Computer Networks

Lecture 19

TCP/Vegas,
TCP Dynamics;
Transport Bandwidth Allocation Framework

11/7/2013
Exam 1 to be returned next Tuesday

Assignment four
  - Questions?
  - Please schedule a design meeting with me or a TF
Recap: TCP/Reno: Big Picture

TD: Triple duplicate acknowledgements
TO: Timeout
Q: when cwnd is cut to half, why sending rate is not?
Recap: TCP/Reno Queueing Dynamics

- Consider congestion avoidance only

There is a filling and draining of buffer process for each TCP flow.
Recap: TCP Throughput Analysis Objective

- To understand the throughput of TCP/Reno as a function of RTT (RTT), loss rate (p) and packet size

- There are typically two approaches in two settings
  - deterministic
  - random losses
Recap: TCP/Reno **Deterministic Throughput** Modeling: Relating $W$ with Loss Rate $p$

- **Consider congestion avoidance only**

Assume one packet loss (loss event) per cycle

Total packets send per cycle = $(W/2 + W)/2 * W/2 = 3W^2/8$

Thus $p = 1/(3W^2/8) = 8/(3W^2)$

$$W = \frac{\sqrt{8/3}}{\sqrt{p}} = \frac{1.6}{\sqrt{p}} \quad \Rightarrow \text{throughput } x = \frac{S}{RTT} \frac{3}{4} \frac{1.6}{\sqrt{p}} = \frac{1.2S}{RTT \sqrt{p}}$$
TCP/Reno Throughput Modeling with Random Losses

\[
\Delta W = \begin{cases} 
\frac{1}{w} & \text{if the packet is not lost} \\
-\frac{w}{2} & \text{if packet is lost}
\end{cases}
\]

\[
\Delta W = (1 - p) \frac{1}{W} + p \left( -\frac{W}{2} \right)
\]

=> mean of W = \(\sqrt{\frac{2(1-p)}{p}}\) \(\approx \frac{1.4}{\sqrt{p}}\), when p is small

=> throughput \(x \approx \frac{1.4S}{RTT\sqrt{p}}\), when p is small
TCP/Reno Dynamics

\[ \Delta W = (1 - p) \frac{1}{W} + p\left(-\frac{W}{2}\right) \]

\[ \dot{x} = \frac{\Delta w_{\text{unit-time}}}{RTT} \]

\[ \Delta w_{\text{pkt}} = (1 - p) \frac{1}{W} + p\left(-\frac{W}{2}\right) \]

\[ \Delta w_{\text{unit-time}} = \Delta w_{\text{pkt}} x = \left[(1 - p) \frac{1}{W} + p\left(-\frac{W}{2}\right)\right] x \]

\[ \dot{x} = \frac{\Delta w_{\text{unit-time}}}{RTT} = \frac{\left[(1 - p) \frac{1}{W} + p\left(-\frac{W}{2}\right)\right] x}{RTT} = (1 - p) \frac{1}{RTT^2} - \frac{1}{2} px^2 \]
Summary: TCP/Reno Throughput Modeling

- They are all *approximate* modeling
  - the details and the exact numbers are not important
  - the objective is to help us understand TCP better
TCP/Reno Depends on Buffer Filling/Draining

Why does TCP/Reno and derived depend on buffer filling up/drain?
TCP/Vegas (Brakmo & Peterson 1994)

- Idea: try to detect congestion by delay before loss
- Objective: not to overflow the buffer; instead, try to maintain a constant number of packets in the bottleneck queue
TCP/Vegas: Key Question

How to estimate the number of packets queued in the bottleneck queue?
Recall: Little’s Law

- For any system with no or (low) loss.

- Assume
  - mean arrival rate $X$, mean service time $T$, and mean number of requests in the system $W$

- Then relationship between $W$, $X$, and $T$:

$$ W = XT $$
Estimating Number of Packets in the Queue
TCP/Vegas CA algorithm

\[ T = T_{\text{prop}} + T_{\text{queueing}} \]

Applying Little’s Law:
\[ x_{\text{vegas}} T = x_{\text{vegas}} T_{\text{prop}} + x_{\text{vegas}} T_{\text{queueing}}, \]
where \( x_{\text{vegas}} = \frac{W}{T} \) is the sending rate.

Then number of packets in the queue is
\[ x_{\text{vegas}} T_{\text{queueing}} = x_{\text{vegas}} T - x_{\text{vegas}} T_{\text{prop}} \]
\[ = W - \frac{W}{T} T_{\text{prop}} \]
TCP/Vegas CA algorithm

maintain a constant number of packets in the bottleneck buffer

for every RTT
\{
  \text{if } W - \frac{W}{RTT} RTT_{\text{min}} < \alpha \text{ then } W \text{ ++}
  \text{if } W - \frac{W}{RTT} RTT_{\text{min}} > \alpha \text{ then } W \text{ --}
\}

for every loss
\quad W := \frac{w}{2}

queue size
TCP/Vegas Dynamics

\[ \Delta w_{\text{RTT}} \approx -(w - xRTT_{\text{min}} - \alpha) \]

\[ \Delta w_{\text{unit-time}} = -\left( \frac{w}{RTT} - \frac{x}{RTT} RTT_{\text{min}} - \frac{\alpha}{RTT} \right) = \frac{x}{RTT} RTT_{\text{min}} + \frac{\alpha}{RTT} - x \]

\[ \dot{x} = \frac{\Delta w_{\text{unit-time}}}{RTT} = \frac{x}{RTT^2} (RTT_{\text{min}} + \frac{\alpha}{x} - RTT) \]
# TCP/Reno vs. TCP/Vegas

<table>
<thead>
<tr>
<th>Congestion signal</th>
<th>TCP/Reno</th>
<th>TCP/Vegas</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>loss rate $p$</td>
<td>queueing delay $T_{queueing}$</td>
</tr>
</tbody>
</table>

## Dynamics ($x'$)

<table>
<thead>
<tr>
<th>TCP/Reno</th>
<th>TCP/Vegas</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\dot{x} = \frac{1}{RTT^2} - \frac{1}{2} p x^2$</td>
<td>$\dot{x} = \frac{x}{RTT^2} (RTT_{min} + \frac{\alpha}{x} - RTT)$</td>
</tr>
</tbody>
</table>

## Equilibrium

<table>
<thead>
<tr>
<th>$x_{reno}$</th>
<th>$x_{vegas}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{\alpha_{reno}}{RTT \sqrt{p}}$</td>
<td>$\frac{\alpha_{vegas}}{T_{queueing}}$</td>
</tr>
</tbody>
</table>

**Discussion:** Why and why not TCP/Vegas?
Interpreting Congestion Measure

- A congestion measure (loss/delay) is a signal from the network to the flows reflecting congestion.

- Another way to think of congestion measure is to think of it as “price”:
  - price goes up as the rate to a link is getting close to capacity.
  - the higher the “price”, the lower the rate.
Interpreting Congestion Measure

\[ p_f = \sum_{l \text{ uses } f} q_l \]

TCP/Reno: \[ \dot{x} = \frac{1}{RTT^2} - \frac{1}{2} px^2 = \frac{1}{2} x^2 \left( \frac{2}{RTT^2x^2} - p \right) \]

TCP/Vegas: \[ \dot{x} = \frac{x}{RTT^2} (RTT_{\text{min}} + \frac{a}{x} - RTT) = \frac{x}{RTT^2} \left( \frac{a}{x} - T_{\text{queueing}} \right) \]
Outline

- Admin and recap
- TCP/Vegas
- Network bandwidth allocation framework
  - motivation
So far our discussion is implicitly on a network with a single bottleneck link; this simplifies design and analysis:

- **Efficiency/optimality (high utilization)**
  - fully utilize the bandwidth of the link

- **Fairness (resource sharing)**
  - each flow receives an *equal* share of the link’s bandwidth
Network Resource Allocation

- It is important to understand and design protocols for a general network topology
  - how will TCP allocate resource in a general topology?
  - how should resource be allocated in a general topology?
Example: TCP/Reno Rates

Rate:

\[ x_1 = 0.26 \]
\[ x_2 = x_3 = 0.74 \]
Example: TCP/Vegas Rates

Rates: $x_1 = 1/3$
$x_2 = x_3 = 2/3$

Diagram:
- $x_1$ to $C = 1$
- $x_2$ to $C = 1$
- $x_3$ to $C = 1$
Example: Maximize Throughput

\[
\begin{align*}
\text{max} & \quad \sum_{f} x_f \\
\text{subject to} & \quad x_1 + x_2 \leq 1 \\
& \quad x_1 + x_3 \leq 1
\end{align*}
\]

Optimal: \( x_1 = 0 \)  \\
\( x_2 = x_3 = 1 \)
Example: Max-min Fairness

Max-min fairness: maximizes the throughput of the flow receiving the minimum (of resources)

• This is a resource allocation scheme used in Asynchronous Transfer Mode and some other network resource allocation proposals
Example: Max-Min

\[
\begin{align*}
\max_{x_f \geq 0} & \quad \min \{x_f\} \\
\text{subject to} & \quad x_1 + x_2 \leq 1 \\
& \quad x_1 + x_3 \leq 1
\end{align*}
\]

Rates: \( x_1 = x_2 = x_3 = 1/2 \)
Network Resource Allocation
Using Utility Functions

- A set of flows $F$
- If $x_f$ is the rate of flow $f$, then the utility to flow $f$ is $U_f(x_f)$, where $U_f(x_f)$ is a concave utility function.
- Maximize aggregate utility, subject to capacity constraints

\[
\begin{align*}
\text{max} & \quad \sum_{f \in F} U_f(x_f) \\
\text{subject to} & \quad \sum_{f : f \text{ uses link } l} x_f \leq c_l \text{ for any link } l \\
\text{over} & \quad x \geq 0
\end{align*}
\]
Example: Proportional Fairness

\[
\begin{align*}
\max_{x_f \geq 0} & \quad \sum_{f} \log x_f \\
\text{subject to} & \quad x_1 + x_2 \leq 1 \\
& \quad x_1 + x_3 \leq 1
\end{align*}
\]

\[
U_f (x_f) = \log (x_f)
\]

Optimal: \( x_1 = 1/3 \)
\( x_2 = x_3 = 2/3 \)

\[
\begin{align*}
x_1 & \quad \text{C} = 1 \\
x_2 & \quad C = 1 \\
x_3 & \quad \text{C} = 1
\end{align*}
\]
Example 3: a Utility Function

\[
\begin{align*}
\text{max} \quad & \quad - \frac{1}{4x_1} - \frac{1}{x_2} - \frac{1}{x_3} \\
\text{subject to} \quad & \quad x_1 + x_2 \leq 1 \\
& \quad x_1 + x_3 \leq 1
\end{align*}
\]

Optimal:
\[
\begin{aligned}
x_1 &= 0.26 \\
x_2 &= x_3 = 0.74
\end{aligned}
\]

\[
U_f(x_f) = -\frac{1}{RTT^2 x_f}
\]
## Summary: Allocation

<table>
<thead>
<tr>
<th>Objective</th>
<th>Allocation (x1, x2, x3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/Reno</td>
<td>0.26 0.74 0.74</td>
</tr>
<tr>
<td>TCP/Vegas</td>
<td>1/3 2/3 2/3</td>
</tr>
<tr>
<td>Max Throughput</td>
<td>0 1 1</td>
</tr>
<tr>
<td>Max-min</td>
<td>1/2 1/2 1/2</td>
</tr>
<tr>
<td>Max sum log(x)</td>
<td>1/3 2/3 2/3</td>
</tr>
<tr>
<td>Max sum of -1/(RTT^2 x)</td>
<td>0.26 0.74 0.74</td>
</tr>
</tbody>
</table>

\[
x_1 \quad x_2 \quad x_3
\]

\[
C = 1
\]
Resource Allocation Frameworks

- **Forward (design) engineering:**
  - how to determine objective functions
  - given objective, how to design effective alg

\[
\begin{align*}
\text{max} & \quad \sum_{f \in F} U_f(x_f) \\
\text{subject to} & \quad \sum_{f: f \text{ uses link } l} x_f \leq c_l \text{ for any link } l \\
\text{over} & \quad x \geq 0
\end{align*}
\]

- **Reverse (understand) engineering:**
  - understand current protocols (what are the objectives of TCP/Reno, TCP/Vegas?)
Outline

- Admin and recap
- TCP/Vegas
- Bandwidth allocation framework
  - Motivation
  - Forward: Nash Bargaining Solution (NBS)
High level picture

- given the feasible set of bandwidth allocation, we want to pick an allocation point that is efficient and fair

The determination of the allocation point should be based on “first principles” (axioms)
Network Bandwidth Allocation: Feasible Region