On the Effectiveness of Delay-based Congestion Avoidance

Ravi Prasad,
Manish Jain,
Constantinos Dovrolis

ravi, jain, dovrolis@cc.gatech.edu

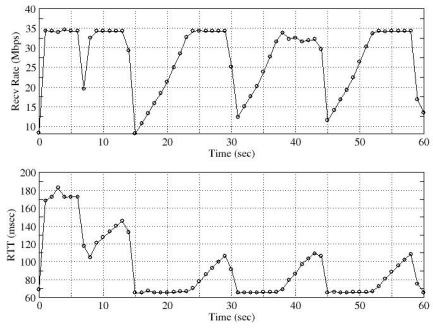
College of Computing
Georgia Tech

Outline

- Loss-based Congestion Avoidance (LCA)
  - Inefficient in high BDP paths
- Delay-based Congestion Avoidance (DCA)
  - Includes TCP Vegas and FAST
- Controllability and observability
  - Does DCA meet these properties?
- Four failure scenarios for DCA
  - Small RTT variations
  - RTT undersampling
  - DCA in highly aggregated paths
  - Random losses

Loss-based Congestion Avoidace (LCA)

- TCP Reno follows the LCA model
- LCA flow increases cwnd until it sees packet loss



- Consequences:
  - Throughput less than available bandwidth
  - Increased loss-rate
  - Large delay variation
  - Large buffering requirement in routers

TCP Reno can be inefficient in high Bandwidth-Delay Product (BDP) paths

- Example: 1Gbps, 100msec, 1500B pkts
  - Large window reduction upon loss
    - Single loss ≈ 4000 pkt window reduction
  - Large recovery time
    - Recovery from 1 pkt loss ≈ 400 sec
  - Need very small loss probability
    - Loss rate must be less than $2*10^{-8}$
- Side comment: previous example assumes unlimited socket buffers
  - See SOBAS for automatic socket buffer sizing to avoid losses

Delay-based Congestion Avoidance (DCA)

- Early paper by R. Jain: ACM CCR – Oct 89
- Follow-up protocols:
  - CARD ('89)
  - Tri-S ('91)
  - DUAL ('92)
  - TCP-Vegas ('94)
  - TCP-BFA ('98)
  - TCP-FAST ('03)
  - SOBAS ('03)

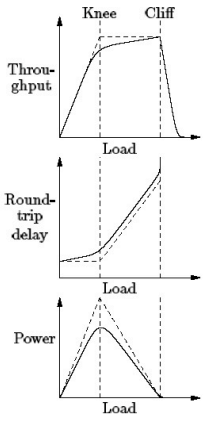


Figure 1: Network performance as a function of the load. Broken curves indicate performance with deterministic service and interarrival times.

TCP Vegas

- Expected rate E = Window / T_{min}
- Measure actual send rate R
- Adjust window based on difference E - R
- If (E-R) is larger than threshold, then:
  - Flow has built up queue at bottleneck
  - RTT T is larger than T_{min}
  - Reduce window to avoid congestion

TCP FAST

- Improved (stabilized) version of TCP Vegas
- Reduce window when RTT increases
- Window decrease factor depends on:
  - Current window size
  - RTT relative increase
- As opposed to constant decrease factor of Vegas
- Scales well in high BDP paths

First concerns about DCA schemes

- Measurement studies showed little or no correlation between increased RTTs and network load
  - Martin, et al., ToN 2003, "Delay-based congestion avoidance for TCP"
  - Biaz and Vaidya, IMC 2003, "Is the Round-Trip Time Correlated with the Number of Packets in Flight?"
- Previous studies showed that correlations of RTT and load are even weaker in high-bandwidth paths
  - Interesting, because DCA is supposed to work better than LCA in high BDP paths!

DCA can fail when:

- RTT variations cannot be reliably measured
  - Cause: small network buffers or noisy RTTs
- DCA flow undersamples RTT variations
  - Cause: flow throughput is small relative to cross traffic throughput
- DCA flow cannot affect RTT
  - Cause: DCA flow competes with many other flows in highly aggregated path
- DCA flow cannot avoid losses
  - Cause: random losses or congestive losses due to cross traffic bursts

Another look at the problem..

- Network state:
  - Tight link queue size Q(t) and available bandwidth A(t)
- Input variables:
  - Window $W_i(t)$ for flow i (DCA or LCA)
  - Instantaneous rate of non-TCP flows
- Output variables:
  - Sampled RTT $T_i(t)$ and loss indicator $L_i(t)$ for flow i

Controllability & Observability

- Major concepts in theory of control systems
- Controllability:
  - A state variable (or output) is controllable if it can be driven to a certain level by an input variable
- Observability:
  - A state variable is observable if it directly affects one or more outputs
- A DCA flow should be able to:
  - Observe the queue size (state variable) through sampled RTT signal (output)
  - Control the queue size through send window (input)
- Under which conditions will a DCA flow meet (or not meet) these properties?

RTT Signal-to-Noise ratio

- Minimum RTT: T_{min}
- Maximum RTT: $T_{min} + B_t/C_t$
  - B_t: Tight link buffer
  - C_t: Tight link capacity
- We need to consider RTT noise n(t)
  - Random queueing at:
    - Non-tight links in forward path
    - Reverse path
  - End-host timestamping resolution & OS noise
- Measured RTT: $T(t) = T_{min} + Q_t(t)/C_t + n(t)$
  - What if noise > signal ?

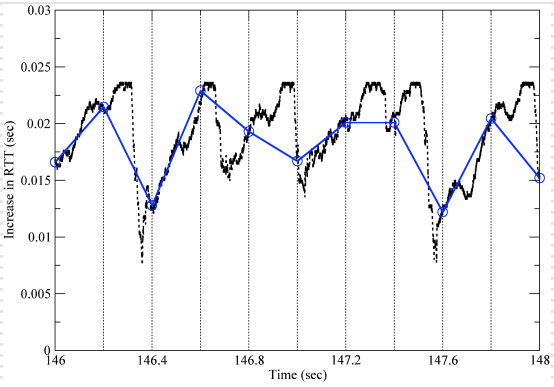
RTT Signal-to-Noise ratio (cont')

- Signal: $Q_t(t)/C_t$
- We have a problem if B_t/C_t = O(noise)
  - RTT variation will not be measured accurately
  - DCA flow will not avoid congestive losses
  - Queue size is not observable, because the output signal (RTT) is too weak
- Lesson:
  - DCA effectiveness depends on link buffer sizes
  - Small buffers and RTT measurement noise can break observability

RTT undersampling

- Suppose that at time t:
  - Available bandwidth = A
  - New cross-traffic flow starts with rate $R_x > A$
  - Tight link buffer will fill up after $B_t/(R_x - A)$
- DCA flow with throughput R samples RTT at rate:
  - L/R, L: packet size
- DCA undersamples RTT, and may not detect queue buildup, if
  - $L/R = O(B_t/(R_x - A))$
  - Or, $R << (R_x - A)$
  - Queue size not observable because input is too sparse

RTT undersampling (cont')



- Lesson: DCA may not work for low-throughput flows (relative to the cross traffic flows)

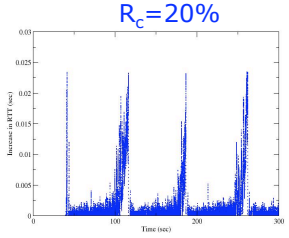
Highly aggregated traffic

- Cross traffic rate: $R_c = C_t - R$
  - Aggregate of many flows
  - Each flow is small share of aggregate
- DCA flow rate: $R << R_c$
- Tight link queue size, and RTT variations, are not controlled by DCA flow
  - RTT variations appear as erratic ("random")
  - Queue size is not controllable because input signal due to any single flow is too weak

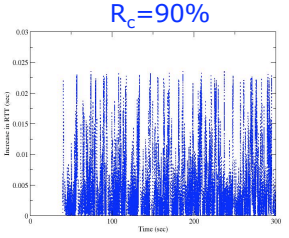
Highly aggregated traffic (cont')

- Cross traffic
  - TCP flows of size 15-20pkts
  - Constant arrival rate of flows

R_c=20%



R_c=90%



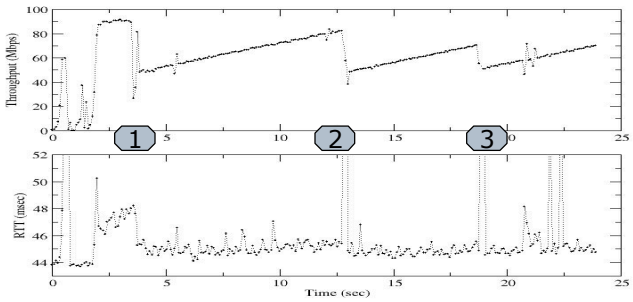
- Lesson: ~~DCA may not work in paths that carry many~~ relatively small flows

Random losses

- A DCA flow can avoid self-induced congestive losses
  - Losses caused and experienced by that flow
- A DCA flow cannot avoid congestive losses caused by high-rate bursts from other flows
  - Queue size controlled by other flows
  - Losses appear as "random" to DCA flow
- But, a DCA flow should still react to such losses to avoid congestion collapse
  - How should DCA flows react to losses?
  - They often do not play fair with TCP Reno in that aspect

Random losses (cont')

- Losses are not always preceded by increasing RTTs
  - RTTs may have increased, but not observed by DCA flow



- DCA flow could avoid losses at 1, but not at 2 or 3
- Lesson: DCA schemes should still expect and react to congestive losses

Concluding remarks

- DCA is probably more efficient than LCA (Reno) under certain conditions:
  - A few high-throughput flows
  - Well-buffered links
  - No significant RTT noise
  - No "random" losses
- Most simulation studies meet previous conditions
- The real Internet does not…

Concluding remarks (cont')

- More research is needed to establish the robustness of DCA schemes
  - Robustness vs efficiency trade-off?
- Robustness in terms of:
  - Different buffer sizes, link capacities, flow RTTs
  - Heterogeneous traffic models and applications
  - Realistic noise sources in RTT measurements