Literature Review Series:
Delay/Rate based
Congestion Avoidance in
TCP

David Hayes
dahayes@swin.edu.au
Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology

Outline

Introduction
Background
  Current TCP congestion avoidance
  Base measurements
Quick early work overview
Algorithm outlines
  CARD
  Packet pair flow control
  TCP-LP
  Vegas
  FAST
  Compound TCP
  DUAL
  Hamilton
Other
Conclusions
Bibliography
Introduction

- Promise low latency zero loss\(^1\)
- Delay based intuition:
  - delay↑ ≡ queue↑  \(\implies\) indicates congestion
- Rate based intuition:
  - Send rate > receive rate  \(\implies\) indicates congestion
- Basic questions:
  - How is congestion determined?
  - and if congested, how should cwnd be adjusted
- Issues:
  - Noise of measurements
  - Correlation of measurements with congestion
  - Compatibility with existing TCP algorithms

\(^1\)congestion related

Background: TCP NewReno congestion avoidance

- Congestion is indicated by packet loss
- The congestion window, cwnd, is adjusted with every ack as follows:

\[
\begin{align*}
    w_{j+1} &= \begin{cases} 
    \beta w_j & \text{packet loss} \\
    w_j + 1/w_j & \text{otherwise}
    \end{cases}
\end{align*}
\]

where in this case \(w\) is in packets.
- Multiplicative decrease
- Additive increase
Background: Base timing measurements

Note: Queueing at FIFO network nodes can increase or decrease the interpacket times.
Background: Base rate measurements

\[ T_1 = \sum w \frac{S}{rtt_i} \]

\[ T_{\text{max}} = \sum w \frac{S}{rtt_{\text{min}}} \]

\[ R_a = \frac{\sum w-a_i}{d_{aw}} A_i \]

Quick early work overview

- [Clark et al., 1985] \& [Clark et al., 1987] NETBLT RFCs 996 \& 998.
- [Jacobson, 1988] \(^a\) – footnote on connectionless rate based AIMD.
- [Jain, 1989] \(^b\) normalised delay gradient.
- [Brakmo and Peterson, 1995] \(^d\) TCP Vegas.


Algorithms: CARD [Jain, 1989]

- CARD - Congestion Avoidance using RTT Delay
- Uses queueing theory to determine knee of throughput graph
- Delay gradient, $\frac{d\text{rtt}}{dw}$
- Conditional increase/decrease of window based on Normalised Delay Gradient:

$$\text{NDG} = \left( \frac{\text{rtt}_j - \text{rtt}_{j-1}}{\text{rtt}_j + \text{rtt}_{j-1}} \right) \left( \frac{\text{w}_j + \text{w}_{j-1}}{\text{w}_j - \text{w}_{j-1}} \right)$$

and

$$\text{w}_{j+1} = \begin{cases} 
\beta_j \text{w}_j & \text{NDG} > 0 \\
\text{w}_j + \alpha & \text{otherwise}
\end{cases}$$

- Algorithm derived using D/D/1 queues
- Use in stochastic networks require enhancements

---

Algorithms: Packet pair flow control [Keshav, 1994]

- Full transport protocol proposal and analysis
- All data is sent as back-to-back pairs
- Available send rate is:

$$T = \frac{\text{size}(p_2)}{\text{pair dispersion}}$$

- Presumes routers use round robin scheduling
Algorithms: TCP-LP [Kuzmanovic and Knightly, 2006]

- Low Priority TCP
- Based on relative one way delay: \( d_i = ts_{rx,i} - ts_{tx,i} \)
  - Send and receive clocks do not need to be synchronised.
  - They do need to be the same frequency.

- Congestion: \( c_i = \begin{cases} 
1 & \bar{d}_i > d_{\min} + \delta(d_{\max} - d_{\min}) \\
0 & \text{otherwise}
\end{cases} \)
  where \( \delta \in (0, 1) \)

- Cwnd adjustment —
  \[ w_i = \begin{cases} 
\frac{w_{i-1}}{2} & (c_i = 1) \land (itti = 0) \\
1 & (c_i = 1) \land (itti = 1) \\
1 + \frac{1}{w_{i-1}} & (c_i = 0) \land (itti = 0)
\end{cases} \]

- itti – interference timeout timer indication (debounce)

- Requires feedback of delay measurement
- Requires accurate estimates of \( d_{\max} - d_{\min} \)

Algorithms: Vegas [Brakmo and Peterson, 1995]

- Iconic rate based TCP
- Defines two rates:
  \( \text{actual} = \sum S \frac{S}{\text{rtt}} \)
  \( \text{expected} = \frac{w}{\text{rtt}_{\min}} \)
  and \( \text{diff} = \text{expected} - \text{actual} \)

- Window adjustment:
  \[ w \leftarrow \begin{cases} 
  w - 1 & \text{diff} > \beta \\
  w + 1 & \text{diff} < \alpha \\
  w & \text{otherwise}
\end{cases} \]

- Usually \( w = \sum S \)
- Then \( \tau_{\text{diff}} = \text{rtt} - \text{rtt}_{\min} \)
- where \( \tau_{\text{diff}} = \text{diff} \left( \frac{\text{rtt} + \text{rtt}_{\min}}{w} \right) \)
  - Requires accurate estimate of \( \text{rtt}_{\min} \)
  - AIAD
Algorithms: FAST [Wei et al., 2006]

- Enhanced Vegas type algorithm
- MIMD — AIMD to slow for high BDP networks
- Uses delay as a rich (non binary) congestion indicator
- Cwnd is updated at regular time intervals ($\Delta t$):

$$w_{t+\Delta t} = \min \left\{ 2w_t, \gamma \left( \frac{\text{rtt}_{\text{min},i}}{\text{rtt}_t} w_t + \alpha \right) + (1 - \gamma)w_t \right\}$$

- For MIMD, $\alpha(w_t, q_i)$
  - increase is proportional to the size of cwnd and the network queueing delay.

Algorithms: Compound TCP [Tan et al., 2006]

- In high speed high BDP networks aims to increase:
  - efficiency
  - RTT fairness and TCP fairness
- In MSW Vista and 7
- Uses Vegas’ rates: $\text{diff} = (\text{expected} - \text{actual})\text{rtt}_{\text{min}}$
- Provides NewReno+ performance throughput
  - The send window, $\text{win}_j$, is calculated as:
    $$\text{win}_j = \min(w_j + \text{dwnd}_j, \text{awnd}_j)$$
  - where $w_j$ is NewReno’s cwnd
  - and dwnd$_j$ is the delay based window.
  - and awnd$_j$ is the receivers advertised window.
The delay window is calculated as follows:

\[
dwnd_{j+1} = \begin{cases} 
    dwnd_j + \alpha ((\text{win}_j)^k - 1)^+ & \text{diff} < \gamma \\
    (dwnd_j - \zeta \text{diff})^+ & \text{diff} \geq \gamma \\
    \text{win}_j(1 - \beta) - \frac{\text{cwnd}}{2} & \text{on loss}
\end{cases}
\]

- Increase rule, where \( \alpha = \frac{1}{8} \) is the multiplicative increase factor relative to window size \((k = 0.75)\)
- Delay decrease rule, relative to diff (the queued data)
- Loss decrease rule, \( \beta = 0.5 \)
- requires accurate estimate of \( \text{rtt}_{\text{min}} \)

\[\text{note: } \text{win}_j = \min(\text{wj} + dwnd_j, \text{awnd}_j)\]


- Designed to supplement loss based congestion control
- Delay based measurements provide “slow tuning” of cwnd every 2\(^{nd}\) RTT

\[w \left\{ \begin{array}{l} \beta w \quad \text{rtt} > \frac{(\text{rtt}_{\text{min}} + \text{rtt}_{\text{max}})}{2} \\
                    w \quad \text{otherwise} \end{array} \right.\]

where \( \beta = \frac{7}{8} \)

- Attempts to keep network buffers half full
- Smaller multiplicative decrease
- Relies on accurate estimates of \( \text{rtt}_{\text{min}} \) and \( \text{rtt}_{\text{max}} \)
Algorithms: Hamilton [Budzisz et al., 2009]

- Designed for coexistence with loss based TCP
- Inspired by Active Queueing techniques (as was PERT [Kotla and Reddy, 2008])

\[ w_{i+1} = \begin{cases} \frac{w_i}{2} & X < g(q_i) \\ w_i + \frac{1}{w_i} & \text{otherwise} \end{cases} \]

Random multiplicative decrease

- Region B stable when queueing delay is high
- Region A stable when queueing delay is low
- AIMD matches NewReno
- Relies on accurate estimates of rtt\(_{\text{min}}\) and rtt\(_{\text{max}}\)

Algorithms: Others of Interest

- [King et al., 2005] — TCP-Africa
  - Two modes: Fast delay based, and slow NewReno based.
  - Compound TCP is based on some of Africa’s ideas
- [Baiocchi et al., 2007] — YeAH-TCP
  - Yet Another Highspeed TCP
  - Two modes like Africa
  - Provides performance improvements on lossy paths.
- A number of schemes propose traffic shaping TCP’s send rate
  - [Karandikar et al., 2000] – ABR like
  - [Wu et al., 2002] – leaky bucket
Conclusions

- Delay can provide an earlier indication of congestion than loss
- As such it will become important in high BDP networks:
  - Even aggressive loss based protocols have very long cwnd oscillations and cannot use the available bandwidth.
- Issues:
  - Compatibility with existing TCPs
  - Inaccurate estimates of $\text{rtt}_\text{min}$ and $\text{rtt}_\text{max}$
  - Send and receive rates are hard to measure (except in FQing networks)
  - Rate based flow control?
- CAIA’s work in the next seminar

Bibliography I


Bibliography II


Bibliography III


[Bibliography V]


