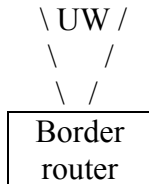## TCP Congestion Control

TCP is not good for short flows.

Definition: TCP friendliness – does a protocol back off when a loss occurs?

TCP is not TCP friendly.

Analogy: elephants vs. mice. Elephants are data intensive connections and mice are short connections. The elephants step on the mice because the data intensive connections get most of the bandwidth. The mice go through slow start each time, rarely getting to the point where they get their fair share of the bandwidth.

Congestion gateway

```
\ UW /
 \    /          The border router should know what congestion there is between
  \  /           the border router and Yahoo based on past connections to Yahoo.
┌─────────┐
│ Border  │
│ router  │
└─────────┘
```

Should control packets be treated differently than data packets?

In the Internet today, every packet is equal. Maybe there should be a bit set for BGP packets to give them priority. This isn't happening today because of legacy issues.

There is one proposal that says that slow start should start at four packets instead of one packet. This would cause the shorter connections to startup faster and acquire a fairer share of bandwidth.

A loss at the beginning of a TCP connection can really reduce throughput for a shorter connection.

Anecdote: At IBM, a study was done that asked people if they would prefer a variable or fixed delay when typing and getting keystrokes echoed back to them. People liked a fixed delay, so adding delay to equalize the delay of the echoed key strokes improved the user experience.

Active Queue Management

Activity: get information back to the sender to get better control.

Goals of active queue management:
1. Efficiency
2. Fairness

Vanilla RED (random early detection) has no signal back to the sending host. It simply drops packets.

Is there are good way to drop packets?
1. Avoid synchronization (all connections backing off at the same time reducing line utilization).
2. Allow larger buffers and low latency.

Increasing the buffer size at routers increases latency because TCP saturates the line and the buffers, causing more packets to be waiting in the larger buffer.

TCP needs a bandwidth delay sized buffer to not increase latency with an increased buffer size.

RED drops packets to increase performance. This seems weird.

Active queue management at the routers is a major layer violation. RED is a mechanism that would enforce using TCP at the end points. End point congestion mechanism makes more sense.

ECN (Explicit Congestion Notification)

Two types:
1. Source quenching (send explicit messages back to source)
2. Mark bit in packet to signal congestion

There is a major deployment issue in that the sender, the receiver, and the router have to change to make ECN work.

Turning off lots of little flows causes loss rates of 5-10%.

Typical loss rate on the Internet is less than 1%.

The point of TCP congestion control is to keep the network from being consistently in a congested state. It shouldn't cause lots of losses.

Penalty Boxes monitor connections sending fast and turn them off to approximate fair queuing (like RED targeted at top flows).

Complexity of Hardware

- FIFO is simple. Add RED and the hardware requirements are not bad.
- Non-FIFO increases the hardware complexity.
- Fair Queuing requires per flow state to be maintained.
- ATM requires per flow monitoring.

More fair yields more work but is more understandable.

Side Note: What if we dropped ACKs? TCP is robust against dropping ACKs since an ACK acknowledges a byte range (up to a certain byte). A subsequent ACK covers for a dropped ACK. Some satellite and some wireless links drop tons of ACKs and there is major trouble until the connection blows up.

Feedback Parameters

- RED marks on average queuing over a window of 500 packets.
- Possibly mark when the average buffer queue is larger than the bandwidth of the round trip. Works for one flow.
- Mark on rate of increase.

The router does not know this stuff.

The process of changing flows in a network are:
1. Make a change.
2. Wait for the change to propagate through the network.
3. Make another change.

It has been shown that any fixed parameters for RED don't work. Developing a self-tuning RED is work in progress.