SHARKFEST '13

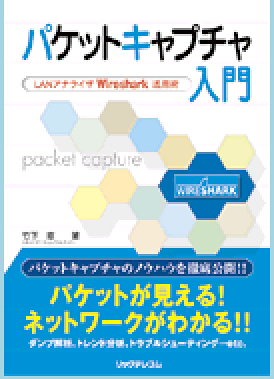
Wireshark Developer and User Conference

PA-3 Debugging Wireless with Wireshark Including Large Trace Files, AirPcap & Cascade Pilot

Megumi Takeshita, ikeriri network service

Ikeriri network service co., ltd
http://www.ikeriri.ne.jp

Megumi Takeshita, ikeriri network service a.k.a. packet otaku



パケットキャプチャ実践技術
Wireshark による パケット解析
packet capture
WIRESHARK
Packet never lies.
すべての答えは
パケットの中に隠されている!
リックテレコム

- Founder, ikeriri network service co.,ltd since 2002 ← Enterprise solution, Nortel networks ←Bay Network
- Wrote 10+ books about packet capturing, analysis, inspection, and consulting (in Japanese)
- Reseller of Riverbed Technology (former CACE technologies) and Metageek in Japan
- Packet capturing Otaku (geek) from Ethereal, 1st Sharkfest !

Ikeriri network service co., ltd.
Packet capture company

守りたい人がいる
陸上自衛隊

Training at JGSDF

Packet never lies.
いけりり★ネットワークサービス
http://www.ikeriri.ne.jp

- Consulting
- Reselling
- Debugging
- Investigating
- Training

Packet Capture

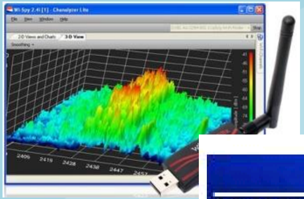
Reseller of Riverbed Technology, Metageek, OSCIUM, Dualcomm etc.

- Ikeriri is one of the reseller of Packet capture / analysis products in Japan
- Riverbed Technology’s AirPcap, TurboCap, Pilot
- Metageek Wi-Spy and Chanalyzer
- OSCIUM products
- Dualcomm products

etc.

OSCIUM

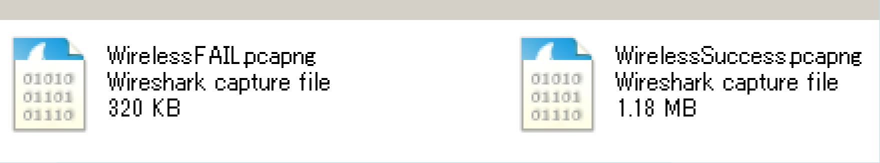
DUALCOMM TECHNOLOGY, INC



Planning for Debugging

boundary value analysis and equivalence partitioning

- Packet capture debugging is like a Black box-test
- Use Pcap/pcapng for boundary value analysis
  two ore more pattern / type of the issue
  OK pcap and NG pcap,
  setON pcap, setOFF pcap



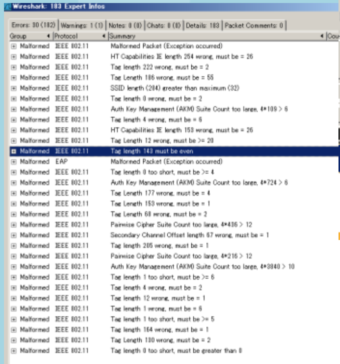
WirelessFAIL.pcapng
Wireshark capture file
320 KB

WirelessSuccess.pcapng
Wireshark capture file
1.18 MB

- Collect Pcap in less experiments using equivalence partitioning (grouping same environment pattern)
  We choose only 1 pcap of them and test

Comparison pcap files

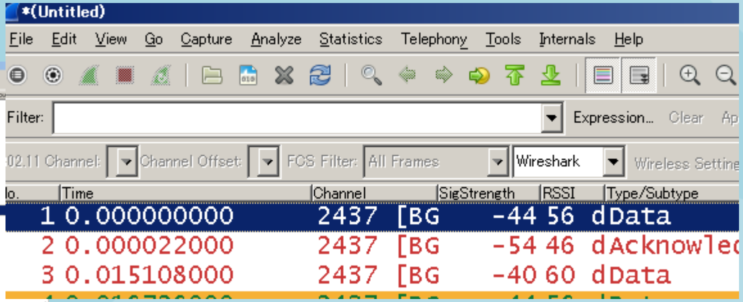
- We should capture comparison pcap files for debugging because there may be clues !
- Using boundary value analysis and equivalence partitioning, collect comparison pcaps.
- Some cases we can easily find the problem, keys, and the answer only watching 2 pattern of pcaps.
- Frame color,
  Expert info is easiest



Wireshark: 183 Expert Infos

Errors: 30 (182) | Warnings: 1 (1) | Notes: 0 (0) | Chats: 0 (0) | Details: 183 | Packet Comments: 0

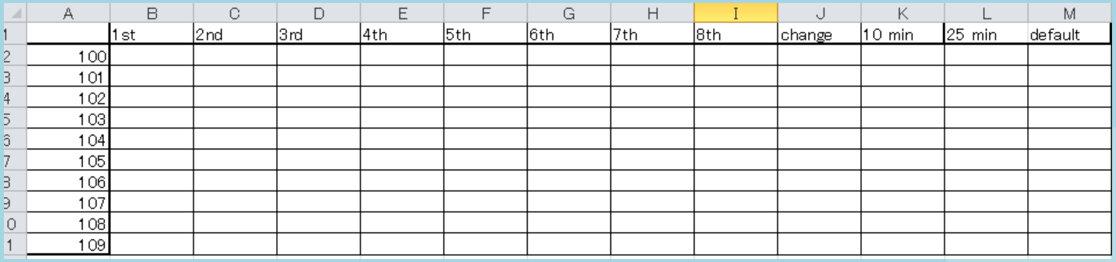
Group	Protocol	Summary	Cou
Malformed	IEEE 802.11	Malformed Packet (Exception occurred)	
Malformed	IEEE 802.11	HT Capabilities IE length 254 wrong, must be = 26	
Malformed	IEEE 802.11	Tag length 222 wrong, must be = 2	
Malformed	IEEE 802.11	Tag Length 186 wrong, must be = 55	
Malformed	IEEE 802.11	SSID length (204) greater than maximum (32)	
Malformed	IEEE 802.11	Tag length 0 wrong, must be = 2	
Malformed	IEEE 802.11	Auth Key Management (AKM) Suite Count too large, 4*109 > 6	
Malformed	IEEE 802.11	Tag length 4 wrong, must be = 6	
Malformed	IEEE 802.11	HT Capabilities IE length 153 wrong, must be = 26	
Malformed	IEEE 802.11	Tag Length 12 wrong, must be >= 20	
Malformed	IEEE 802.11	Tag length 143 must be even	
Malformed	EAP	Malformed Packet (Exception occurred)	
Malformed	IEEE 802.11	Tag length 0 too short, must be >= 4	
Malformed	IEEE 802.11	Auth Key Management (AKM) Suite Count too large, 4*724 > 6	
Malformed	IEEE 802.11	Tag Length 177 wrong, must be = 4	
Malformed	IEEE 802.11	Tag Length 153 wrong, must be = 1	
Malformed	IEEE 802.11	Tag Length 68 wrong, must be = 2	
Malformed	IEEE 802.11	Pairwise Cipher Suite Count too large, 4*436 > 12	
Malformed	IEEE 802.11	Secondary Channel Offset length 67 wrong, must be = 1	
Malformed	IEEE 802.11	Tag length 205 wrong, must be = 1	1
Malformed	IEEE 802.11	Pairwise Cipher Suite Count too large, 4*216 > 12	1
Malformed	IEEE 802.11	Auth Key Management (AKM) Suite Count too large, 4*3840 > 10	1
Malformed	IEEE 802.11	Tag length 1 too short, must be >= 6	1
Malformed	IEEE 802.11	Tag length 4 wrong, must be = 2	1
Malformed	IEEE 802.11	Tag length 12 wrong, must be = 1	1
Malformed	IEEE 802.11	Tag length 1 wrong, must be = 6	1
Malformed	IEEE 802.11	Tag length 1 too short, must be >= 5	1
Malformed	IEEE 802.11	Tag length 164 wrong, must be = 1	1
Malformed	IEEE 802.11	Tag Length 130 wrong, must be = 2	1
Malformed	IEEE 802.11	Tag length 0 too short, must be greater than 0	1



*(Untitled)

```
lo.  Time          Channel   SigStrength  RSSI  Type/Subtype
  1 0.000000000    2437 [BG     -44 56  dData
  2 0.000022000    2437 [BG     -54 46  dAcknowled
  3 0.015108000    2437 [BG     -40 60  dData
```

Gathering information and making table



	A	B	C	D	E	F	G	H	I	J	K	L	M
1		1st	2nd	3rd	4th	5th	6th	7th	8th	change	10 min	25 min	default
2	100												
3	101												
4	102												
5	103												
6	104												
7	105												
8	106												
9	107												
10	108												
11	109												

- Hearing the customer in deep, address (MAC,IP) port (TCP,UDP) log message, how to ? How many ?
- Host type, OS, Software version
  *Android is difficult (many variation)
  iOS (iPhone and iPad) is simple
  Windows 7/8 may be in same result
- Frequency is also important

Create plan of Experiment

Test capture procedure

the iteration number

test kind, types

test configuration

Standards and protocol and sequence

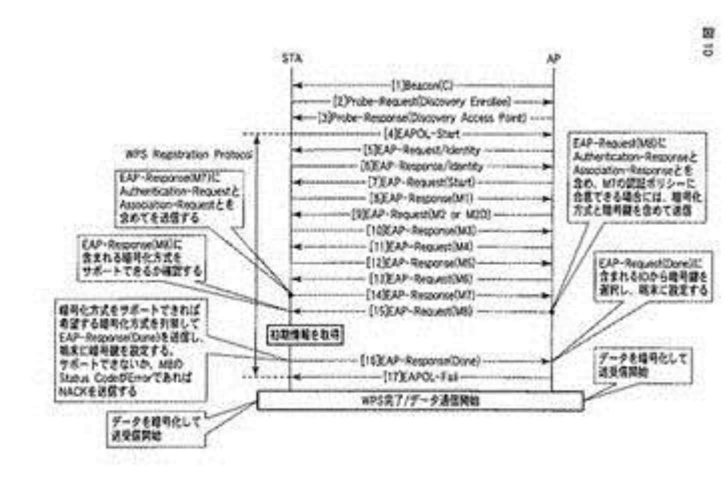


図10
STA
AP
[1]Beacon(C)
[2]Probe-Request(Discovery Enrollee)
[3]Probe-Response(Discovery Access Point)
[4]EAPOL-Start
[5]EAP-Request/Identity
[6]EAP-Response/Identity
[7]EAP-Request(Start)
[8]EAP-Response(M1)
[9]EAP-Request(M2 or M2D)
[10]EAP-Response(M3)
[11]EAP-Request(M4)
[12]EAP-Response(M5)
[13]EAP-Request(M6)
[14]EAP-Response(M7)
[15]EAP-Request(M8)
[16]EAP-Response(Done)
[17]EAPOL-Fail
WPS Registration Protocol
EAP-Response(M7)にAuthentication-RequestとAssociation-Requestとを含めてを送信する
EAP-Response(M6)に含まれる暗号化方式をサポートできるか確認する
暗号化方式をサポートできれば希望する暗号化方式を列挙してEAP-Response(Done)を送信し、端末に暗号鍵を設定する。サポートできないか、M8のStatus CodeがErrorであればNACKを送信する
初期情報を取得
データを暗号化して送受信開始
EAP-Request(M8)にAuthentication-ResponseとAssociation-Responseとを含め、M7の認証ポリシーに合意できる場合には、暗号化方式と暗号鍵を含めて送信
EAP-Request(Done)に含まれるIDから暗号鍵を選択し、端末に設定する
データを暗号化して送受信開始
WPS完了/データ通信開始

- Standards, protocol helps us debugging, using documents, White Papers in IEEE, RFCs in IANA and other sites
- Sequence diagram is very much hint for debug for checking and comparing

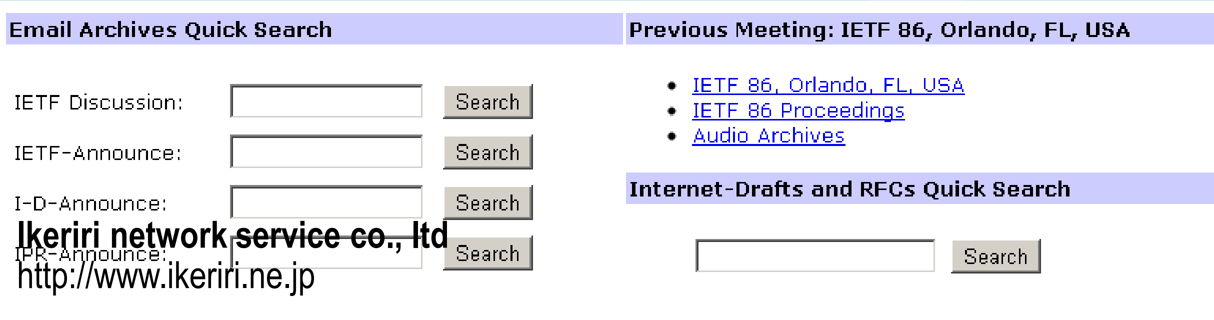
IEEE

Advancing Technology for Humanity

The world's largest professional association for the adv

About IEEE | Membership & Services | Societies & Communities | Publications & Standards

Search IEEE Google™ Custom Search Search



Email Archives Quick Search

IETF Discussion: [ ] Search

IETF-Announce: [ ] Search

I-D-Announce: [ ] Search

IPR-Announce: [ ] Search

Previous Meeting: IETF 86, Orlando, FL, USA

- IETF 86, Orlando, FL, USA
- IETF 86 Proceedings
- Audio Archives

Internet-Drafts and RFCs Quick Search

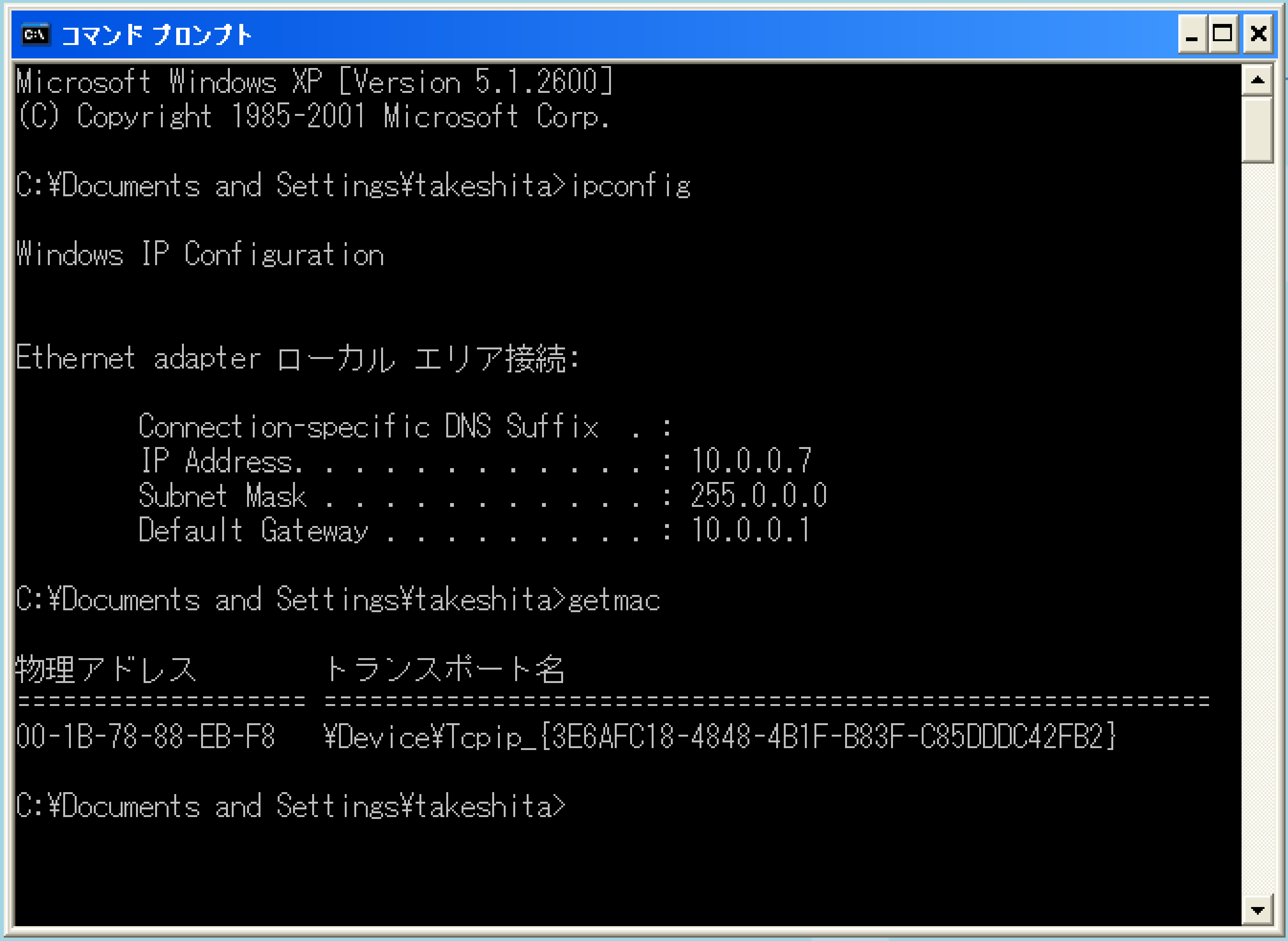
[ ] Search

Before Debug capturing

Before capturing

- Clear browser cache for capturing all communication packet.
- DNS cache is also clear if you need to get DNS query-response packet
- Disable or turn off Windows firewall and personal firewall etc.
- Stop and exit software and service of sending packet like VPN(keep alive), UPnP(SSDP discovery), iTunes
- Record Date, IP address, tcp port and MAC address for inspecting later.

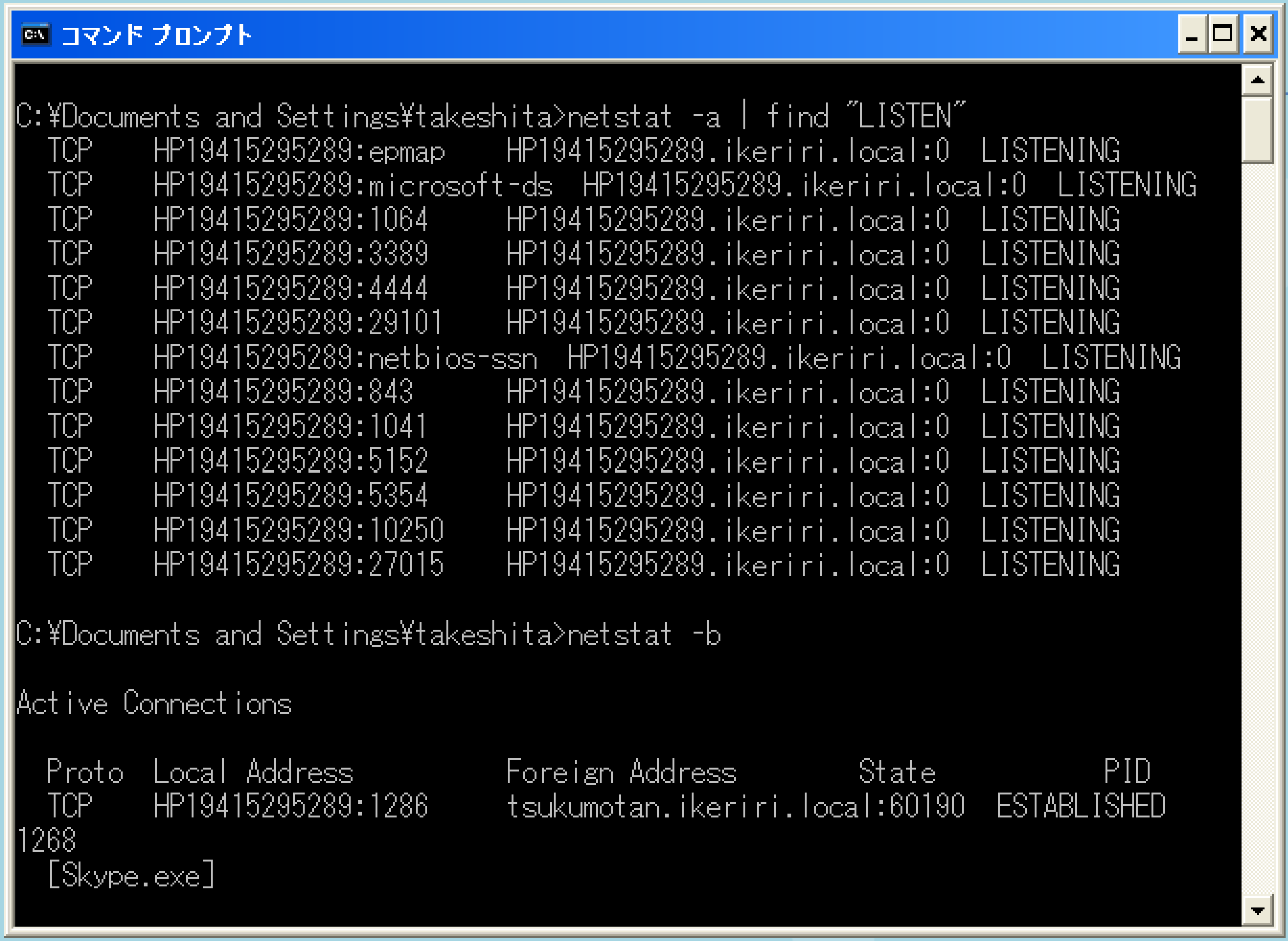
Tips1 redirecting information



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:¥Documents and Settings¥takeshita>ipconfig

Windows IP Configuration


Ethernet adapter ローカル エリア接続:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 10.0.0.7
        Subnet Mask . . . . . . . . . . . : 255.0.0.0
        Default Gateway . . . . . . . . . : 10.0.0.1

C:¥Documents and Settings¥takeshita>getmac

物理アドレス        トランスポート名
=================== ==========================================================
00-1B-78-88-EB-F8   ¥Device¥Tcpip_{3E6AFC18-4848-4B1F-B83F-C85DDDC42FB2}

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

- Executing ipconfig and getmac command and redirecting help us inspecting later

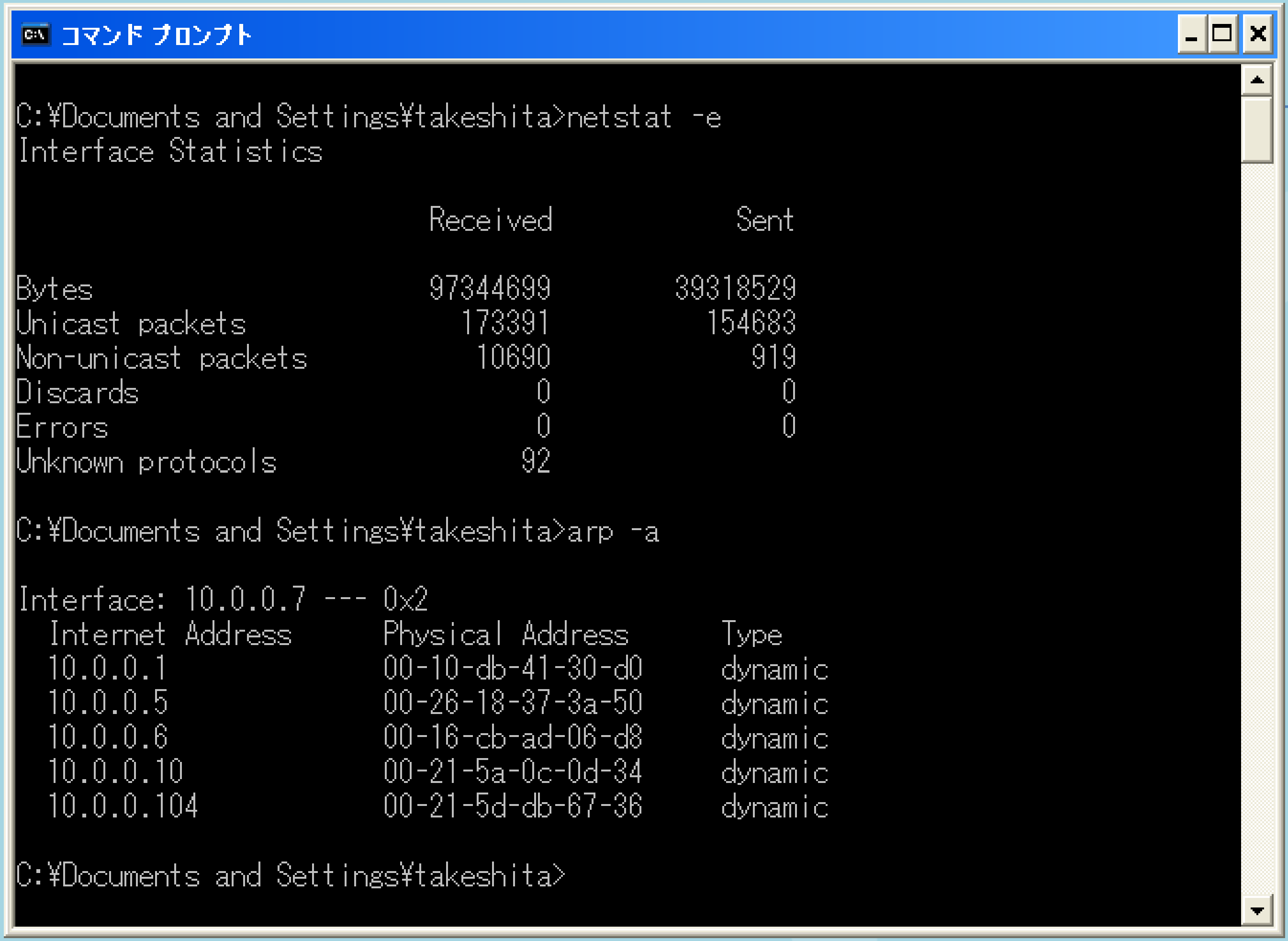
TIPS2: netstat –a and netstat -b



```
C:¥Documents and Settings¥takeshita>netstat -a | find "LISTEN"
  TCP    HP19415295289:epmap    HP19415295289.ikeriri.local:0  LISTENING
  TCP    HP19415295289:microsoft-ds  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
  TCP    HP19415295289:29101    HP19415295289.ikeriri.local:0  LISTENING
  TCP    HP19415295289:netbios-ssn  HP19415295289.ikeriri.local:0  LISTENING
  TCP    HP19415295289:843      HP19415295289.ikeriri.local:0  LISTENING
  TCP    HP19415295289:1041     HP19415295289.ikeriri.local:0  LISTENING
  TCP    HP19415295289:5152     HP19415295289.ikeriri.local:0  LISTENING
  TCP    HP19415295289:5354     HP19415295289.ikeriri.local:0  LISTENING
  TCP    HP19415295289:10250    HP19415295289.ikeriri.local:0  LISTENING
  TCP    HP19415295289:27015    HP19415295289.ikeriri.local:0  LISTENING

C:¥Documents and Settings¥takeshita>netstat -b

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    HP19415295289:1286     tsukumotan.ikeriri.local:60190  ESTABLISHED
1268
  [Skype.exe]
```

- Show tcp/udp connections using netstat,
  and I recommend piping and find matching (LISTEN)
  netstat –b tells bind application to socket.

TIPS3

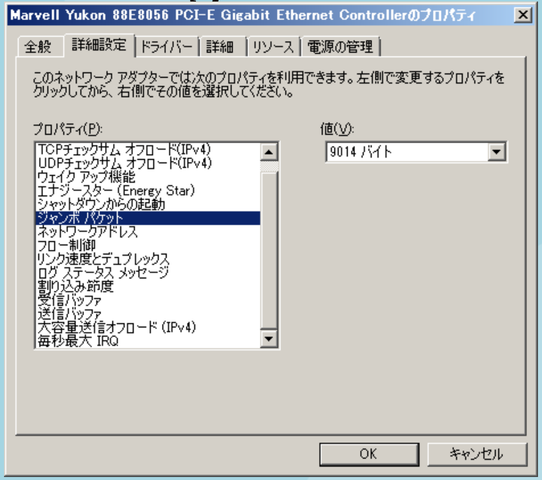


```
C:¥Documents and Settings¥takeshita>netstat -e
Interface Statistics

                           Received            Sent

Bytes                      97344699        39318529
Unicast packets              173391          154683
Non-unicast packets           10690             919
Discards                          0               0
Errors                            0               0
Unknown protocols                92

C:¥Documents and Settings¥takeshita>arp -a

Interface: 10.0.0.7 --- 0x2
  Internet Address      Physical Address      Type
  10.0.0.1              00-10-db-41-30-d0     dynamic
  10.0.0.5              00-26-18-37-3a-50     dynamic
  10.0.0.6              00-16-cb-ad-06-d8     dynamic
  10.0.0.10             00-21-5a-0c-0d-34     dynamic
  10.0.0.104            00-21-5d-db-67-36     dynamic

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

- Please check your NIC status (including Error and Discard frames) using netstat –e command.

Check settings in NIC

- Today almost NICs offload tcp, udp/ip function.
- Almost NICs support Gigabit Ethernet and carrier extension (over 1500MTU ex. 9kb MTU)
- Wireshark read pcap stream from WinPcap
- Please check offload settings in properties in NIC (from device manager)
- Also please check MTU setting too. (Jumbo frame or MTU)

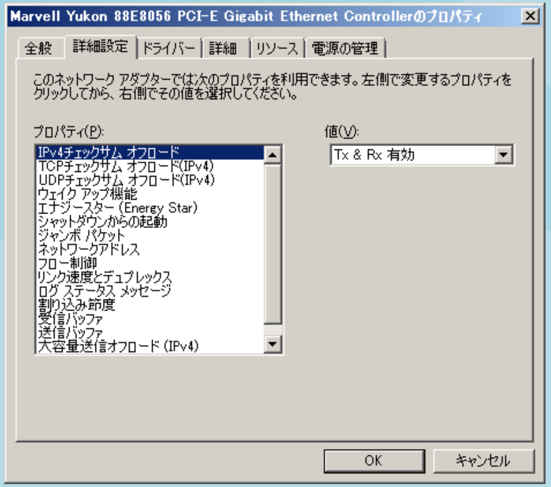


Marvell Yukon 88E8056 PCI-E Gigabit Ethernet Controllerのプロパティ

このネットワーク アダプターでは次のプロパティを利用できます。左側で変更するプロパティをクリックしてから、右側でその値を選択してください。

プロパティ(P): ジャンボ パケット

値(V): 9014 バイト



Marvell Yukon 88E8056 PCI-E Gigabit Ethernet Controllerのプロパティ

このネットワーク アダプターでは次のプロパティを利用できます。左側で変更するプロパティをクリックしてから、右側でその値を選択してください。

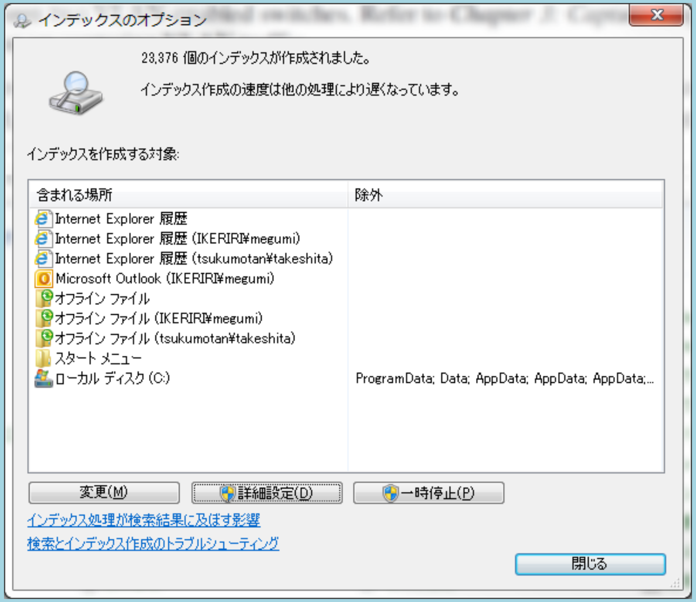
プロパティ(P): IPv4チェックサム オフロード

値(V): Tx & Rx 有効

全般
詳細設定
ドライバー
詳細
リソース
電源の管理
IPv4チェックサム オフロード
TCPチェックサム オフロード(IPv4)
UDPチェックサム オフロード(IPv4)
ウェイク アップ機能
エナジースター (Energy Star)
シャットダウンからの起動
ジャンボ パケット
ネットワークアドレス
フロー制御
リンク速度とデュプレックス
ログ ステータス メッセージ
割り込み節度
受信バッファ
送信バッファ
大容量送信オフロード (IPv4)
OK
キャンセル

Use Windows Search Index

- To add extension of cap and pcap,
  set type as clear text search,
  We can search pcap/cap files like Google !
  off course in multibytes (in Japanese)
- Control panel -> index option / folder option



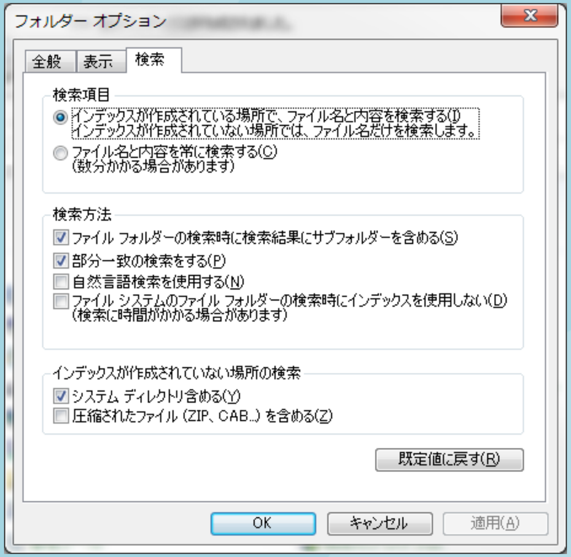
23,376 個のインデックスが作成されました。

インデックス作成の速度は他の処理により遅くなっています。

インデックスを作成する対象:

含まれる場所	除外
Internet Explorer 履歴	
Internet Explorer 履歴 (IKERIRI¥megumi)	
Internet Explorer 履歴 (tsukumotan¥takeshita)	
Microsoft Outlook (IKERIRI¥megumi)	
オフライン ファイル	
オフライン ファイル (IKERIRI¥megumi)	
オフライン ファイル (tsukumotan¥takeshita)	
スタート メニュー	
ローカル ディスク (C:)	ProgramData; Data; AppData; AppData; AppData;...

変更(M)
詳細設定(D)
一時停止(P)
インデックス処理が検索結果に及ぼす影響
検索とインデックス作成のトラブルシューティング
閉じる



検索項目

● インデックスが作成されている場所で、ファイル名と内容を検索する(I)
インデックスが作成されていない場所では、ファイル名だけを検索します。

○ ファイル名と内容を常に検索する(C)
(数分かかる場合があります)

検索方法

☑ ファイル フォルダーの検索時に検索結果にサブフォルダーを含める(S)
☑ 部分一致の検索をする(P)
□ 自然言語検索を使用する(N)
□ ファイル システムのファイル フォルダーの検索時にインデックスを使用しない(D)
(検索に時間がかかる場合があります)

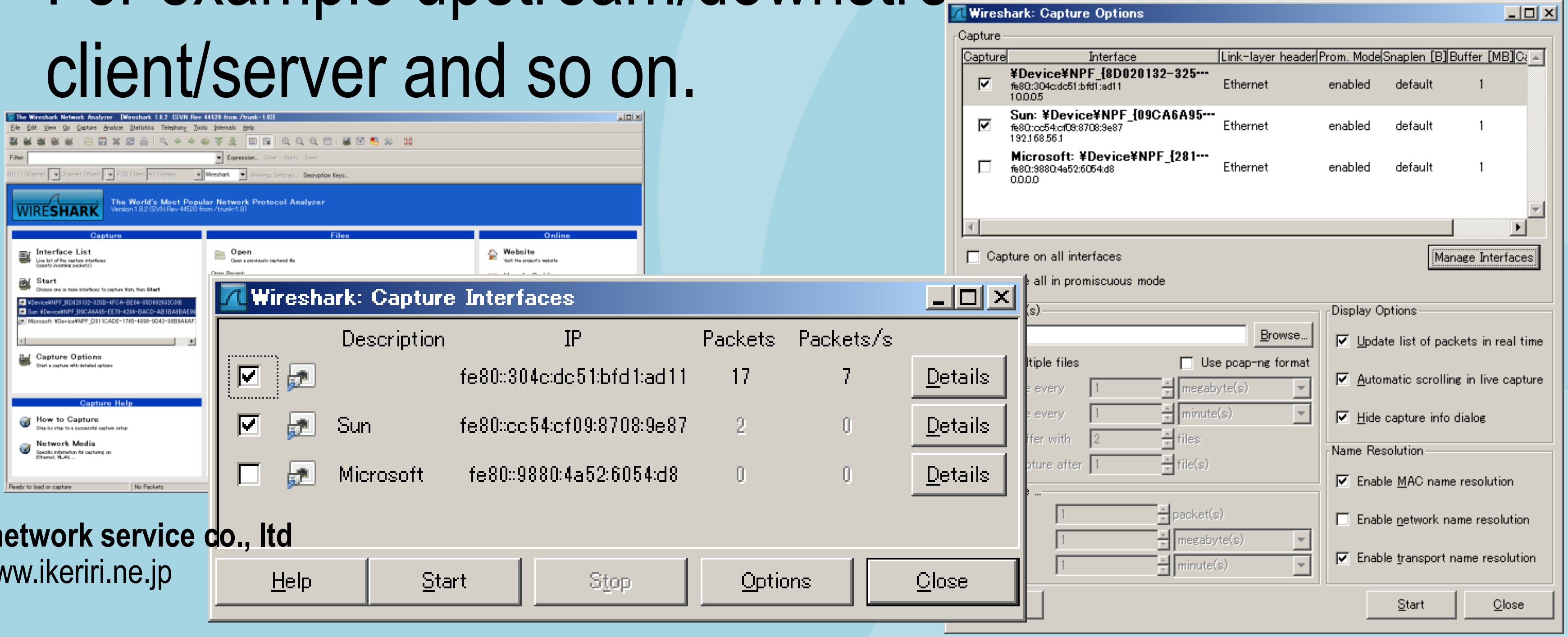
インデックスが作成されていない場所の検索

☑ システム ディレクトリ含める(Y)
□ 圧縮されたファイル (ZIP、CAB...) を含める(Z)

Wireshark setting

Capturing many interface in one time
Check multiple interface and capture

- In case of checking many interface in the same time, now check multiple interface and start capture.
- Trace file is combined with multiple interface
- For example upstream/downstream from router, client/server and so on.



The Wireshark Network Analyzer [Wireshark 1.8.2 (SVN Rev 44520 from /trunk-1.8)]
WIRESHARK
The World's Most Popular Network Protocol Analyzer
Version 1.8.2 (SVN Rev 44520 from /trunk-1.8)
Capture
Interface List
Start
Capture Options
Capture Help
How to Capture
Network Media
Files
Open
Website
Online

Wireshark: Capture Interfaces

	Description	IP	Packets	Packets/s	
☑		fe80::304c:dc51:bfd1:ad11	17	7	Details
☑	Sun	fe80::cc54:cf09:8708:9e87	2	0	Details
☐	Microsoft	fe80::9880:4a52:6054:d8	0	0	Details

Help
Start
Stop
Options
Close

Wireshark: Capture Options

Capture

Capture	Interface	Link-layer header	Prom. Mode	Snaplen [B]	Buffer [MB]	Ca
☑	¥Device¥NPF_{8D020132-325--- fe80::304c:dc51:bfd1:ad11 10.0.0.5	Ethernet	enabled	default	1	
☑	Sun: ¥Device¥NPF_{09CA6A95--- fe80::cc54:cf09:8708:9e87 192.168.56.1	Ethernet	enabled	default	1	
☐	Microsoft: ¥Device¥NPF_{281--- fe80::9880:4a52:6054:d8 0.0.0.0	Ethernet	enabled	default	1	

☐ Capture on all interfaces

e all in promiscuous mode

☐ Use pcap-ng format

Display Options

☑ Update list of packets in real time

☑ Automatic scrolling in live capture

☑ Hide capture info dialog

Name Resolution

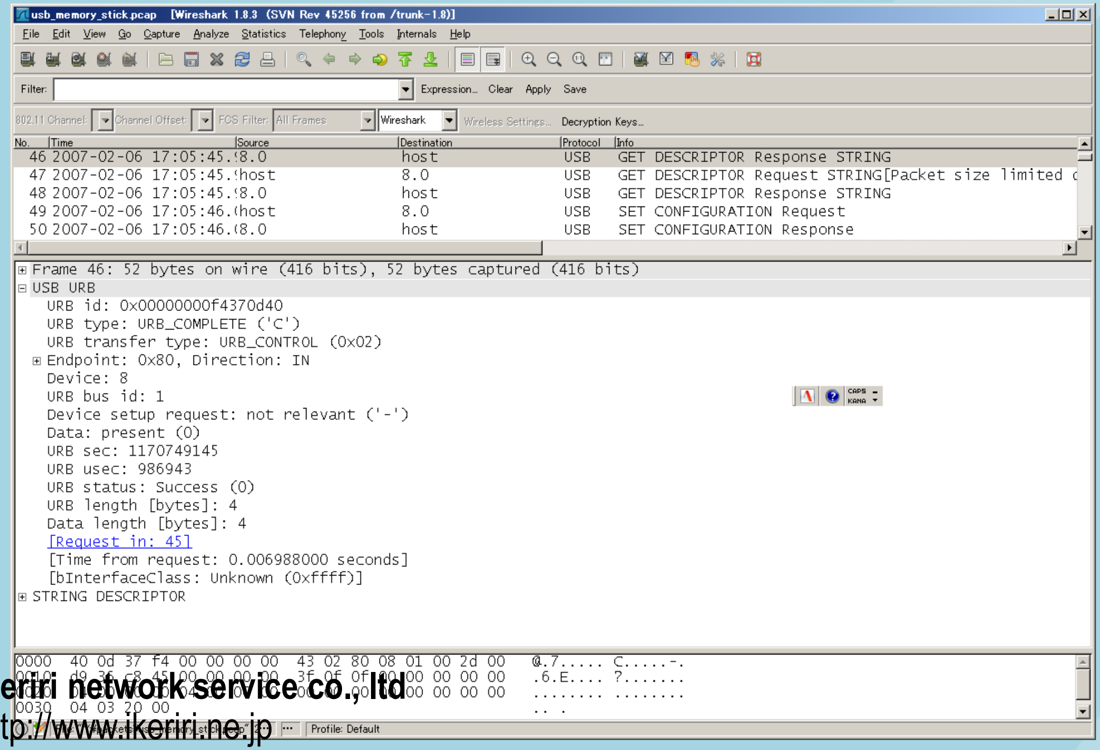
☑ Enable MAC name resolution

☐ Enable network name resolution

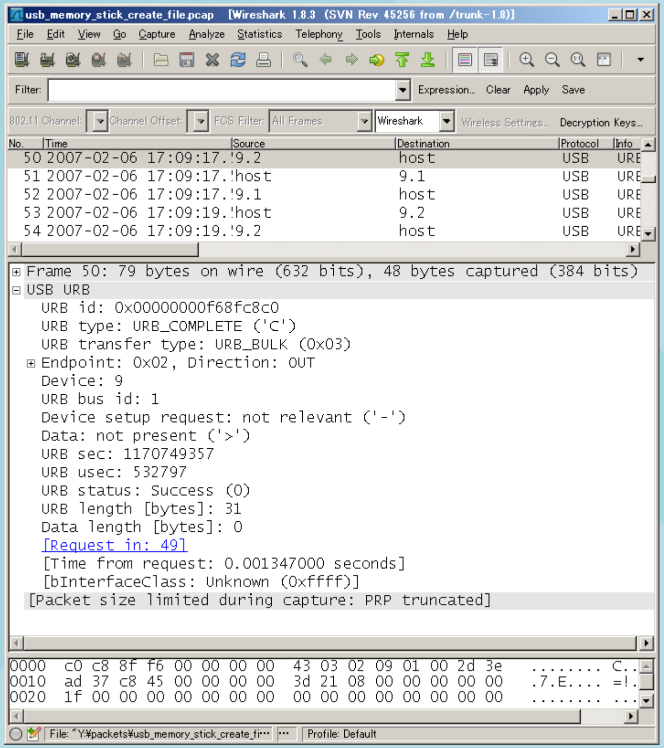
☑ Enable transport name resolution

USB Debugging

- We can capture USB frames using Linux
- VMware environment also works


usb_memory_stick.pcap [Wireshark 1.8.3 (SVN Rev 45256 from /trunk-1.8)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...
Profile: Default

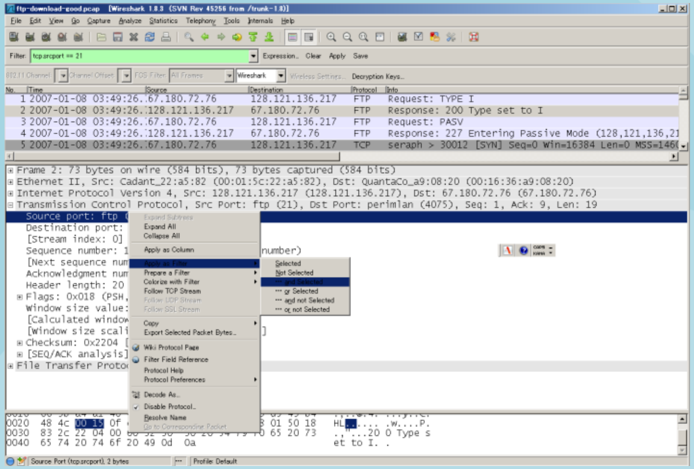
```
No. Time                  Source     Destination  Protocol Info
 46 2007-02-06 17:05:45.!8.0       host         USB      GET DESCRIPTOR Response STRING
 47 2007-02-06 17:05:45.!host      8.0          USB      GET DESCRIPTOR Request STRING[Packet size limited c
 48 2007-02-06 17:05:45.!8.0       host         USB      GET DESCRIPTOR Response STRING
 49 2007-02-06 17:05:46.(host      8.0          USB      SET CONFIGURATION Request
 50 2007-02-06 17:05:46.(8.0       host         USB      SET CONFIGURATION Response

⊞ Frame 46: 52 bytes on wire (416 bits), 52 bytes captured (416 bits)
⊟ USB URB
   URB id: 0x00000000f4370d40
   URB type: URB_COMPLETE ('C')
   URB transfer type: URB_CONTROL (0x02)
 ⊞ Endpoint: 0x80, Direction: IN
   Device: 8
   URB bus id: 1
   Device setup request: not relevant ('-')
   Data: present (0)
   URB sec: 1170749145
   URB usec: 986943
   URB status: Success (0)
   URB length [bytes]: 4
   Data length [bytes]: 4
   [Request in: 45]
   [Time from request: 0.006988000 seconds]
   [bInterfaceClass: Unknown (0xffff)]
⊞ STRING DESCRIPTOR

0000  40 0d 37 f4 00 00 00 00  43 02 80 08 01 00 2d 00   @.7..... C.....-.
0010  d9 36 c8 45 00 00 00 00  3f 0f 0f 00 00 00 00 00   .6.E.... ?.......
0020  [illegible] 00 00 00 00   ........ ........
0030  04 03 20 00                                        .. .
```


usb_memory_stick_create_file.pcap [Wireshark 1.8.3 (SVN Rev 45256 from /trunk-1.8)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...
File: "Y:¥packets¥usb_memory_stick_create_fi···
Profile: Default

```
No. Time                  Source     Destination  Protocol Info
 50 2007-02-06 17:09:17.!9.2       host         USB      URE
 51 2007-02-06 17:09:17.!host      9.1          USB      URE
 52 2007-02-06 17:09:17.!9.1       host         USB      URE
 53 2007-02-06 17:09:19.!host      9.2          USB      URE
 54 2007-02-06 17:09:19.!9.2       host         USB      URE

⊞ Frame 50: 79 bytes on wire (632 bits), 48 bytes captured (384 bits)
⊟ USB URB
   URB id: 0x00000000f68fc8c0
   URB type: URB_COMPLETE ('C')
   URB transfer type: URB_BULK (0x03)
 ⊞ Endpoint: 0x02, Direction: OUT
   Device: 9
   URB bus id: 1
   Device setup request: not relevant ('-')
   Data: not present ('>')
   URB sec: 1170749357
   URB usec: 532797
   URB status: Success (0)
   URB length [bytes]: 31
   Data length [bytes]: 0
   [Request in: 49]
   [Time from request: 0.001347000 seconds]
   [bInterfaceClass: Unknown (0xffff)]
  [Packet size limited during capture: PRP truncated]

0000  c0 c8 8f f6 00 00 00 00  43 03 02 09 01 00 2d 3e   ........ C..
0010  ad 37 c8 45 00 00 00 00  3d 21 08 00 00 00 00 00   .7.E.... =!.
0020  1f 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ...
```

Using display filter

- Protocol.field.value style
- Easiest way is taking use of actual header field (right click and show submenu and set/prepare filter)
- Condition of multiple format &&(AND) ||(OR) parameter value can be compared (gt ge / lt le)
- Automatic complication will help you to create
- Contains keyword http.request.url contains ikeriri



ftp-download-good.pcap [Wireshark 1.8.3 (SVN Rev 45256 from /trunk-1.8)]

Filter: tcp.srcport == 21

```
No. Time                    Source           Destination      Protocol Info
  1 2007-01-08 03:49:26.:67.180.72.76    128.121.136.217  FTP      Request: TYPE I
  2 2007-01-08 03:49:26.:128.121.136.217 67.180.72.76     FTP      Response: 200 Type set to I
  3 2007-01-08 03:49:26.:67.180.72.76    128.121.136.217  FTP      Request: PASV
  4 2007-01-08 03:49:26.:128.121.136.217 67.180.72.76     FTP      Response: 227 Entering Passive Mode (128,121,136,21
  5 2007-01-08 03:49:26.:67.180.72.76    128.121.136.217  TCP      seraph > 30012 [SYN] Seq=0 Win=16384 Len=0 MSS=146(
```

```
⊞ Frame 2: 73 bytes on wire (584 bits), 73 bytes captured (584 bits)
⊞ Ethernet II, Src: Cadant_22:a5:82 (00:01:5c:22:a5:82), Dst: QuantaCo_a9:08:20 (00:16:36:a9:08:20)
⊞ Internet Protocol Version 4, Src: 128.121.136.217 (128.121.136.217), Dst: 67.180.72.76 (67.180.72.76)
⊟ Transmission Control Protocol, Src Port: ftp (21), Dst Port: perimlan (4075), Seq: 1, Ack: 9, Len: 19
    Source port: ftp (
    Destination port:
    [Stream index: 0]
    Sequence number: 1                         number)
    [Next sequence num
    Acknowledgment num
    Header length: 20
  ⊞ Flags: 0x018 (PSH,
    Window size value:
    [Calculated window
    [Window size scali                         ]
  ⊞ Checksum: 0x2204 [
  ⊞ [SEQ/ACK analysis]
⊞ File Transfer Protoc
```

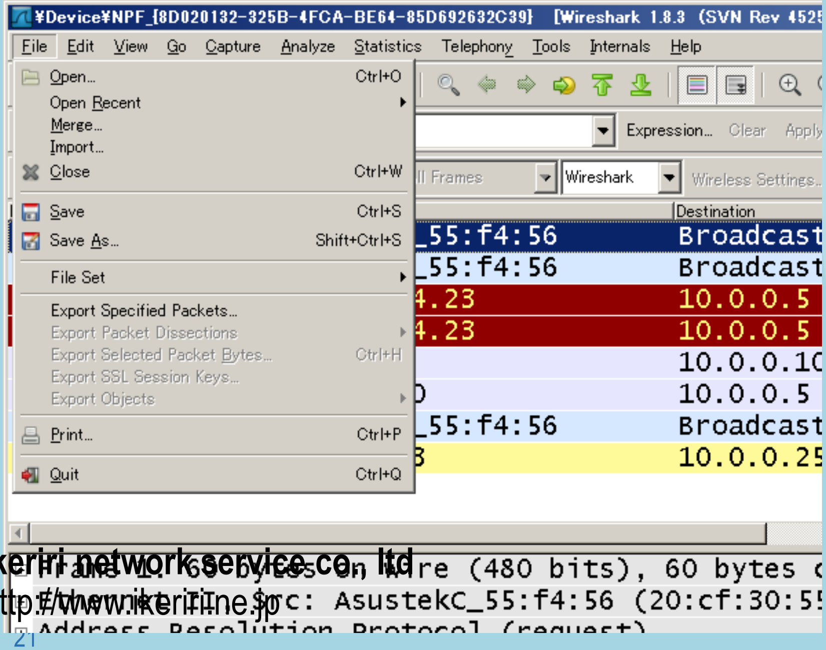
```
0020  48 4c 00 15 0f              3 01 50 18    HL.... .W....P.
0030  83 2c 22 04 00              0 65 20 73    .,"...20 0 Type s
0040  65 74 20 74 6f 20 49 0d  0a               et to I. .
```

Source Port (tcp.srcport), 2 bytes

Profile: Default

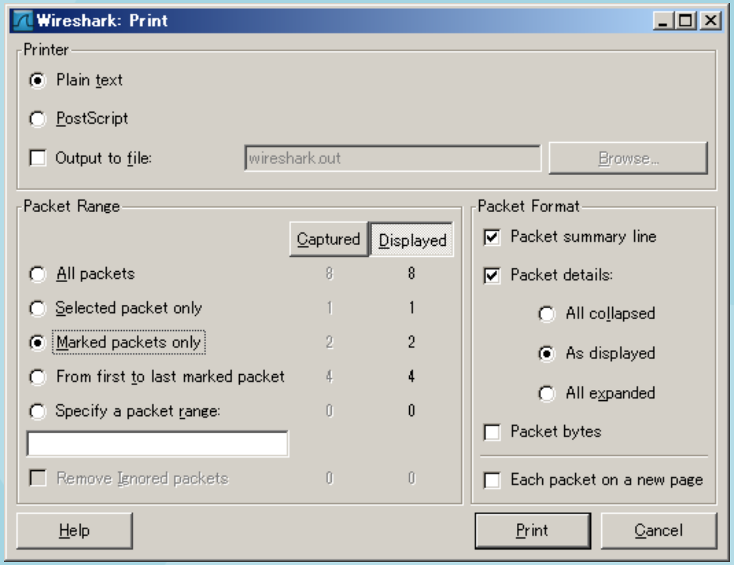
Mark and export specified packet
Print packet information to text file

- Marking packet is important for good report.
- Export specified packet and create good trace file.
- Text-based packet information is usable to send email and making report.


¥Device¥NPF_{8D020132-325B-4FCA-BE64-85D692632C39} [Wireshark 1.8.3 (SVN Rev 4525
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Open... Ctrl+O
Open Recent
Merge...
Import...
Close Ctrl+W
Save Ctrl+S
Save As... Shift+Ctrl+S
File Set
Export Specified Packets...
Export Packet Dissections
Export Selected Packet Bytes... Ctrl+H
Export SSL Session Keys...
Export Objects
Print... Ctrl+P
Quit Ctrl+Q
Expression... Clear Apply
ll Frames
Wireshark
Wireless Settings...
Destination

```
_55:f4:56      Broadcast
_55:f4:56      Broadcast
4.23           10.0.0.5
4.23           10.0.0.5
               10.0.0.10
0              10.0.0.5
_55:f4:56      Broadcast
3              10.0.0.25
```

```
Frame 1: 60 bytes on wire (480 bits), 60 bytes c
Ethernet II, Src: AsustekC_55:f4:56 (20:cf:30:55
Address Resolution Protocol (request)
```


Wireshark: Print

Printer

● Plain text
○ PostScript
□ Output to file: wireshark.out Browse...

Packet Range

	Captured	Displayed
○ All packets	8	8
○ Selected packet only	1	1
● Marked packets only	2	2
○ From first to last marked packet	4	4
○ Specify a packet range:	0	0
□ Remove Ignored packets	0	0

Packet Format

☑ Packet summary line
☑ Packet details:
○ All collapsed
● As displayed
○ All expanded
□ Packet bytes
□ Each packet on a new page

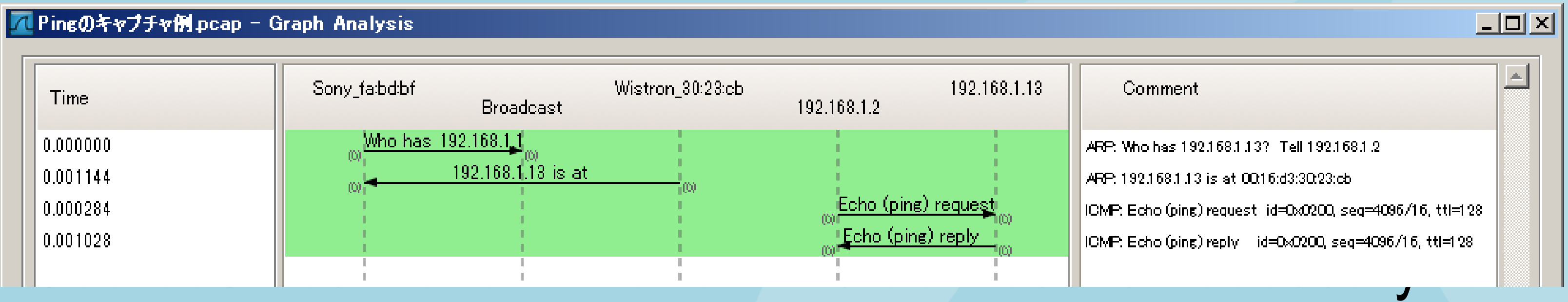
TIPS：Useful shortcut

Shortcut	contents
Ctrl+↑,↓	Set mouse in packet detail pane, easy to go next / back previous packet (useful !!)
←,→	Expand / collapse information
Ctrl+O,Ctrl+W,Ctrl+P, Ctrl+P,Ctrl+S,Ctrl+Q	Open, Window Close, Print, Save,Quit
Ctrl+H	Output Hex data (for exporting raw data)
Ctrl+F	Find packet
Ctrl+T	Set reference time (for calculating response time)
Ctrl+Shift+P, Ctrl+Shift+A	preference Profile
Ctrl+[Space]	Immediately clear temporary coloring rules.

Debugging packet size issue using ICMP

Capturing PING(ICMP) packet

- Start capturing, then test ping command



Pingのキャプチャ例.pcap - Graph Analysis

Time	Comment
0.000000	ARP: Who has 192.168.1.13? Tell 192.168.1.2
0.001144	ARP: 192.168.1.13 is at 00:16:d3:30:23:cb
0.000284	ICMP: Echo (ping) request id=0x0200, seq=4096/16, ttl=128
0.001028	ICMP: Echo (ping) reply id=0x0200, seq=4096/16, ttl=128

- communication under TCP/IP
- ARP request / response loop make address resolution.
- ARP result is remembered and cached for 120 seconds in each PCs
- ICMP echo request / response loop check layer 3 connectivity.

Ikeriri network service co., ltd
http://www.ikeriri.ne.jp

IP trace file analysis

- Check identification field of IP header
  same Identification number means re-send packet, fragmentation, and security problem.
- TTL field is the hint of hop counts (always the node uses 128/64)
- Check DF/MF bit and offset field in IP header.
- Compare IP length field and MTU size.

ping a.b.c.d –l 1500 -f

- original

Ethernet II (14)	IP (20)	ICMP (8)	Message 1500

- Fragment1

Ethernet II (14)	IP(20) DF=　　MF＝ offset＝	ICMP (8)	Message 1472

- Fragment2

Ethernet II (14)	IP(20) DF=　　MF＝ offset＝	Message 28

Count packet size (MTU1500)

ICMP -28

Ethernet II (14)	IP (20)	ICMP (8)	Message 1472(MTU=1500)

- ping IP –l size ※-f fragment disabled

TCP HTTP and many protocols -40

Ethernet II (14)	IP (20)	TCP (20)	Segment size MSS＝1460

UDP VOIP and video transmission -28

Ethernet II (14)	IP (20)	UDP (8)	Datagram size 1472(MTU=1500)

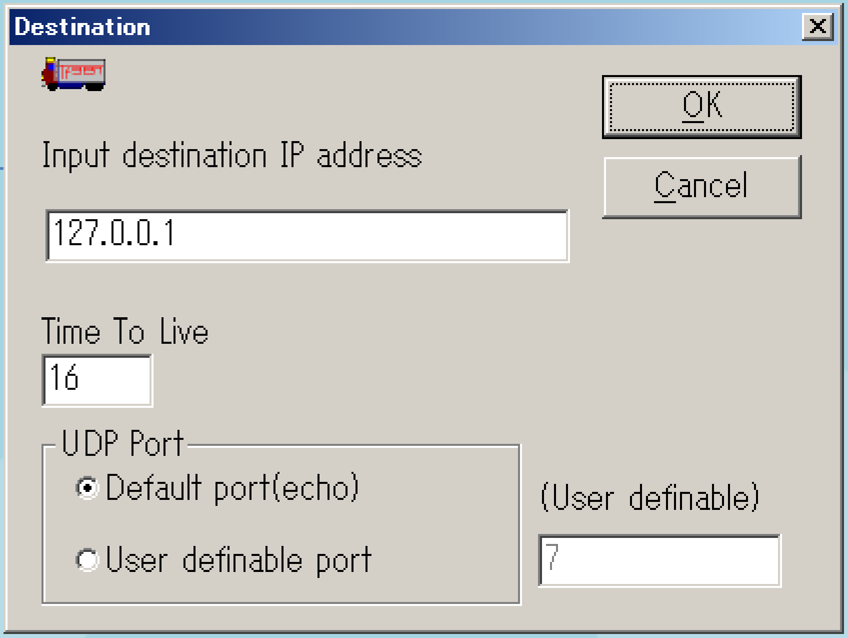
PPPoE Header and MTU size according to Japanese ISPs

- NTT east flets
  MTU 1454Bytes MSS 1414Bytes
- NTT west flets premium
  MTU 1438Bytes MSS 1398Bytes
- GRE + IPsec (transport mode) 1440 Bytes
  GRE + IPsec (tunneling mode) 1420 Byte
- UDP(NAT Traversal）
  IP(20) UDP(8) , PPPoE, PPP header

Debugging Upper Layer4

tfgen

- For checking TCP vs. UDP it is very useful



Destination

Input destination IP address: 127.0.0.1

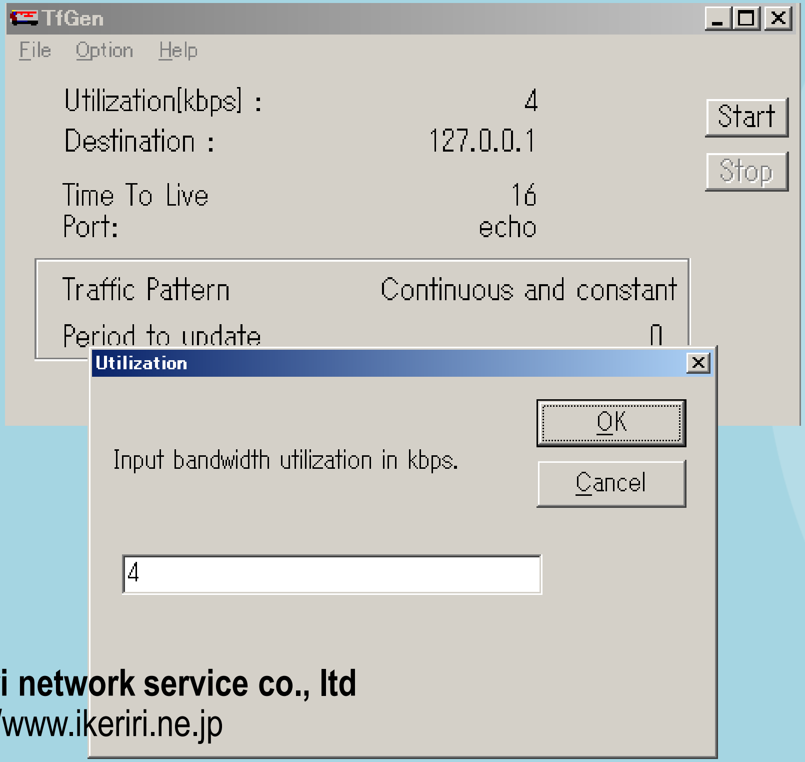
Time To Live: 16

UDP Port

● Default port(echo)

○ User definable port

(User definable): 7



TfGen

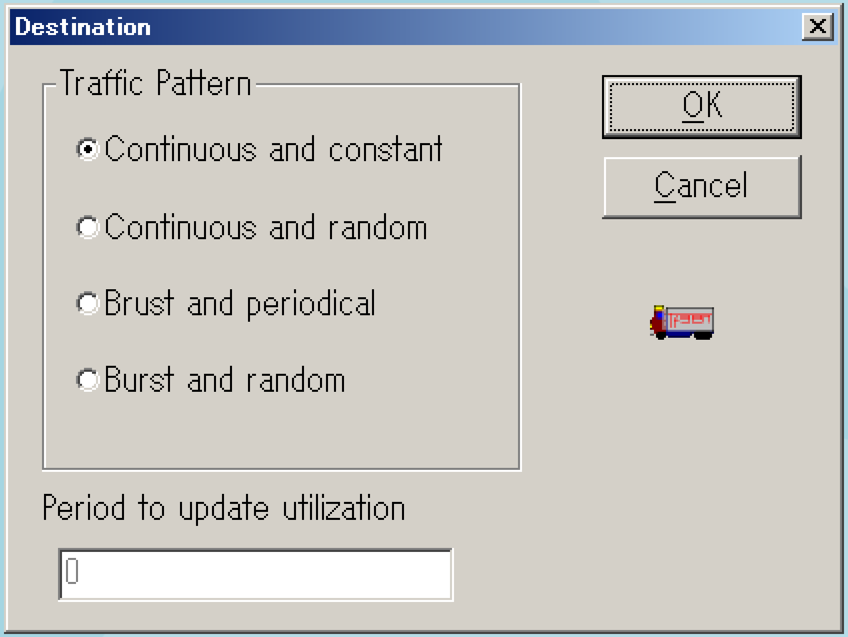
Utilization[kbps] : 4
Destination : 127.0.0.1
Time To Live 16
Port: echo

Traffic Pattern Continuous and constant
Period to update 0

Utilization

Input bandwidth utilization in kbps.

4



Destination

Traffic Pattern

● Continuous and constant

○ Continuous and random

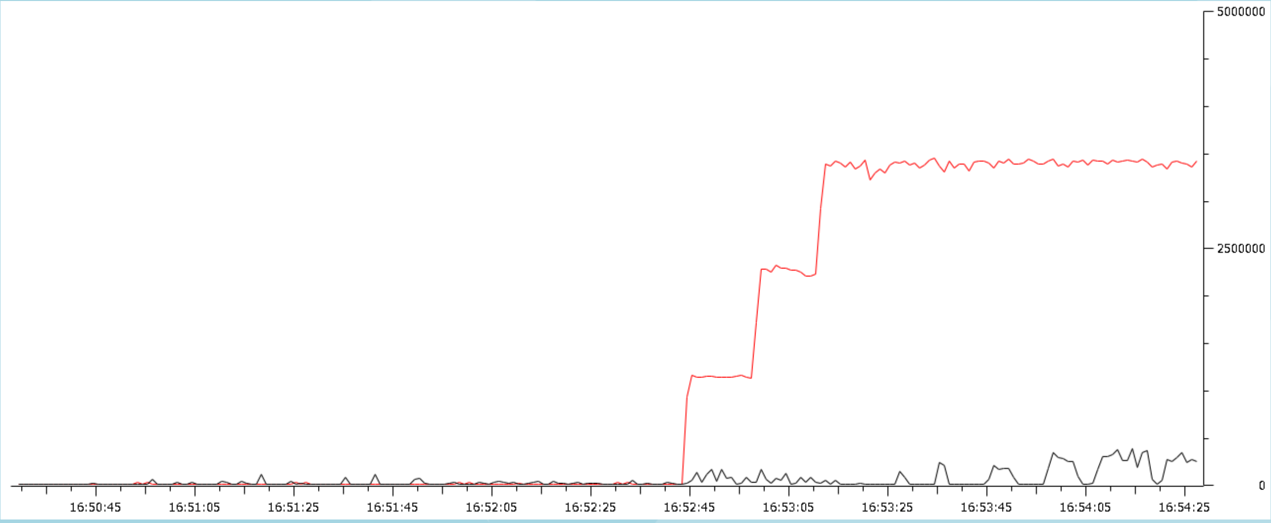
○ Brust and periodical

○ Burst and random

Period to update utilization: 0

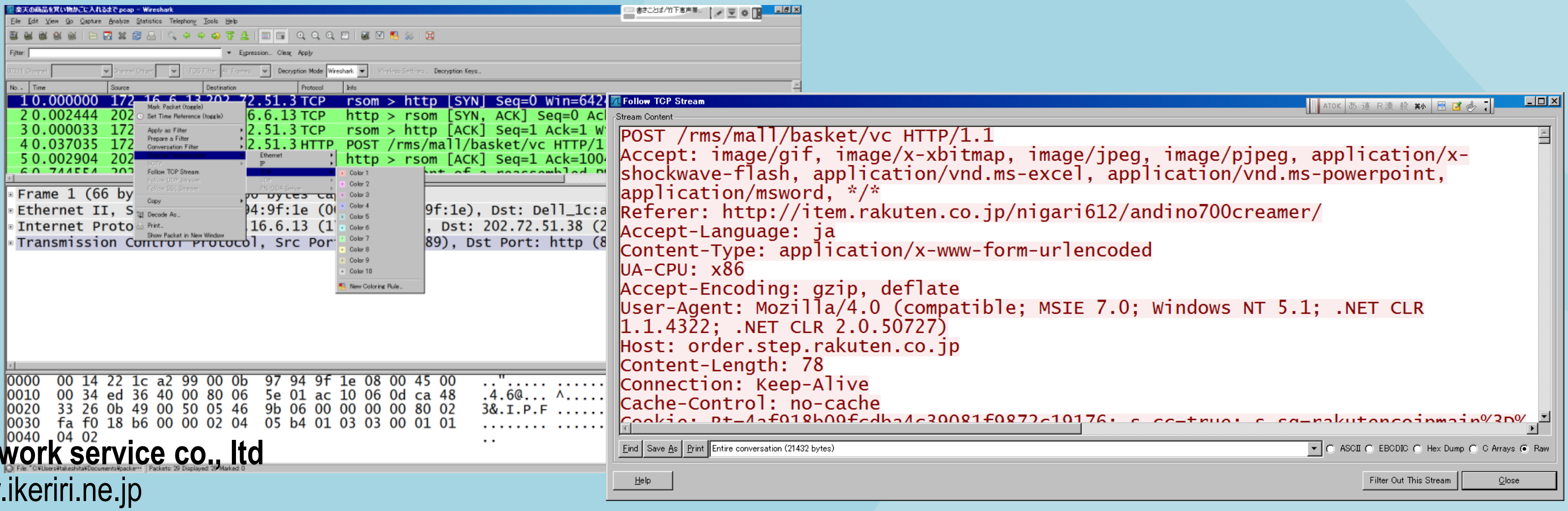
Create IO graph and visualize

- Compare TCP (connection oriented) and UDP (connectionless protocol) and visualize.
- Lets use IO graph function and filter packet by protocol
- Set X axis to seconds and Y axis to bit/tick (means bps)



Check streams TCP/UDP

- Wireshark set stream ID (tcp.stream) in each TCP connection automatically.
- Filter by tcp stream number and colorize conversation.
- Check bytes using “Follow TCP Stream”
- UDP stream is also analyzing by “Follow UDP Stream”



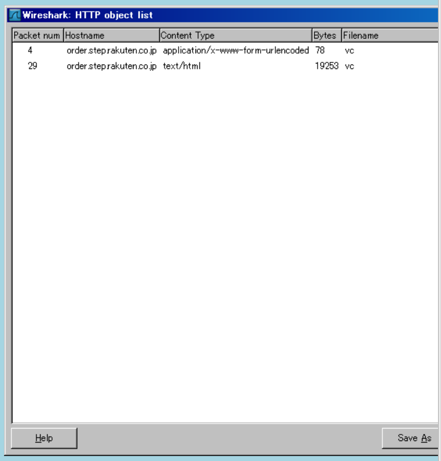
```
1 0.000000  172.16.6.13 202.72.51.3 TCP   rsom > http [SYN] Seq=0 Win=642
2 0.002444  202          6.6.13     TCP   http > rsom [SYN, ACK] Seq=0 Ac
3 0.000033  172          2.51.3     TCP   rsom > http [ACK] Seq=1 Ack=1 W
4 0.037035  172          2.51.3     HTTP  POST /rms/mall/basket/vc HTTP/1
5 0.002904  202                            http > rsom [ACK] Seq=1 Ack=100

Frame 1 (66 by
Ethernet II, S          94:9f:1e (0(          9f:1e), Dst: Dell_1c:a
Internet Proto          16.6.13 (1            , Dst: 202.72.51.38 (2
Transmission Control Protocol, Src Por        89), Dst Port: http (8

0000  00 14 22 1c a2 99 00 0b  97 94 9f 1e 08 00 45 00   .."..... ......
0010  00 34 ed 36 40 00 80 06  5e 01 ac 10 06 0d ca 48   .4.6@... ^......
0020  33 26 0b 49 00 50 05 46  9b 06 00 00 00 00 80 02   3&.I.P.F ......
0030  fa f0 18 b6 00 00 02 04  05 b4 01 03 03 00 01 01   ........ ......
0040  04 02                                              ..

File: "C:¥Users¥takeshita¥Documents¥packe... Packets: 29 Displayed: 29 Marked: 0
```

```
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
Cookie: Rt=4af918b09fcdba4c39081f9872c19176; s_cc=true; s_sq=rakutencoinmain%3D%
```

Ikeriri network service co., ltd
http://www.ikeriri.ne.jp

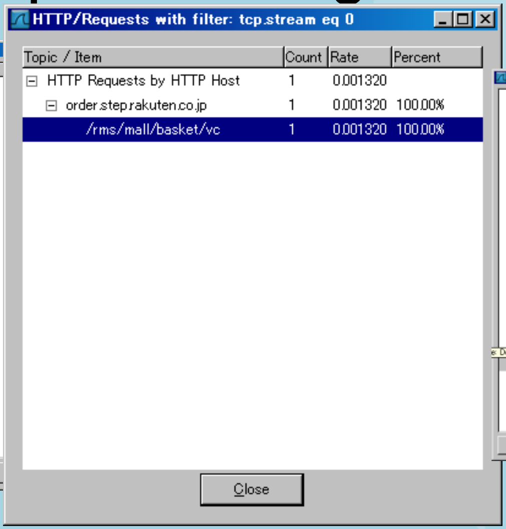
Export function is very good for HTTP

- We can restore HTTP data from WEB communication pcap/pcapng files by File>Export>Object>HTTP
- HTTP statistics is important
  the count value means Web application performance
  1 image map vs. 100 gif file



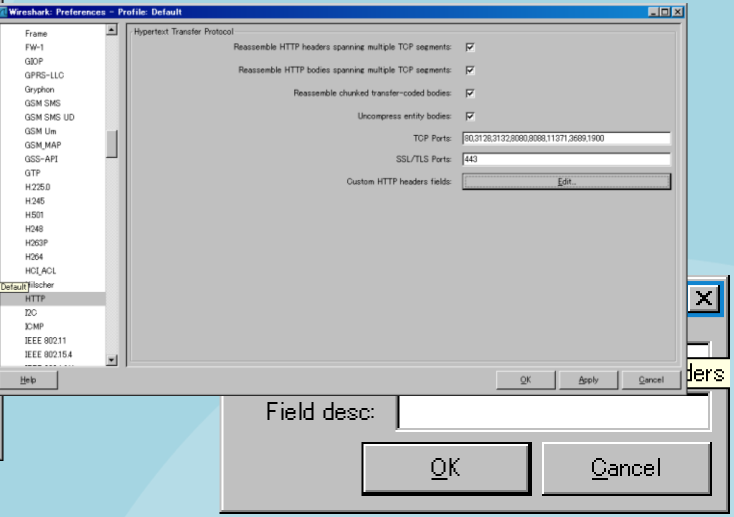
Wireshark: HTTP object list

Packet num	Hostname	Content Type	Bytes	Filename
4	order.step.rakuten.co.jp	application/x-www-form-urlencoded	78	vc
29	order.step.rakuten.co.jp	text/html	19253	vc



HTTP/Requests with filter: tcp.stream eq 0

Topic / Item	Count	Rate	Percent
⊟ HTTP Requests by HTTP Host	1	0.001320	
⊟ order.step.rakuten.co.jp	1	0.001320	100.00%
/rms/mall/basket/vc	1	0.001320	100.00%



Wireshark: Preferences - Profile: Default

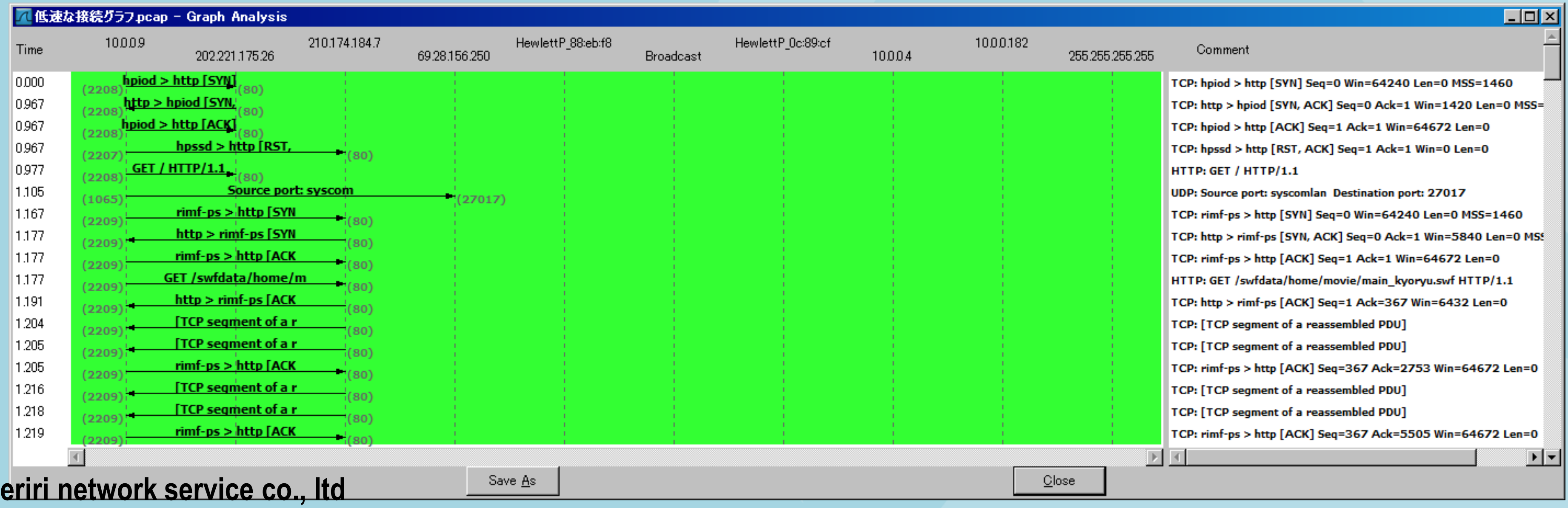
Hypertext Transfer Protocol

Reassemble HTTP headers spanning multiple TCP segments: ☑
Reassemble HTTP bodies spanning multiple TCP segments: ☑
Reassemble chunked transfer-coded bodies: ☑
Uncompress entity bodies: ☑
TCP Ports: 80,3128,3132,8080,8088,11371,3689,1900
SSL/TLS Ports: 443
Custom HTTP headers fields: Edit...

Field desc:

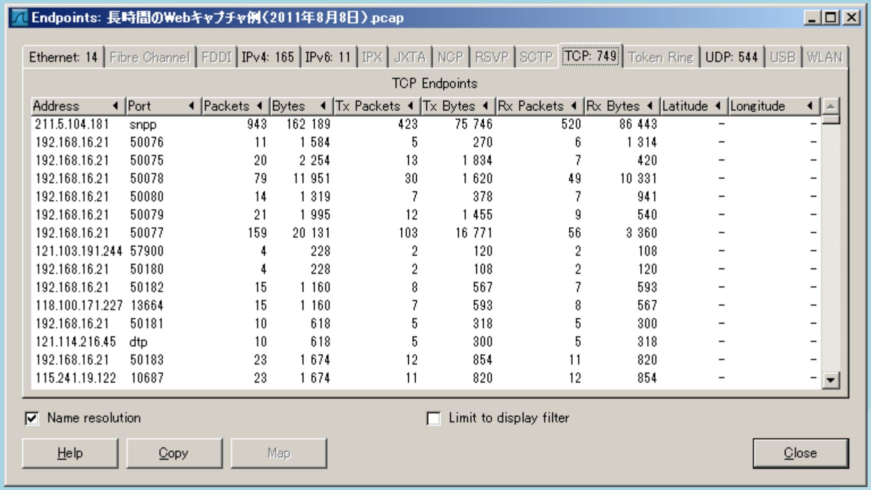
FlowGraph gives you a new look of debugging

- Statistics>FlowGraph and maximize the screen
- Display filter is very good ways to create good visualization.
- If you need to follow TCP, set graph to TCP graph.
- Compare behavior with RFC and standards



```
0.000  TCP: hpiod > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460
0.967  TCP: http > hpiod [SYN, ACK] Seq=0 Ack=1 Win=1420 Len=0 MSS=
0.967  TCP: hpiod > http [ACK] Seq=1 Ack=1 Win=64672 Len=0
0.967  TCP: hpssd > http [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
0.977  HTTP: GET / HTTP/1.1
1.105  UDP: Source port: syscomlan  Destination port: 27017
1.167  TCP: rimf-ps > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460
1.177  TCP: http > rimf-ps [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS
1.177  TCP: rimf-ps > http [ACK] Seq=1 Ack=1 Win=64672 Len=0
1.177  HTTP: GET /swfdata/home/movie/main_kyoryu.swf HTTP/1.1
1.191  TCP: http > rimf-ps [ACK] Seq=1 Ack=367 Win=6432 Len=0
1.204  TCP: [TCP segment of a reassembled PDU]
1.205  TCP: [TCP segment of a reassembled PDU]
1.205  TCP: rimf-ps > http [ACK] Seq=367 Ack=2753 Win=64672 Len=0
1.216  TCP: [TCP segment of a reassembled PDU]
1.218  TCP: [TCP segment of a reassembled PDU]
1.219  TCP: rimf-ps > http [ACK] Seq=367 Ack=5505 Win=64672 Len=0
```

Trend analysis BASIC
TopN style, and drilled down in details



Endpoints: 長時間のWebキャプチャ例（2011年8月8日）.pcap

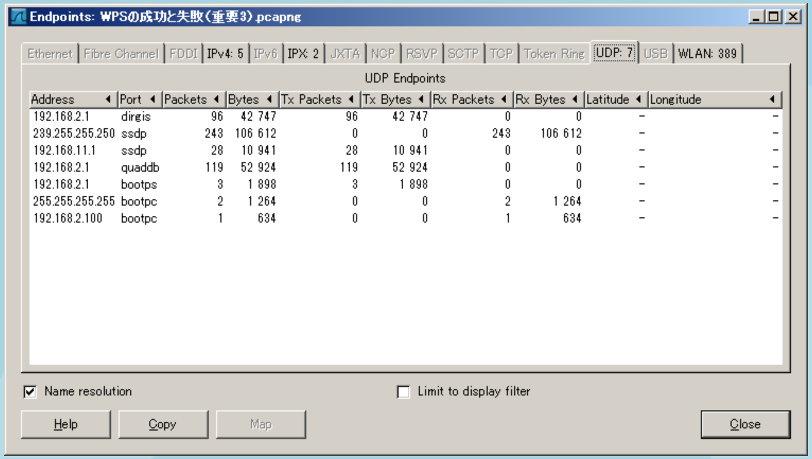
Ethernet: 14 | Fibre Channel | FDDI | IPv4: 165 | IPv6: 11 | IPX | JXTA | NCP | RSVP | SCTP | TCP: 749 | Token Ring | UDP: 544 | USB | WLAN

TCP Endpoints

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude	Longitude
211.5.104.181	snpp	943	162 189	423	75 746	520	86 443	-	-
192.168.16.21	50076	11	1 584	5	270	6	1 314	-	-
192.168.16.21	50075	20	2 254	13	1 834	7	420	-	-
192.168.16.21	50078	79	11 951	30	1 620	49	10 331	-	-
192.168.16.21	50080	14	1 319	7	378	7	941	-	-
192.168.16.21	50079	21	1 995	12	1 455	9	540	-	-
192.168.16.21	50077	159	20 131	103	16 771	56	3 360	-	-
121.103.191.244	57900	4	228	2	120	2	108	-	-
192.168.16.21	50180	4	228	2	108	2	120	-	-
192.168.16.21	50182	15	1 160	8	567	7	593	-	-
118.100.171.227	13664	15	1 160	7	593	8	567	-	-
192.168.16.21	50181	10	618	5	318	5	300	-	-
121.114.216.45	dtp	10	618	5	300	5	318	-	-
192.168.16.21	50183	23	1 674	12	854	11	820	-	-
115.241.19.122	10687	23	1 674	11	820	12	854	-	-

☑ Name resolution

□ Limit to display filter



Endpoints: WPSの成功と失敗（重要3）.pcapng

Ethernet | Fibre Channel | FDDI | IPv4: 5 | IPv6 | IPX: 2 | JXTA | NCP | RSVP | SCTP | TCP | Token Ring | UDP: 7 | USB | WLAN: 389

UDP Endpoints

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude	Longitude
192.168.2.1	dirgis	96	42 747	96	42 747	0	0	-	-
239.255.255.250	ssdp	243	106 612	0	0	243	106 612	-	-
192.168.11.1	ssdp	28	10 941	28	10 941	0	0	-	-
192.168.2.1	quaddb	119	52 924	119	52 924	0	0	-	-
192.168.2.1	bootps	3	1 898	3	1 898	0	0	-	-
255.255.255.255	bootpc	2	1 264	0	0	2	1 264	-	-
192.168.2.100	bootpc	1	634	0	0	1	634	-	-

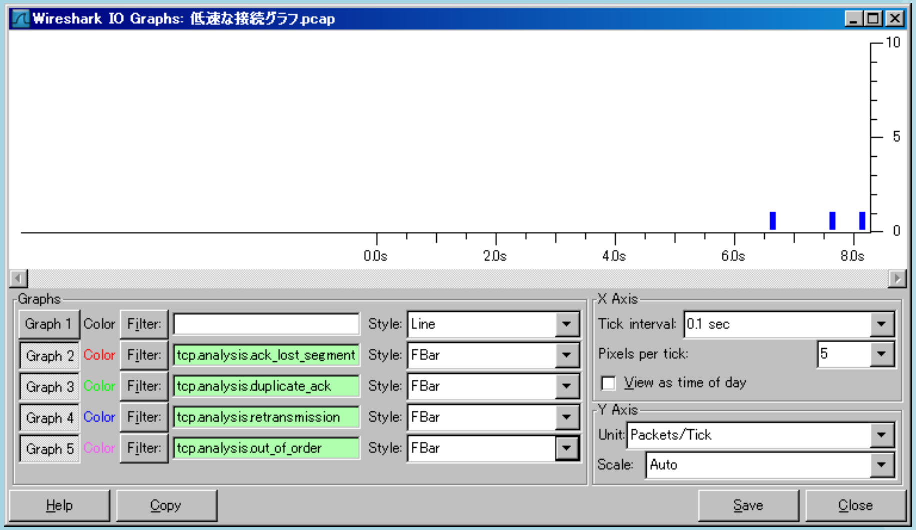
☑ Name resolution

□ Limit to display filter

1. Create TOPN list table of Endpoint and filtered
2. Create N<>other list table of Conversation
3. Then create protocol hierarchy and check stream

Utilize IO graph in two ways

- Set packets to Y axis to create ERROR graph
  Histogram style is good for Frequency graph
- Set bit to Y axis to create BPS graph
  line style is good for amount graph.



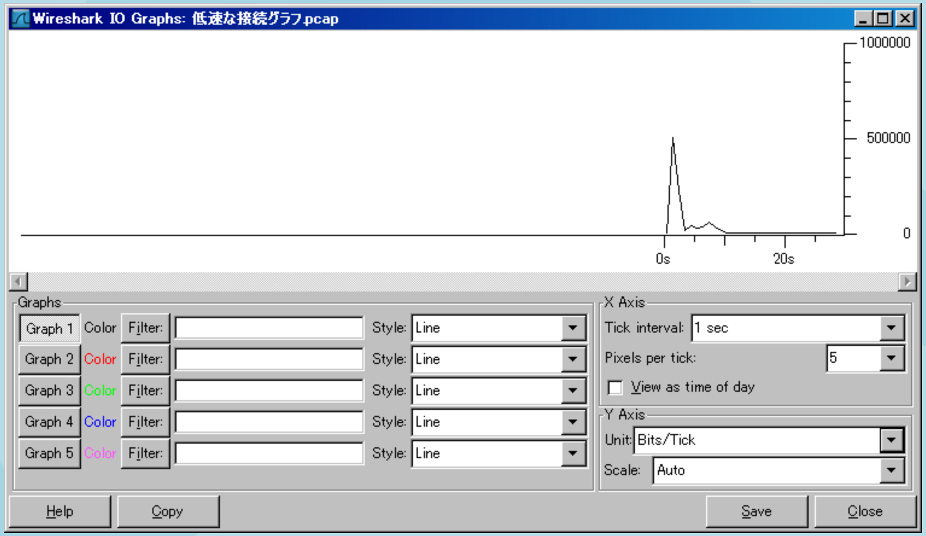
Wireshark IO Graphs: 低速な接続グラフ.pcap

Graphs

Graph 1	Color	Filter:		Style: Line
Graph 2	Color	Filter:	tcp.analysis.ack_lost_segment	Style: FBar
Graph 3	Color	Filter:	tcp.analysis.duplicate_ack	Style: FBar
Graph 4	Color	Filter:	tcp.analysis.retransmission	Style: FBar
Graph 5	Color	Filter:	tcp.analysis.out_of_order	Style: FBar

X Axis
Tick interval: 0.1 sec
Pixels per tick: 5
□ View as time of day

Y Axis
Unit: Packets/Tick
Scale: Auto



Wireshark IO Graphs: 低速な接続グラフ.pcap

Graphs

Graph 1	Color	Filter:		Style: Line
Graph 2	Color	Filter:		Style: Line
Graph 3	Color	Filter:		Style: Line
Graph 4	Color	Filter:		Style: Line
Graph 5	Color	Filter:		Style: Line

X Axis
Tick interval: 1 sec
Pixels per tick: 5
□ View as time of day

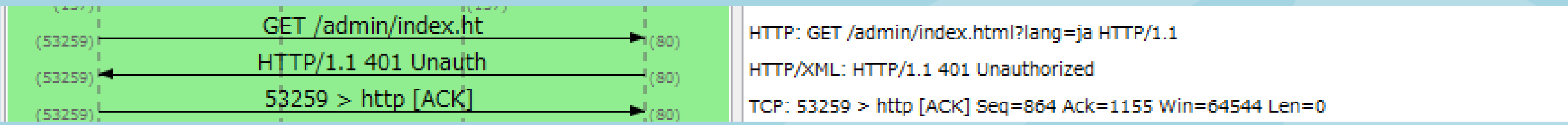
Y Axis
Unit: Bits/Tick
Scale: Auto

Ikeriri network service co., ltd
http://www.ikeriri.ne.jp

Digest Auth SUCCESS/FAIL

アクセス－認証（失敗）－認証（成功）.pcapng

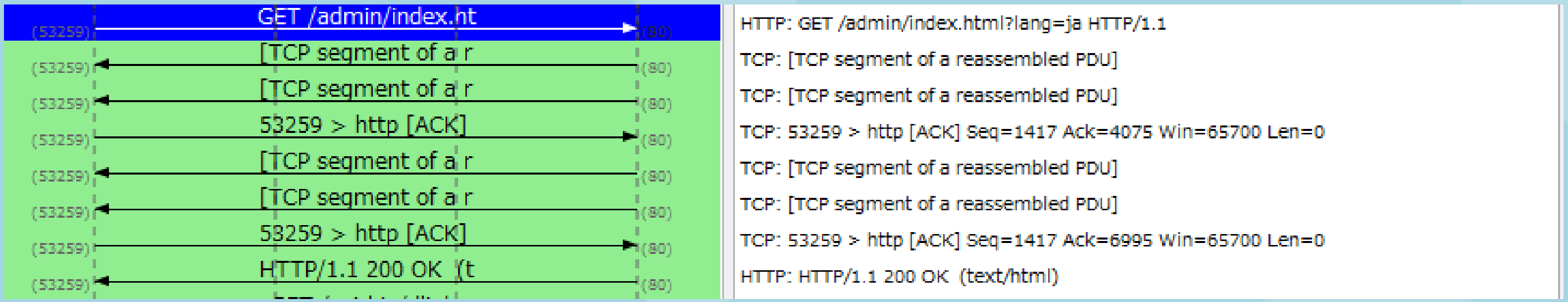
- Digest authentication will be failed when ID/Password mismatch



```
(53259)  GET /admin/index.ht        (80)    HTTP: GET /admin/index.html?lang=ja HTTP/1.1
(53259)  HTTP/1.1 401 Unauth        (80)    HTTP/XML: HTTP/1.1 401 Unauthorized
(53259)  53259 > http [ACK]         (80)    TCP: 53259 > http [ACK] Seq=864 Ack=1155 Win=64544 Len=0
```

401 Unauthorized

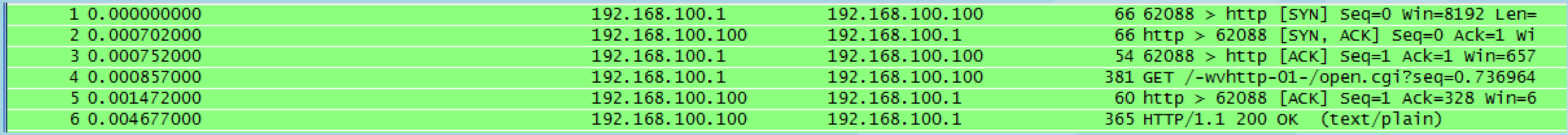
- If success



```
(53259)  GET /admin/index.ht        (80)    HTTP: GET /admin/index.html?lang=ja HTTP/1.1
(53259)  [TCP segment of a r        (80)    TCP: [TCP segment of a reassembled PDU]
(53259)  [TCP segment of a r        (80)    TCP: [TCP segment of a reassembled PDU]
(53259)  53259 > http [ACK]         (80)    TCP: 53259 > http [ACK] Seq=1417 Ack=4075 Win=65700 Len=0
(53259)  [TCP segment of a r        (80)    TCP: [TCP segment of a reassembled PDU]
(53259)  [TCP segment of a r        (80)    TCP: [TCP segment of a reassembled PDU]
(53259)  53259 > http [ACK]         (80)    TCP: 53259 > http [ACK] Seq=1417 Ack=6995 Win=65700 Len=0
(53259)  HTTP/1.1 200 OK  (t        (80)    HTTP: HTTP/1.1 200 OK  (text/html)
```

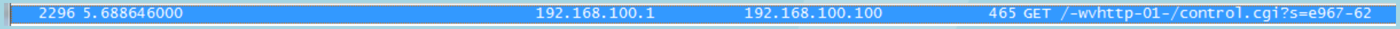
Sample trace

- Try to click “Home” and check trace file.



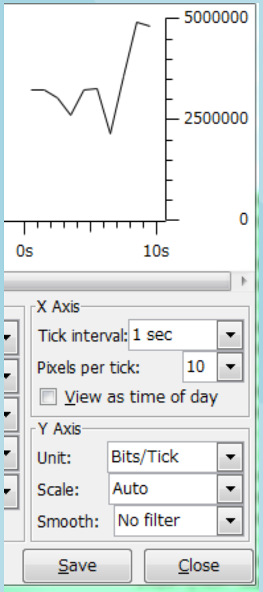
```
 1 0.000000000      192.168.100.1      192.168.100.100      66 62088 > http [SYN] Seq=0 Win=8192 Len=
 2 0.000702000      192.168.100.100    192.168.100.1        66 http > 62088 [SYN, ACK] Seq=0 Ack=1 Wi
 3 0.000752000      192.168.100.1      192.168.100.100      54 62088 > http [ACK] Seq=1 Ack=1 Win=657
 4 0.000857000      192.168.100.1      192.168.100.100     381 GET /-wvhttp-01-/open.cgi?seq=0.736964
 5 0.001472000      192.168.100.100    192.168.100.1        60 http > 62088 [ACK] Seq=1 Ack=328 Win=6
 6 0.004677000      192.168.100.100    192.168.100.1       365 HTTP/1.1 200 OK  (text/plain)
```

- Once called control.cgi and c.1.ae.brightness==0 c.1.wb==auto c.1.shade==off   c.1.focus==auto c.1.zoom==6040 c.1.pan:=-4014   c.1.tilt:=-153 value send to the server



```
2296 5.688646000      192.168.100.1      192.168.100.100      465 GET /-wvhttp-01-/control.cgi?s=e967-62
```

- Moving picture needs 5Mbps how about creating IO graph and set Y axis as a bit/tick



Tick interval: 1 sec
Pixels per tick: 10
□ View as time of day
Unit: Bits/Tick
Scale: Auto
Smooth: No filter

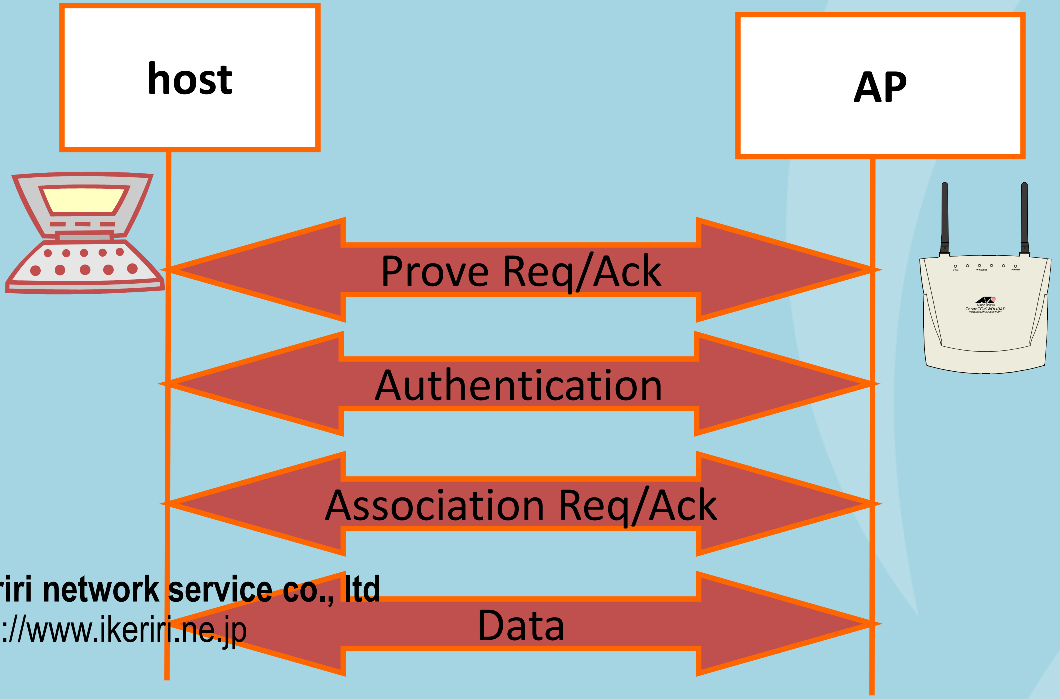
Wireless specific debugging

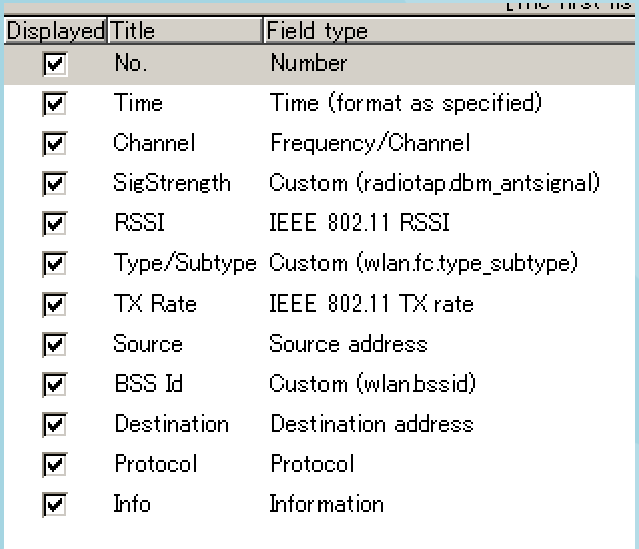
Use AirPcap and set clear text if possible

- Need Jumbo frame or IEEE802.11a/n go NX
- We have to capture their own 4 way handshake to decrypt pcap file secured by WPA2-PSK,
- Its terrible troublesome to match between the WPA2 handshake and the communication packet.
- Set free channel
  in test capture
  (android 14ch NG)

Type/Subtype, TX rate, BSSID, CH, RSSI

- In Wireless environment, please watch important field of IEEE802.11 header, physical (radiotap/PPI) header (Type/Subtype, TX Rate, BSSID, CH, RSSI)
- Many troubles are occurred before Data exchange





Displayed	Title	Field type
☑	No.	Number
☑	Time	Time (format as specified)
☑	Channel	Frequency/Channel
☑	SigStrength	Custom (radiotap.dbm_antsignal)
☑	RSSI	IEEE 802.11 RSSI
☑	Type/Subtype	Custom (wlan.fc.type_subtype)
☑	TX Rate	IEEE 802.11 TX rate
☑	Source	Source address
☑	BSS Id	Custom (wlan.bssid)
☑	Destination	Destination address
☑	Protocol	Protocol
☑	Info	Information

Ikeriri network service co., ltd
http://www.ikeriri.ne.jp

Between deployments and standards

- IEEE802.11 and related standards, protocols are not so punctual and irritate rules (they are not described in detail and all step, procedure, but just set the summary)
- For example WPS is famous and many user use the PIN or button settings, but the deployments in Wireless devices differs a lot
- We have to check sequences in detail for debugging

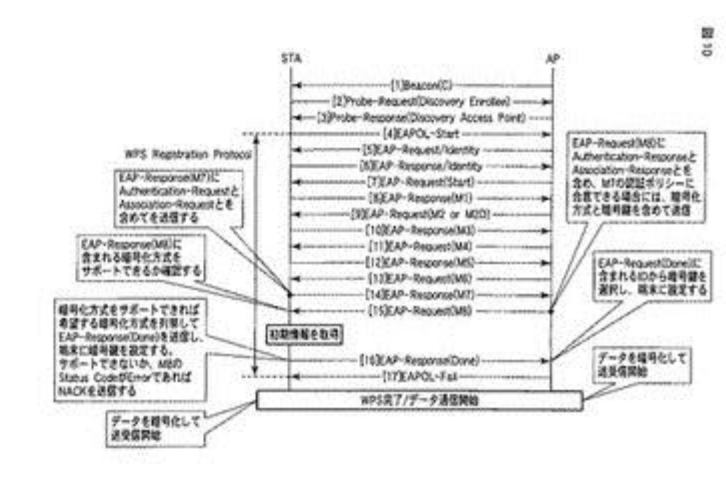
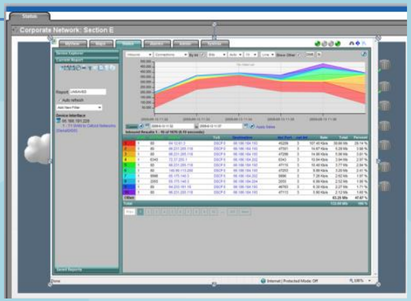


図10
STA
AP
[1]Beacon(C)
[2]Probe-Request(Discovery Enrollee)
[3]Probe-Response(Discovery Access Point)
[4]EAPOL-Start
[5]EAP-Request/Identity
[6]EAP-Response/Identity
[7]EAP-Request(Start)
[8]EAP-Response(M1)
[9]EAP-Request(M2 or M2D)
[10]EAP-Response(M3)
[11]EAP-Request(M4)
[12]EAP-Response(M5)
[13]EAP-Request(M6)
[14]EAP-Response(M7)
[15]EAP-Request(M8)
[16]EAP-Response(Done)
[17]EAPOL-Fail
WPS Registration Protocol
EAP-Response(M7)に Authentication-Requestと Association-Requestとを 含めてを送信する
EAP-Response(M8)に 含まれる暗号化方式を サポートできるか確認する
暗号化方式をサポートできれば 希望する暗号化方式を列挙して EAP-Response(Done)を送信し、端末に暗号鍵を設定する。サポートできないか、M8の Status CodeがErrorであれば NACKを送信する
初期情報を取得
WPS完了/データ通信開始
データを暗号化して 送受信開始
EAP-Request(M8)に Authentication-Responseと Association-Responseとを 含め、M1の認証ポリシーに 合意できる場合には、暗号化 方式と暗号鍵を含めて送信
EAP-Request(Done)に 含まれるIDから暗号鍵を 選択し、端末に設定する
データを暗号化して 送受信開始

Ikeriri network service co., ltd
http://www.ikeriri.ne.jp

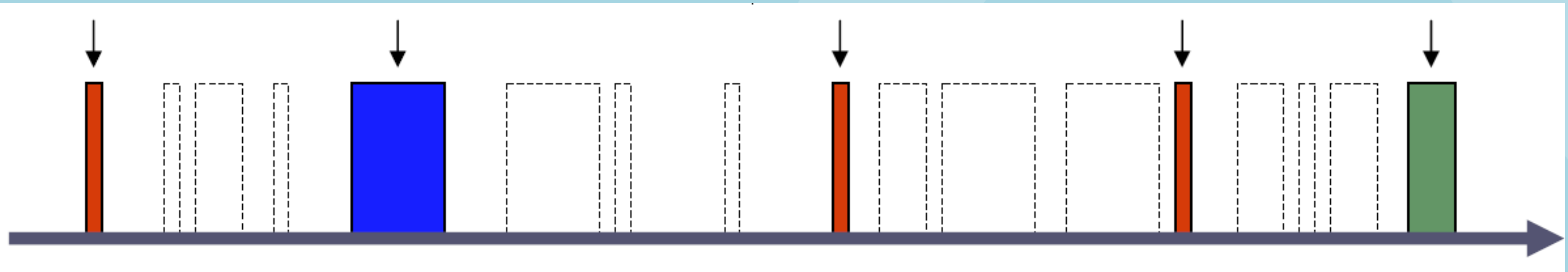
Huge packet debugging

Huge packet case



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

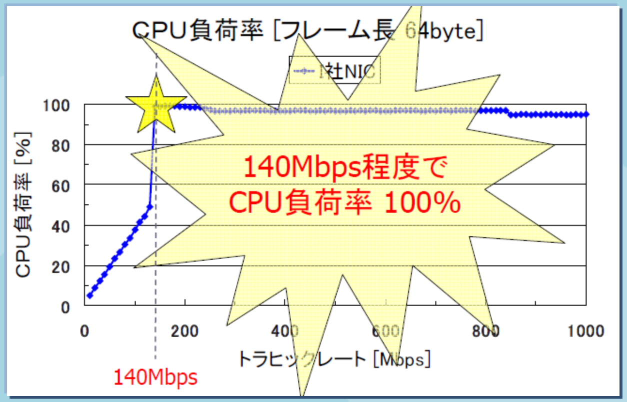
Ignored Ignored Ignored May be

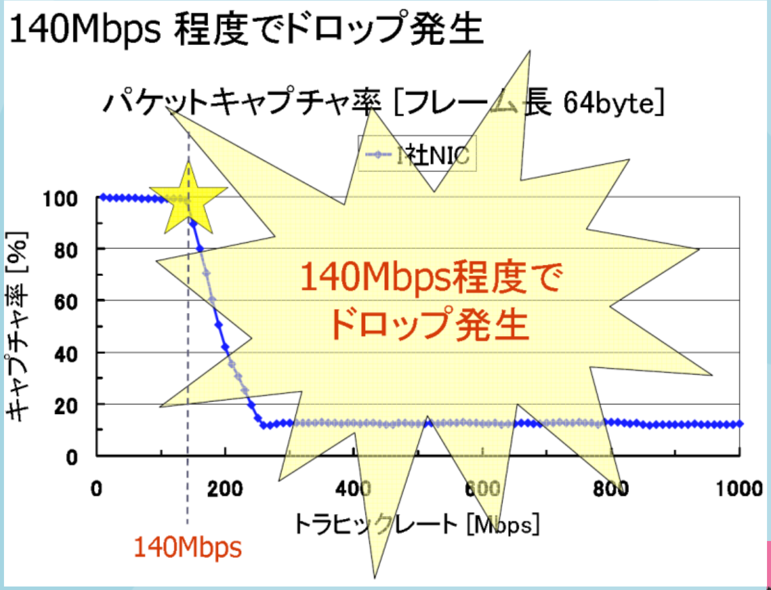


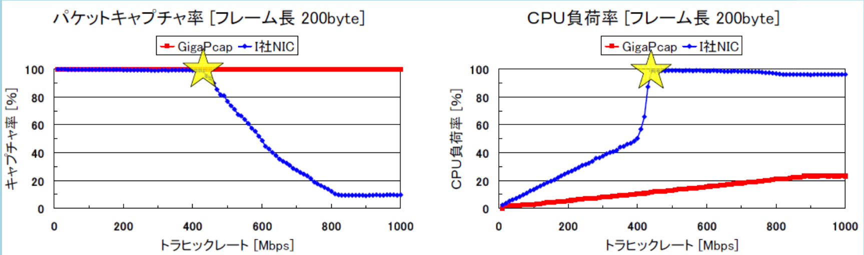
- But small packet (64 bytes – 100 bytes) may be ignored. Some small packet is important symptom of analysis (ARP / TCP SYN / HTTP GET and others)

We need TurboCap

- Typical Intel's GigaNIC (e1000),
  typical Dell PowerEdge2850 / Xeon 2.8GHz
  RAM 1GB (PC3200, DDR2, 400MHz)
- **Threadshould is 140Mbps in Frame size = 64**
- **Frame size = 200 , actual rate 400Mbps**
- Frame size = 1500 , **may be ok**, no problem.
- We need TurboCap



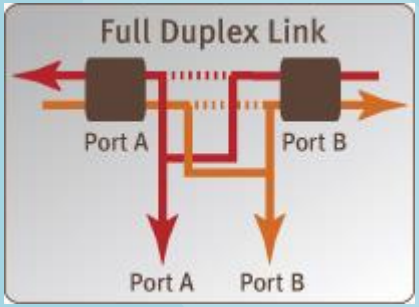




Ikeriri network service co., ltd
http://www.ikeriri.ne.jp

Debugging Environment

- Using TurboCap, MMMM packets received by the application NNNN packets accepted by the filter and dumped to disk

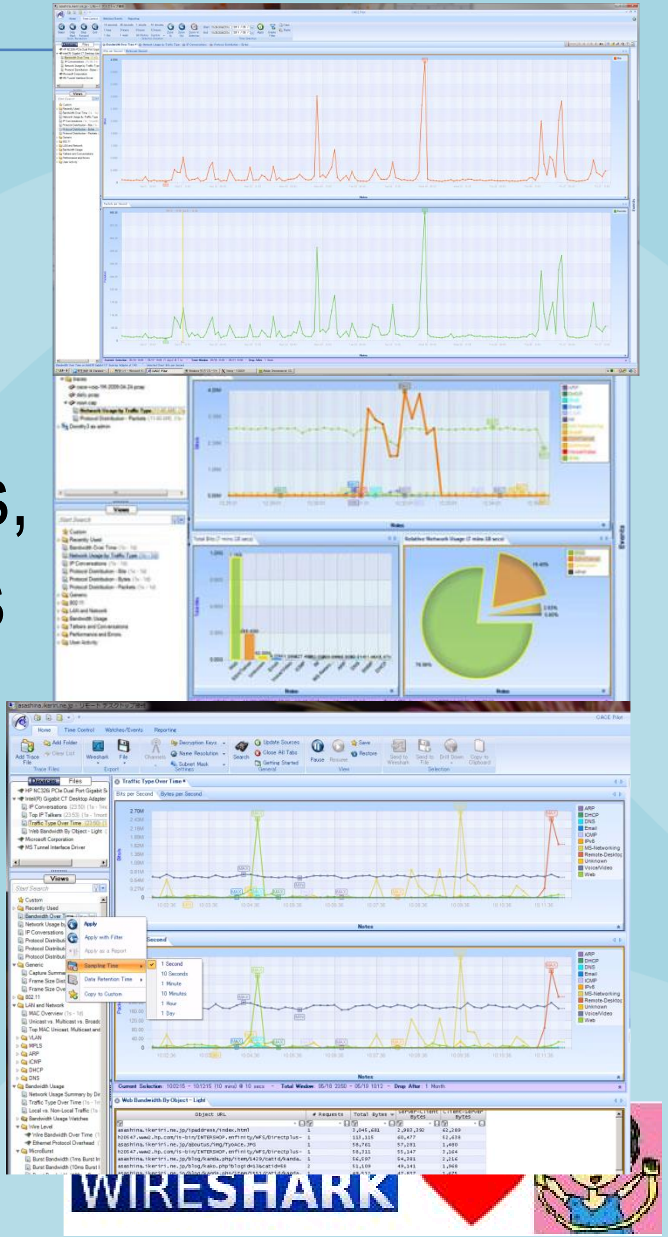


- To fix, Optimize I/O access flow
  packet -> IRQ -> SVC -> driver -> OS
- Use 6 cores Xeon-L5640 and 24GB RAM !
  (power resolve things and no page files)
- Stop tcpdump and create program using pcap libraries in C/C++ (dumpcap.exe)
- Pcap -> standard output -> FIFO -> SQLite
- 3 month no problem

Ikeriri network service co., ltd
http://www.ikeriri.ne.jp

Driving 250GB pcap file with Pilot

- We use 250GB pcap file,
  huge huge file with Cascade
  PilotPE installed into NotePC
- Use view to check macro analysis,
  and finally check the actual pcaps
  using Wireshark
- Only, best, easiest way to
  drive huge pcap file

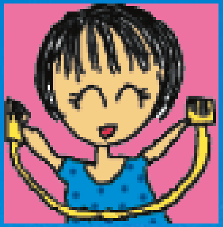


QA and Demonstration

SHARKFEST '13

Wireshark Developer and User Conference

Thank You !!



ikeriri network service

http://www.ikeriri.ne.jp