Troubleshooting Application Performance Issues
June 15, 2011

Mike Canney
Principal Network Analyst | getpackets.com
Who am I?

• Mike Canney, President  getpackets.com
• canney@getpackets.com
• 319.265.0170
Agenda

• Issues with troubleshooting applications
• Creating a CDA (Capture to Disk Appliance)
• Using Pilot for “back in time” troubleshooting with a CDA and Wireshark
• Application QA Lifecycle
• Top Causes for Application Performance issues
  – Application Turns
  – TCP Window Size
  – Application Block Size issues (Inflight Data)
  – TCP Retransmissions
• Using Wireshark to create custom profiles to troubleshoot CIFS/SMB
Issues with Troubleshooting Applications (from a Network Perspective)

• Application performance issues can impact your business/customers ability to make money.
• “It’s not the Network!” - The Network is guilty until proven innocent.
• User Response time is “Relative”.
• Intermittent performance issues (often a moving target).
The “moving target”

• Analyzer placement
  – Two options
    • Move the analyzers as needed
    • Capture anywhere and everywhere

• To defend the Network, multiple capture points of the problem is the best solution.
Commercial Vs. Free Capture

• Define your capture strategy
  – Data Rates
  – What path does the application traverse?
  – What are my goals? Troubleshooting vs. Statistical information.
  – Do I need to capture every packet?
Capture to Disk Appliance (on a budget)

• What is needed?
  – dumpcap is a command line utility included with the Wireshark download to enable capture using a ring buffer.
  – Use an inexpensive PC or laptop (best to have 2 NICs or more).
  – Basic batch file to initiate capture.
  – Pilot (optional but recommended)
dumpcap example

cd \program files (x86)\wireshark
dumpcap -i 1 -s 128 -b files:100 -b filesize: 2000000 –w c:\traces\internet\sliced.pcap

This is a basic batch file that will capture off of interface 1, slice the packets to 128 bytes, write 100 trace files of ~2 Gigabytes, and write the trace file out to a pcap file.
So why did I write multiple 2 Gig trace files?

• Pilot!
• Pilot can easily read HUGE trace files.
• This allows us to utilize our CDA is ways no other analyzer can.
• I personally have sliced and diced 50 GB trace files in Pilot in a matter of seconds.
So how does this all work together?

• Directory full of 2GB trace files, each file time stamped based on when they were written to disk.

• A user calls in and complains that “the network” is slow.

• Locate that trace file based on time and date and launch Pilot.
Troubleshooting user “Network Issue”
Think about what you just saw.

• From a 2 GB trace file we were able to:
  – Look at the total Network throughput.
  – See what applications were consuming the bandwidth.
  – Identify the user that was responsible for consuming the bandwidth.
  – Identify the URI’s the user was hitting and what the response times were.
  – Drill down to the packets involved in the slow web response time in Wireshark.

• All in a matter of a few seconds.
Why focus on the Application?

• Applications are typically developed in a “golden” environment
  – Fastest PCs
  – High Bandwidth/low latency
• When applications move from test (LAN) to production (WAN) the phone starts ringing
• Usually applications go through QA Lifecycle
• Typical QA/App developers test the following:
  – Functional
  – Regression
  – Stress (server)
  – Rinse and Repeat
• What is often missing is “Networkability” testing
• All QA Lifecycles should include Networkability testing
Application Networkability Assessment (ANA)

• Identify business transactions, number of users and network conditions the application will be deployed in.
• Simulation vs. Emulation
• Simulation is very quick, often gives you rough numbers of how an application will perform over different network conciliations.
• Emulation is the only way to determine when an application will “fail”.
• Combination of both is recommended.
Top Causes for Poor Application Performance

• Application Turns
• An Application Turn is a request/response pair.
• For each “turn” the application must wait the full round trip delay.
• The greater the number of turns, the worse the application will perform over a WAN (latency bound).
App Turn

GET /assets/images/riverbed_logo.png HTTP/1.1
[TCP segment of a reassembled PDU]
[TCP segment of a reassembled PDU]
49222 > http [ACK] Seq=573 Ack=2921 Win=16060 Len=0
[TCP Window Update] 49222 > http [ACK] Seq=573 Ack=2921 Win=1
HTTP/1.1 200 OK (PNG)
GET /assets/photos/Riverbed_Cascade_home_010211.png HTTP/1.1
App Turns and Latency

• It is fairly easy to determine App Turns impact on end user response time
  – Multiply the number of App Turns by the round trip delay:
    • 10,000 turns * .050 ms delay = 500 seconds due to latency

• Note, this has nothing to do with Bandwidth or the Size of the WAN Circuit
So what causes all these App Turns?

• Size of a fetch in a Data Base call
• Number of files that are being accessed
• Loading single images in a Web Page instead of using an image map
• Number of bytes being retrieved and how they are being retrieved (block size)
TCP Window Size

• The TCP Window Size defines the host’s receive buffer.
• Large Window Sizes can sometimes help overcome the impact of latency.
• Depending on how the application was written, advertised TCP Window Size may not have an impact at all (more on this later).
TCP Inflight Data

- The amount of unacknowledged TCP data that is on the wire at any given time.
- TCP inflight data in limited by the following:
  - TCP Retransmissions
  - TCP Window Size
  - Application block size
- The amount of TCP inflight data will never exceed the receiving devices advertised TCP Window Size.
TCP Inflight Data

Wireshark 10: Graphs: SMB Read Client.pcap
TCP Retransmissions

• Every time a TCP segment is sent, a retransmission timer is started.
• When the Acknowledgement for that segment is received the timer is stopped.
• If the retransmission timer expires before the Acknowledgement is received, the TCP segment is retransmitted.
TCP Retransmissions

• Excessive TCP Retransmissions can have a huge impact on application performance.

• Not only does the data have to get resent, but TCP flow control (Slow Start) kicks into action.
Troubleshooting CIFS/SMB

• Arguably the most common File Transfer method used in businesses today.
• SMB definitely not developed with the WAN in mind.
• One of the most “chatty” protocols/applications I run into (with the exception of poorly written SQL).
Quiz

• What is faster using MS File Sharing?
  – Pushing a file to a file server?
  – Pulling a file from a file server?
Demo of SMB Profile
### My personal SMB Profile

#### Wireshark: Preferences - Profile: SMB

**Columns**

<table>
<thead>
<tr>
<th>Displayed</th>
<th>Title</th>
<th>Field type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No.</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>Time (format as specified)</td>
</tr>
<tr>
<td></td>
<td>Bytes</td>
<td>Packet length (bytes)</td>
</tr>
<tr>
<td></td>
<td>SMB Byte Count</td>
<td>Custom (emb.bcc)</td>
</tr>
<tr>
<td></td>
<td>SMB Time</td>
<td>Custom (emb.time)</td>
</tr>
<tr>
<td></td>
<td>SMB CMD</td>
<td>Custom (emb.cmd)</td>
</tr>
<tr>
<td></td>
<td>Block Size</td>
<td>Custom (emb.file rw.length)</td>
</tr>
<tr>
<td></td>
<td>TCP WIN</td>
<td>Custom (tcp.window_size)</td>
</tr>
<tr>
<td></td>
<td>INFLIGHT</td>
<td>Custom (tcp.analysis.bytes_in_flight)</td>
</tr>
<tr>
<td></td>
<td>File Data</td>
<td>Custom (emb.file)</td>
</tr>
<tr>
<td></td>
<td>SMB Bytes</td>
<td>Custom (emb.offset)</td>
</tr>
<tr>
<td></td>
<td>Source</td>
<td>Source address</td>
</tr>
<tr>
<td></td>
<td>Destination</td>
<td>Destination address</td>
</tr>
<tr>
<td></td>
<td>Info</td>
<td>Information</td>
</tr>
</tbody>
</table>

#### Properties

- **Field type**: Number
- **Field name**: 

---

**SHARKFEST ’11 | Stanford University | June 13–16, 2011**
Take Away Points

• Building your own CDA is easy to do and may fit in a majority of the areas you need to capture from
• Pilot, Pilot, Pilot, it’s not just a fancy reporting engine for Wireshark!
• Test your applications “Networkability” before they hit production.
• Use the Wireshark Profiles, they will save you a ton of time.