Visualizations and Correlations in Troubleshooting

Kevin Burns
Comcast

kevin_burns@cable.comcast.com
Comcast Technology Groups

- Cable CMTS, Modem, Edge Services
- Backbone Transport, Routing
- Converged Regional Area Networks
- Metro Ethernet Services
- National Data Center
- Digital Voice Services (CDV)
- Domain Name / DHCP Services
- Product Engineering / Development
- Home Services (TV, Internet, Security)
- Business / Commercial Services (L2VPN, MPLS, etc)
How Do It Know ???

- **HINT:** It don’t.
- Computers only know as much as **YOU** do.
- We can only apply what **WE** know.
- How do we get from “I don’t know” to solving the problem??

解答符号
Why am I talking about Visualization?

- Troubleshooting is more of an art than a science. This presentation is about how I describe my own “art”. Everyone will develop their own art (ie: methodologies).
- Nobody can teach you this. They can only help you learn to how to incorporate ideas and techniques into your own art.
- A lot can be gained from looking at different types of thinking and methods to incorporate into your own set of tools and techniques.
- Visualizing problems is the most common process people are involved in during a troubleshooting effort.
- To be a successful problem solver you need to understand how the components of visualization fit together.
Understanding Visualization Components

- In order to visualize a problem we must:
  - Be able to recognize how a problem manifests itself to its users, engineers and inside of packet captures.
  - Categorize the problem based on its manifestation behavior to users and protocol interactions
  - Determine what technical indicators exist that allow us to correlate information to visualize the problem.
  - Understand the impact of various technical indicators.

- The goal of visualization is to determine how a problem manifests itself and correlate its technical indicators to produce a diagnosis.
- Visualization is about seeing and recognizing patterns on several different levels.
- Problem solving is about utilizing visualization techniques to resolve an issue.
Components of Visualization

- **Problem Manifestation**
  - The outward or perceptible indication of a problem.
  - Determine how the problem manifests inside of a packet capture.
  - Categorize of the problem and it’s behavior.

- **Technical Indicators**
  - Characteristics of a problem’s manifestation.
  - Identify a problem’s technical indicators

- **Correlation**
  - Correlation of various technical indicators.
  - Correlate technical indicators with a problem’s manifestation
  - Look for repeatable patterns.

- **Diagnosis**
  - The foundation of a definitive diagnosis is based on correlation of a problem’s manifestation and it’s technical indicators.
You will iterate around this cycle many times as new information is gathered.
Problem Manifestation

- How is it known the problem exists?
- How is the problem viewed?
  - By users
  - By engineers
  - By management
  - In packets
- What technical indicators does the problem manifest itself through?
  - Retransmissions
  - Time-outs, Delays
  - Application Messages
- What tools can help you uncover more methods of how the problem manifests itself?
- What techniques can you use to look for patterns?
- Understand how different technical indicators relate to impact!!
Problem Categorization by OSI Model

- Problems will manifest themselves in one or more layers of the OSI Model.
- Problems are almost always isolated to a single layer.
- The first and most important step in troubleshooting is to determine what layer of the OSI model the problem lives in. If you don’t want to understand the OSI model at least understand the protocol dependencies you are dealing with.
- OSI teaches us about dependancies, that is why it’s useful.
Problem Categorization by Type

- **Loss of Connectivity**
  - Complete and total loss of end to end connectivity at one or more layers.
  - Application failures, TCP Resets, Ping failures

- **Intermittent Connectivity**
  - Inconsistent end to end connectivity at one or more layers.
  - Dropped packets, sessions

- **Degraded Performance**
  - End to end connectivity is good but performance over the connection is suffering
  - Low Throughput, Latency impact

- **Unknown**
  - Technical indicators are unknown (concentrate on changes in state).
Case Study: Manifestation of Server Delay

Application
Remedy Ticketing System

Symptoms
- User experiencing minute long delays when performing lookups.
- Network path appears to be clean. No loss or latency.

Manifestation
- Problem manifests as delay
- Delay is obvious in the packets
Case Study: Manifestation of Server Delay

What do you see as the manifestation of the problem? Does it correlate with the user experience?
Why is there a 79 second pause between the client request and server response? Take note of the TCP Delayed ACK as well.
**Server to Backend DB Visualization**

### Technical Indicators:
- **Delay**
- **TCP SEQ+LEN=ACK**
- **Application Request/Response Behavior**

Why does the Remedy Server stop talking to the Database for 59 seconds after ACKing all responses???
Correlation of Client/Server Visualizations

Client → Remedy Server → Database Server

RPC Call
Sport: EPH Dport: 1605

Client IP: 172.30.16.163
Server IP: 172.30.1.134

95ms

TCP ACK

Client IP: 172.30.16.163
Server IP: 172.30.1.134

DB REQUEST
Sport: EPH Dport: 1521

Client IP: 172.30.1.134
Server IP: 172.30.0.71

79 sec

DB RESPONSE
Sport: 1521 Dport: EPH

Client IP: 172.30.1.134
Server IP: 172.30.0.71

95ms

RPC REPLY
Sport: EPH Dport: 1605

Client IP: 172.30.16.163
Server IP: 172.30.1.134

79 sec

DB RESPONSE
Sport: 1521 Dport: EPH

Client IP: 172.30.1.134
Server IP: 172.30.0.71

79 sec
What are Technical Indicators?

- Assuming the correct packets have been captured, the problem will always exist inside of the packets.
- Technical Indicators are feedback mechanisms found in packet communications. *(sometimes you really have to dig for them)*
- They are not symptoms.
  - *I tend to avoid using the word symptom as people tend to associate it with being the cause.*
- Problems will exist inside of packets in several ways
  - Explicit packet feedback mechanisms
  - Implicit packet feedback mechanisms
  - Extrapolated Data and Measurements
  - Behavior and Relationship Based (Correlation)
Feedback Mechanisms

- Assuming the correct packets have been captured, the problem will always exist inside of the packets.
- Problems will exist inside of packets in several ways
  - Explicit packet feedback mechanisms:
    - TCP (FIN, RST)
    - Application Messages
    - ICMP return types/codes.
  - Implicit packet feedback mechanisms:
    - Timing
    - Behavior
    - Other Correlative Factors
  - Extrapolated Data and Measurements
    - Latency
    - Throughtput
    - Examples, Behavior, Relationships
Explicit Feedback Mechanisms

Why is TCP waiting 3 seconds to retransmit the first lost segment?

Technical Indicators:
Explicit: Application Feedback
Implicit: Timing (Delay)

Application Gives Up!
Implicit Feedback Mechanisms

Why did .32 wait 200ms before retransmitting the lost segment?

Why did .32 not Fast Retransmit after receiving 3 duplicate ACKs?

Technical Indicators:
Explicit: TCP Retransmission
Implicit: Timing (Delay)
Behavior (not Fast Retransmitting)
Extrapolated Data & Measurements

Technical Indicators:
Lost Packets and TCP Retransmissions

Packet Loss = 0.003 (.3%)
## Finding Round Trip Latency

### Technical Indicators:

**Timing (Round Trip Time)**

<table>
<thead>
<tr>
<th>No.</th>
<th>Delta.t</th>
<th>Destination</th>
<th>Source</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>7812</td>
<td>0.049236298</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>9086</td>
<td>0.049224853</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>8245</td>
<td>0.049217225</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>21</td>
<td>0.049198151</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>63</td>
<td>0.049182892</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>7037</td>
<td>0.049177170</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>37</td>
<td>0.049169540</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>8187</td>
<td>0.049148560</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>8935</td>
<td>0.049148559</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>7051</td>
<td>0.049144745</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>10</td>
<td>0.049143923</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>9209</td>
<td>0.049139023</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>8876</td>
<td>0.049133301</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>8791</td>
<td>0.049131393</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
<tr>
<td>9138</td>
<td>0.049125672</td>
<td>72.27.37.13</td>
<td>172.28.85.156</td>
<td>TCP</td>
<td>46646 &gt; comnplex-link [ACK] seq=9171745 Ack=1 win=588801 seq=9171746 Ack=1 win=588801</td>
</tr>
</tbody>
</table>

**Round Trip Latency = 49ms**

Sorting by Delta Time lets us see the round trip latency!!
Throughput Measurement

http://www.switch.ch/network/tools/tcp_throughput/

Maximum throughput with a TCP window of 64 KByte and RTT of 49.0 ms <= 10.45 Mbit/sec.

Throughput is at 70% of theoretical max using 64K Buffers

Window scaling is enabled. Shouldn’t it have more TCP tx buffers to use?
Useful Technical Indicators

- **Timing Based**
  - Delta Time
    - Latency / Delay measurements
  - Relative Time
    - Throughput and Response Times
  - Absolute Time
    - Correlation to log files

- **TCP Based**
  - SYN, FIN, Reset
  - Retransmissions & Out of Order Packets
  - ACKs: Dup, Triple, Delayed, SACK
  - Windowing: Window Size & Window Full Messages

- **Application Based**
  - Transaction ID’s
  - Control Messages
    - Open, Close, Abort

- **Measurements**
  - Service Response Time
  - Latency & Throughput
  - Other Delay
Measurement Techniques

- **Standard Columns**
  - Delta Time: Sorting to find latency
  - Relative Time: Find request/response delays

- **Custom Columns**
  - IP: ip.ttl, ip.id
  - TCP: tcp.seq, tcp.ack, tcp.len, tcp.options.sack
  - Application Specific (transaction/message IDs)

- **Service Response Times**
  - Use to find application delays

- **Expert**
  - Best used to look for TCP behavior (reactions to conditions on the wire)
Protocol Layer Techniques

- **IP Based**
  - Use TTL column to visualize packet flow through routers
  - Use IPID column to visualize packet loss.

- **ICMP Based**
  - Destination Unreachable (Fragmentation, TTL, ACL’s, Host)

- **TCP Based**
  - Out of Order Packets: Look for SACKs in opposite direction. Indicates possible packet loss or network queuing or asynchronous routing issues.
  - ACK: Useful to prove a request arrived at a destination
  - Dup ACKs: Triple Dup ACKs indicate host not using Fast Retransmit algorithm.
  - Delayed ACKs: Indicates TCP waiting for an application.
  - Windowing: Full windows may indicate application problems or lack of TCP buffering (scaling needed).
Case Study: Assisting the Manifestation Process

Application
Radius authentication was suffering due to “perceived” packet loss

Symptoms
- WIFI clients were unable to authenticate through a specific site/router
- Network path appears to be clean. No obvious loss or latency.

Manifestation
- Problem manifests as loss of connectivity
- Location of problem local to a single site.
# Case Study: Radius Authentication Failures

<table>
<thead>
<tr>
<th>No.</th>
<th>del.t</th>
<th>rel.t</th>
<th>Destination</th>
<th>Source</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000</td>
<td>0.00000</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=463)</td>
</tr>
<tr>
<td>2</td>
<td>0.00008</td>
<td>0.00008</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=463)</td>
</tr>
<tr>
<td>3</td>
<td>0.00013</td>
<td>0.00021</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=463)</td>
</tr>
<tr>
<td>4</td>
<td>0.00075</td>
<td>0.00078</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>RADIUS</td>
<td>Accounting-Response(5) (id=118, l=20)</td>
</tr>
<tr>
<td>5</td>
<td>0.00002</td>
<td>0.00009</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>RADIUS</td>
<td>Accounting-Response(5) (id=118, l=20)</td>
</tr>
<tr>
<td>6</td>
<td>1.45403</td>
<td>1.45484</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=510)</td>
</tr>
<tr>
<td>7</td>
<td>0.00013</td>
<td>1.45485</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=510)</td>
</tr>
<tr>
<td>8</td>
<td>0.00012</td>
<td>1.45486</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=510)</td>
</tr>
<tr>
<td>9</td>
<td>0.00069</td>
<td>1.45556</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>RADIUS</td>
<td>Accounting-Response(5) (id=118, l=20)</td>
</tr>
<tr>
<td>10</td>
<td>0.67394</td>
<td>2.12950</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=491)</td>
</tr>
<tr>
<td>11</td>
<td>0.00012</td>
<td>2.12951</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=491)</td>
</tr>
<tr>
<td>12</td>
<td>0.00015</td>
<td>2.12953</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=491)</td>
</tr>
<tr>
<td>13</td>
<td>0.01140</td>
<td>2.14094</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>RADIUS</td>
<td>Accounting-Response(5) (id=118, l=20)</td>
</tr>
<tr>
<td>14</td>
<td>0.00028</td>
<td>2.14097</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>RADIUS</td>
<td>Accounting-Response(5) (id=118, l=20)</td>
</tr>
<tr>
<td>15</td>
<td>1.354180</td>
<td>3.495152</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=510)</td>
</tr>
<tr>
<td>16</td>
<td>0.00017</td>
<td>3.495169</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=510)</td>
</tr>
<tr>
<td>17</td>
<td>0.00004</td>
<td>3.495173</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=510)</td>
</tr>
<tr>
<td>18</td>
<td>0.00121</td>
<td>3.496392</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>RADIUS</td>
<td>Accounting-Response(5) (id=118, l=20)</td>
</tr>
<tr>
<td>19</td>
<td>1.693790</td>
<td>5.190182</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=534)</td>
</tr>
<tr>
<td>20</td>
<td>0.00013</td>
<td>5.190195</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=534)</td>
</tr>
<tr>
<td>21</td>
<td>0.00004</td>
<td>5.190199</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=534)</td>
</tr>
<tr>
<td>22</td>
<td>0.00081</td>
<td>5.190191</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Response(5) (id=118, l=20)</td>
</tr>
<tr>
<td>23</td>
<td>0.892666</td>
<td>6.083678</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=506)</td>
</tr>
<tr>
<td>24</td>
<td>0.00015</td>
<td>6.083693</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=506)</td>
</tr>
<tr>
<td>25</td>
<td>0.00017</td>
<td>6.083710</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>RADIUS</td>
<td>Accounting-Request(4) (id=118, l=506)</td>
</tr>
<tr>
<td>26</td>
<td>0.000820</td>
<td>6.084530</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>RADIUS</td>
<td>Accounting-Response(5) (id=118, l=20)</td>
</tr>
<tr>
<td>27</td>
<td>0.00052</td>
<td>6.084582</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>RADIUS</td>
<td>Accounting-Response(5) (id=118, l=20)</td>
</tr>
</tbody>
</table>

How do we find and visualize packet loss?
### Case Study: Visualize Sessions

#### Technical Indicator

Number of packets in each session.

#### Technique

Use Columns to Visualize Sessions

<table>
<thead>
<tr>
<th>No.</th>
<th>del.t</th>
<th>rel.t</th>
<th>Destination</th>
<th>Source</th>
<th>dst.port</th>
<th>src.port</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>0.000000</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>21503</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.000008</td>
<td>0.000008</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>21503</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.000013</td>
<td>0.000021</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>21503</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.000759</td>
<td>0.000780</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.000029</td>
<td>0.000089</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.454033</td>
<td>1.454842</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.000013</td>
<td>1.454855</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.000012</td>
<td>1.454867</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.000698</td>
<td>1.455565</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.673942</td>
<td>2.129507</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.000012</td>
<td>2.129519</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.000015</td>
<td>2.129534</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.011410</td>
<td>2.140944</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.000028</td>
<td>2.140972</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1.354180</td>
<td>3.495152</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.000017</td>
<td>3.495169</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.000004</td>
<td>3.495173</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Request(4) (id=118,</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0.001219</td>
<td>3.496392</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>1.693790</td>
<td>5.190182</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.000013</td>
<td>5.190195</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0.000004</td>
<td>5.190199</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>0.000813</td>
<td>5.191012</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0.892666</td>
<td>6.083678</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.000015</td>
<td>6.083693</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0.000017</td>
<td>6.083710</td>
<td>172.30.16.147</td>
<td>69.252.208.133</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0.000820</td>
<td>6.084530</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>0.000052</td>
<td>6.084582</td>
<td>69.252.208.133</td>
<td>172.30.16.147</td>
<td>21503</td>
<td>1813</td>
<td>RADIUS Accounting-Response(5) (id=118,</td>
<td></td>
</tr>
</tbody>
</table>
Notice the 2 second delays manifest themselves after packets are filtered down to a single session!
Case Study: Visualization of Packet Loss

Technique

Use IP ID and TTL to track packet flow through a router

TTLs allow us to see packet loss inside of the router. IPID=56388 is never shown with TTL=254

IPID=53855 routed successfully (TTL -1 )
Case Study: Correlating for Visibility

**Technique**

Correlation of data from different columns in Wireshark allows us to visualize the packet loss inside the router and the attempts by the application to recover from it.
Extrapolation Techniques

- Application Based
  - Always attempt to decode the application layer.
  - Look for hints in the packet hex bytes that may indicate what the protocol is.
  - Look for explicit messages that indicate application behavior or reactions to conditions on the wire.
  - Find protocol fields that allow you to track requests and responses.
  - Associate application messages and behavior to reactions and recovery mechanisms in the transport layer (ie: TCP).
Case Study: Manifestation of Server Delay

Application
Performance degradation with database transactions.

Symptoms
- Transactions which should take less than one second are taking up to (5) seconds causing the application to disconnect.
- Network path appears to be clean. No obvious loss or latency.

Manifestation
- Problem manifests as delay
- Location of Delay uncertain.
Case Study: Slow Database Transactions

Technical Indicators:
Expert shows no obvious or relevant indicators found.
Case Study: Slow Database Transactions

**Technique**

Look in Hex Data for a hint on what the protocol may be.
Case Study: Slow Database Transactions

Technical Indicators (after protocol forcing to LDAP)
LDAP Bind Time is very slow.
Case Study: Slow Database Transactions

Why does it take the LDAP server nearly 5 seconds to respond to the Bind request??
Case Study: Manifestation of Client Delay

Application
UNIX servers and VMs.

Symptoms
- UNIX admins are reporting very slow response times running SUDO level commands.
- Network path appears to be clean. No obvious loss or latency.

Manifestation
- Problem manifests as delay
- Location of Delay uncertain.
Case Study: Sudo Command Slow

LDAP Service Response Time statistics: sudo_slow.pcap

<table>
<thead>
<tr>
<th>Index</th>
<th>Procedure</th>
<th>Calls</th>
<th>Min SRT</th>
<th>Max SRT</th>
<th>Avg SRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Search</td>
<td>2104</td>
<td>0.000423</td>
<td>0.070427</td>
<td>0.036691</td>
</tr>
<tr>
<td>0</td>
<td>Bind</td>
<td>1</td>
<td>0.000214</td>
<td>0.000214</td>
<td>0.000214</td>
</tr>
</tbody>
</table>

Wireshark: Expert Infos

Errors: 0 (0)  Warnings: 0 (0)  Notes: 0 (0)  Chats: 3 (4)  Details: 4  Packet Comments: 0

Close
## Case Study: Sudo Command Slow

![Wireshark screenshot showing packet analysis]

### Technical Indicators
- Large delay seen in delta time

**Sorting by Delta Time manifests obvious delays!**
Case Study: Sudo Command Slow

Technical Indicators

LDAP Unbind Time is very slow.

Client waits 97 seconds before unbinding the LDAP connection.
What is Correlation?

- The goal of correlation is to map the problem’s method of manifestation to what is happening in the packets!!
- The process of correlating technical indicators must be understood, you cannot automate anything you have never done manually.
- You need to understand the protocols and the tools, know how Wireshark thinks!!

The Network is Slow!

INFO

Problem Manifestation

Technical Indicators

Correlation

Diagnosis
Correlation Best Practices

- The correlation process starts by understanding how a problem manifests itself.
- Get as much information from the users and technical staff as possible.
- Ask how it is known the problem actually exists.
- Always analyze from the client’s perspective first.
- Look for small patterns that can represent the problem as a whole.
  - A complex problem can often be represented by 10 packets or less.
- Visualize and understand requests and responses. Be the app!!
  - You cannot automate this part unless you understand how to do it manually.
- Understand the relationship between different technical indicators.
- Use visualization techniques for large amounts of packets.
  - Graphs, expert, column sorting.
Packet Based Correlations

<table>
<thead>
<tr>
<th>No.</th>
<th>delta.t</th>
<th>Destination</th>
<th>Source</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000</td>
<td>68.85.204.170</td>
<td>76.96.35.70</td>
<td>TCP</td>
<td>40335 &gt; afs3-vlserver [SYN] Seq=0 Win=5840 Len=0 MSS=1460</td>
</tr>
<tr>
<td>2</td>
<td>0.00019</td>
<td>76.96.35.70</td>
<td>68.85.204.170</td>
<td>TCP</td>
<td>afs3-vlserver &gt; 40335 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0</td>
</tr>
<tr>
<td>3</td>
<td>0.00115</td>
<td>68.85.204.170</td>
<td>76.96.35.70</td>
<td>TCP</td>
<td>40335 &gt; afs3-vlserver [ACK] Seq=1 Ack=1 Win=6144 Len=0</td>
</tr>
<tr>
<td>4</td>
<td>0.00018</td>
<td>68.85.204.170</td>
<td>76.96.35.70</td>
<td>TCP</td>
<td>[TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>5</td>
<td>0.00015</td>
<td>76.96.35.70</td>
<td>68.85.204.170</td>
<td>TCP</td>
<td>afs3-vlserver &gt; 40335 [ACK] Seq=1 Ack=257 Win=6912 Len=0</td>
</tr>
<tr>
<td>6</td>
<td>0.00014</td>
<td>76.96.35.70</td>
<td>68.85.204.170</td>
<td>TCP</td>
<td>[TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>7</td>
<td>0.00011</td>
<td>76.96.35.70</td>
<td>68.85.204.170</td>
<td>TCP</td>
<td>[TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>8</td>
<td>0.99976</td>
<td>76.96.35.70</td>
<td>68.85.204.170</td>
<td>HTTP</td>
<td>HTTP/1.1 100 Continue</td>
</tr>
<tr>
<td>9</td>
<td>0.00006</td>
<td>76.96.35.70</td>
<td>68.85.204.170</td>
<td>TCP</td>
<td>afs3-vlserver &gt; 40335 [RST, ACK] Seq=27 Ack=257 Win=6912 Len=0</td>
</tr>
<tr>
<td>10</td>
<td>0.00031</td>
<td>68.85.204.170</td>
<td>76.96.35.70</td>
<td>TCP</td>
<td>[TCP Dup ACK 4#1] 40335 &gt; afs3-vlserver [ACK] Seq=257 Ack=257</td>
</tr>
</tbody>
</table>

Technical Indicators

TCP Reset
Delay (delta time)

TCP Reset correlates to a 1 second time out!
Behavior Based Correlations

TCP columns allow us to prove all keepalives were received yet the application still times out after 10 seconds and closes the connection.

<table>
<thead>
<tr>
<th>No.</th>
<th>rel.t</th>
<th>Destination</th>
<th>Source</th>
<th>Protocol</th>
<th>Info</th>
<th>tcp.seq</th>
<th>tcp.ack</th>
<th>tcp.len</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>67.178.2.242</td>
<td>68.87.8.74</td>
<td>COPS</td>
<td>COPS Keep-Alive (KA)</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>0.003990</td>
<td>68.87.8.74</td>
<td>67.178.2.242</td>
<td>COPS</td>
<td>COPS Keep-Alive (KA)</td>
<td>1</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>0.203947</td>
<td>67.178.2.242</td>
<td>68.87.8.74</td>
<td>TCP</td>
<td>pktcable-cops &gt; 51454 [ACK]</td>
<td>9</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2.873947</td>
<td>76.96.180.242</td>
<td>68.87.8.74</td>
<td>COPS</td>
<td>COPS Keep-Alive (KA)</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>2.875502</td>
<td>68.87.8.74</td>
<td>76.96.180.242</td>
<td>COPS</td>
<td>COPS Keep-Alive (KA)</td>
<td>1</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>3.072271</td>
<td>76.96.180.242</td>
<td>68.87.8.74</td>
<td>TCP</td>
<td>pktcable-cops &gt; 54298 [ACK]</td>
<td>9</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>4.256548</td>
<td>67.178.2.242</td>
<td>68.87.8.74</td>
<td>COPS</td>
<td>COPS Keep-Alive (KA)</td>
<td>1</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>4.260500</td>
<td>68.87.8.74</td>
<td>67.178.2.242</td>
<td>COPS</td>
<td>COPS Keep-Alive (KA)</td>
<td>9</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>4.460304</td>
<td>67.178.2.242</td>
<td>68.87.8.74</td>
<td>TCP</td>
<td>pktcable-cops &gt; 51454 [ACK]</td>
<td>9</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>10.004980</td>
<td>67.178.2.242</td>
<td>68.87.8.74</td>
<td>COPS</td>
<td>COPS Client-Close (CC)</td>
<td>17</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>10.005114</td>
<td>67.178.2.242</td>
<td>68.87.8.74</td>
<td>TCP</td>
<td>pktcable-cops &gt; 51454 [FIN, PSH]</td>
<td>33</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>10.009087</td>
<td>68.87.8.74</td>
<td>67.178.2.242</td>
<td>TCP</td>
<td>51454 &gt; pktcable-cops [FIN, ACK]</td>
<td>17</td>
<td>33</td>
<td>0</td>
</tr>
</tbody>
</table>

Technical Indicators
- TCP SEQ-ACK-LEN
- Relative Time
Data Extrapolation Revisited

7Mb/sec with .3% packet loss
Measurement Based Correlations

Correlation of Bytes in Flight and Receiver Window Size indicates inefficient use of available receiver buffers...... but why?
Measurement Based Correlations

Can you spot the correlation that visualizes the problem?
Correlation Techniques

- Know where the analyzer is
  - Use TTL value to determine the location of packet collection

- Identify Client and Server
  - Always analyze from the perspective of the client first

- Identify Requests and Responses
  - Important to be able to measure transaction times and understand application behavior.

- Associate Packets to Process
  - Look for manifestation behavior in the packets
  - Utilize hex data to learn more about the application

- Look for obvious timing indicators that can be correlated with behavior. Common timers are: 1, 2, 5, 10, 30, 60, 120… (seconds)

- Reduce the scope of the problem to as few packets as possible.
  - Concentrate on single sessions.
Useful Visualizations
TCP Sequence and Acknowledgements

Sender (Sequence + Length) = Receiver ACK Number

SEQ(1) + LEN(8) = ACK(9)

TCP sequence, acknowledgement, and length fields are invaluable at proving a packet arrived at a destination.
TCP Session Visualization
TCP Selective Acknowledgements

Filtering on TCP Selective Acknowledgement packets allows us to see the manifestation of unidirectional packet loss.
Filtering on a single direction and sorting by the IP ID field allows us to visualize unidirectional packet loss.
Validation using IP Identification

Wireshark is confused by duplicate packets and thinks there are DUP ACKs and Retransmissions occurring. IP ID field allows us to see the duplicate IP packets.
Case Study: Bad TCP Stack

Application
Web Server admins complaining about drop in throughput due to packet loss

Symptoms
- Throughput was perceived as being overly impacted due to packet loss
- Network path tested out as clean. No obvious loss or latency. Extremely minor levels of packet loss were seen (< .1%)

Manifestation
- Problem manifests as performance degradation
- Issues appeared to be localized to a single hosting site
### Case Study: Bad TCP Stack

Round Trip Time is roughly 52-60ms

Web Server does not support TCP Selective Acknowledgements
Case Study: Bad TCP Stack (Expert)
Case Study: Web Server Bad RTO

<table>
<thead>
<tr>
<th>No</th>
<th>del.t</th>
<th>rel.t</th>
<th>Destination</th>
<th>Source</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>17327</td>
<td>0.000001000</td>
<td>2.124852000</td>
<td>Client</td>
<td>Bad Web Server</td>
<td>TCP</td>
<td>[TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>17328</td>
<td><em>REF</em></td>
<td><em>REF</em></td>
<td>Client</td>
<td>Bad Web Server</td>
<td>TCP</td>
<td>[TCP Previous segment not captured] [TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>17329</td>
<td>0.000160000</td>
<td>0.000160000</td>
<td>Client</td>
<td>Bad Web Server</td>
<td>TCP</td>
<td>[TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>17330</td>
<td>0.000184000</td>
<td>0.000200000</td>
<td>Client</td>
<td>Bad Web Server</td>
<td>TCP</td>
<td>[TCP Dup ACK 17329#1] 51669-80 [ACK] Seq=205 Ack=18543101 Wnd=3845632 Len=0</td>
</tr>
<tr>
<td>17331</td>
<td>0.000100000</td>
<td>0.000210000</td>
<td>Bad Web Server</td>
<td>Client</td>
<td>TCP</td>
<td>[TCP segment of a reassembled PDU]</td>
</tr>
</tbody>
</table>

262ms before retransmitting the lost segment??
Case Study: Bad Web Server Performance

Web Server does not appear to react very well to any level of packet loss
How should TCP recover from packet loss?

TCP should enable and use SACK to send/receive better feedback about performance. Retransmissions should occur as close to the RTT as possible.
Good TCP Recovery vs Bad TCP Recovery

Good Web Server transmits 87292 bytes at RTT (44ms) during recovery

Bad Web Server only transmits 7678 bytes at RTT (60ms) during recovery
RTT is barely visible in the delta time since recovery ramps up quickly.
Bad TCP Recovery Visualization (I/O Graph)

RTT is very visible during a very inefficient recovery.
Bad TCP Stack Recovering with Slow Start

Bad TCP stack using slow start instead of congestion control after packet loss.
Thank You !!

Questions?

(feel free to email me at kevin_burns@cable.comcast.com)