

# Wireshark Developer and User Conference

## BI-8 Wireshark Software Case Studies

4:45p – 6:00p Mon June 25 2012

**Megumi Takeshita**

Founder | ikeriri network service co.,ltd.

**SHARKFEST '12**

UC Berkeley

June 24-27, 2012

# About Megumi Takeshita 竹下恵

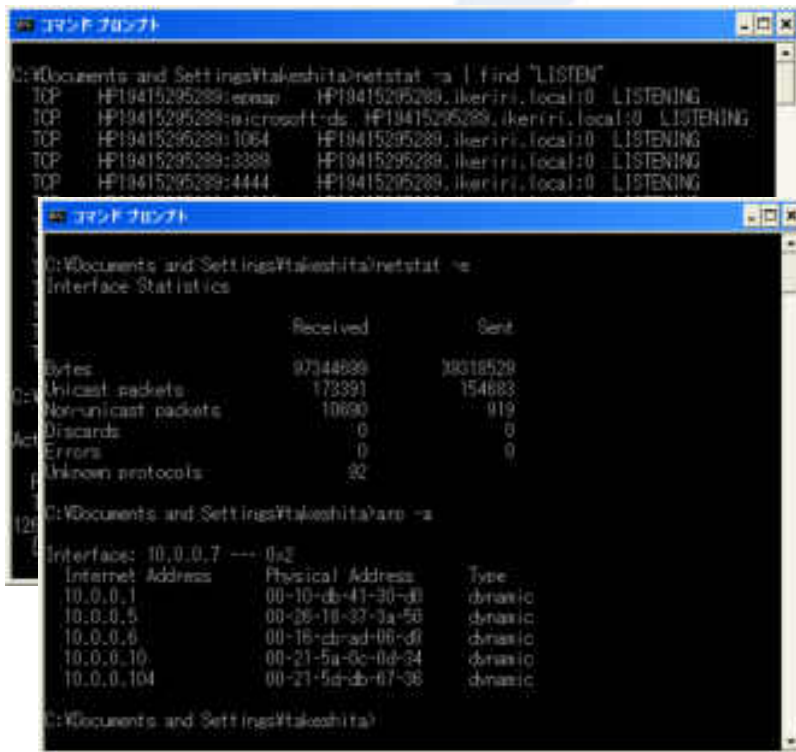


- Founder, ikeriri network service co.,ltd ← Enterprize solution, nortel networks ← IS Bay Network
- 10+ books about packet capturing, analysis, inspection, and consulting
- Reseller of Riverbed Technology ( former CACE technologies ) and Metageek in Japan
- Packet capturing Otaku ( geek ) from Ethereal, 1<sup>st</sup> Sharkfest !



# before capturing

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



```
C:\Documents and Settings\takeshita>netstat -a | find "LISTEN"
TCP    HP19415295289:esmap  HP19415295289:ikeriri.local:0 LISTENING
TCP    HP19415295289:microsofts HP19415295289:ikeriri.local:0 LISTENING
TCP    HP19415295289:1064    HP19415295289:ikeriri.local:0 LISTENING
TCP    HP19415295289:3389    HP19415295289:ikeriri.local:0 LISTENING
TCP    HP19415295289:4444    HP19415295289:ikeriri.local:0 LISTENING

C:\Documents and Settings\takeshita>netstat -e
Interface Statistics

              Received              Sent
Bytes          97344939          38318529
Unicast packets 173391            154683
Non-unicast packets 10690            919
Discards        0                  0
Errors          0                  0
Unknown protocols 32

C:\Documents and Settings\takeshita>arp -a
Interface: 10.0.0.7 --- 0x2
Internet Address      Physical Address      Type
10.0.0.1               00-10-d6-41-30-d0     dynamic
10.0.0.5               00-26-10-37-3a-56     dynamic
10.0.0.6               00-16-cd-ad-06-df     dynamic
10.0.0.10              00-21-5a-0c-0a-34     dynamic
10.0.0.104             00-21-5a-d6-b7-36     dynamic

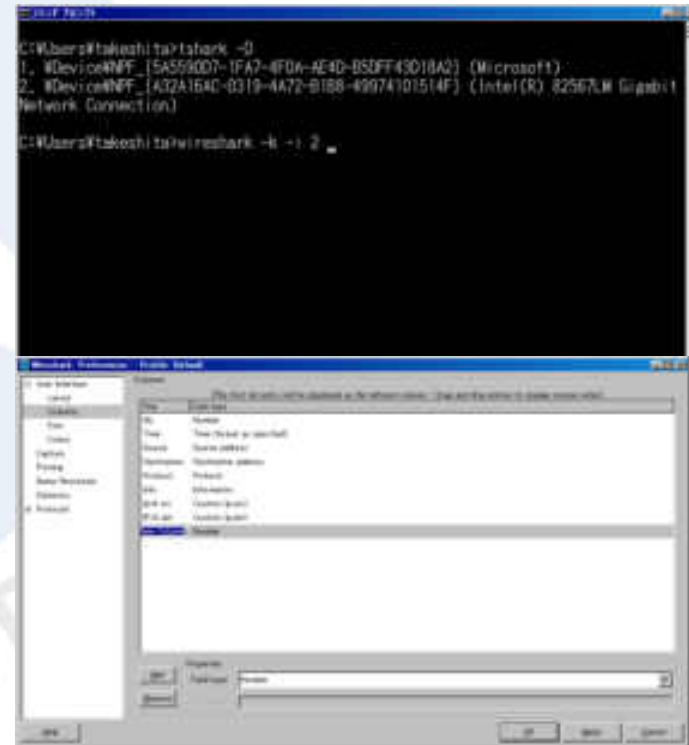
C:\Documents and Settings\takeshita>
```

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



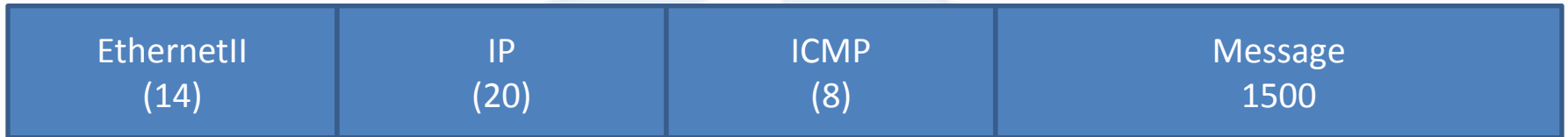
# Setting Wireshark

- 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

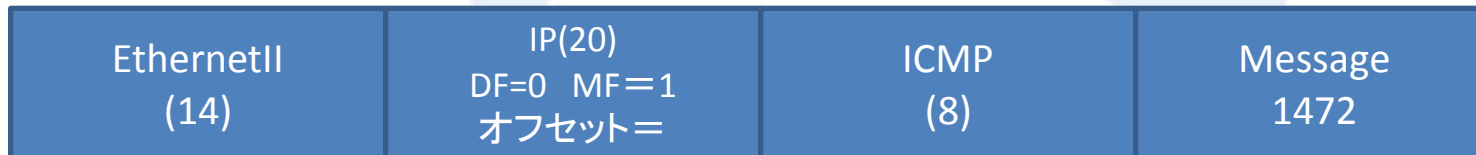


# Fragment

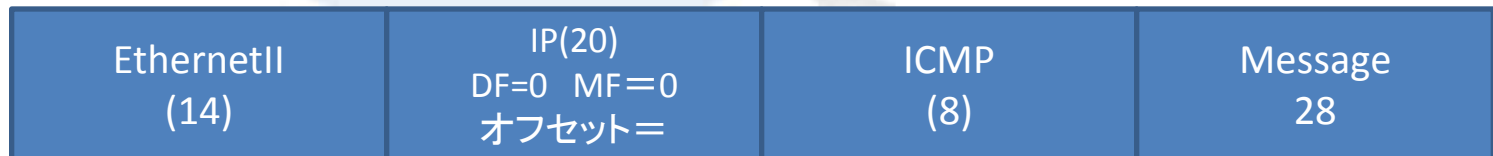
- Original frame



- Fragment 1/2 in Ethernet MTU



- Fragment 2/2 in Ethernet MTU



# Testing packet size

**ICMP** • ping host -l message size (MTU-28) -f

EthernetII (14)	IP (20)	ICMP (8)	message MTU=1500->1472
--------------------	------------	-------------	---------------------------

## TCP

EthernetII (14)	IP (20)	TCP (20)	Segment size MSS=1460
--------------------	------------	-------------	--------------------------

## UDP

EthernetII (14)	IP (20)	UDP (8)	Datagram size MTU=1500->1472
--------------------	------------	------------	---------------------------------

- 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

(初期ウィンドウサイズはOS等で指定)

- Window Size, SACK, MSS, Window Scaling
- Some router may rewrite this section via NAT
- Follow TCP stream and Use coloring and Ctrl+Space

The image displays a Wireshark network traffic analysis. The main pane shows a list of captured packets. The first two packets are highlighted in green, representing the initial SYN and SYN-ACK of a TCP handshake. The third packet is highlighted in blue, representing the ACK. The fourth packet is highlighted in red, representing an HTTP POST request. The details pane for the selected packet (No. 4) shows the following information:

```
POST /rms/mall/basket/vc HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Referer: http://item.rakuten.co.jp/nigari612/andino700creamer/
Accept-Language: ja
Content-Type: application/x-www-form-urlencoded
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
Host: order.step.rakuten.co.jp
Content-Length: 78
Connection: Keep-Alive
Cache-Control: no-cache
```

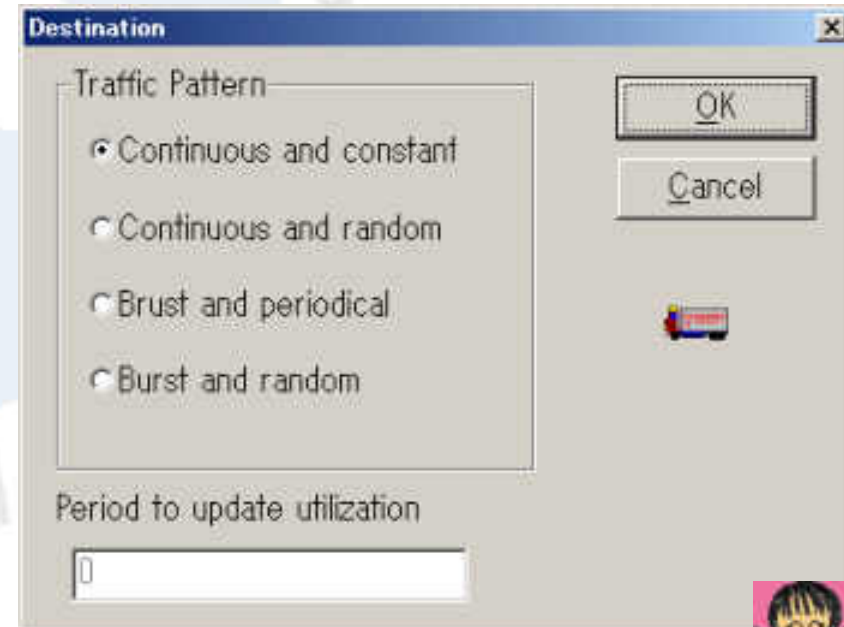
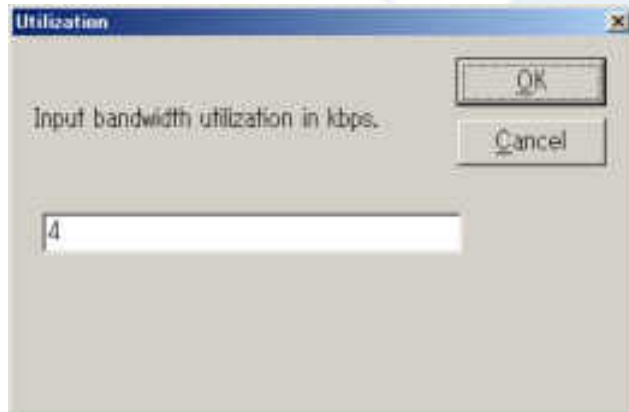
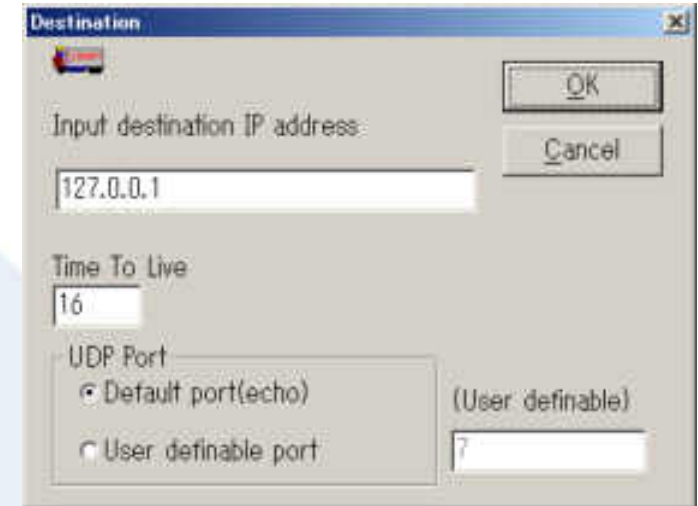
The packet bytes pane at the bottom shows the raw data of the selected packet, including Ethernet II, Internet Protocol, and Transmission Control Protocol headers.





# Using iperf and tfgen (made in Japan)

- Throughput ->iperf
- Traffic ->tfgen





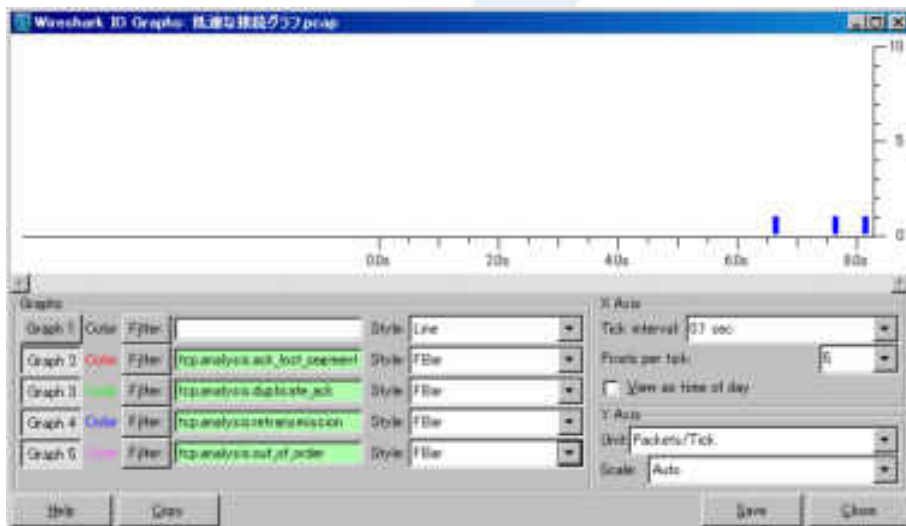
# Overview of troubleshoot

- 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 location  
to see packet in macro range ( statistics )
- Expert Info say many things automatically
- Think of packet lost -> Ignore (Ctrl+I)



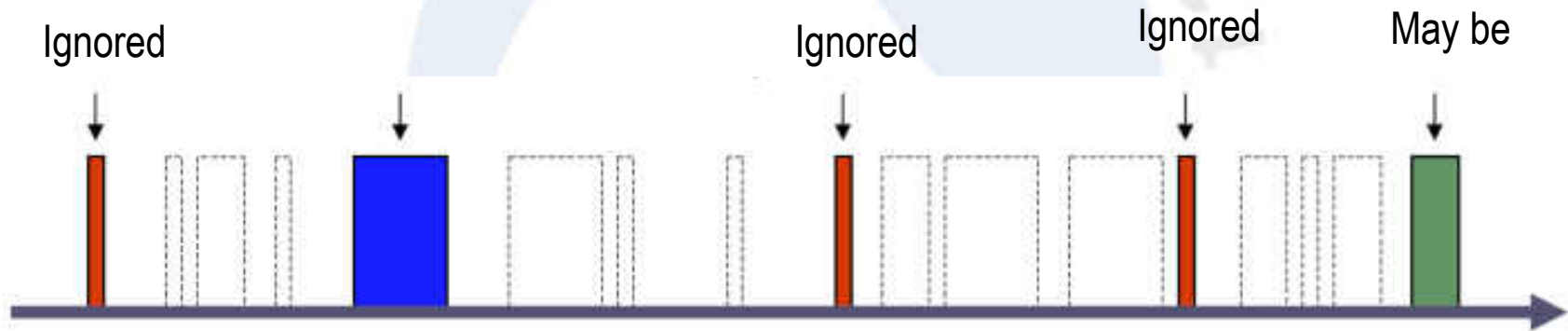
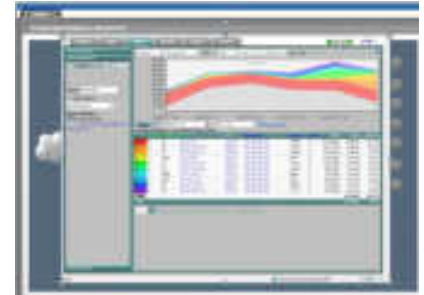
# Expertize IO graph

- 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 )



# No sampling, non-sampling

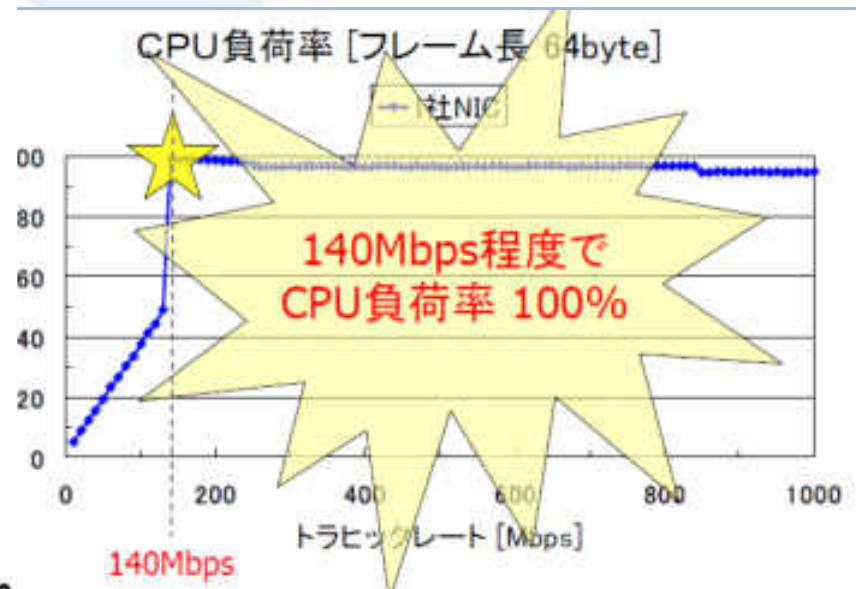
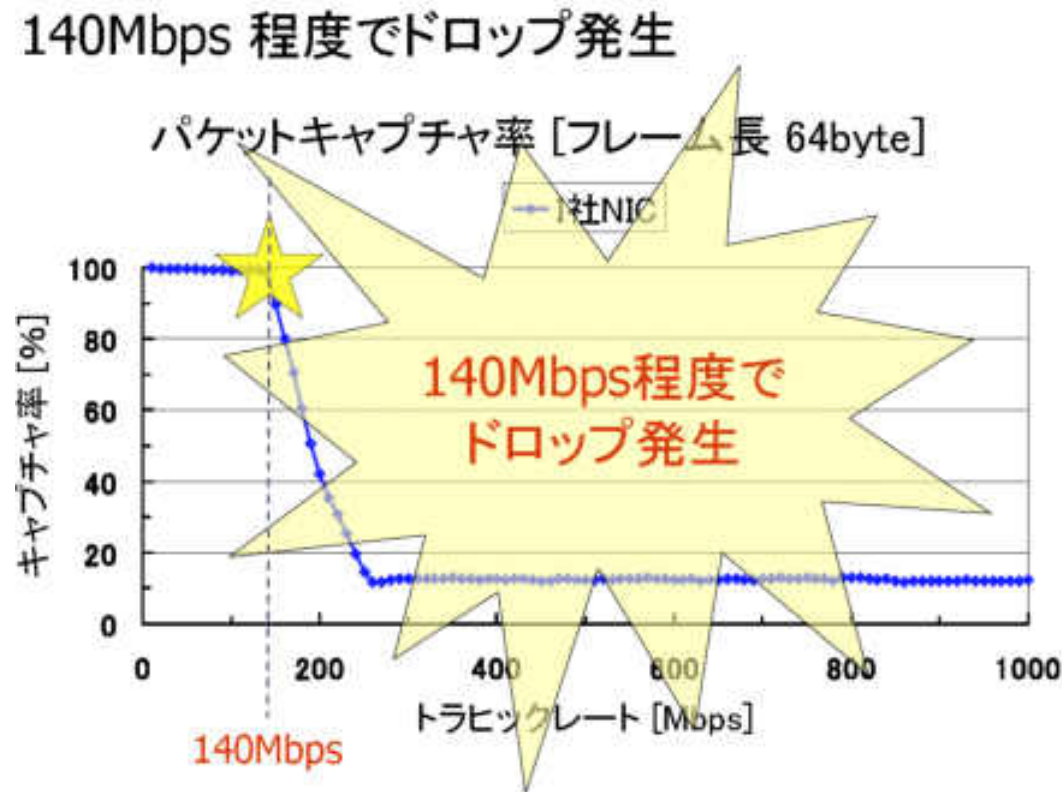
- In old days we use sampling technologies like SNMP, MRTG, and many flow analysis such as Cisco NetFlow, sFlow, iFlow



- 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 )

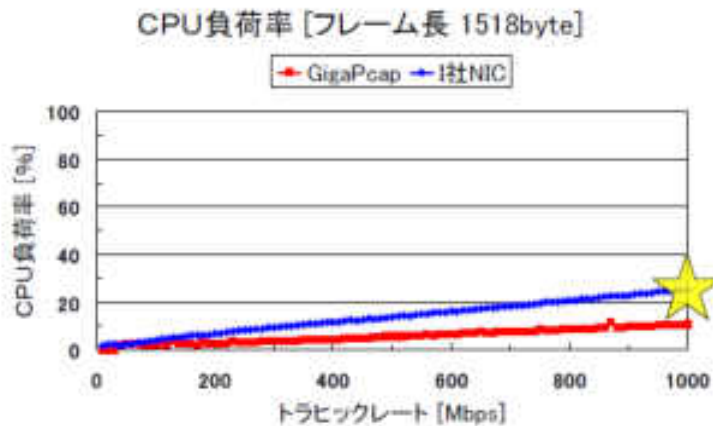
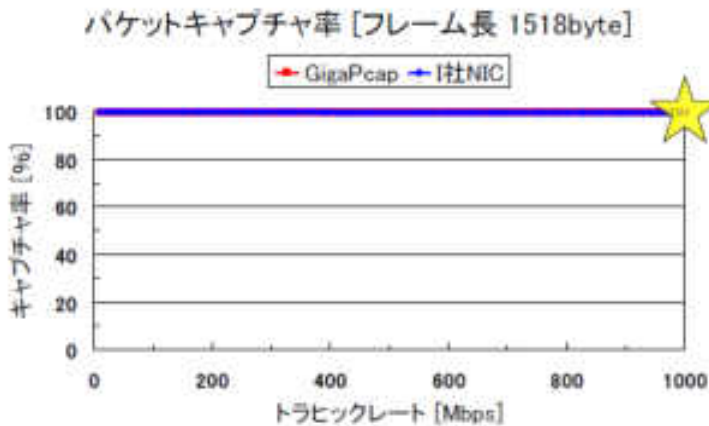
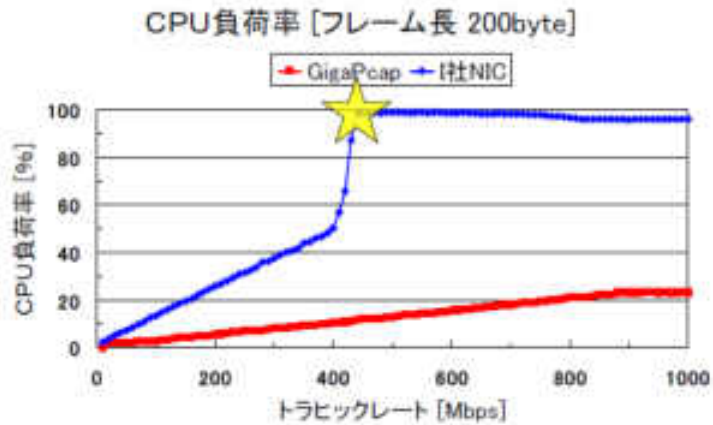
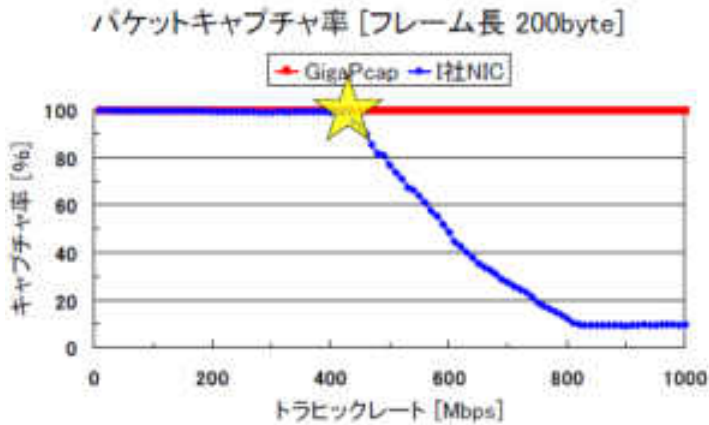
# Actual capture rate

- Typical Intel's GigaNIC (e1000), typical Dell PowerEdge2850 / Xeon 2.8GHz RAM 1GB (PC3200, DDR2, 400MHz)
- Threadshoud is 140Mbps in Frame size = 64



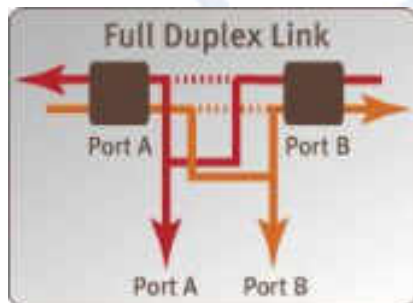
# Another frame size

- 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

