

T2-1

**Analyzer Placement and Baseline
Techniques**

March 31, 2008

Tony Fortunato

Sr Network Specialist | The Technology Firm

SHARKFEST '08

Foothill College

March 31 - April 2, 2008

About your Presenter

Tony Fortunato, Sr Network Specialist, The Technology Firm

Website: www.thetechfirm.com

A Senior Network Specialist with experience in performance testing, network design, implementation, and troubleshooting LAN/WAN/Wireless networks, desktops and servers since 1989.

Tony has taught at Colleges/Universities, Network/Interop and many onsite corporate settings to thousands of analysts.

Tony is an authorized and certified Fluke Networks and Wireshark Instructor, but trains and uses Sniffer, MRTG and many other products. Tony always demonstrates his vendor neutral approach to network design, support and implementations.

Tony has architected, installed and supported various types of Residential Wireless High Speed as well as hundreds of WIFI hotspots.. Tony combines custom programs, open source and commercial software to ensure a simple support infrastructure.

Tony works on networks from 2 to 120,000 nodes and specializes in post installation performance/design review. This process involves using various tools (Protocol analyzers, traffic generators and network management) and working on multi-vendor equipment (switches, routers, servers, etc).

Tony works at customer sites within a range of capacities from project management, network design, consulting, troubleshooting, designing customized courses and assisting with installing physical equipment.

Baselining vs Troubleshooting

I personally believe that it doesn't matter if you are troubleshooting or application baselining, the techniques and issues described in the next few slides apply to both.

I always say that, *“if you do a baseline correctly, you will find something to investigate or tweak. In some cases, you may discover that the application has always had issues, but no one has ever looked into it.”*

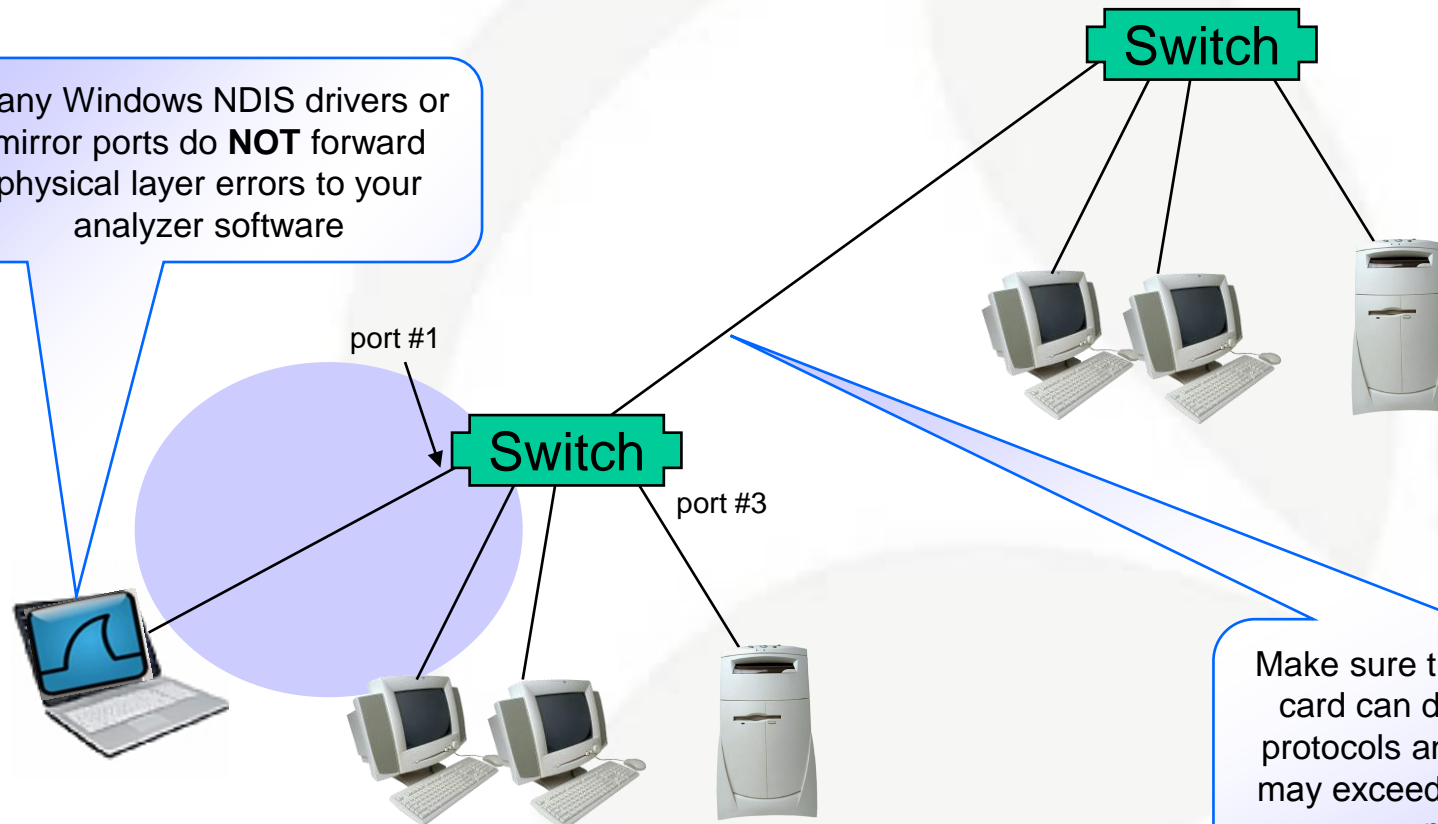
Methodology In a Nutshell

The overall methodology required to properly analyze a problem or document an application behavior falls into the following categories;

- Concisely identify the issue, purpose or goal of the exercise
- Position yourself to properly capture the data
- Configure the tool to capture efficiently
- Use various reports or techniques to investigate any issues
- Document any anomalies and recommended changes
- Test..... And Document!!!!!!!!!!!!!!!

Analyzer Placement: Switches

Many Windows NDIS drivers or mirror ports do **NOT** forward physical layer errors to your analyzer software

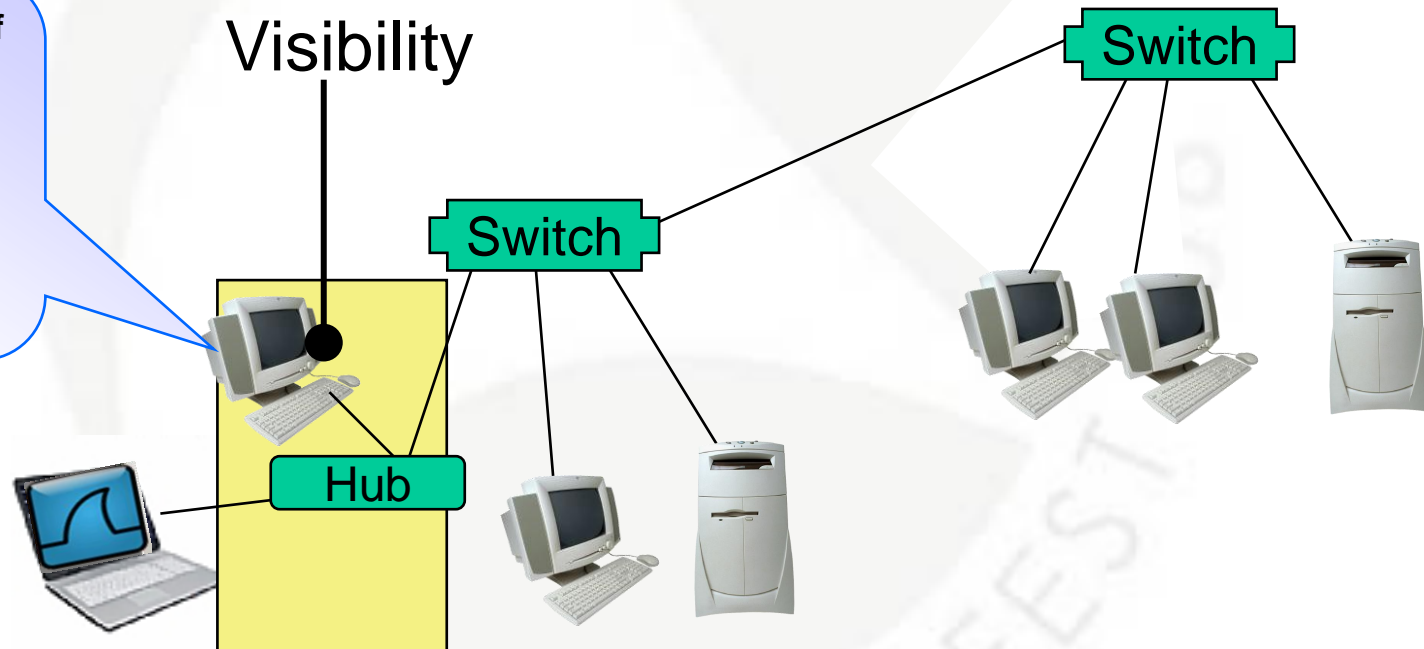


Make sure that your analyzer and card can decode trunk specific protocols and packets since they may exceed Ethernet's maximum packet Size

“Hubbing Out”

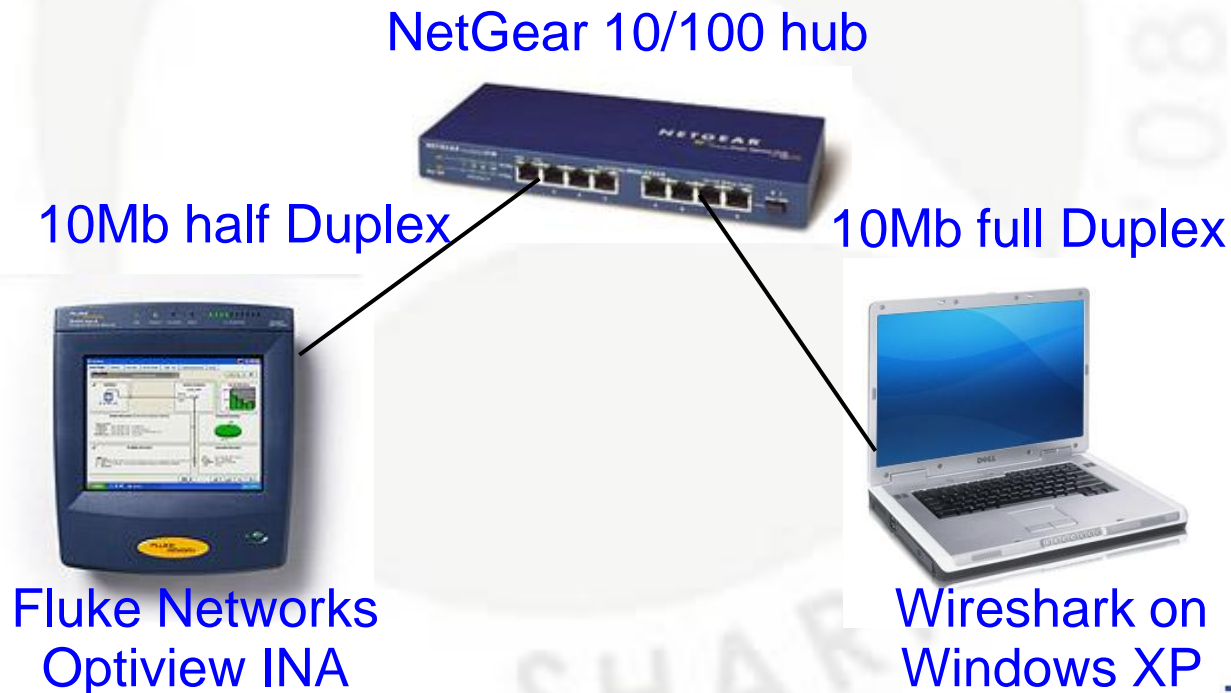
Hubbing-out refers to using a hub to see traffic from a host. The hub in this case is acting as a half-duplex tap.

Since hubs are **Half Duplex**, double check switch ports and workstation settings are set to half duplex to avoid duplex mismatch issues.



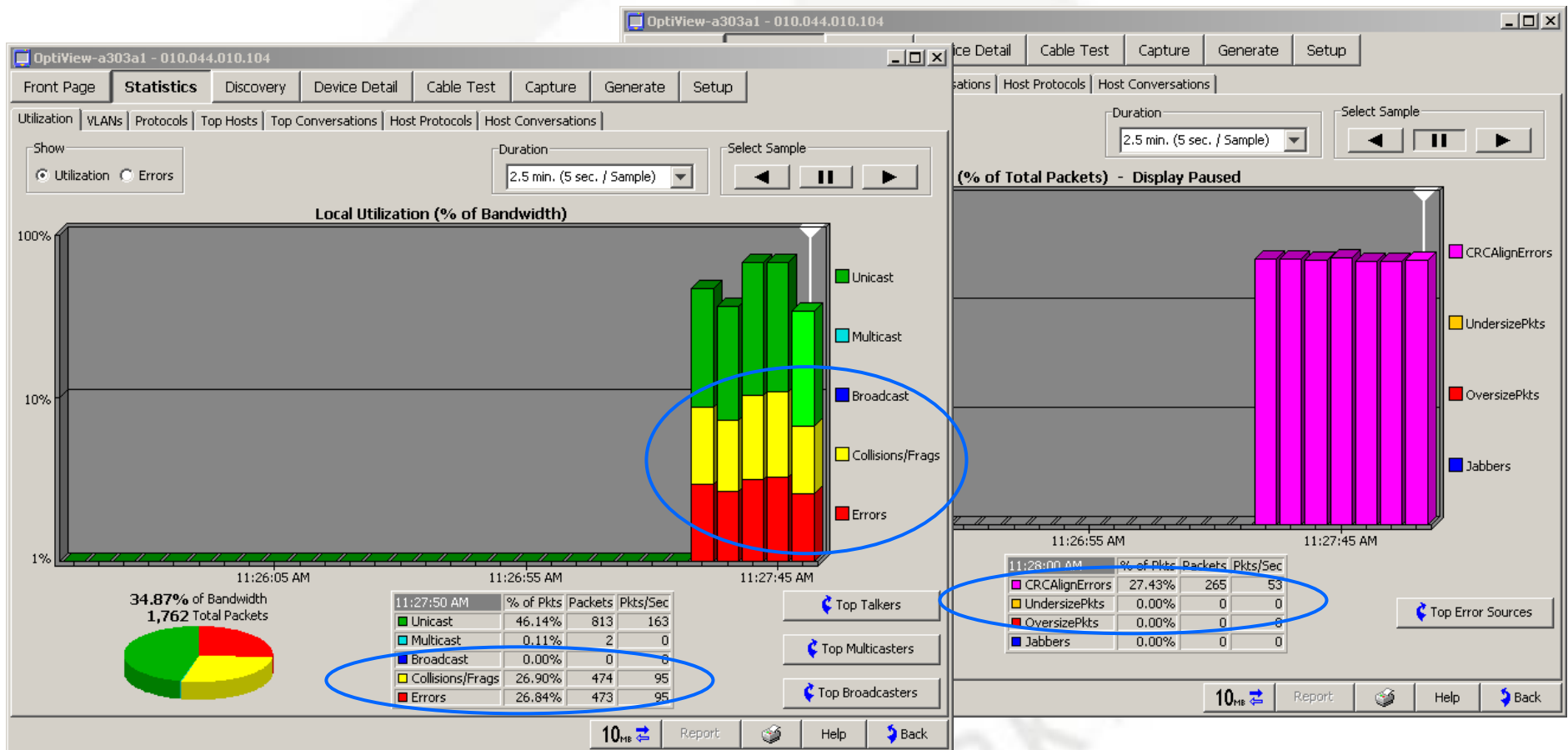
Lab Setup – Physical Level Errors

- I wanted to see the effects of half/full duplex mismatch and the resulting effects with various tools
- I simply performed a file transfer between both devices and intentionally set a duplex mismatch between the laptop (full duplex) and the hub (half duplex)



Lab Setup

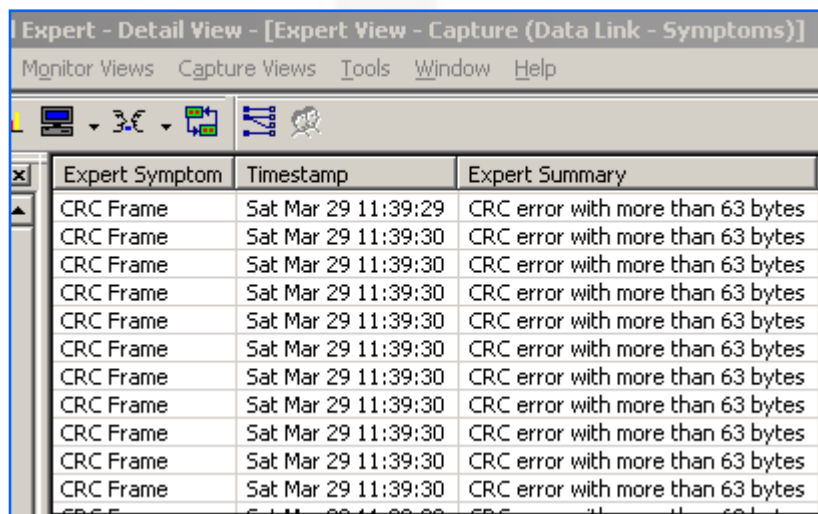
I wanted to confirm via my Fluke Analyzer that I was causing physical level errors and obviously I was.



Lab Setup

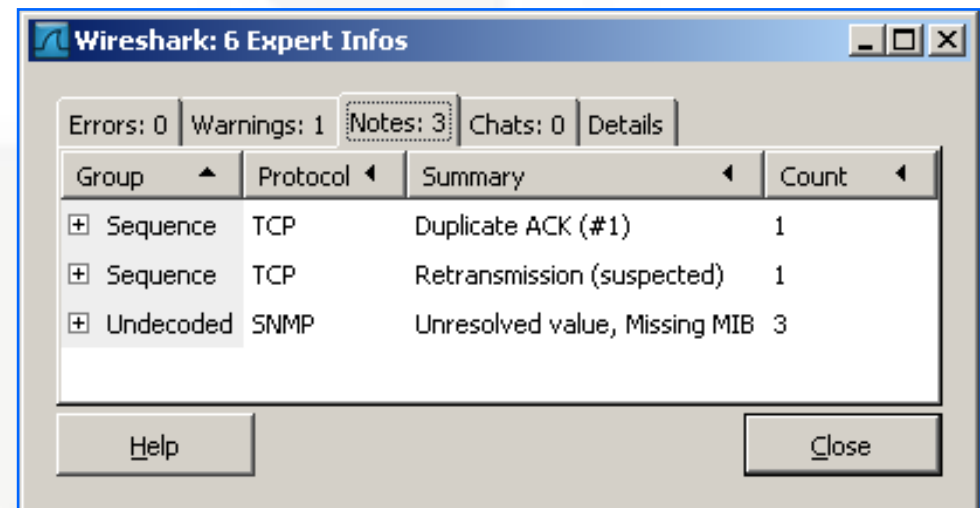
Even at the packet level, the hardware based analyzer did see the errors, where Wireshark (with the Windows default NDIS driver) did not.

With Wireshark, we have to learn to spot the effects of physical errors. For example, in this case we see ***Duplicate ACK's*** and ***Retransmissions***.



Expert - Detail View - [Expert View - Capture (Data Link - Symptoms)]

Expert Symptom	Timestamp	Expert Summary
CRC Frame	Sat Mar 29 11:39:29	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes
CRC Frame	Sat Mar 29 11:39:30	CRC error with more than 63 bytes



Wireshark: 6 Expert Infos

Errors: 0 | Warnings: 1 | Notes: 3 | Chats: 0 | Details

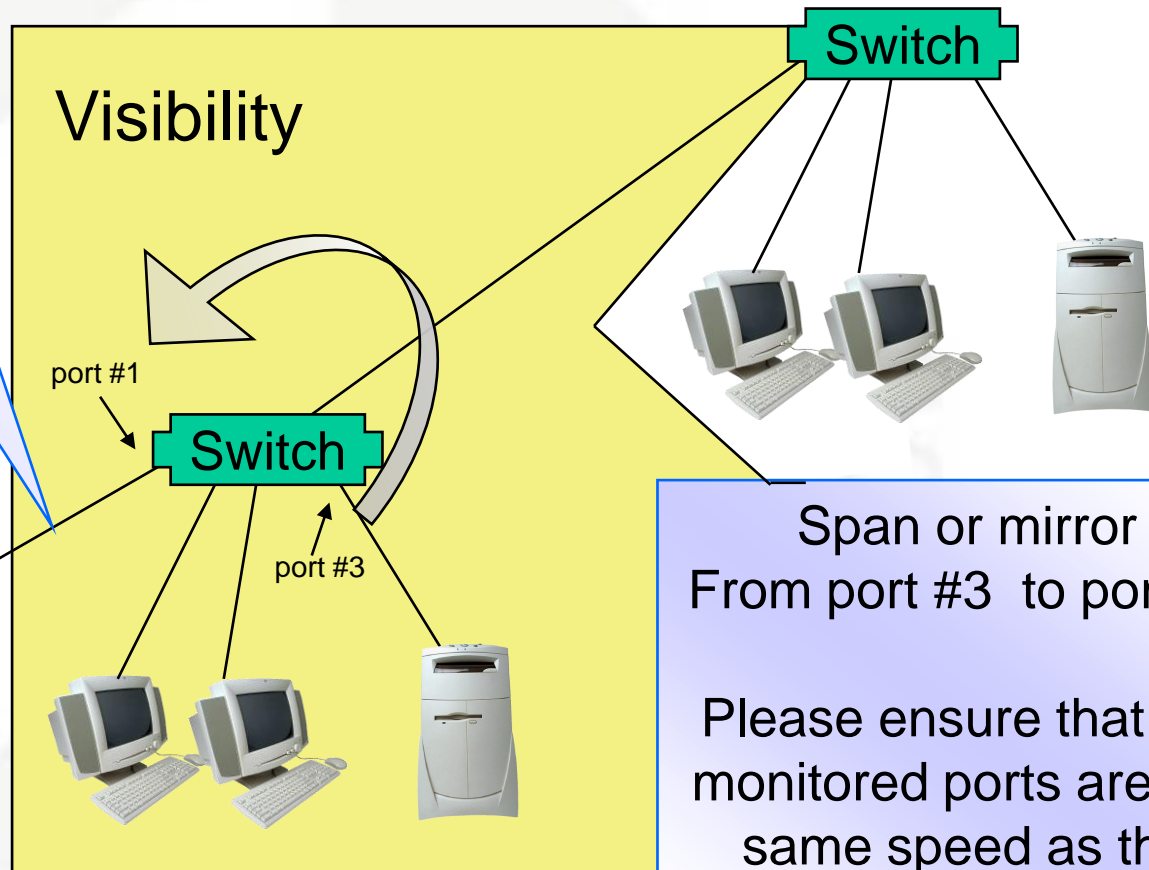
Group	Protocol	Summary	Count
+	Sequence	TCP Duplicate ACK (#1)	1
+	Sequence	TCP Retransmission (suspected)	1
+	Undecoded	SNMP Unresolved value, Missing MIB	3

Help Close

Port Spanning or Mirroring

Many monitor ports do not forward packets with physical level errors.

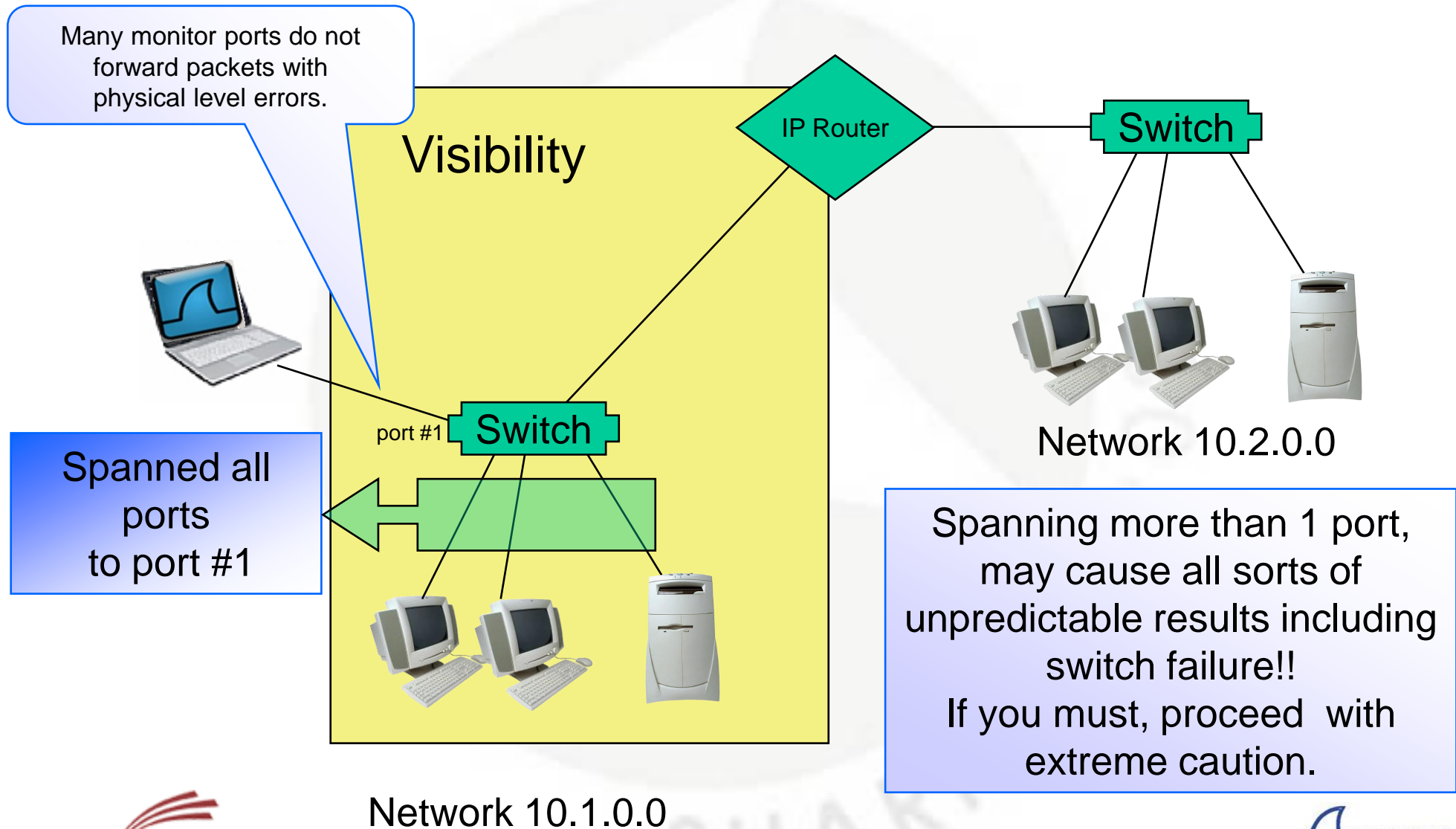
Try to avoid mirroring or spanning multiple ports or entire Vlans



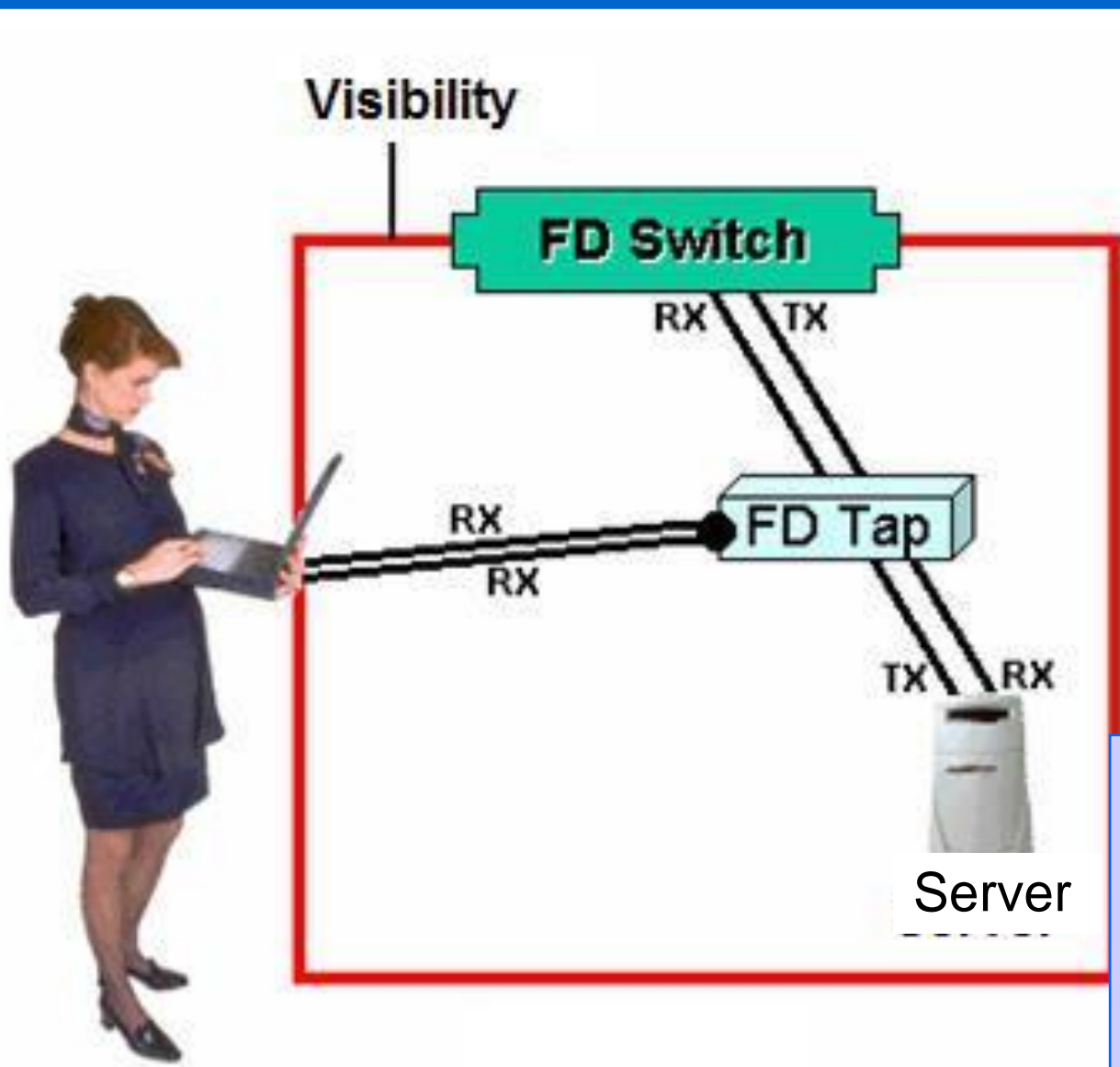
Span or mirror
From port #3 to port #1

Please ensure that the monitored ports are the same speed as the source ports

Port Spanning or Mirroring



Full Duplex Links



iTap GigaBit Copper
Dual Port Aggregator



10/100BaseT
Dual Port Aggregator Tap



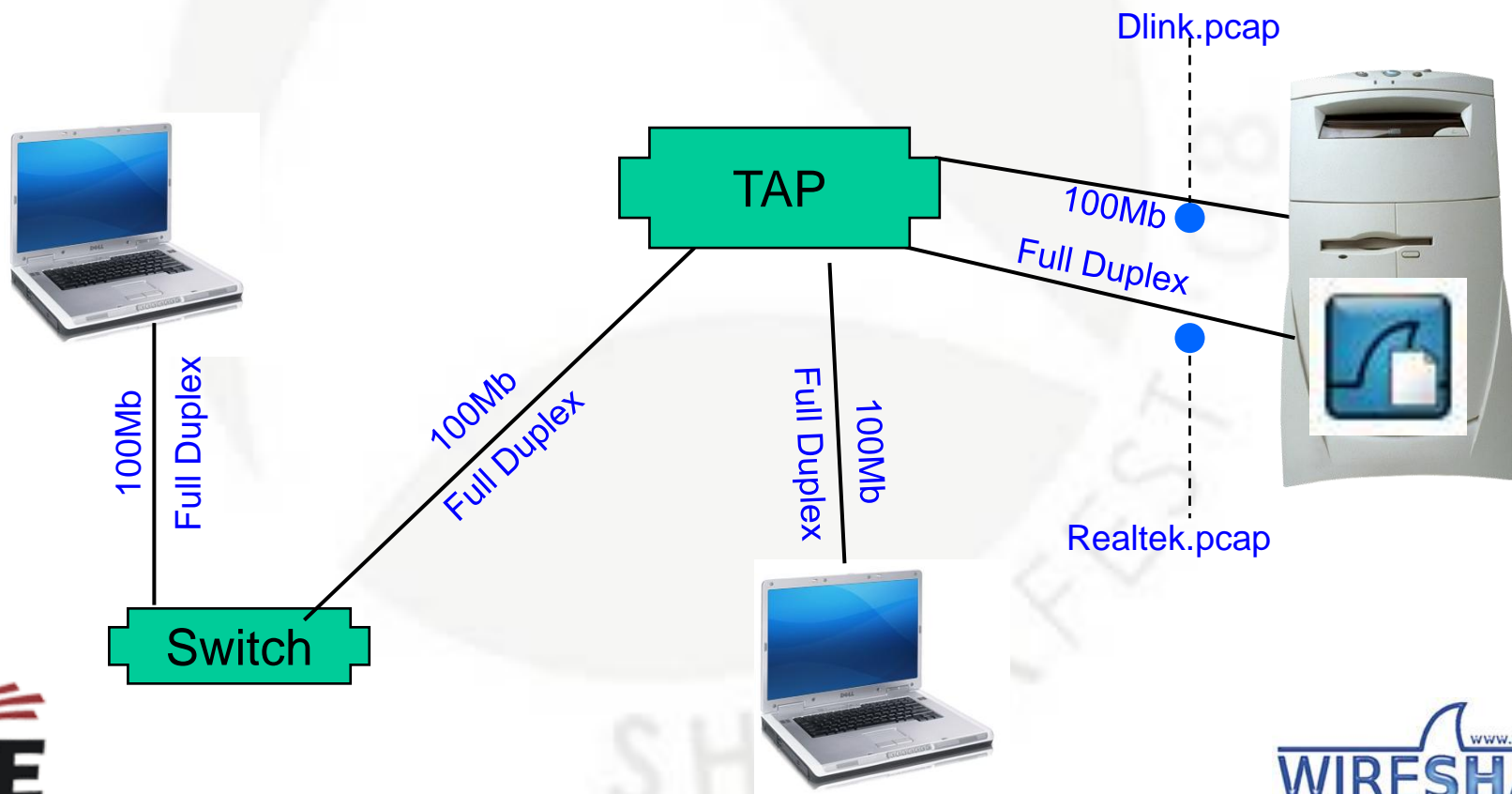
10/100BaseT
Port Aggregator Tap



- Aggregating taps will combine both Rx channels to one monitoring port
- Non-aggregating taps will require 2 NICS on the monitoring tool

Multiple trace files

In this example I captured a full duplex conversation using a full duplex, non-aggregating tap. The capture is a Windows XP PC and as a Dlink and Realtek NIC.



Assembling The Data

Now that we have 2 trace files, there are 3 ways to put them together;

- GUI
- Command Line
- Drag and Drop (Windows)

mergcap - Command Line (Windows)

Syntax

Usage: mergcap [options] -w <outfile>-> <infile> ...

Example

```
mergcap -v -w combined.pcap realtek.pcap dlink.pcap
```

Where

-v is verbose mode to get some feedback while files are being processed

-w is the output file 'combined'

Output

.....

```
C:\ Record: 1050
```

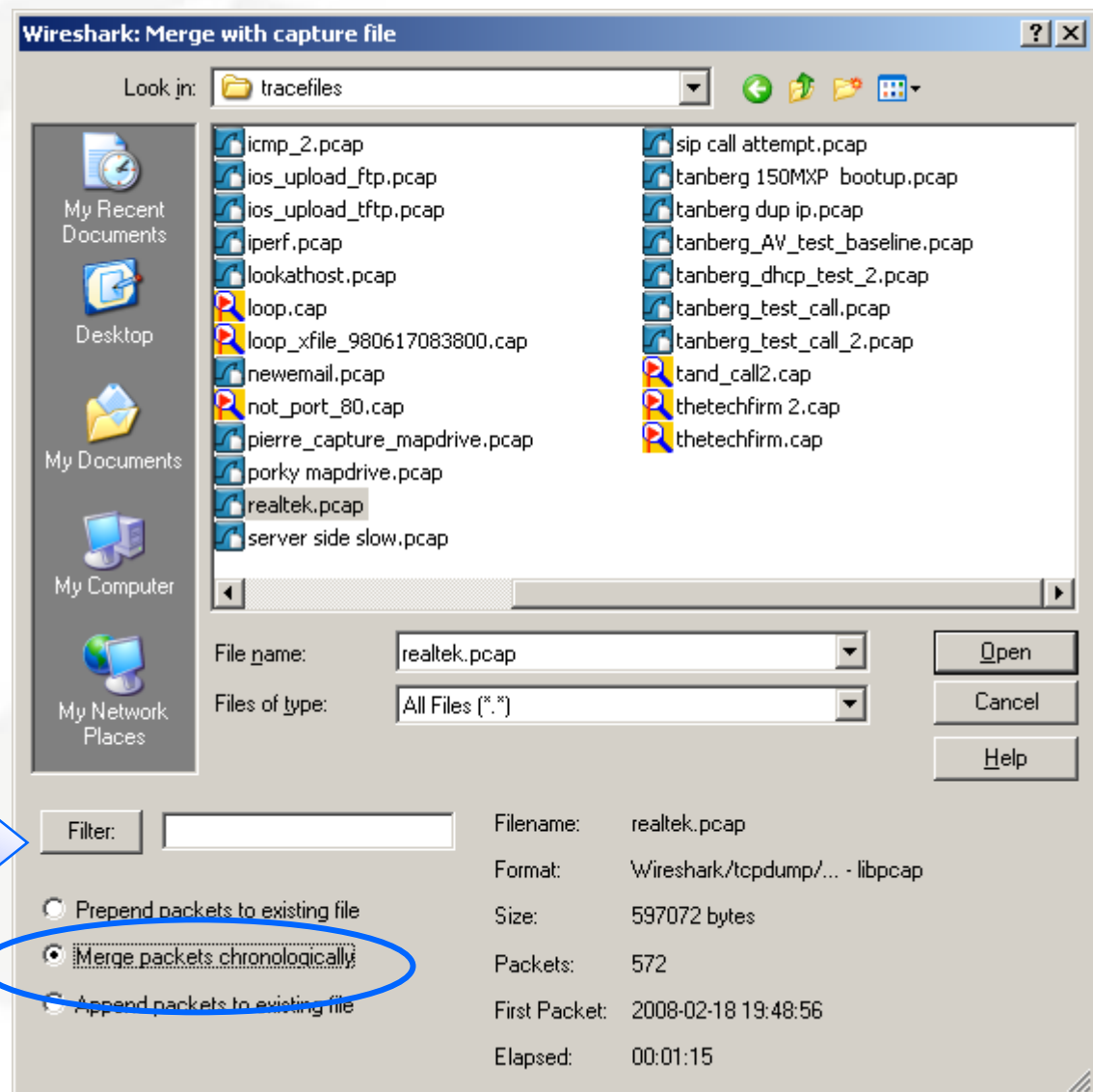
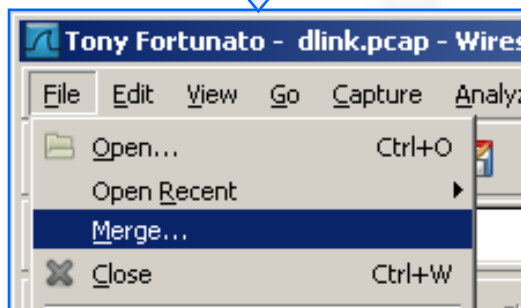
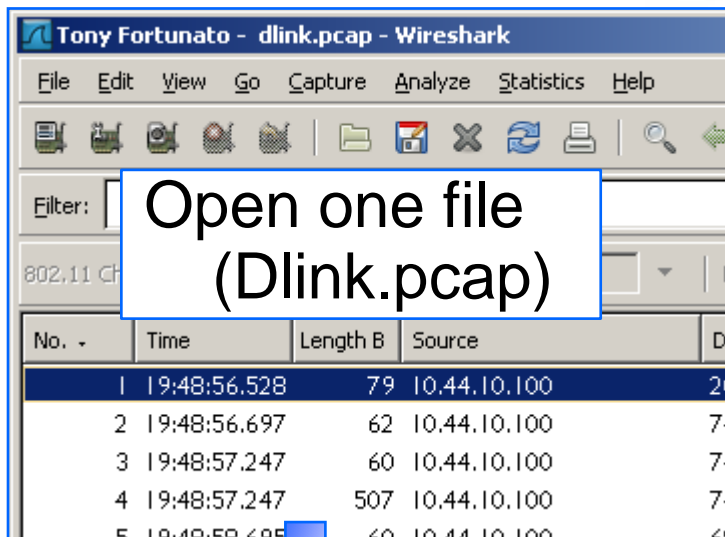
```
C:\ dir combined.pcap
```

```
Volume in drive C has no label.
```

```
Directory of C:\
```

```
03/30/2008 01:54 PM          676,680 combined.pcap
```

mergcap - GUI



Drag and Drop

Drag and drop both files into the Wireshark application

File Name	Size	Type	Date Modified
homecam.pcap	349 KB	Wireshark file	10/30/2007 1
iperf.pcap	2,457 KB	Wireshark file	10/30/2007 1
email.pcap	5 KB	Wireshark file	10/30/2007 8
	6 KB	Wireshark file	11/8/2007 12
	78 KB	Wireshark file	2/18/2008 8:5

Merging the following files:

- C:/Documents and Settings/Tony Fortunato/Desktop/dnd/tracefiles/iperf.pcap
- C:/Documents and Settings/Tony Fortunato/Desktop/dnd/tracefiles/homecam.pcap

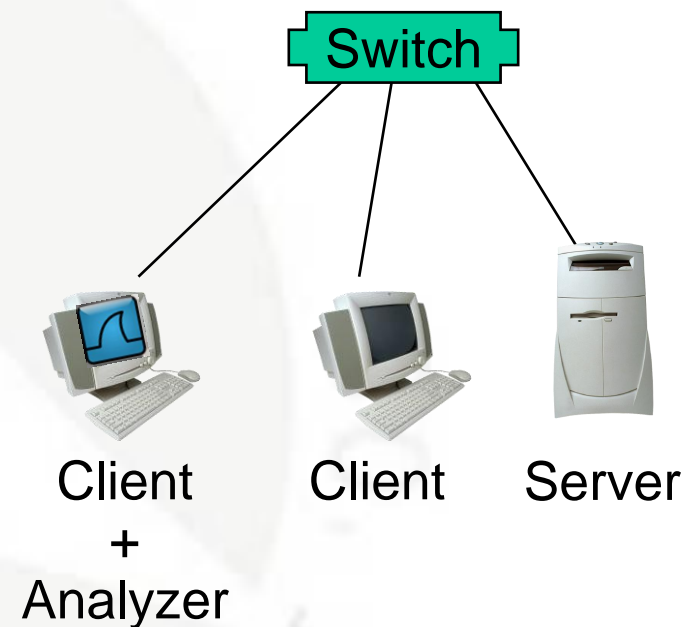
The packets in these files will be merged chronologically into a new temporary file.

OK

Software

Install Wireshark on the client or server experiencing the problem.

Installing any software on a production server is more difficult to accomplish, but sometimes a development server is available for testing and may exhibit the same symptoms you are trying to troubleshoot.



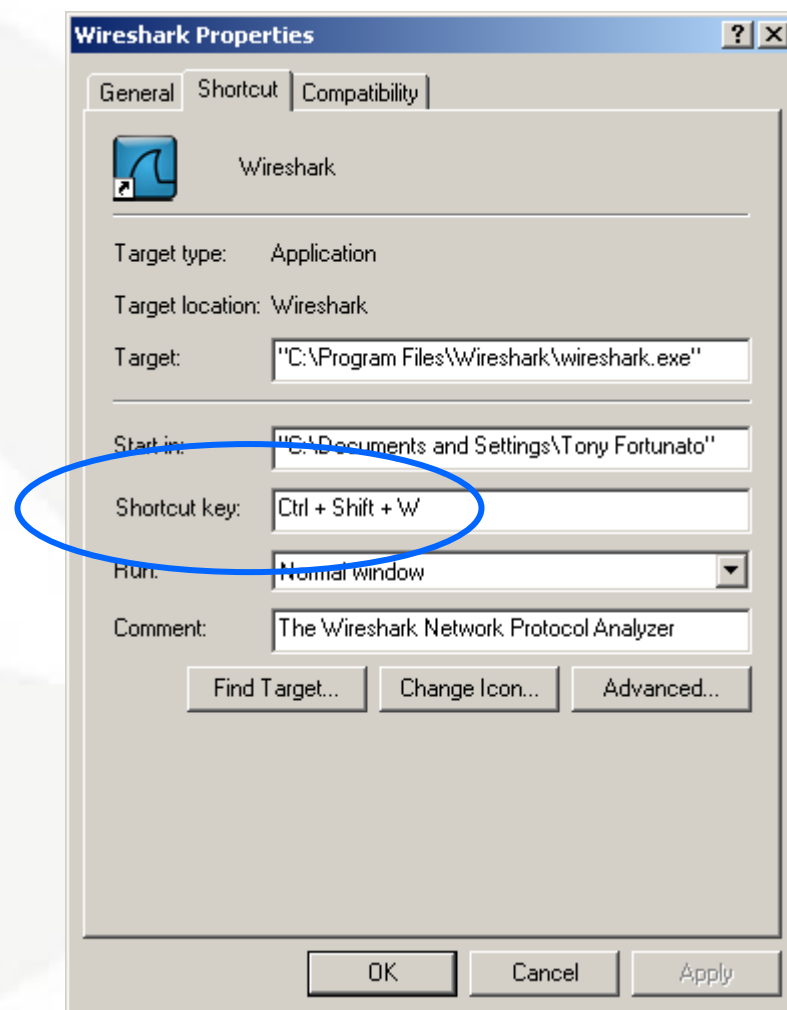
Make Wireshark More Convenient

(windows)

Assign a Shortcut key to Wireshark (Windows) for easier access.

This tip is applicable to all tools you use frequently

- Command prompt
- Notepad
- SNMP browser



Capturing From Your VPN Driver (windows)

If you select your VPN driver you can capture your decrypted data to help diagnose application or connectivity issues.

The screenshot displays the Wireshark interface with a capture of network traffic. The main window shows a list of captured packets with columns for No., Time, Length B, Source, Destination, Protocol, and Info. The 'Capture Options' dialog box is open, showing the selected interface and IP address.

No.	Time	Length B	Source	Destination	Protocol	Info
1	0.000000	35	Send_02	Send_02	PPP LCF	Configuration Request
2	0.837891	71	Receive_02	Receive_02	PPP LCF	Configuration Request
3	0.000000	49	Send_02	Send_02	PPP LCF	Configuration Reject
4	0.706054	40	Receive_02	Receive_02	PPP LCF	Configuration Request
5	0.000000	40	Send_02	Send_02	PPP LCF	Configuration Ack
6	0.000000	40	Send_02	Send_02	PPP LCF	Identification
7	0.000000	40	Send_02	Send_02	PPP LCF	Identification
8	0.000000	40	Send_02	Send_02	PPP CH	Challenge
9	0.000000	40	Send_02	Send_02	PPP CH	Response
10	0.000000	40	Send_02	Send_02	PPP CH	Success (MESSAGE='\$=2829C8E5D0E
11	0.000000	20	Receive_02	Receive_02	PPP CBC	Callback Request
12	0.000000	20	Send_02	Send_02	PPP CBC	Callback Response
13	0.688477	20	Receive_02	Receive_02	PPP CBC	Callback Ack
14	0.000000	24	Receive_02	Receive_02	PPP CCF	Configuration Request
15	0.000000	24	Send_02	Send_02	PPP CCF	Configuration Request

File: "C:\DOCUME~1\TONYFO~1\LOCAL5~1\Temp\etherXXXa038..." Packets: 47 Displayed: 47 Marked: 0 Dropped: 0

Capturing From Your Wireless Card

If you select your Wireless Card, you can capture only the data packets.

If you need to capture wireless management packets such as Beacon, Acknowledgement and Probe packets you need to use a product like AirPcap.

Frame 5 (46 bytes on wire, 46 bytes captured)					
Radiotap Header v0, Length 32					
Header revision: 0	1	0.000	46	Cisco-Li_db:06:7d (RA)	IEEE 802.11 Clear-to-send, Flags=.....C
Header pad: 0	3	0.000	46	Cisco-Li_db:06:7d (RA)	IEEE 802.11 Acknowledgement, Flags=.....C
Header length: 32	5	0.000	46	Cisco-Li_11:ea:16 (RA)	IEEE 802.11 Acknowledgement, Flags=.....C
Present flags: 0x000058ef	7	0.000	46	AskeyCom_01:8d:96 (RA)	IEEE 802.11 Acknowledgement, Flags=.....C
MAC timestamp: 184770912	9	0.000	46	Cisco-Li_11:ea:16 (RA)	IEEE 802.11 Acknowledgement, Flags=.....C
Flags: 0x10	10	0.000	46	Cisco-Li_db:06:7d (RA)	IEEE 802.11 Clear-to-send, Flags=.....C
Data Rate: 24.0 Mb/s	12	0.000	46	Cisco-Li_db:06:7d (RA)	IEEE 802.11 Acknowledgement, Flags=.....C
Channel frequency: 2437 [BG]	14	0.000	46	Cisco-Li_11:ea:16 (RA)	IEEE 802.11 Acknowledgement, Flags=.....C
Channel type: 802.11g (pure-)	15	0.004	46	Cisco-Li_db:06:7d (RA)	IEEE 802.11 Clear-to-send, Flags=.....C
SSI Signal: -65 dBm	17	0.000	46	Cisco-Li_db:06:7d (RA)	IEEE 802.11 Acknowledgement, Flags=.....C
SSI Noise: -91 dBm	19	0.000	46	Cisco-Li_11:ea:16 (RA)	IEEE 802.11 Acknowledgement, Flags=.....C
Signal Quality: 88	21	0.000	46	AskeyCom_01:8d:96 (RA)	IEEE 802.11 Acknowledgement, Flags=.....C
Antenna: 0	23	0.000	46	02:11	Acknowledgement, Flags=.....C
SSI Signal: 26 dB	24	0.000	46	02:11	Clear-to-send, Flags=.....C
802.11 FCS: 0x3b1fcc64 [correct]					

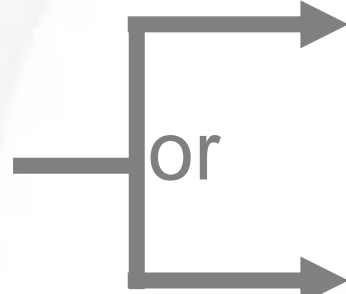
Captured via AirPcap driver and interface

Saving To A File

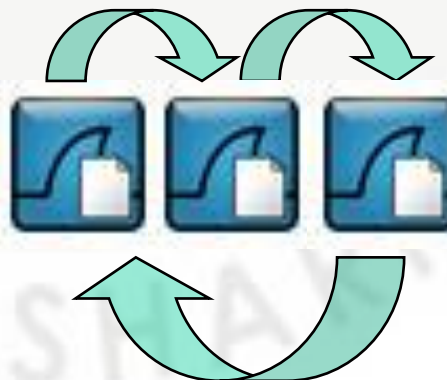
Single File



File Set



Ring Buffer



Capture File Configurations

File Name

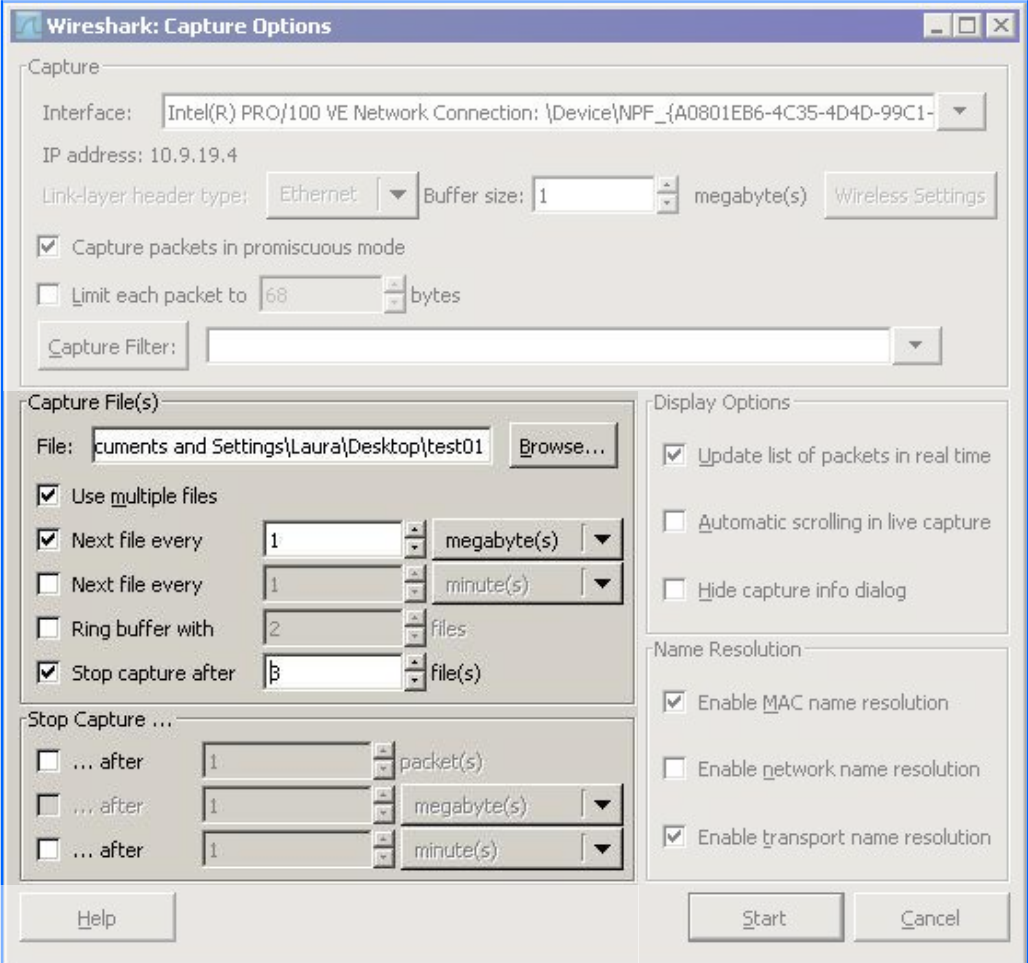
Multiple Files

Stop based on

- Data captured
- Packets captured
- Time

Ring buffer number

File count



The image shows the 'Wireshark: Capture Options' dialog box. It is divided into several sections:

- Capture:** Interface: Intel(R) PRO/100 VE Network Connection: {Device}\NPF_{A0801EB6-4C35-4D4D-99C1-...}; IP address: 10.9.19.4; Link-layer header type: Ethernet; Buffer size: 1 megabyte(s); Capture packets in promiscuous mode; Limit each packet to 68 bytes; Capture Filter: (empty).
- Capture File(s):** File: cuments and Settings\Laura\Desktop\test01; Use multiple files; Next file every 1 megabyte(s); Next file every 1 minute(s); Ring buffer with 2 files; Stop capture after 3 file(s).
- Stop Capture ...:** ... after 1 packet(s); ... after 1 megabyte(s); ... after 1 minute(s).
- 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.

Buttons: Help, Start, Cancel.

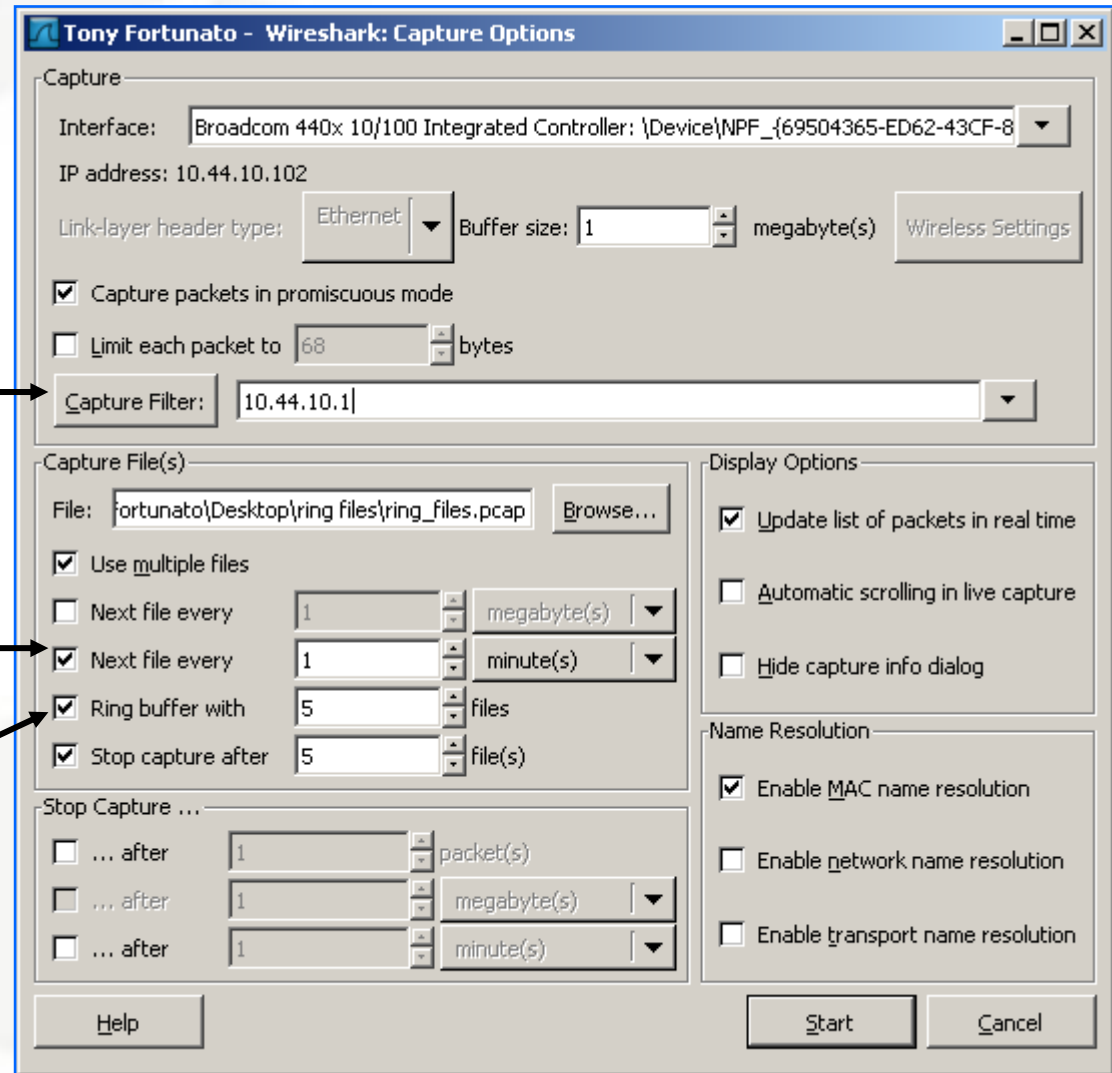
Capture Example Using The GUI

The following will have
the last 24 hours of
packets

To or from 10.44.10.1

Every minute

Ringing 5 files,
showing the last 5
minutes worth of
data



Capture Example From The Command Line

(windows)

Capturing from the command line allows us to start captures more quickly and consistently.

The same syntax can be used in a startup, batch file or desktop shortcut, so anyone can start a capture with very little Wireshark experience.

To capture from the command line, we will use *tshark*

The only thing to determine is the interface number of the adapter you want to use. Simply go to the Wireshark program directory and type *tshark -D*. This number will be used later

The command to capture using the same parameters as the previous slide is;

```
tshark -i number -f "host 10.0.12.15" -w hourly.pcap -b duration:3600 -b files:24
```

What To Filter On?

I always suggest that when you are baselining or troubleshooting a device or application for the first time, try to filter on the lowest address

- MAC ADDRESS; Through Layer 2 Switches
- IP ADDRESS; Through Routers
- TCP or UDP address or Data; Through Firewalls

Accelerators (Keyboard Shortcuts)

TAB

- Move from Packet List, Detail and Bytes

Ctrl+Down Arrow

- Move to next packet in (even if packet list is not in focus)

The screenshot shows the Wireshark interface with the following data in the packet list pane:

No.	Time	Source	Destination	Protocol	Info
1	0.0	Sony_f4:3a:09	Dell_be:9d:fd	ARP	10.1.0.99 is at 08:00
2	0.0	10.1.0.1	10.1.0.99	UDP	Source port: 32799 D
3	0.0	10.1.0.99	10.1.0.1	ICMP	Destination unreachab
4	0.0	10.1.0.1	10.1.0.99	TCP	2664 > netbios-ssn [S

The detail pane shows the following information for the selected packet:

- Frame 1 (42 bytes on wire, 42 bytes captured)
- Ethernet II, Src: Sony_f4:3a:09 (08:00:46:f4:3a:09), Dst: Dell_be:9d:fd (00:14:22:be:9d:fd)
- Address Resolution Protocol (reply)

The bytes pane shows the following hex data:

```
0000 00 14 22 be 9d fd 08 00 46 f4 3a 09 08 06 00 01  .."..... F.:.....
0010 08 00 06 04 00 02 08 00 46 f4 3a 09 0a 01 00 63  ..... F.:.....c
0020 00 14 22 be 9d fd 0a 01 00 01
```

Optimizing Wireshark

Capture

- Update List of Packets in Real Time
- Capture Dialog Window
- Name Resolution
- Buffer Size (Windows)
- Protocol Tasks
- Command-Line Capture

Display

- Number of Columns
- Split the trace file

P: 342343 D: 342343 M: 0 Drops: 9348

Name Resolution Preferences

MAC name resolution

Network name resolution

- Concurrent DNS name resolution
- Maximum concurrent requests

Transport name resolution

Name Resolution

Enable MAC name resolution:

Enable network name resolution:

Enable transport name resolution:

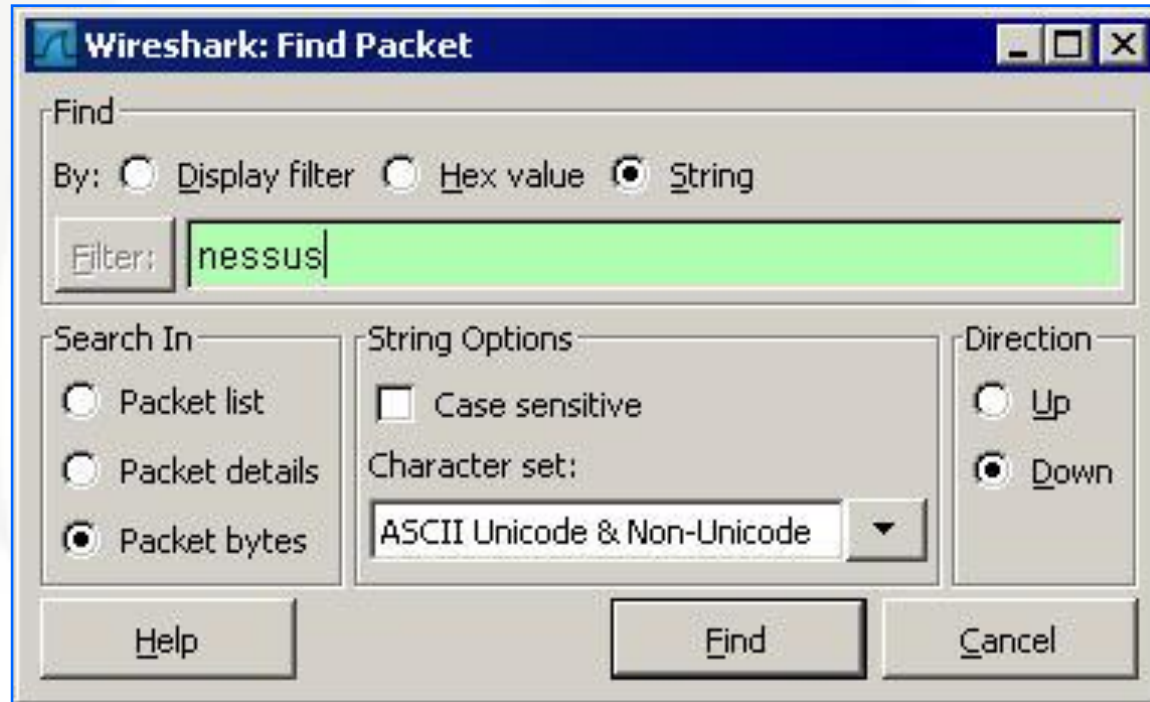
Enable concurrent DNS name resolution:

Maximum concurrent requests:

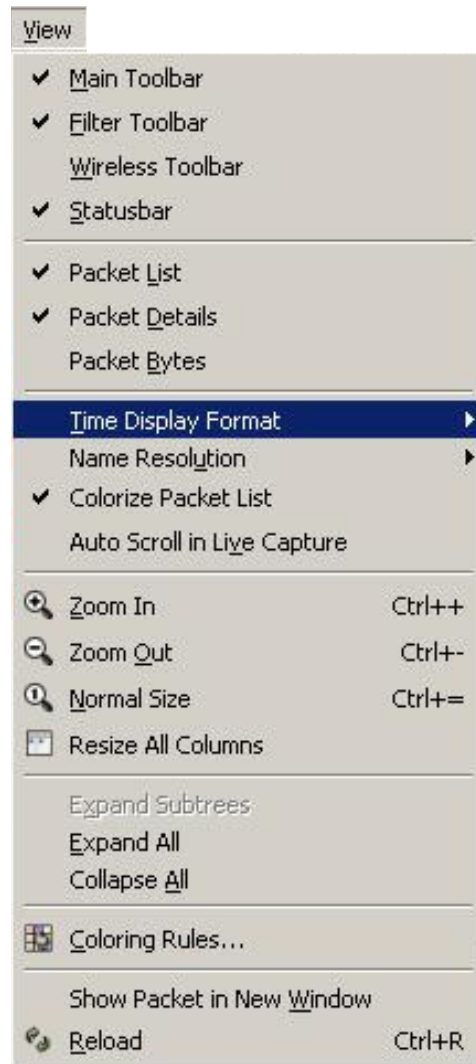
Find a Packet

Find packets based on

- Display filter
- Hex Value
- String



Configuring Your Time Settings

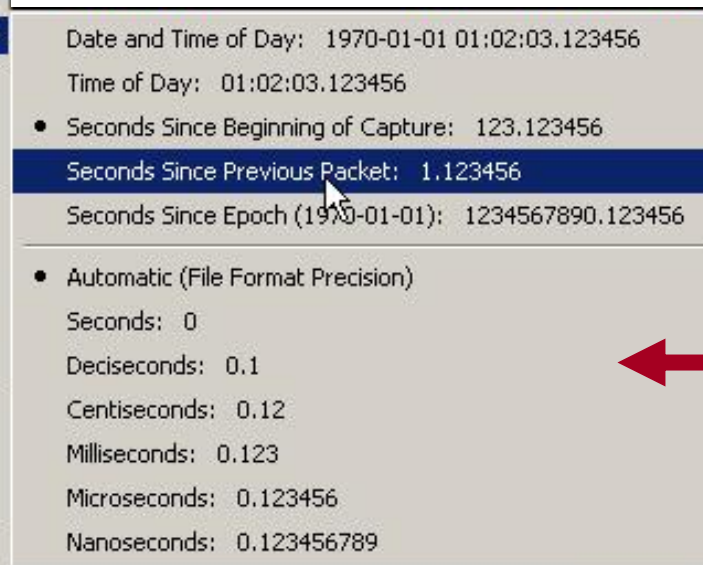


Date/Time of Day or just Time of Day

Secs. Since Beginning of Capture

Secs. Since Previous Packets

Secs. Since Epoch



Time Precision

- Automatic
- Seconds
- Deciseconds
- Centiseconds
- Milliseconds
- Microseconds
- Nanoseconds

Use the spacebar to select multiple entries

Using the Time Reference

Used to determine the time between specific packets.

Need to have time set to “Seconds Since Beginning of Capture”. If you do not, Wireshark will ask you if its ok for Wireshark to change it to that format.

Can create multiple reference points

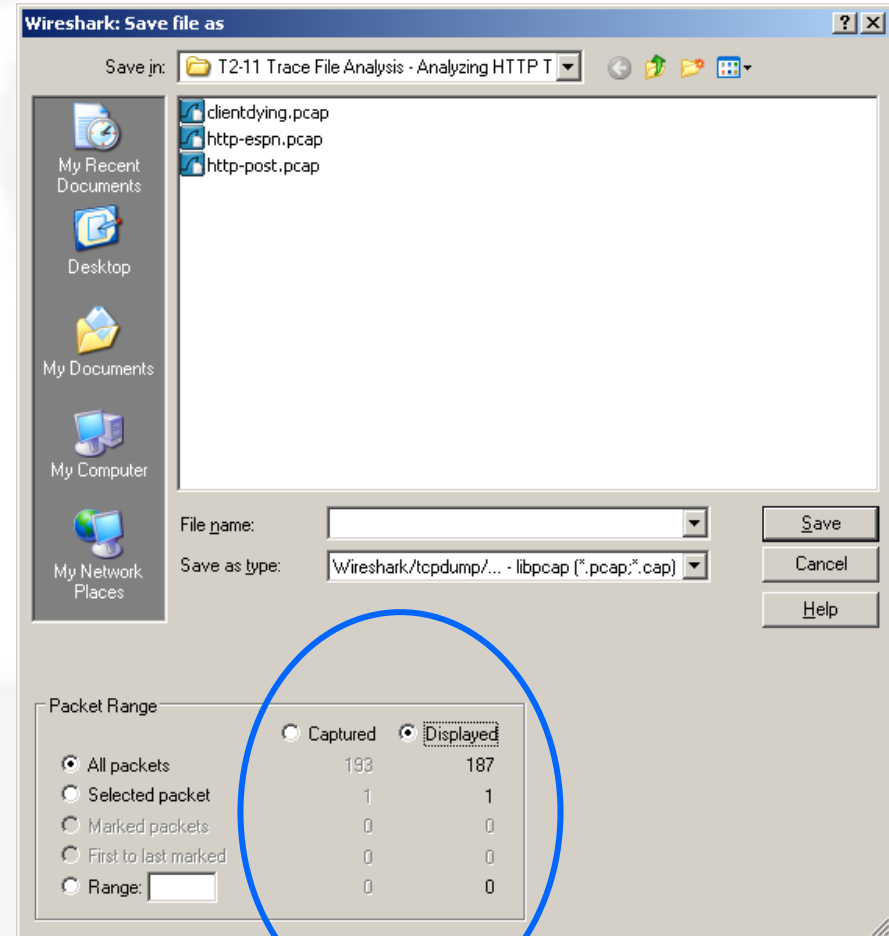
The screenshot shows the Wireshark interface with a dialog box titled "Switch to the appropriate Time Display Format?". The dialog contains a warning icon and the text: "Time References don't work well with the currently selected Time Display Format. Do you want to switch to 'Seconds Since Beginning of Capture' now?". There are "Yes" and "No" buttons. A blue arrow points from the "Set Time Reference (toggle)" menu item to the dialog. Another blue arrow points from the dialog to the packet list table.

No.	Time	Length B	Source	Destination	Protocol	Info
25	40.237	62	216.49.88.118	216.49.88.118	TCP	1033 > 80 [RST] Seq=...
26	40.329	60	216.49.88.118	24.6.125.19	TCP	80 > 1033 [SYN, ACK] Seq=...
27	40.329	60	24.6.125.19	216.49.88.118	TCP	1033 > 80 [ACK] Seq=...
28	*REF*	243	24.6.125.19	216.49.88.118	HTTP	GET /apps/Agent/en...
29	0.100	60	216.49.88.118	24.6.125.19	TCP	80 > 1033 [ACK] Seq=...
30	0.106	306	216.49.88.118	24.6.125.19	HTTP	HTTP/1.1 200 OK (...

Taking Frames Out Of Your Trace

After 'cleaning' up your trace with extensive filtering you may want to save this 'cleaner' version

Simply ensure you have selected the 'Displayed' option at the bottom of the Save Dialogue box



Difference Between Prepare and Apply

When you right click on a report or field name, you typically have the option to 'Apply' or 'Prepare' a filter.

Apply

- Takes whatever you have selected and immediately 'applies' it

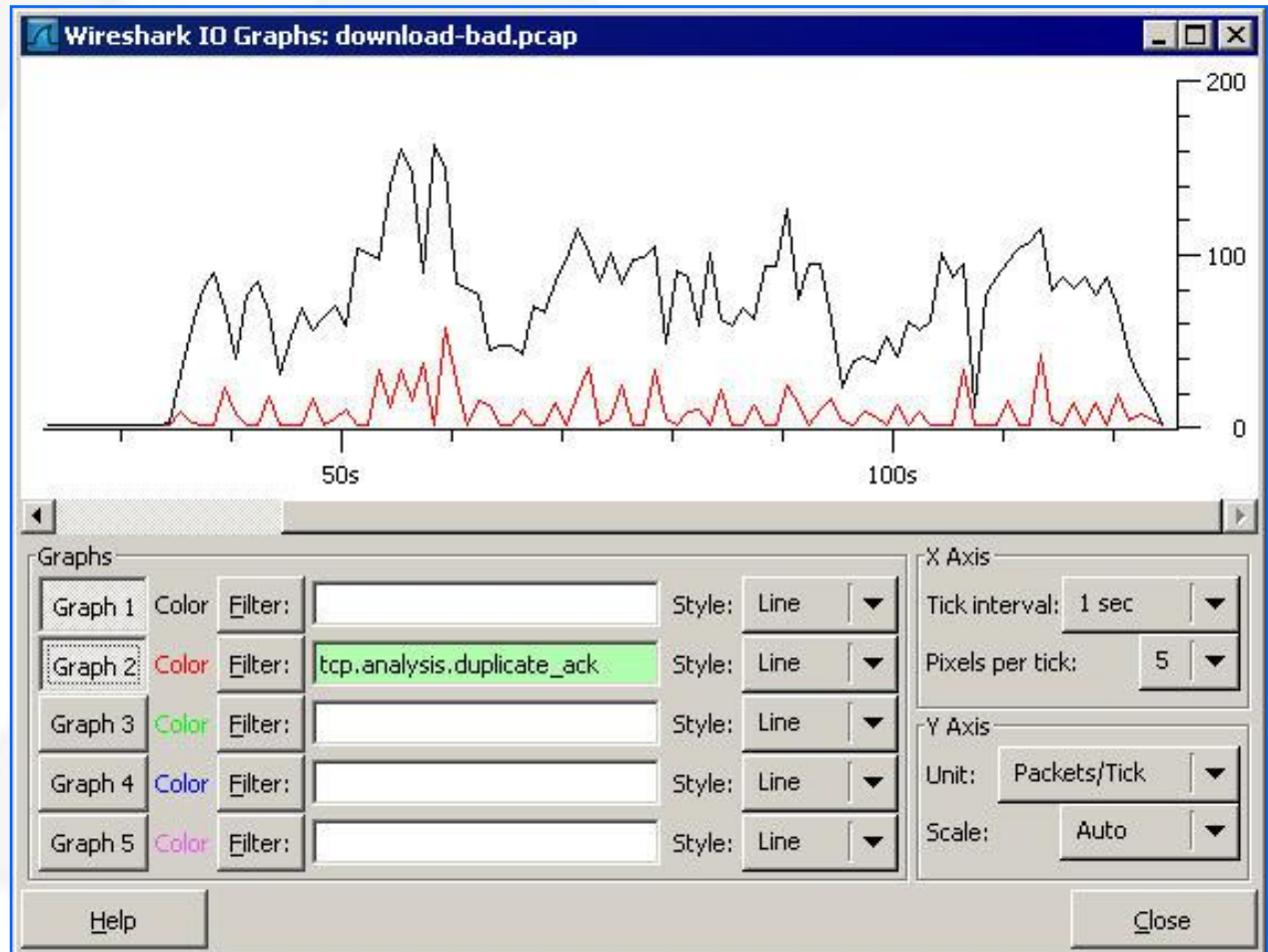
Prepare

- Takes whatever you have selected and simply inputs it into the display filter area
- Now you can modify it prior to invoking the display criteria

IO Graphs

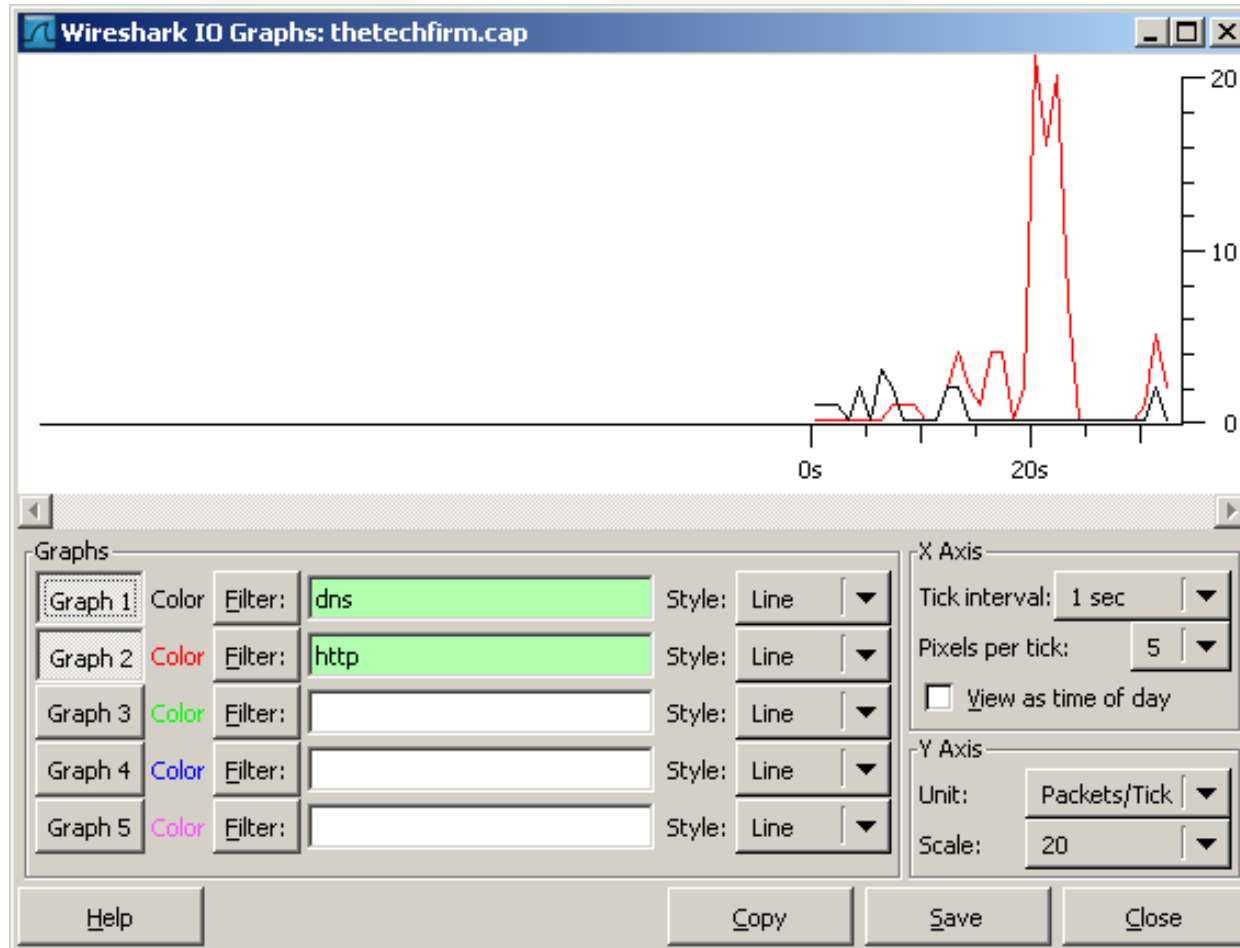
IO Graphs provide a visual representation of the traffic rate.

Consider 'color assumptions' when assigning filters.

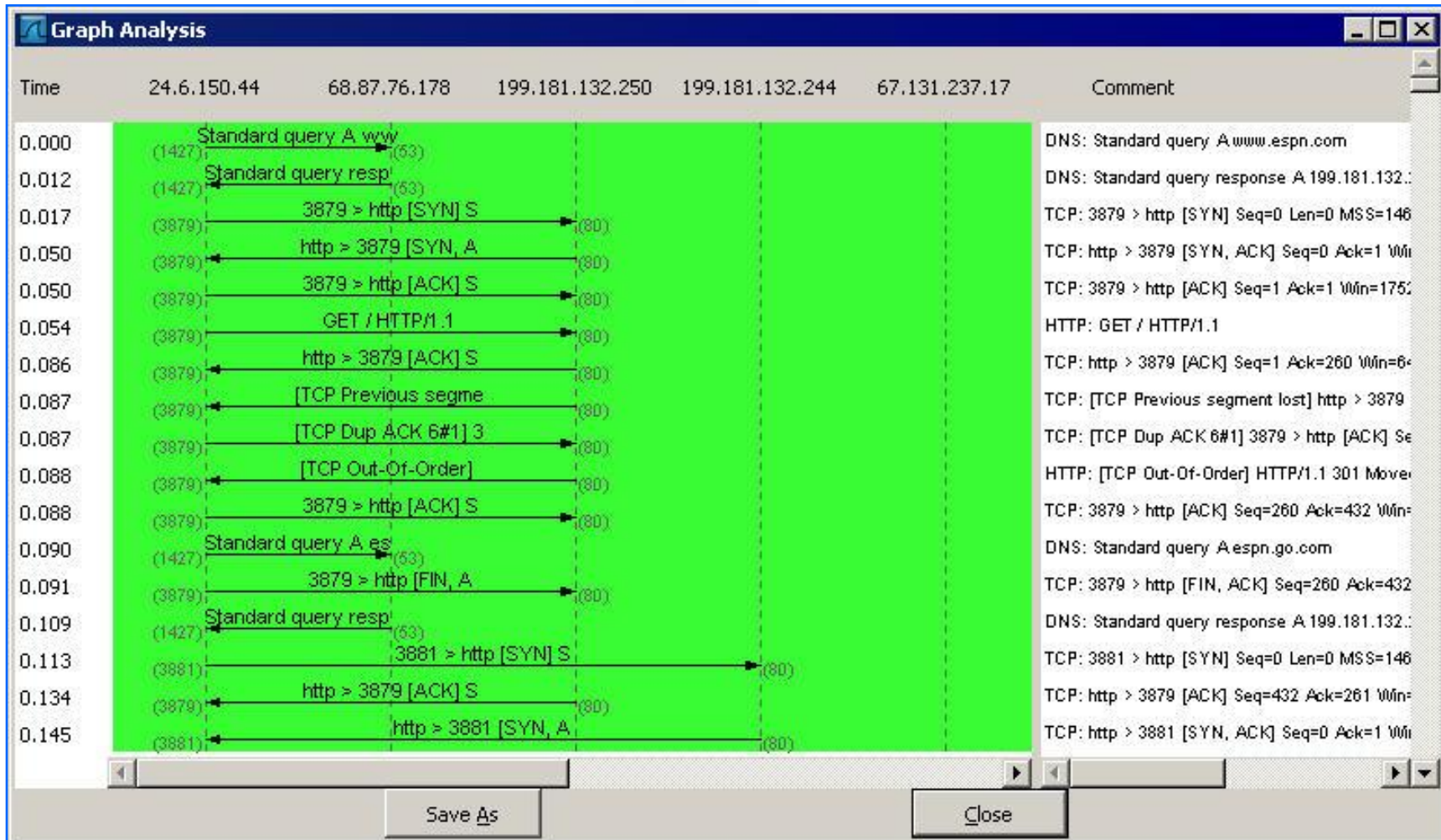


IO Graphs – Application Documentation

In the graph below I wanted to illustrate when DNS and HTTP was active.



HTTP Flow Graphing



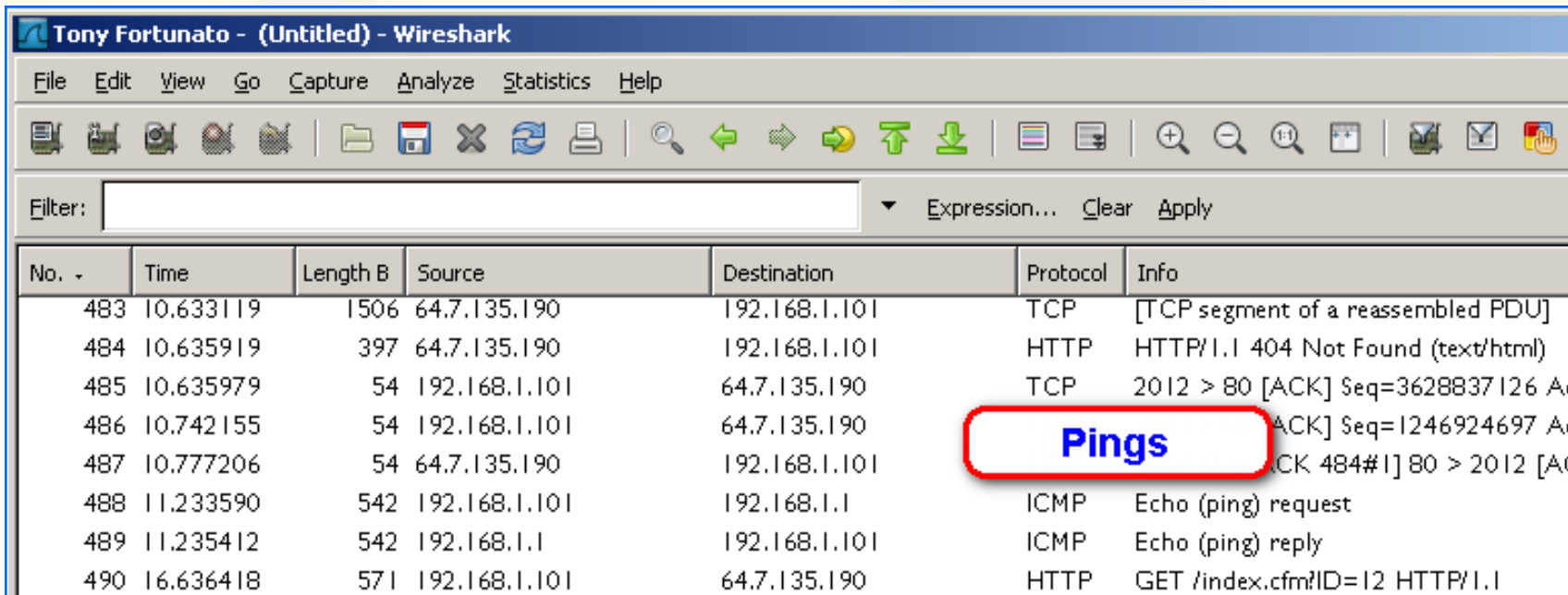
Creating a Packet Bookmark

Sometimes when you are capturing packets, you may need to create some kind of 'bookmark'.

To do so, simply ping something that has nothing to do with your troubleshooting.

For example ping your router with a 500 Byte payload.

The syntax for windows; ***ping ipaddress -l 500***



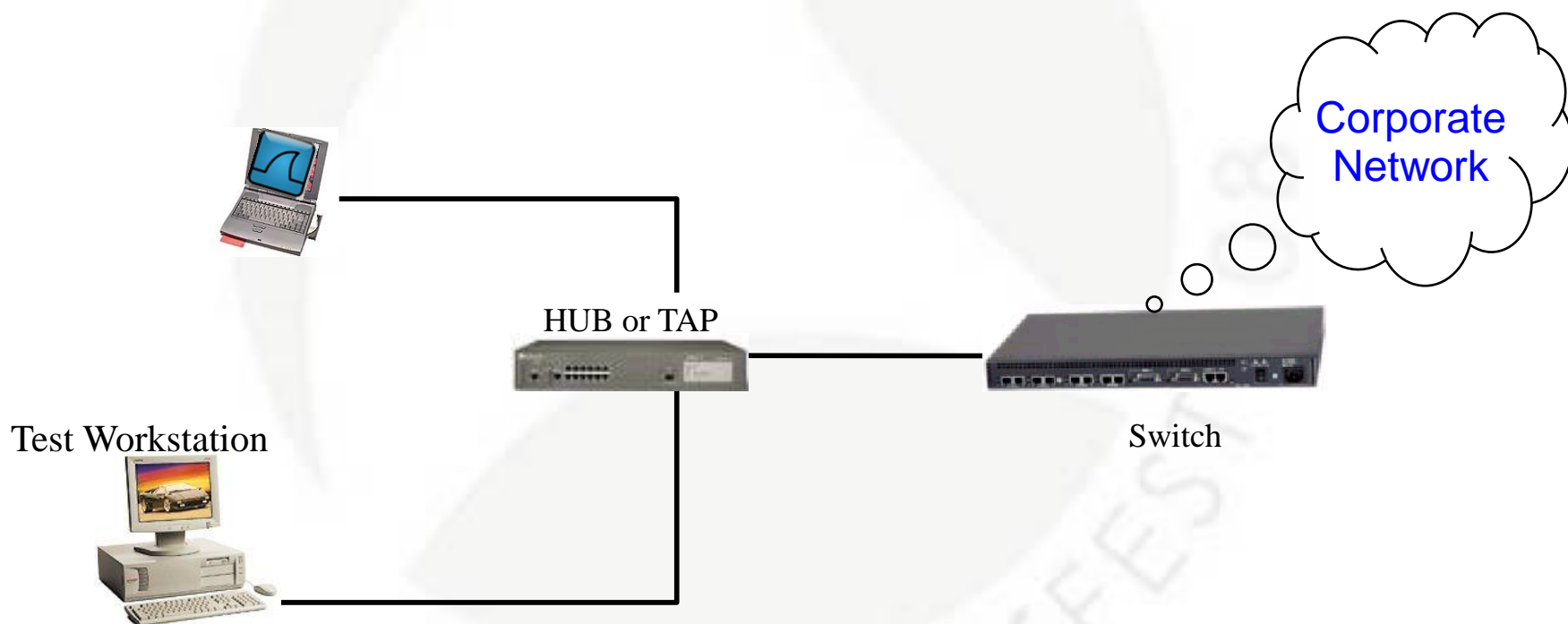
The screenshot shows the Wireshark interface with a packet capture. A red box highlights the word "Pings" in the packet list pane, which is positioned over the ICMP Echo (ping) request packet (No. 488). The packet list pane shows the following data:

No. -	Time	Length B	Source	Destination	Protocol	Info
483	10.633119	1506	64.7.135.190	192.168.1.101	TCP	[TCP segment of a reassembled PDU]
484	10.635919	397	64.7.135.190	192.168.1.101	HTTP	HTTP/1.1 404 Not Found (text/html)
485	10.635979	54	192.168.1.101	64.7.135.190	TCP	2012 > 80 [ACK] Seq=3628837126 A
486	10.742155	54	192.168.1.101	64.7.135.190	TCP	[ACK] Seq=1246924697 A
487	10.777206	54	64.7.135.190	192.168.1.101	TCP	[ACK 484#1] 80 > 2012 [A
488	11.233590	542	192.168.1.101	192.168.1.1	ICMP	Echo (ping) request
489	11.235412	542	192.168.1.1	192.168.1.101	ICMP	Echo (ping) reply
490	16.636418	571	192.168.1.101	64.7.135.190	HTTP	GET /index.cfm?ID=12 HTTP/1.1

Boot-up Baseline

This baseline observes a device's boot-up process and provides clues as to the configuration of that device.

Most common example is to baseline your new PC build.



Boot-up Example Findings

Servers	Protocol	Bytes/Packets
10.10.10.11	DNS	
10.10.10.2	DHCP	
10.10.10.1	Default Gateway	
10.10.22.10	PDC	
18.12.14.14	Time server	
10.10.10.3	LDAP/Kerberos	

Application Baseline Example

Task	Start Frame #	End Frame #	Bytes
Launch acme Data Entry Application	0	1,000	100,829
Login	1,002	3,121	6,232,232
Query for account 123	3,231	5,764	13,123,385
Change name and submit	6,000	6456	213,489

Roll your Own

With a PC and 2 network interface cards, you can easily design a remote capture tool.

Using Remote Desktop or VNC, you can connect from your PC to your Wireshark and capture from the other interface



Pathping (windows)

Use Microsoft's pathping command to document packet loss and response time

```
C:\ Documents and Settings\Tony Fortunato>pathping 10.0.29.1 -n

Tracing route to 10.0.29.1 over a maximum of 30 hops

  0  10.44.10.102
  1  10.44.10.1
  2  172.17.4.1
  3  10.0.29.1

Computing statistics for 75 seconds...

Hop  RTT      Source to Here   This Node/Link   Address
  0                               Lost/Sent = Pct  Lost/Sent = Pct
  0                               0/ 100 = 0%     0/ 100 = 0%     10.44.10.102
  1    2ms      0/ 100 = 0%     0/ 100 = 0%     |               10.44.10.1
  2  102ms    2/ 100 = 2%     2/ 100 = 2%     |               172.17.4.1
  3  100ms    4/ 100 = 4%     0/ 100 = 0%     |               10.0.29.1

Trace complete.
```

Netstat results (windows)

Use Netstat to see what TCP/UDP ports and IP addresses your application is using

```
C:\>netstat -n -b
```

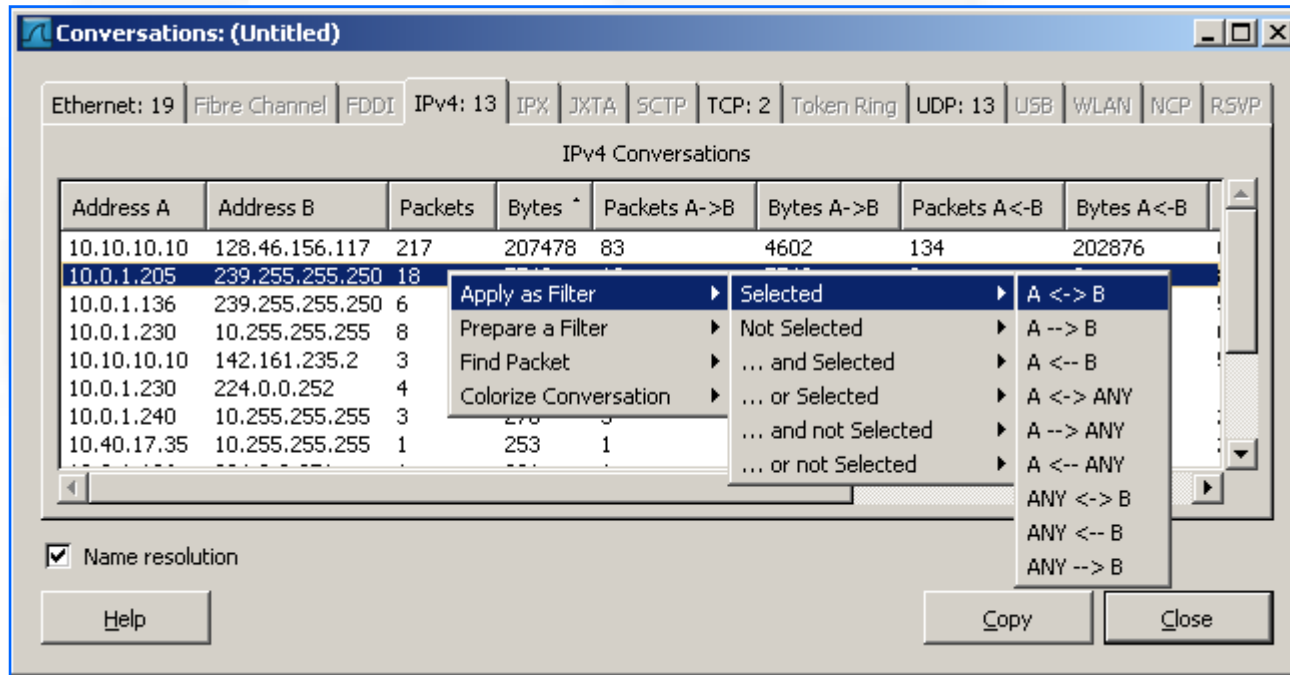
Active Connections

Proto	Local Address	Foreign Address	State	PID
TCP	10.10.10.10:2716	142.161.235.2:17879	ESTABLISHED	2724

[Skype.exe]

Specific Display Filters

Once you have captured your packets, use Statistics->Conversation->Display Filters to understand your application behavior.



The screenshot shows the 'Conversations: (Untitled)' window in Wireshark. The 'IPv4: 13' tab is active, displaying a table of IPv4 conversations. The selected conversation is between 10.0.1.205 and 239.255.255.250. A context menu is open over this row, with 'Apply as Filter' selected, and a sub-menu is open showing 'A <-> B' as the chosen filter.

Address A	Address B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B
10.10.10.10	128.46.156.117	217	207478	83	4602	134	202876
10.0.1.205	239.255.255.250	18	278	0	0	18	278
10.0.1.136	239.255.255.250	6	120	0	0	6	120
10.0.1.230	10.255.255.255	8	160	0	0	8	160
10.10.10.10	142.161.235.2	3	60	0	0	3	60
10.0.1.230	224.0.0.252	4	80	0	0	4	80
10.0.1.240	10.255.255.255	3	60	0	0	3	60
10.40.17.35	10.255.255.255	1	253	1	253	0	0