T2-6: Trace File Analysis - The Elephant Coming From Behind: Full Window, Window Update and TCP Keep-Alive’s

**Laura Chappell**
Founder  |  Wireshark University

**Betty DuBois**
Principal Consultant  |  DuBois Training & Consulting, LLC

**SHARKFEST ’08**
Foothill College
March 31 - April 2, 2008
Using Wireshark to isolate a TCP Window issue

- Expert Info
  - Errors
  - Warnings
  - Notes
    - Zero Window
    - Window Update
    - Keep Alive
    - Window Full
- Chat
- Time Column
  - Delta
  - Delta Displayed
Best Practices for Protocol Analysis

Onsite v. offsite analysis
Create a baseline when performance is acceptable
Analyze application traffic before deployment (capacity planning)

Troubleshooting Tips:
- Who complained?
- Begin as close to the user as possible
- Name captures appropriately (sue1, sue2, sue3mac, etc.)
- Move analyzer as needed or use multiple analyzers and agents
- Time-sync if using multiple analyzers
- Have taps/hubs in place for when the need arises
- Focus on the time column (delta time setting)
- Consider command-line capture (nmcap/tshark)

Security Tips:
- Baseline protocols, applications, traffic patterns
- Examine summary and protocol information for anomalies
- Look for signatures in questionable traffic
- Snort website has many signatures in the rule sets
Expert Info – Errors

Error (red): serious problem, e.g. [Malformed Packet]

- **Malformed**: malformed packet or dissector has a bug, dissection of this packet aborted

Expert Errors do NOT always mean a network or communication error, they may indicate a dissector errors
Warn (yellow): warning, e.g. application returned an "unusual" error code like a connection problem

Are the fast retransmissions the issue?

- Calculate percentage of retransmitted vs. normal packets.
- Over 1% error ratio merits further investigation. That is not the case here.

Attend Session T2-8 for further info on Retransmissions and Fast Retransmissions
Note (cyan): notable things, e.g. an application returned an "usual" error code like HTTP 404

- Zero window
- Window update
- Keep-Alive
- Duplicate ACK (Covered in Session T2-8)
The key to determining when a Zero Window is truly an issue, is how long does it stay at zero before updating.

The example below shows a host processing its data and updating the TCP window very quickly.

tcp-window-good.pcap
Wireshark creates an Expert Note whenever the TCP window size increases.

Warning: Wireshark does NOT create the Note when the window size is decreasing.

<table>
<thead>
<tr>
<th>No.</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Bytes</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>0.000034</td>
<td>192.168.1.112</td>
<td>192.168.1.103</td>
<td>TCP</td>
<td>60</td>
<td>1939 &gt; 2707 [ACK] Seq=14227933</td>
</tr>
<tr>
<td>98</td>
<td>0.000033</td>
<td>192.168.1.112</td>
<td>192.168.1.103</td>
<td>TCP</td>
<td>60</td>
<td>1939 &gt; 2707 [ACK] Seq=14227933</td>
</tr>
<tr>
<td>99</td>
<td>0.000038</td>
<td>192.168.1.112</td>
<td>192.168.1.103</td>
<td>TCP</td>
<td>60</td>
<td>1939 &gt; 2707 [ACK] Seq=14227933</td>
</tr>
<tr>
<td>100</td>
<td>0.000035</td>
<td>192.168.1.112</td>
<td>192.168.1.103</td>
<td>TCP</td>
<td>60</td>
<td>1939 &gt; 2707 [ACK] Seq=14227933</td>
</tr>
<tr>
<td>101</td>
<td>0.000043</td>
<td>192.168.1.112</td>
<td>192.168.1.103</td>
<td>TCP</td>
<td>60</td>
<td>1939 &gt; 2707 [ACK] Seq=14227933</td>
</tr>
<tr>
<td>102</td>
<td>0.000035</td>
<td>192.168.1.112</td>
<td>192.168.1.103</td>
<td>TCP</td>
<td>60</td>
<td>1939 &gt; 2707 [ACK] Seq=14227933</td>
</tr>
<tr>
<td>103</td>
<td>0.000159</td>
<td>192.168.1.112</td>
<td>192.168.1.103</td>
<td>TCP</td>
<td>60</td>
<td>1939 &gt; 2707 [ACK] Seq=14227933</td>
</tr>
<tr>
<td>104</td>
<td>0.000235</td>
<td>192.168.1.112</td>
<td>192.168.1.103</td>
<td>TCP</td>
<td>60</td>
<td>[TCP ZeroWindow] 1939 &gt; 2707</td>
</tr>
<tr>
<td>105</td>
<td>0.009293</td>
<td>192.168.1.112</td>
<td>192.168.1.103</td>
<td>TCP</td>
<td>60</td>
<td>[TCP Window Update] 1939 &gt; 2707</td>
</tr>
<tr>
<td>106</td>
<td>0.000071</td>
<td>192.168.1.103</td>
<td>192.168.1.112</td>
<td>TCP</td>
<td>1514</td>
<td>2707 &gt; 1939 [ACK] Seq=27110054</td>
</tr>
<tr>
<td>107</td>
<td>0.000060</td>
<td>192.168.1.103</td>
<td>192.168.1.112</td>
<td>TCP</td>
<td>1514</td>
<td>2707 &gt; 1939 [ACK] Seq=27110068</td>
</tr>
</tbody>
</table>
Expert Info – Notes – Keep-Alive

Normal Keep-Alive’s should be sent at regular intervals when no data is being transferred to keep the session alive.

In this trace, what do you notice about the Delta times?

<table>
<thead>
<tr>
<th>No.</th>
<th>Delta</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Bytes</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>375</td>
<td>16.074234</td>
<td>madheifer.pacific.net.au</td>
<td>10.0.52.164</td>
<td>TCP</td>
<td>60</td>
<td>TCP Keep-Alive</td>
</tr>
<tr>
<td>373</td>
<td>8.064197</td>
<td>madheifer.pacific.net.au</td>
<td>10.0.52.164</td>
<td>TCP</td>
<td>60</td>
<td>TCP Keep-Alive</td>
</tr>
<tr>
<td>371</td>
<td>4.128068</td>
<td>madheifer.pacific.net.au</td>
<td>10.0.52.164</td>
<td>TCP</td>
<td>60</td>
<td>TCP Keep-Alive</td>
</tr>
<tr>
<td>369</td>
<td>2.198012</td>
<td>madheifer.pacific.net.au</td>
<td>10.0.52.164</td>
<td>TCP</td>
<td>60</td>
<td>TCP Keep-Alive</td>
</tr>
<tr>
<td>367</td>
<td>1.133508</td>
<td>madheifer.pacific.net.au</td>
<td>10.0.52.164</td>
<td>TCP</td>
<td>60</td>
<td>TCP Keep-Alive</td>
</tr>
<tr>
<td>5966</td>
<td>0.681690</td>
<td>madheifer.pacific.net.au</td>
<td>10.0.52.164</td>
<td>HTTP</td>
<td>1514</td>
<td>TCP Retransmission Continuation</td>
</tr>
<tr>
<td>365</td>
<td>0.667083</td>
<td>madheifer.pacific.net.au</td>
<td>10.0.52.164</td>
<td>TCP</td>
<td>60</td>
<td>TCP Keep-Alive</td>
</tr>
</tbody>
</table>
Using the Time columns makes isolating faults much more efficient.

- Delta previous packet may not be seen
- Delta Displayed only care about seen packets
Receiver Congestion

SEQ 3000 – 1460 bytes of data

SEQ 2000 – 0 bytes of data; ACK 4460, Win=5840

SEQ 4460 – 1460 bytes of data

SEQ 5920 – 1460 bytes of data

SEQ 7380 – 1460 bytes of data

SEQ 2000 – 0 bytes of data; ACK 5920, Win=4380

SEQ 2000 – 0 bytes of data; ACK 7380, Win=2920

SEQ 8840 – 1460 bytes of data

SEQ 10300 – 1460 bytes of data

SEQ 2000 – 0 bytes of data; ACK 8840, Win=1460

SEQ 2000 – 0 bytes of data; ACK 8840, Win=0
Lab: Congestion

File: download-bad.cap

Select View > Time Display Format > Seconds Since Previous Displayed Packet

Sort the packets on the Time column

What is the cause of the highest delays in this trace file?

Re-sort by the number column

- Which packet filled the receive buffer?
- What was the total delay time caused by receiver congestion?
- Was packet loss and high latency the biggest problem in this trace file?
What’s Next?

Laura’s Lab Kit v9
In show bags as well as…
ISO image: www.novell.com/connectionmagazine/laurachappell.html
Wireshark University: www.wiresharkU.com
Laura’s Blog: laurachappell.blogspot.com/