New Techniques for Making TCP Robust to Corruption-Based Loss

—

Wesley Eddy, Shawn Ostermann

Ohio University

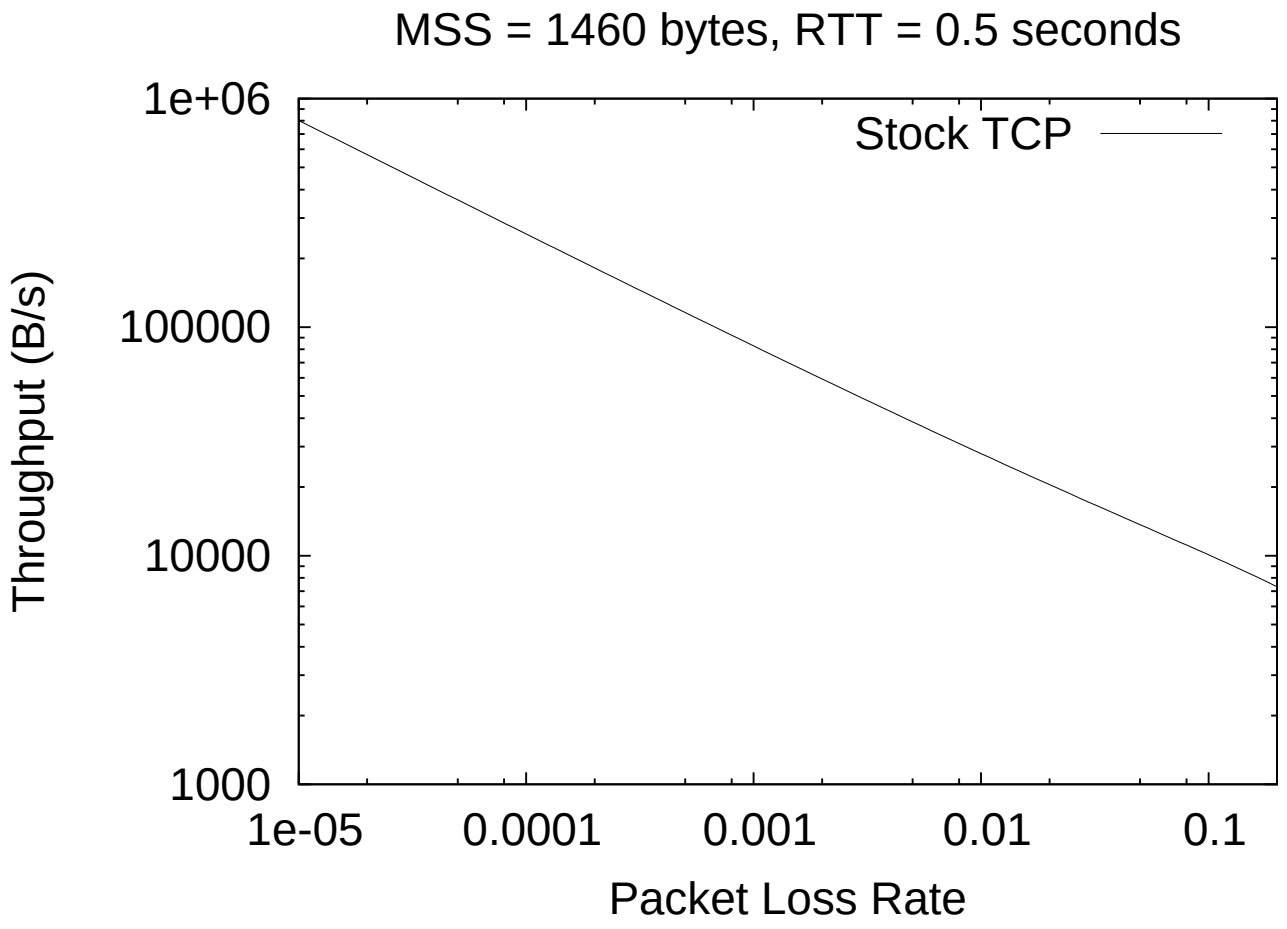
Mark Allman

BBN Technologies/NASA GRC

Background

- Based on BBN ETEN work
  - ETEN is - **E**xplicit **T**ransmission **E**rror **N**otification
- We focus on:
  - Bulk-transfer
    - Long duration flows with lots of data to send
  - Wireless / space environment
    - Possible high packet error rate leads to poor TCP througphut

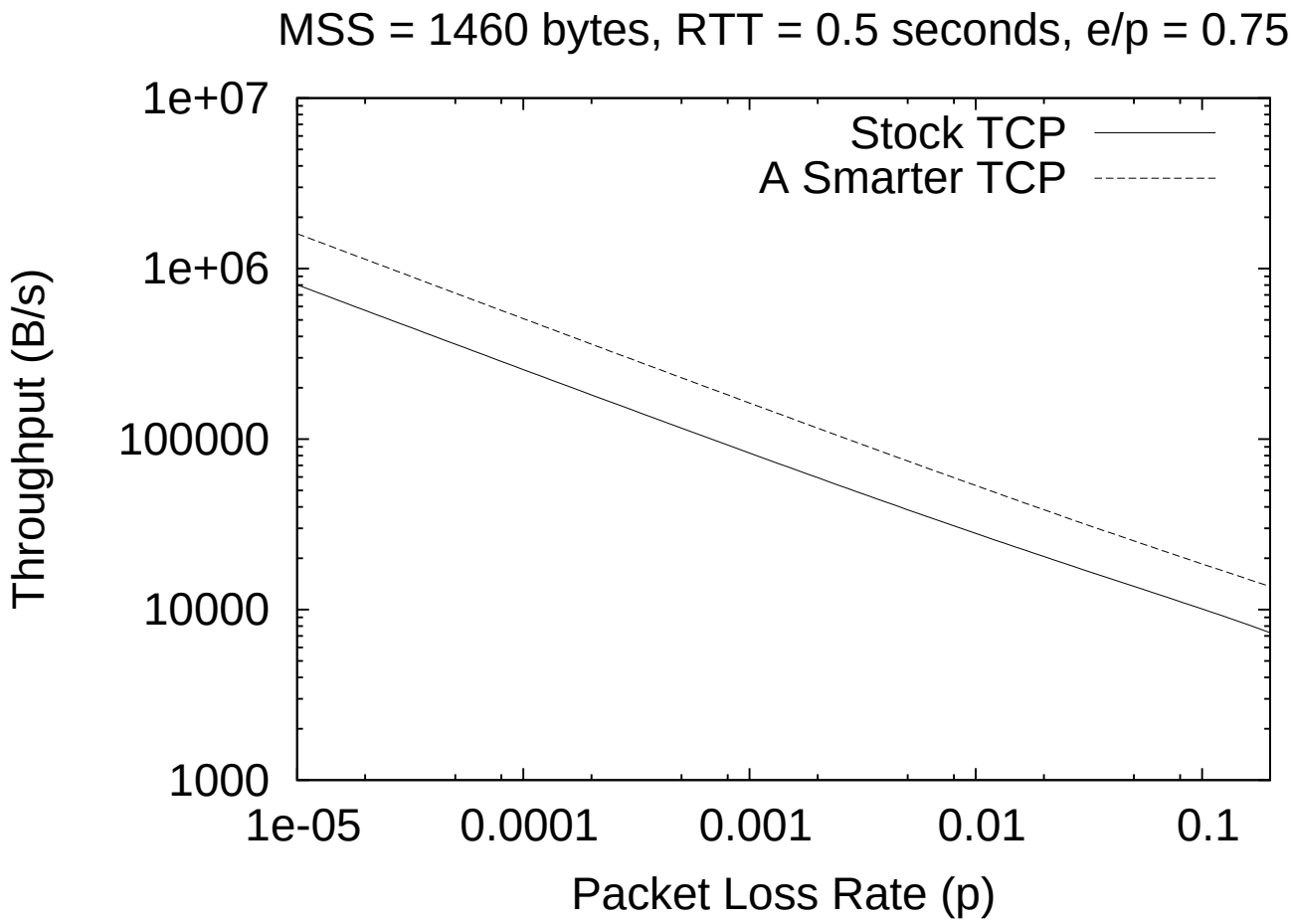
TCP's Response to Packet Loss



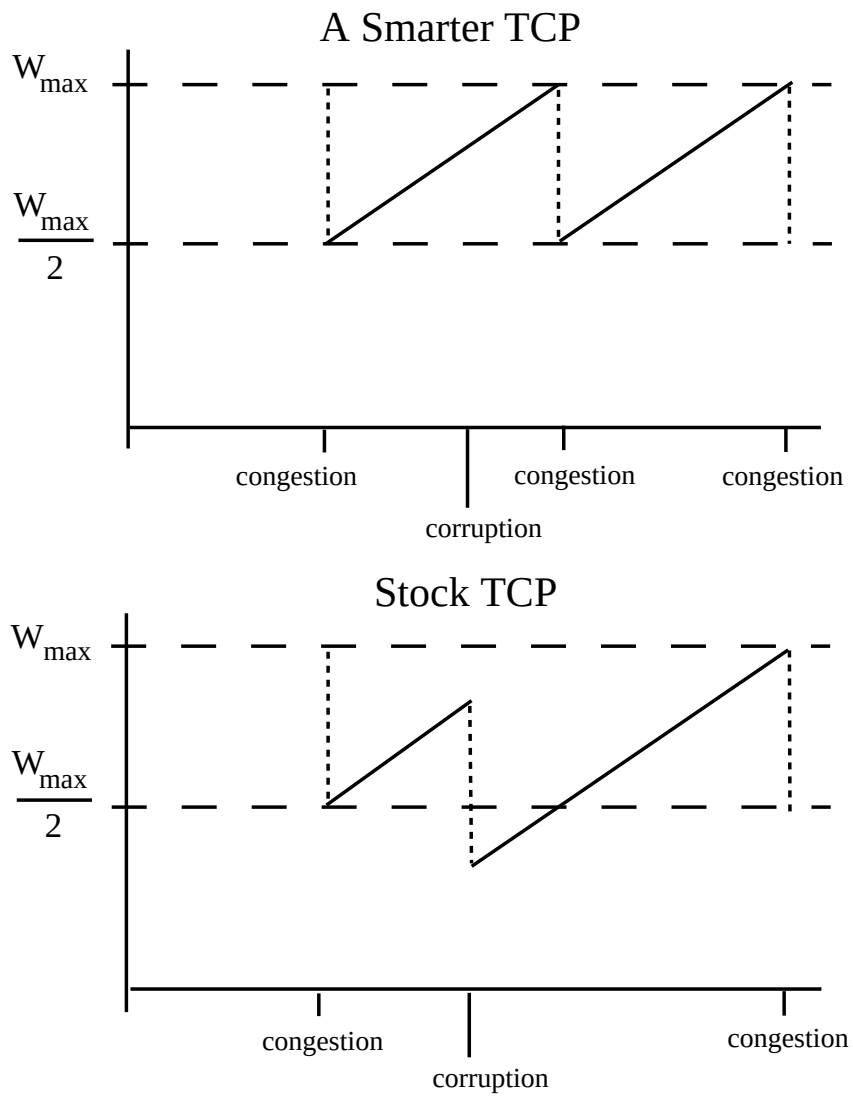
Packet Errors Considered Harmful

- Total packet loss rate is determined by congestion and errors ($p = c + e$)
  - Need to slow down for congestion, not always for errors
  - TCP can't tell the difference
  - If p is dominated by e . . . ouch!

How Much Better Can It Be?



With Perfect Knowledge



Cumulative Explicit Transmission Error Notification (CETEN)

- Each hop in the network path knows the error rate its link sees, so let's have them record it in an extra packet header
- Encode as a survival probability
  - Start with a 1 at the sender
  - Each router updates this field by multiplying its existing value by the error rate of the last hop

Help Where You LEAST Expect It

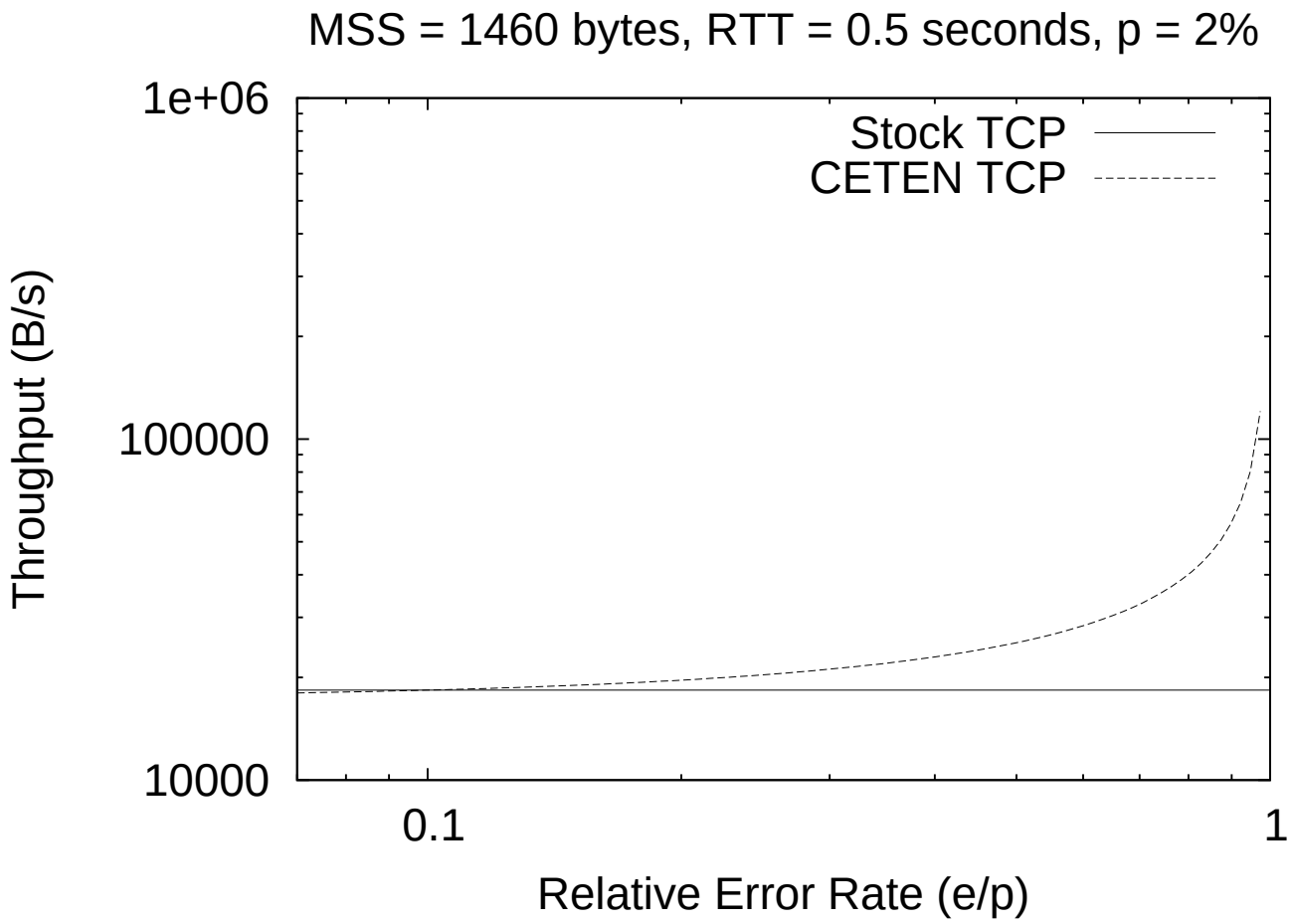
- Empirically, total retransmissions alone are not a good estimate of lost packets
- LEAST is **L**oss **E**stimation **A**lgorithm**S** for **T**CP
  - The basic idea is to use retransmits and unneeded duplicate transmissions to estimate the total packet loss rate
  - Works great (details out of scope here)
- Percent of losses due to errors

$$\frac{e}{p} = \frac{\text{CETEN estimate from routers}}{\text{LEAST estimate from sender}}$$

Modified Congestion Window Update

- Probabilistically
  - Could use $\frac{e}{p}$ to "guess" how often to use standard slowdown procedure (divide cwnd by two)
- Deterministically
  - Instead of $\frac{1}{2}$, multiply cwnd by $\frac{(1+\frac{e}{p})}{2}$

Gain in Throughput



Future Work

- Fairness/Friendliness
- Other adjusted congestion control algorithms?
  - There may be something better that we haven't thought of
- Exact method for computing error rate averages (time scale, etc)
- Effects of a misbehaving receiver
  - One who echoes: "I always lose 100% of your packets due to errors, send faster"