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 DECbit set in each window.
  - If <50% set, increase rate additively.
  - If >=50% set, decrease rate multiplicatively.

![Queue Length at router](image-url)
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.

\[
\text{AvgLen}_{n+1} = (1-\alpha) \times \text{AvgLen}_n + \alpha \times \text{Length}_n
\]

i.e. \( \text{AvgLen}_{n+1} = \sum_{i=1}^{n} \text{Length}(\alpha)(1-\alpha)^{n-i} \)

---

RED Drop Probabilities

\[
\text{If minTh < AvgLen < maxTh:} \\
\hat{p}_{\text{AvgLen}} = \maxP \left[ \frac{\text{AvgLen} - \text{minTh}}{\text{maxTh} - \text{minTh}} \right] \\
\Pr(\text{Drop Packet}) = \frac{\hat{p}_{\text{AvgLen}}}{1 - \text{count} \times \hat{p}_{\text{AvgLen}}}
\]

\text{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

A synchronization diagram showing the RTT and the delay caused by $N \times RTT$. The sources are labeled A, B, C, and D, with Source A being the focal point of the diagram.
Synchronization of sources

Desynchronized sources

Winter 2008 CS244a Handout B
Desynchronized sources

RTT

Avg

N × RTT

Aggregate Flow

Winter 2008 CS244a Handout B