Analysis of a window-based flow control mechanism based on TCP Vegas in heterogeneous network environment

Hiroyuki Ohsaki

Cybermedia Center, Osaka University, Japan

oosaki@cmc.osaka-u.ac.jp

Contents

- Introduction
- Analytic model
  - Window-based flow control based on TCP Vegas
  - Heterogeneous network
- Analysis of throughput and fairness
- Analysis of stability and transient behavior
- Numerical examples
  - Two bottleneck links with 30 TCP connections
- Conclusion

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 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. Measures RTT for a specific packet
  2. Estimates **severity of congestion**
  3. Changes window size
- Packet loss can be **prevented**

Objectives

- Analyze a window-based flow control
  - Congestion avoidance mechanism of **TCP Vegas**
  - Connections with **different propagation delays**
  - Several **bottleneck links**
  - Using a control theoretic approach
- Show numerical examples
  - **Throughput** and **fairness**
  - **Stability** and transient behavior

Congestion avoidance 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)***

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

Assumptions

- Network topology is **arbitrary**
- All routers employ **static routing**
- All routers have a FIFO buffer for each port
- All TCP connections are **symmetry**
- Backward path is never congested

State transition equation: window size

- **Window size** of connection c : w_c
  - δ_c: **control parameter** that determines the amount of increase/decrease in window size

$$w_c(k) = \begin{cases} \left[w_c\left(k - \frac{\tau_c}{\tau}\right) + \delta_c\left(\gamma_c - d_c(k)\right) \right]^+ & \text{if } k \equiv 0 \ (\text{mod } \frac{\tau_c}{\tau}) \\ w_c(k-1) & \text{otherwise} \end{cases}$$

$$d_c(k) = \left(\frac{w_c\left(k - \frac{\tau_c}{\tau}\right)}{\tau_c} - \frac{w_c\left(k - \frac{\tau_c}{\tau}\right)}{r_c(k)} \right) \tau_c$$

State transition equation: queue length

- **Queue length** of the buffer for link *l* : q_l
  - *b(c,l)* : the previous link of link *l* for connection *c*
  - l_c : the access link of connection *c*

$$q_l(k) = \left[q_l(k-1) + \left(\sum_{c \in C(l)} A_{c,l}(k-1) - \mu_l \right) \tau \right]^+$$

$$A_{c,l}(k) = \begin{cases} \dfrac{w_c(k)}{r_c(k)} & \text{if } l = l_c \\ \dfrac{\mu_l A_{c,b(c,l)}(k - \Delta_{b(c,l)})}{\sum_{d \in C(l)} A_{d,b(d,l)}(k - \Delta_{b(d,l)})} & \text{if } l \neq l_c \text{ and } q_l(k) > 0 \\ A_{c,b(c,l)}(k - \Delta_{b(c,l)}) & \text{if } l \neq l_c \text{ and } q_l(k) = 0 \end{cases}$$

Throughput and fairness

- Focus on equilibrium values (denoted by *)
  - ρ_c^* : **throughput** of connection *c*
  - θ_c^* : sum of **all queuing delays** of connection *c*

$$\gamma_c = \rho_c^* \theta_c^*$$

- Can be regarded as a **Little's law**

$$N = \lambda T$$

- Fairness between TCP connections *c* and *c'*

$$\frac{\rho_c^*}{\rho_{c'}^*} = \frac{\gamma_c}{\gamma_{c'}} \times \frac{\theta_{c'}^*}{\theta_c^*}$$

Stability and transient behavior

- Obtain a **linearized** model
  - $x(t)$: state vector (current state – equilibrium state)
  - Δ_{LCM} : Lowest Common Multiple of all $\tau c/\tau$'s

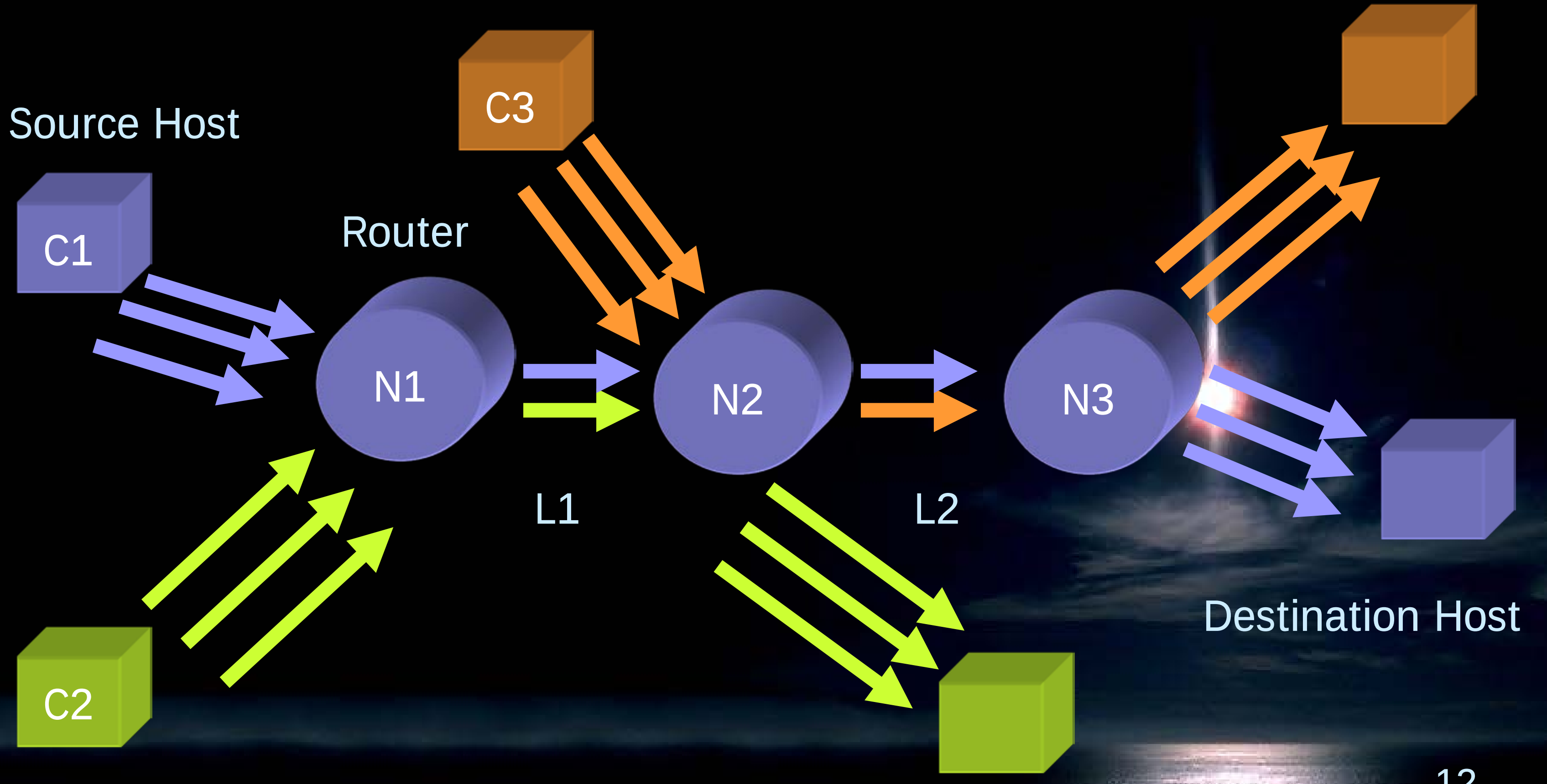
$$\mathbf{x}(k+\Delta_{LCM}) = \mathbf{A}\,\mathbf{x}(k)$$

$$\mathbf{x}(k) \equiv \begin{bmatrix} w_{c1}(k) - w_{c1}^{*} \\ \vdots \\ w_{c|N|}(k) - w_{c|N|}^{*} \\ q_{l1}(k) - q_{l1}^{*} \\ \vdots \\ q_{l|L|}(k) - q_{l|L|}^{*} \end{bmatrix}$$

- **Eigenvalues** of **A** determine stability and transient behavior

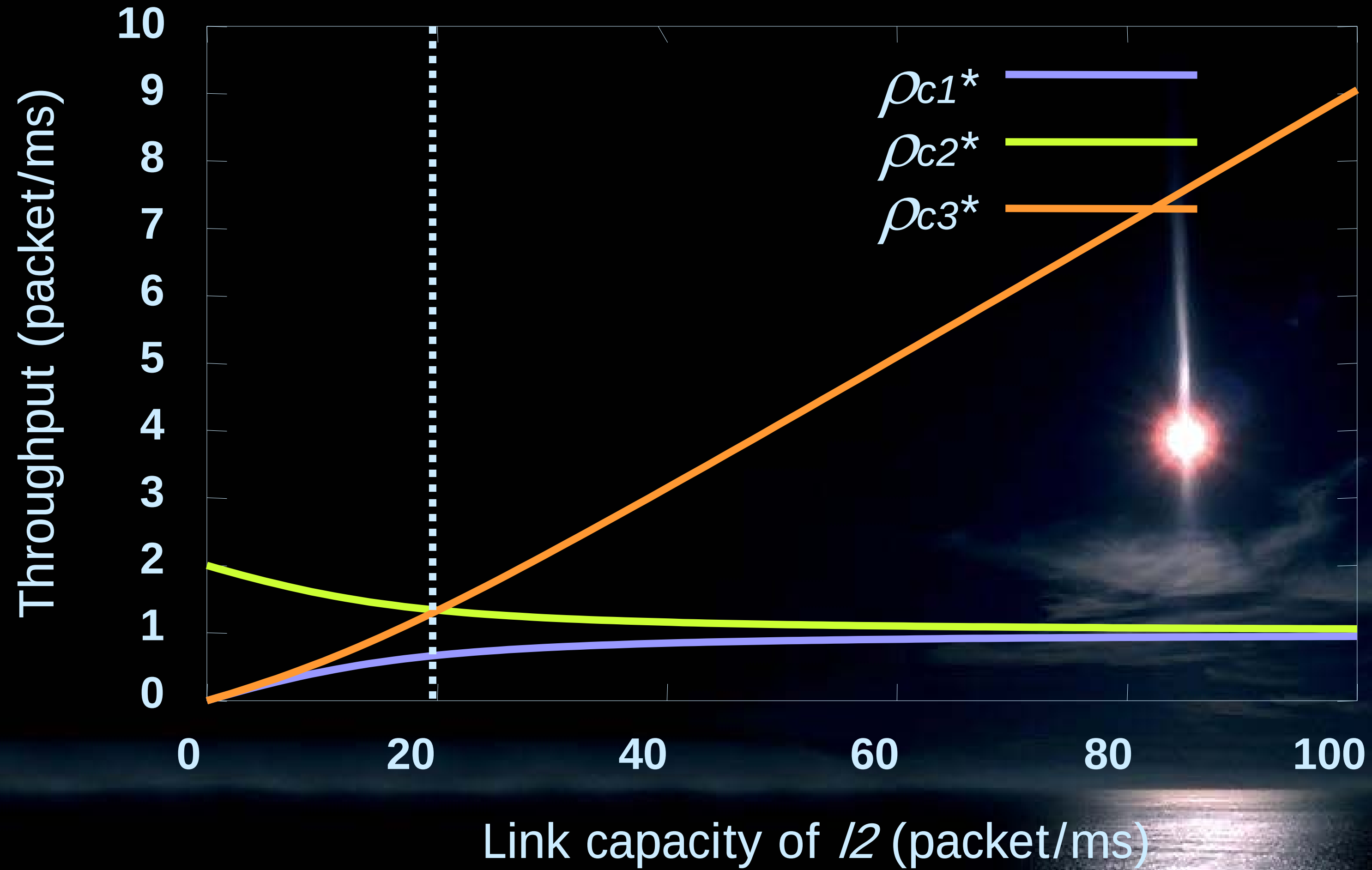
Numerical example: network model

- Two bottleneck links and 30 TCP connections



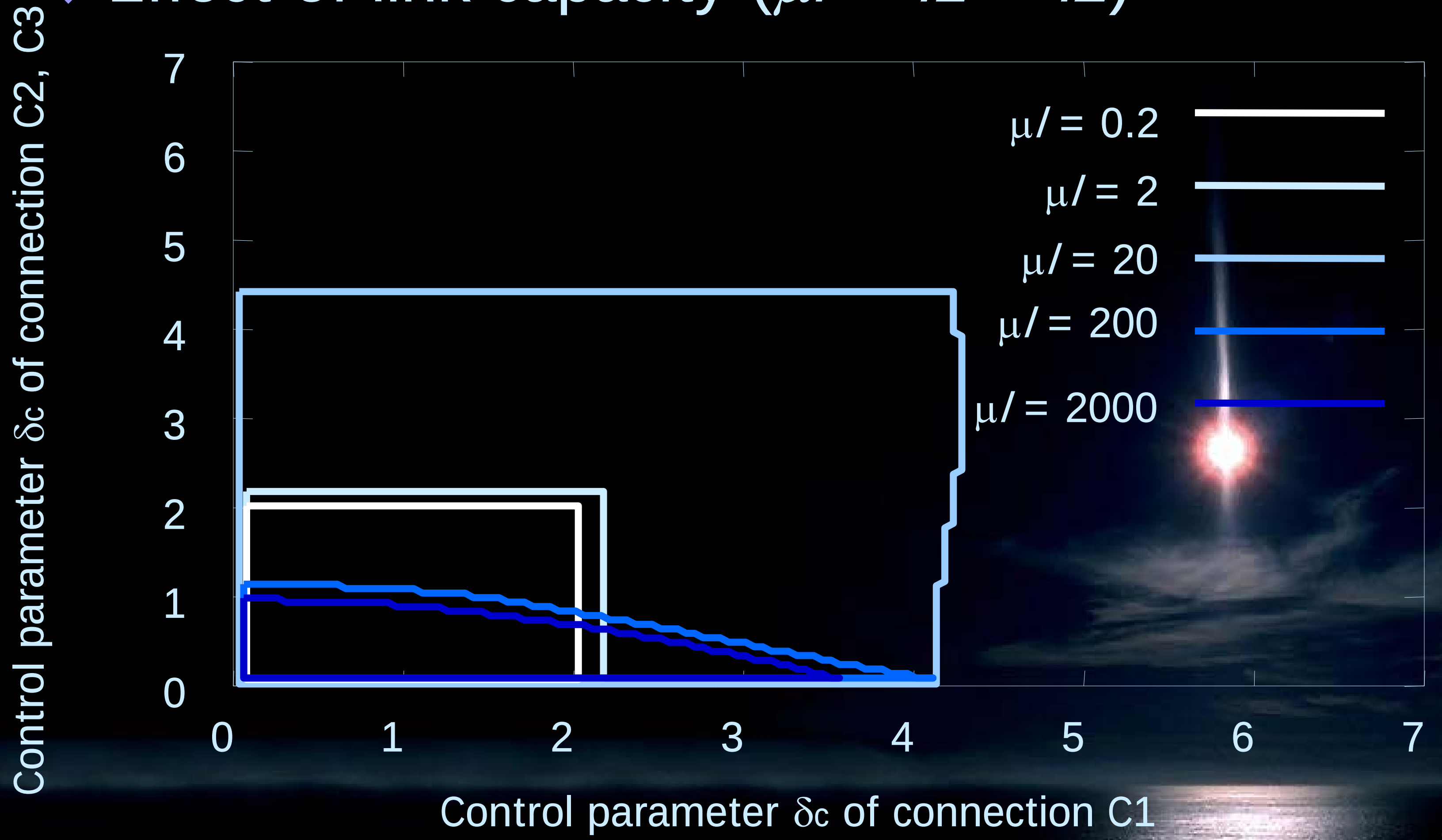
Numerical example: throughput

- Effect of difference in link capacities (μ/1 = 20)



Numerical example: stability

- Effect of link capacity (*μl = l1 = l2*)



Conclusion

- Analytic model
  - Window-based flow control based on **TCP Vegas**
  - **Homogeneous network**
- Throughput and fairness
  - Can be explained by **Little's law**
  - Has a bias against **link capacity** and **# of bottleneck links**
  - Window size in steady state
    - (bandwidth) x (propagation delay) + (control parameter γc)

Conclusion (cont.)

- Stability and transient behavior
  - Determined by **eigenvalues** of state transition matrix **A**
  - Link capacity significantly affects stability
  - Investigation using **trajectories of eigenvalues**
- Future work
  - More simulation studies
  - Extension to TCP Tahoe or TCP Reno