Linux 2.4 Implementation of Westwood+ TCP with Rate-Halving: A Performance Evaluation over the Internet

Angelo Dell'Aera
Luigi Alfredo Grieco
Saverio Mascolo

Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari
Via Orabona 4
70125 Bari, Italy

ICC2004 Paris - Jun 22, 2004
Outline

➢ Background of TCP Westwood and TCP Westwood+
➢ End-to-End Bandwidth Estimation by filtering the ACK stream
➢ Linux implementation of TCP Westwood+
➢ Internet measurements
TCP Westwood Congestion Control

Key idea: use end-to-end bandwidth estimation to adaptively set $cwnd$ and $ssthresh$ after congestion instead of standard "blind" multiplicative window decrease
Standard TCP (Van Jacobson)

Typical cwnd dynamics following the AIMD paradigm
TCP Westwood Adaptive Decrease vs TCP (New) Reno blind by $\frac{1}{2}$ window shrinking
Known drawbacks of TCP Reno

- Low throughput over wireless links because losses due to unreliable links are misinterpreted as congestion

- Reno throughput is proportional to $1/RTT$, i.e. it is not fair
End-to-End Bandwidth Estimation

➢ TCP Westwood+ algorithm is based on end-to-end estimation of the bandwidth available along the TCP connection path.

➢ The estimate is obtained by filtering the stream of returning ACK packets and it is used to adaptively set the control windows when network congestion is experienced.
End-to-End Bandwidth Estimation
TCP Westwood+
Pseudo Code

➢ When 3 DUPACKs are received by the sender
  sthresh = max(2, (BWE * RTTmin) / MSS);
cwnd = sthresh;

➢ When coarse timeout expires
  sthresh = max(2, (BWE * RTTmin) / MSS);
cwnd = 1;

➢ When ACKs are successfully received
  cwnd increases as stated in RFC2581
  the end-to-end bandwidth estimate BWE is computed
TCP Westwood+

- TCP Westwood+ follows an Additive Increase Adaptive Decrease paradigm
- TCP Westwood+ improves the stability of the standard TCP multiplicative decrease algorithm
- The congestion window is decreased enough in presence of heavy congestion and not too much in presence of light congestion or losses not due to congestion
The adaptive setting of the control windows increases the fair allocation of the available bandwidth to different TCP flows.

- Setting $cwnd$ to $\text{BWE} \times \text{RTTmin}$ sustains a transmission rate $((\text{BWE} \times \text{RTTmin}) / \text{RTT})$ smaller than the bandwidth estimated at the time of congestion thus leaving room in the buffers for coexisting flows.

- This improves statistical multiplexing and fairness.
Warning…

- ACKs reach the TCP sender compressed
- Bandwidth samples

\[ b_j = \frac{d_j}{t_j - t_{j-1}} \]

contain high frequency components that cannot be filtered out by a discrete-time filter due to aliasing

\[ t_j - t_{j-1} = \text{ACK interarrival time} \]
ACK compression effects

- ACK pairs give information about the bandwidth of the last link traversed on the backward path

- To smooth ACK compression we accumulate ACKs over an RTT and then compute a bandwidth sample
An anti-aliasing filter in packet networks

\[ b_j = \frac{d_j}{\Delta_j} \quad \text{Antialiased samples} \]

\[ \Delta_j = \text{Last RTT} \]

\[ d_j = \text{all data acknowledged in the last RTT} \]
End-to-End Bandwidth Estimate

TCP Westwood+

BWE

Low pass filter

Anti ACK compression

Segments sent

Internet

Returning ACKs
End-to-End Bandwidth Estimate

The following time-invariant low-pass filter is used

\[ b_k = \frac{d_k}{RTT_k} \]

\[ \hat{b}_k = \alpha \cdot \hat{b}_{k-1} + (1 - \alpha) \cdot b_k \]
ACK compression effects

➢ We have found that **ACK compression** has very important effects on TCP

➢ **ACK compression** must be considered when doing simulation
Topology with ACK compression effects (10 Mbps)

Forward traffic:
20 TCP Westwood connections

Reverse traffic:
10 TCP long-lived NewReno connections
The 20 Westwood+ connections estimate a best-effort available bandwidth that reasonably approaches the fair share of 0.5 Mbps.
Westwood overestimates up to 100 times the fair share due to ACK compression.
Summary on bandwidth estimate

➢ TCP Westwood : one bandwidth sample computed for each ACK
   (leads to bandwidth overestimate in presence of ACK compression)

➢ TCP Westwood+ : one bandwidth sample computed for each RTT
Linux Implementation of TCP Westwood+

- Linux 2.4 TCP implementation supports the Rate-Halving congestion control algorithm

- The Rate-Halving congestion control algorithm adjusts the window by sending one segment per two acknowledgments for exactly one round trip
Linux Implementation of TCP Westwood+

➢ Rate-Halving sets the window to exactly one half of the data which was actually held in the network during the congested round trip.

➢ Rate-Halving has been slightly modified by setting the lower bound for $cwnd$ decrease to $BWE \times RTT_{min}$.
The patch has been developed to be as less intrusive as possible. In fact, it's possible to enable/disable TCP Westwood+ through the sysctl `net.ipv4.tcp_westwood`.

The sysctl allows to switch between TCP New Reno and TCP Westwood+.
Linux Implementation of TCP Westwood+

- TCP Westwood+ support was integrated in the Linux kernel release 2.4.26-pre1 (kernel 2.4) and kernel release 2.6.3-rc1 (kernel 2.6)

http://www.kernel.org
http://buffer.antifork.org/westwood/westwood.html
Internet measurements

➢ More than 4000 FTP from Bari, South Italy to:

  • panther.cs.ucla.edu (UCLA)
  • signserv.signal.uu.se (Uppsala)
  • main.penguin.it (Parma)
Uploads to panther.cs.ucla.edu (1)

File size=3.2MB, From: rigel.poliba.it, To: panther.cs.ucla.edu, Total number of uploads = 197, Average New Reno Goodput = 16.86Kbyte/s, Average Westwood+ Goodput = 25.21Kbyte/s
Uploads to *panther.cs.ucla.edu* (2)

File size = 32MB

- **Westwood+**
- **New Reno**

Goodput (KB/s)

<table>
<thead>
<tr>
<th>Date</th>
<th>Goodput (KB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fri Feb 21</td>
<td>18:15:56</td>
</tr>
<tr>
<td>Sat Feb 22</td>
<td>06:04:31</td>
</tr>
<tr>
<td>Sat Feb 22</td>
<td>20:26:44</td>
</tr>
<tr>
<td>Sun Feb 23</td>
<td>07:24:58</td>
</tr>
<tr>
<td>Sun Feb 23</td>
<td>19:06:51</td>
</tr>
<tr>
<td>Mon Feb 24</td>
<td>03:52:51</td>
</tr>
</tbody>
</table>

2003
Uploads to panther.cs.ucla.edu (3)

File size = 32MB

Goodput (KB/s)
Uploads to *panther.cs.ucla.edu* (4)
Main References

➢ L. A. Grieco, S. Mascolo
“Performance Comparison of Reno, Vegas, and Westwood+TCP Congestion Control”
ACM CCR Vol. 34 No. 2, April 2004

➢ S. Mascolo, C. Casetti, M. Gerla, S. Lee, M. Sanadidi
“TCP Westwood: bandwidth estimation for enhanced transport over wireless links”
ACM Mobicom 2001 and Winet Journal 2002

➢ L. A. Grieco and S. Mascolo,
“End-to-End Bandwidth Estimation for Congestion Control in Packet Networks”
Second International Workshop QoS-IP 2003
Thanks for the attention

Questions?