BI-8 Wireshark Software Case Studies
4:45p – 6:00p  Mon June 25 2012

Megumi Takeshita
Founder  |  ikeriri network service co.,ltd.
About Megumi Takeshita 竹下恵

• Founder, ikeriri network service co.,ltd ← Enterprize solution, nortel networks ← IS Bay Network

• 10+ books about packet capturing, analysis, inspection, and consulting

• Reseller of Riverbed Technology (former CACE technologies) and Metageek in Japan

• Packet capturing Otaku (geek) from Ethereal, 1st Sharkfest!
before capturing

- Clear DNS C:\>ipconfig /cleardns
- Stop firewalls, anti-spywares and others
- Stop service like UPnP(SSID), VPN and many

- C:\>netstat -a | find "LISTEN" ; netstat -ab
- Check NIC error, discards

C:\>netstat -e
Setting Wireshark

- Adding Wireshark program path into system variable (set Path=%Path%;C:\Pro...)
- Check interface index number (thark -D)
- Add columns according to the field catching up
- To see latency, add fields tcp.time_delta
- Set Time display format previous displayed packet
### Fragment

- **Original frame**

<table>
<thead>
<tr>
<th>EthernetII (14)</th>
<th>IP (20)</th>
<th>ICMP (8)</th>
<th>Message 1500</th>
</tr>
</thead>
</table>

- **Fragment 1/2 in Ethernet MTU**

<table>
<thead>
<tr>
<th>EthernetII (14)</th>
<th>IP (20)</th>
<th>ICMP (8)</th>
<th>Message 1472</th>
</tr>
</thead>
</table>

- **Fragment 2/2 in Ethernet MTU**

<table>
<thead>
<tr>
<th>EthernetII (14)</th>
<th>IP (20)</th>
<th>ICMP (8)</th>
<th>Message 28</th>
</tr>
</thead>
</table>
## Testing packet size

### ICMP
- ping host –l message size (MTU-28) -f

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Message Size</th>
<th>MTU</th>
</tr>
</thead>
<tbody>
<tr>
<td>EthernetII</td>
<td>IP</td>
<td>ICMP</td>
</tr>
<tr>
<td>(14)</td>
<td>(20)</td>
<td>(8)</td>
</tr>
</tbody>
</table>

### TCP
- Segment size
  - MSS = 1460

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Message Size</th>
<th>MTU</th>
</tr>
</thead>
<tbody>
<tr>
<td>EthernetII</td>
<td>IP</td>
<td>TCP</td>
</tr>
<tr>
<td>(14)</td>
<td>(20)</td>
<td>(20)</td>
</tr>
</tbody>
</table>

### UDP
- Datagram size
  - MTU=1500–1472

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Message Size</th>
<th>MTU</th>
</tr>
</thead>
<tbody>
<tr>
<td>EthernetII</td>
<td>IP</td>
<td>UDP</td>
</tr>
<tr>
<td>(14)</td>
<td>(20)</td>
<td>(8)</td>
</tr>
</tbody>
</table>

- NTT East MTU 1454Bytes (MSS 1414)
- NTT West FTTH MTU 1438Bytes (MSS 1398)
- GRE + IPsec (transport mode) 1440
- GRE + IPsec (tunneling mode) 1420
Check negotiation of TCP

See first 2 packet of 3way handshake
（初期ウインドウサイズはOS等で指定）

- Window Size, SACK, MSS, Window Scaling
- Some router may rewrite this section via NAPT
- Follow TCP stream and Use coloring and Ctrl+Space
Using iperf and tfgen (made in Japan)

- Throughput -> iperf
- Traffic -> tfgen
Overview of troubleshoot

- If we know the error obviously, see difference from OK and NG packet to see packet in micro range (field).
- No idea of trouble, capture packet at more than 2 locations to see packet in macro range (statistics).
- Expert Info say many things automatically.
- Think of packet lost -> Ignore (Ctrl+I).
• To see errors and counting the number of packet, set Y axis to packet/sec (histogram)

• To see performance and throughput, set Y axis to bit/sec (line)
• In old days we use sampling technologies like SNMP, MRTG, and many flow analysis such as Cisco NetFlow, sFlow, iFlow

• But small packet (64 bytes – 100 bytes) may be ignored. Some small packet is important symptom of analysis (ARP / TCP SYN / HTTP GET and others)
Actual capture rate

- Typical Intel’s GigaNIC (e1000), typical Dell PowerEdge2850 / Xeon 2.8GHz RAM 1GB (PC3200, DDR2, 400MHz)

- Thread should be 140Mbps in Frame size = 64

140Mbps程度でドロップ発生

140Mbps程度でドロップ発生

140Mbps

CPU負荷率 [フレーム長 64byte]

140Mbps程度で
CPU負荷率 100%

140Mbps

CPU負荷率 [フレーム長 64byte]
Another frame size

- Frame size = 200, actual rate 400Mbps
- Frame size = 1500, may be ok, no problem.
Ooops in non-sampling

• In case of frame size is 1500, there may be some drops (it is not non-sampling)
• Actually, customer want to see most highest point of traffic, so if the pcap file do not contains all packet (some ignored) no use.
• Off course please order TurboCap from us (^^)
For non-sampling inspection

- MMMM packets received by the application
- NNNN packets accepted by the filter and dumped to disk ummm
- Optimise I/O access flow
- packet -> IRQ -> SVC -> driver -> OS
- Use 6 cores Xeon-L5640 and 24GB RAM! (power resolve things and no page files)
- Stop tcpdump and create program using pcap libraries in C/C++
- Pcap -> standard output -> FIFO -> SQLite
- 3 month no problem
Thank you