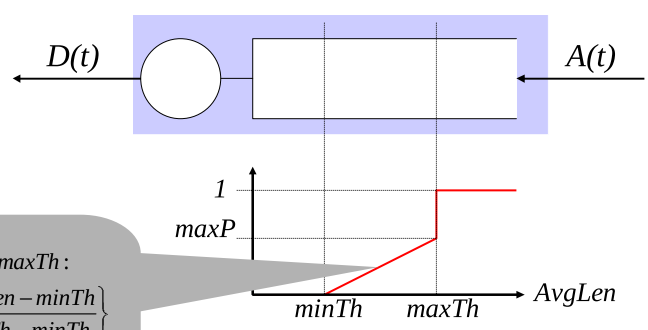
Random Early Detection (RED)

- RED is based on DECbit, and was designed to work well with TCP.
- RED implicitly notifies sender by dropping packets.
- Drop probability is increased as the *average* queue length increases.
- (Geometric) moving average of the queue length is used so as to detect long term congestion, yet allow short term bursts to arrive.

$$AvgLen_{n+1} = (1-\alpha) \times AvgLen_n + \alpha \times Length_n$$

i.e. $$AvgLen_{n+1} = \sum_{i=1}^{n} Length_i (\alpha)(1-\alpha)^{n-i}$$

RED Drop Probabilities



If $minTh < AvgLen < maxTh$:

$$\hat{p}_{AvgLen} = maxP \left\{ \frac{AvgLen - minTh}{maxTh - minTh} \right\}$$

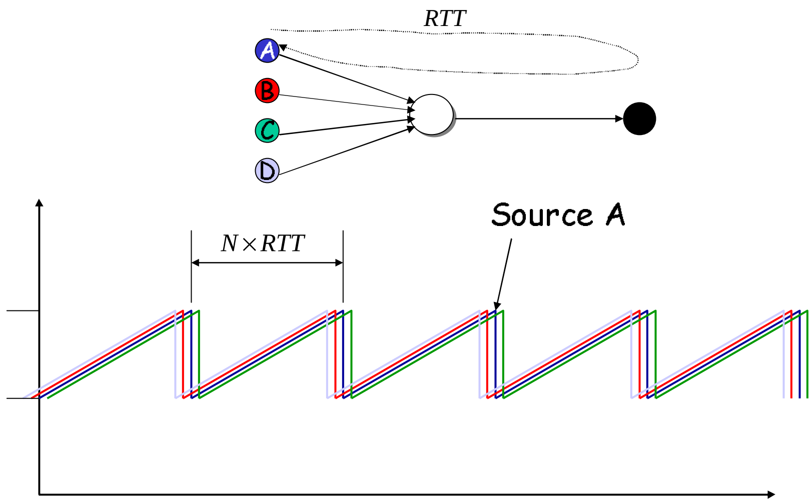
$$\Pr(\textbf{Drop Packet}) = \frac{\hat{p}_{AvgLen}}{1 - count \times \hat{p}_{AvgLen}}$$

count counts how long we've been in $minTh < AvgLen < maxTh$ since we last dropped a packet. i.e. drops are spaced out in time, reducing likelihood of re-entering slow-start.

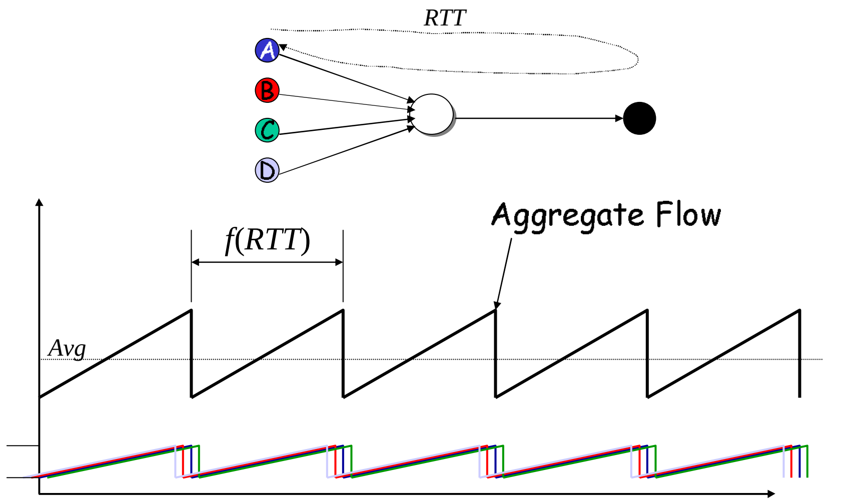
Properties of RED

- Drops packets before queue is full, in the hope of reducing the rates of some flows.
- Drops packet for each flow *roughly* in proportion to its rate.
- Drops are spaced out in time.
- Because it uses average queue length, RED is tolerant of bursts.
- Random drops hopefully desynchronize TCP sources.

Synchronization of sources

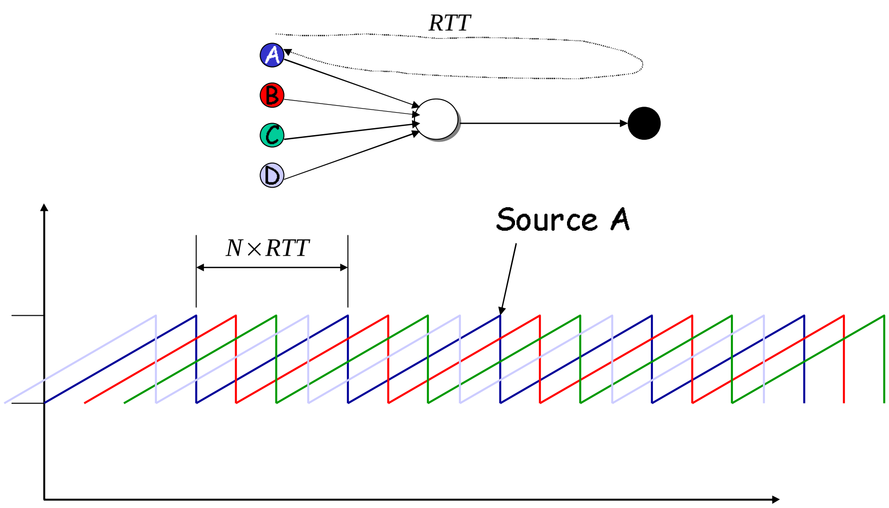


Synchronization of sources



Winter 2008 CS244a Handout 8 9

Desynchronized sources



Winter 2008 CS244a Handout 8 10

Desynchronized sources