TCP Veno
TCP Enhancement for Transmission Over Wireless Access Networks

C. P. Fu and S. C. Liew
IEEE JSAC, 2003

Presented by J. L. Chiang
July 10, 2003
Outline

- Introduction
- TCP Veno
- Simulation Study
- Conclusions
Introduction

Conventional TCP treats packet loss as congestion notification, however, this cannot apply in error-prone wireless networks.

TCP (Reno) also suffers from heterogeneity on the internet.

- Reliability: optical v.s. wireless
- Asymmetry: ADSL
- Propagation delay: satellite v.s. terrestrial
TCP Vegas

- Proactive congestion control

\[
\begin{align*}
\text{Expected} &= \frac{cwnd}{\text{BaseRTT}} \\
\text{Actual} &= \frac{cwnd}{\text{RTT}} \\
\text{Diff} &= \text{Expected} - \text{Actual}
\end{align*}
\]

Let \( N \) be the backlog at bottleneck link:

\[
\text{RTT} = \text{BaseRTT} + \frac{N}{\text{Actual}}
\]

\[
N = \text{Actual} \times (\text{RTT} - \text{BaseRTT}) = \text{Diff} \times \text{BaseRTT}
\]

- Use \( N \) to adjust TCP window size to avoiding packet loss due to buffer overflow.
- 37%~71% improvement over Reno
Problems of Vegas

- Vegas cannot coexists with Reno, since it is less aggressive than Reno’s policy, which continues to increase window size until packet loss occurs.

- The measured backlog is not necessary the data backlog in asymmetric networks.
  - Underutilization on the wider forward path
TCP Veno

- Veno = Ve(gas) + (Re)no
- Veno use the measurement of \( N \) not as a way to adjust window size proactively, but rather as an indication of whether the connection is in a congestive state:
  - \( N < \beta \): random loss
  - \( N > \beta \): congestive loss
- Only the sender stack is modified.
- With 1% random packet loss rate, improvement up to 80% is achieved.
Modified Additive Increase

- Slow Start (unchanged)
- Additive Increase:

If \( N < \beta \) // available bandwidth not fully utilized
set \( cwnd = cwnd + 1/cwnd \) when each new ACK is received
else if \( N \geq \beta \) // available bandwidth fully utilized
set \( cwnd = cwnd + 1/cwnd \) when every other new ACK is received
Modified Additive Increase

- Fewer oscillations
- Longer large-window region
- Less congestion loss
Modified Multipliciative Decrease

- Timeout (unchanged)
  1) \( \text{ssthresh} = \frac{\text{cwnd}}{2} \);
  2) \( \text{cwnd} = 1 \);

- Fast retransmit

  1) Retransmit the missing packet
     \[ \text{ssthresh} = \frac{\text{cwnd}}{2} \]
     \[ \text{cwnd} = \text{ssthresh} + 3 \]
  2) Each time another dup ACK arrives, increment \( \text{cwnd} \) by one packet.
  3) When the next ACK acknowledging new data arrives, set \( \text{cwnd} \) to \( \text{ssthresh} \) (value in step 1).
Modified Multiplicative Decrease
Simulation A: Distinguishing Packet Loss
Simulation A: Distinguishing Packet Loss

- Random loss detection accuracy: $\frac{83}{84} = 99\%$
- Misdiagnosing rate of congestion: $\frac{2}{12} = 17\%$

<table>
<thead>
<tr>
<th>Actual type of packet loss</th>
<th>Veno TCP</th>
<th>No background traffic</th>
<th>Background traffic (UDP) with sending rate of 500kb/s</th>
<th>Background traffic (UDP) with sending rate of 1Mb/s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Congestion</td>
<td>Random</td>
</tr>
<tr>
<td>First</td>
<td>#FF4/5</td>
<td>0</td>
<td>1</td>
<td>83</td>
</tr>
<tr>
<td>Run</td>
<td>#FF1/2</td>
<td>6</td>
<td>10</td>
<td>146</td>
</tr>
<tr>
<td>Second</td>
<td>#FF4/5</td>
<td>0</td>
<td>1</td>
<td>78</td>
</tr>
<tr>
<td>Run</td>
<td>#FF1/2</td>
<td>3</td>
<td>8</td>
<td>102</td>
</tr>
<tr>
<td>Third</td>
<td>#FF4/5</td>
<td>0</td>
<td>1</td>
<td>67</td>
</tr>
<tr>
<td>Run</td>
<td>#FF1/2</td>
<td>3</td>
<td>9</td>
<td>139</td>
</tr>
</tbody>
</table>
Simulation B: Single Connection

Packet

$\text{Src } N$

$\cdot$

10Mbps

$\text{Src } 1$

Router

$\mu_f, \tau_f, B_f$

$\mu_r, \tau_r, B_r$

$\text{Dst } N$

$\text{Dst } 1$

10Mbps

Ack

Controlled loss from $10^{-4}$ to $10^{-1}$
Simulation B: Single Connection

TCP Veno

TCP Reno
Simulation C: Co-existing Connections

- 4R v.s. 2R2V
Simulation C: Co-existing Connections

- Fair-share of R-V and V-V
- Same total throughput
Simulation C: Co-existing Connections

- Non-interference between R and V
- Loss-tolerant of V (increased throughput)
Simulation D: Internet Measurements

<table>
<thead>
<tr>
<th></th>
<th>Time slots</th>
<th>10:00–11:00</th>
<th>11:00–12:00</th>
<th>12:00–13:00</th>
<th>14:00–15:00</th>
<th>15:00–16:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Th(KB/s)</td>
<td></td>
<td>65.5/64.4</td>
<td>51.3/60.3</td>
<td>45.1/54.5</td>
<td>51.5/58.7</td>
<td>70.2/73.9</td>
</tr>
<tr>
<td># TO</td>
<td>8.1/6.5</td>
<td>7.6/5.8</td>
<td>8.9/5.3</td>
<td>9.5/5.8</td>
<td>6.5/6.4</td>
<td></td>
</tr>
<tr>
<td># FF</td>
<td>19.9/11.5</td>
<td>21.6/10.5</td>
<td>24.6/18.5</td>
<td>28.3/18.7</td>
<td>14.5/14.0</td>
<td></td>
</tr>
<tr>
<td># Retran. Pkts</td>
<td>185.8/127.5</td>
<td>166.7/103.9</td>
<td>170.1/137.3</td>
<td>156.6/137.2</td>
<td>98.5/97.1</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

- Veno is desirable for:
  - Deployability
  - Compatibility
  - Flexibility

- What TCP Veno proposes is to refine Reno’s AIMD evolution over heterogeneous networks by using the complete judgment of network state estimation – congestive state or non-congestive state, rather than merely depending on packet loss occurrence.