A control theoretical analysis of a window-based flow control mechanism for TCP connections with different propagation delays

Hiroyuki Ohsaki
Cybermedia Center
Osaka University, Japan
oosaki@cmc.osaka-u.ac.jp

Contents

TCP (Transmission Control Protocol)

- Packet retransmission mechanism
  - Retransmit lost packets in the network
- Congestion avoidance mechanism
  - A window-based flow control mechanism
- Several versions of TCP
  - TCP Tahoe
  - TCP Reno
  - TCP Vegas

TCP Reno

- Implemented in BSD UNIX
- Widely used in the current Internet
- Use packet loss as feedback information
  1. Source host continuously increases window size
  2. Packet loss occurs at the bottleneck router
  3. Source host detects packet loss by duplicate ACK
  4. Source host reduces its window size to 1/2
- Packet loss is inevitable

TCP Vegas

- Advantages over TCP Reno
  - A new retransmission mechanism
  - An improved congestion avoidance mechanism
  - A modified slow-start mechanism
- Uses measured RTT as feedback information
  1. Source host measures the RTT for a specific packet
  2. Source host estimates severity of congestion
  3. Source host changes window size
- Packet loss can be prevented

Objectives

- Analyze a window-based flow control
  - Congestion avoidance mechanism of TCP Vegas
  - Connections have different propagation delays
  - Stability and transient behavior using a control theoretic approach
- Show numerical examples
  - Parameter tuning of TCP Vegas

Congestion avoidance mechanism of TCP Vegas

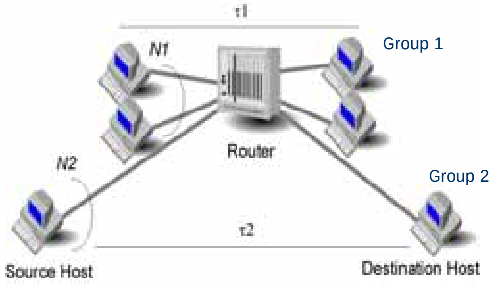
- Source host maintains the minimum RTT: τ
- Source host measures the actual RTT: *r*(k)

$$d(k) = \frac{w_n(k)}{\tau} - \frac{w_n(k)}{r(k)}$$

- Window size is changed based on *d*(k)

$$w_n(k+1) = \begin{cases} w_n(k)+1 & \text{if } d(k) < \alpha \\ w_n(k)-1 & \text{if } \beta < d(k) \\ w_n(k) & \text{otherwise} \end{cases}$$

Analytic Model (M = 2)



Assumptions

- A single bottleneck router in the network
- TCP connections in a group are synchronized
- All TCP connections are greedy

Derivation of state transition equations

- Window size: *$w_{m,n}$*(k)
  - □ *$\delta_{m,n}$*: control parameter (i.e., feedback gain)
  - □ *Δ_m*: Frequency of window size change

$$w_{m,n}(k+\Delta_m) = \max\left(w_{m,n}(k) + \delta_{m,n}(\gamma_{m,n} - d_{m,n}(k)), 0\right)$$

- Queue length: *q*(k)

$$q(k+1) = \min\left[\max\left\{\sum_{m=1}^{M} N_m\left(w_m(k) - \frac{w_m(k)B\Delta_m\tau}{\sum_{m=1}^{M} N_m w_m(k)}\right), 0\right\}, L\right]$$

- System state

$$\left(w_1(k) \quad w_2(k) \quad \ldots \quad w_M(k) \quad q(k)\right)$$

Stability and transient behavior analysis

- Obtain a linearized model
  - **x**(k) : state vector (current state – equilibrium state)

$$\mathbf{x}(k+\Delta_{LCM}) = \mathbf{A}\,\mathbf{x}(k)$$

- Eigenvalues of **A** determine stability and transient behavior
  - s: the maximum eigenvalues of **A**

$$s = \max_i(s_i)$$

  - *s* < 1: stable
  - *s* > 1: unstable
  - smaller *s*: better transient performance

$$\mathbf{x}(k) \equiv \begin{bmatrix} w_1(k) - w_1^* \\ \vdots \\ w_M(k) - w_M^* \\ q(k) - q^* \end{bmatrix}$$

Numerical examples

- Network parameters
  - *M* = 2: 2 groups of TCP connections (short and large delay)
  - *N_1*=10: # of TCP connections in group 1
  - *N_2*=10: # of TCP connections in group 2
  - *B*=150Mbps: processing speed of the bottleneck router
- Control parameters
  - $\gamma_1=\gamma_2=3$: control parameter adjusting # of in-flight packets
  - δ_1, δ_2: control parameter adjusting a feedback gain

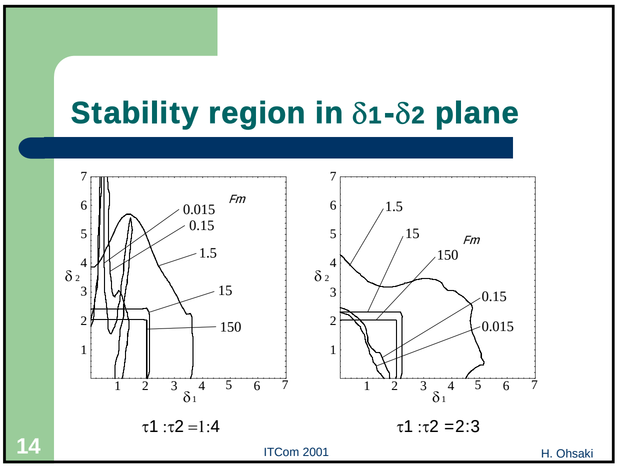
Queuing delay ratio: *Fm*

- Ratio of queuing delay to propagation delay
  - $N_m \gamma_m$: # of packets in the router's buffer

$$F_m = \frac{N_m \gamma_m}{B\tau}$$

  - Large *Fm*: the queuing delay is not negligible
  - Small *Fm*: the queuing delay is negligible
- If *Fm* is identical, stability and transient behavior are not changed

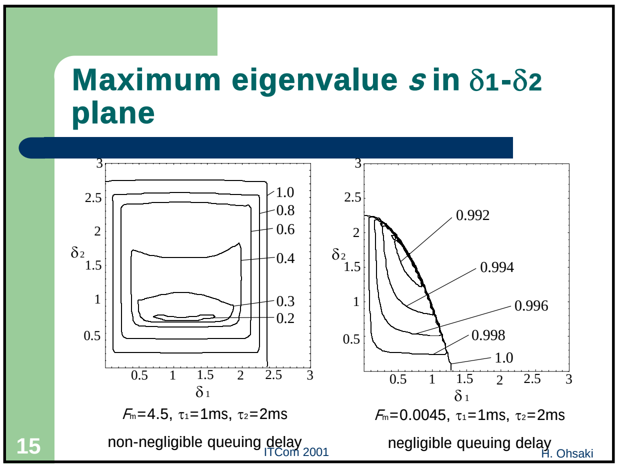
---



Stability region in δ1-δ2 plane

$\tau_1 : \tau_2 = 1:4$

$\tau_1 : \tau_2 = 2:3$



Maximum eigenvalue *s* in δ1-δ2 plane

$F_m = 4.5$, $\tau_1 = 1$ms, $\tau_2 = 2$ms

non-negligible queuing delay

$F_m = 0.0045$, $\tau_1 = 1$ms, $\tau_2 = 2$ms

negligible queuing delay

Conclusion

- A window-based flow control based on TCP Vegas
- TCP connections have different propagation delays
- Stability and transient behavior analysis
- if *Fm* is small (i.e., propagation delay > queuing delay)…
  - Parameter δ should be proportional to TCP's propagation delay
  - Transient behavior cannot be improved
- if Fm is large (i.e., propagation delay < queuing delay)…
  - Parameter δ should be between 0 and 2
  - Transient behavior can be greatly improved