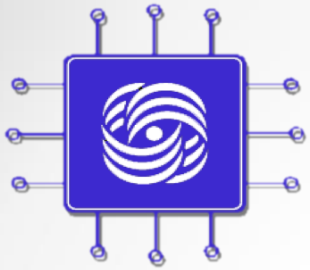


APPLIED
RESEARCH
CENTER FOR
COMPUTER
NETWORKS

Системы моделирования компьютерных сетей #2

Доп. главы Компьютерных сетей и телекоммуникации

к. ф.-м. н. Антоненко В.А.

Вопросы

1. Что такое имитационное моделирование?
2. Что такое аналитическое моделирование?
3. Что такое агентное моделирование?
4. Что такое системная динамика?
5. Что такое дискретно событийная модель?

План лекции

1. Основы контейнерной виртуализации – Docker
2. Tcpdump
3. NPS
4. Задание №1
5. Wireshark

Introduction to Docker

DOCKER HISTORY ……

- A dotCloud (PAAS provider) project
- Initial commit January 18, 2013
- Docker 0.1.0 released March 25, 2013
- 18,600+ github stars, 3800+ forks,740 Contributors…. and continues
- dotCloud pivots to docker inc. October 29, 2013

What is Docker ?!!!

- Open platform for developers and sysadmins to build, ship and run distributed applications
- Can run on popular 64-bit Linux distributions with kernel 3.8 or later
- Supported by several cloud platforms including Amazon EC2, Google Compute Engine, and Rackspace.

Features….

- Light-Weight
  - Minimal overhead (*cpu/io/network*)
  - Based on Linux containers
  - Uses layered filesystem to save space (AUFS/LVM)
  - Uses a copy-on-write filesystem to track changes
- Portable
  - Can run on any Linux system that supports LXC (today).
  - 0.7 release includes support for RedHat/Fedora family.
  - Raspberry pi support.
  - Future plans to support other container tools (lmctfy, etc.)
  - Possible future support for other operating systems (Solaris, OSX, Windows?)
- Self-sufficient
  - A Docker container contains everything it needs to run
  - Minimal Base OS
  - Libraries and frameworks
  - Application code
  - A docker container should be able to run anywhere that Docker can run.

The Challenge……

Multiplicity of Stacks

Static website

nginx 1.5 + modsecurity + openssl + bootstrap 2

User DB

postgresql + pgv8 + v8

Queue

Redis + redis-sentinel

Analytics DB

hadoop + hive + thrift + OpenJDK

Background workers

Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

Web frontend

Ruby + Rails + sass + Unicorn

API endpoint

Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

Do services and apps interact appropriately?

Multiplicity of hardware environments

Development VM

QA server

Customer Data Center

Public Cloud

Disaster recovery

Production Servers

Production Cluster

Contributor's laptop

Can I migrate smoothly and quickly?

Cargo Transport Pre-1960……

Multiplicity of Goods

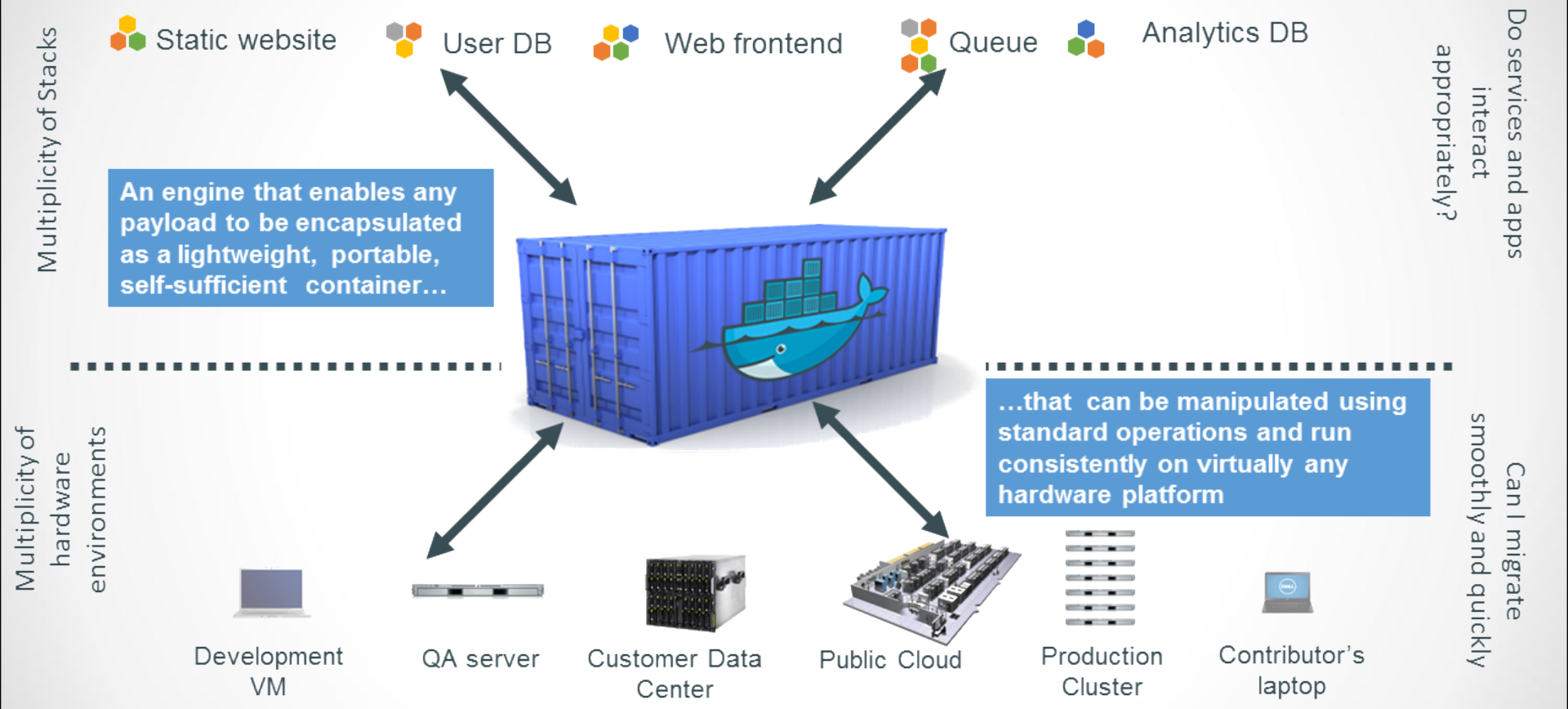
Do I worry about how goods interact (e.g. coffee beans next to spices)

Multipilicity of methods for transporting/storing

Can I transport quickly and smoothly (e.g. from boat to train to truck)

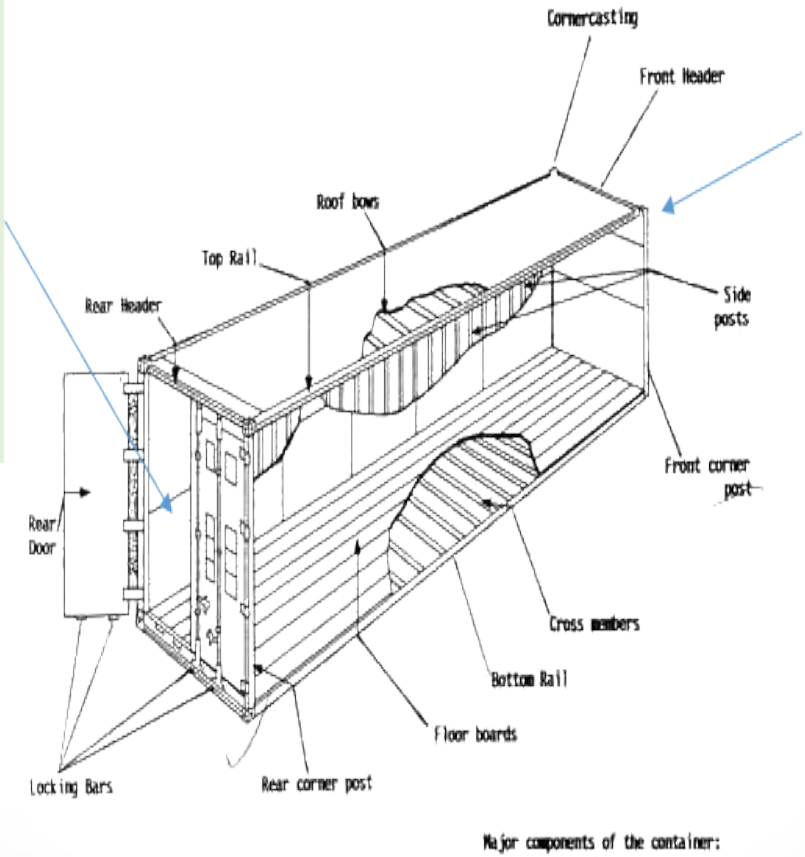
Solution: Intermodal Shipping Container……

Docker is a Container System for Code……



Why it Works: Separation of Concerns……

- Dan the Developer
  - Worries about what's "inside" the container
    - His code
    - His Libraries
    - His Package Manager
    - His Apps
    - His Data
  - All Linux servers look the same

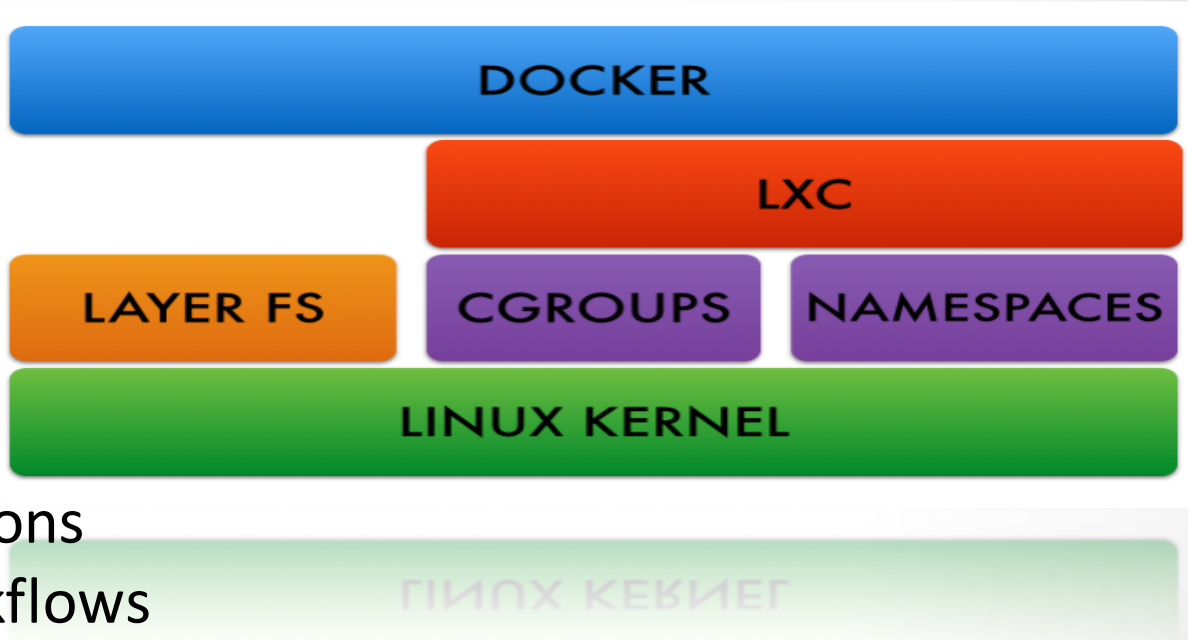


Major components of the container:

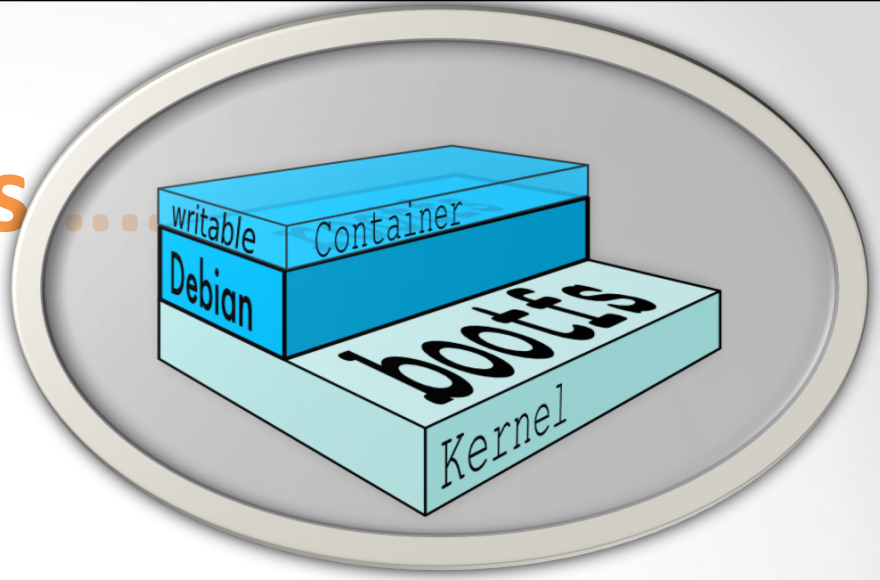
- Oscar the Ops Guy
  - Worries about what's "outside" the container
    - Logging
    - Remote access
    - Monitoring
    - Network config
  - All containers start, stop, copy, attach, migrate, etc. the same way

Docker Architecture……

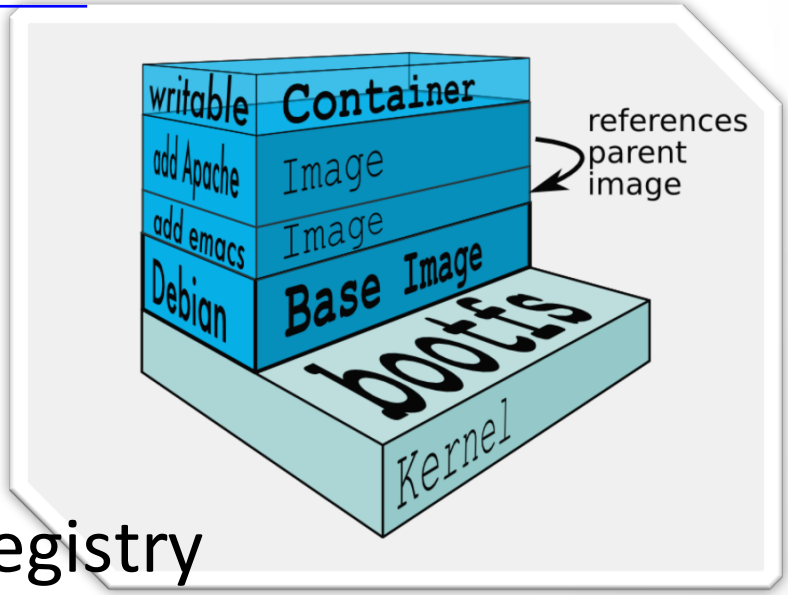
- Docker Engine
  - CLI
  - Docker Daemon
  - Docker Registry
- Docker Hub
  - Cloud service
    - Share Applications
    - Automate workflows
    - Assemble apps from components
- Docker images
- Docker containers



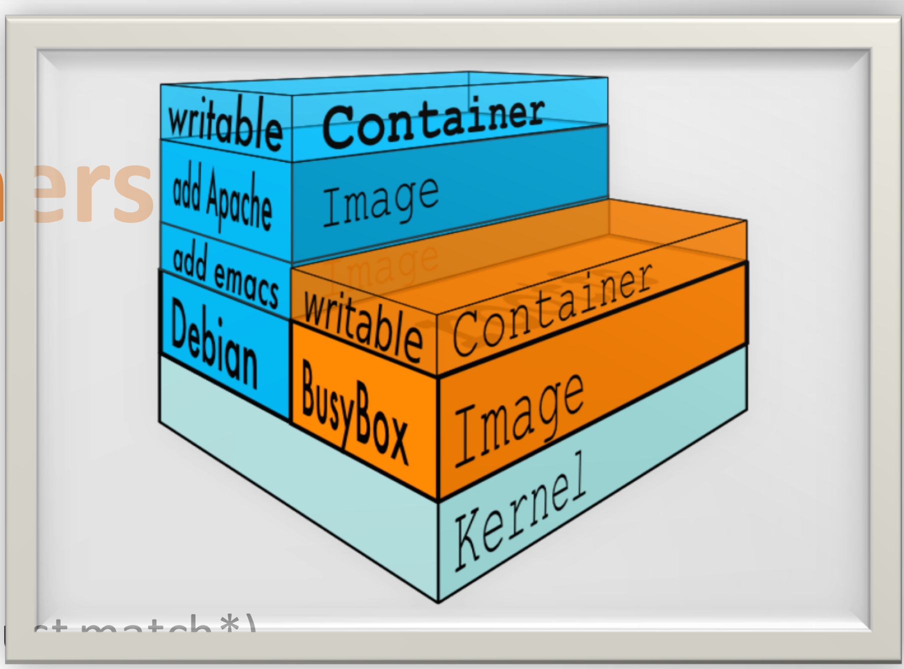
Docker images



- NOT A VHD
- NOT A FILESYSTEM
- uses a *Union File System*
- a read-only *Layer*
- do not have state
- Basically a tar file
- Has a hierarchy
  - Arbitrary depth
- Fits into the Docker Registry



Docker Containers



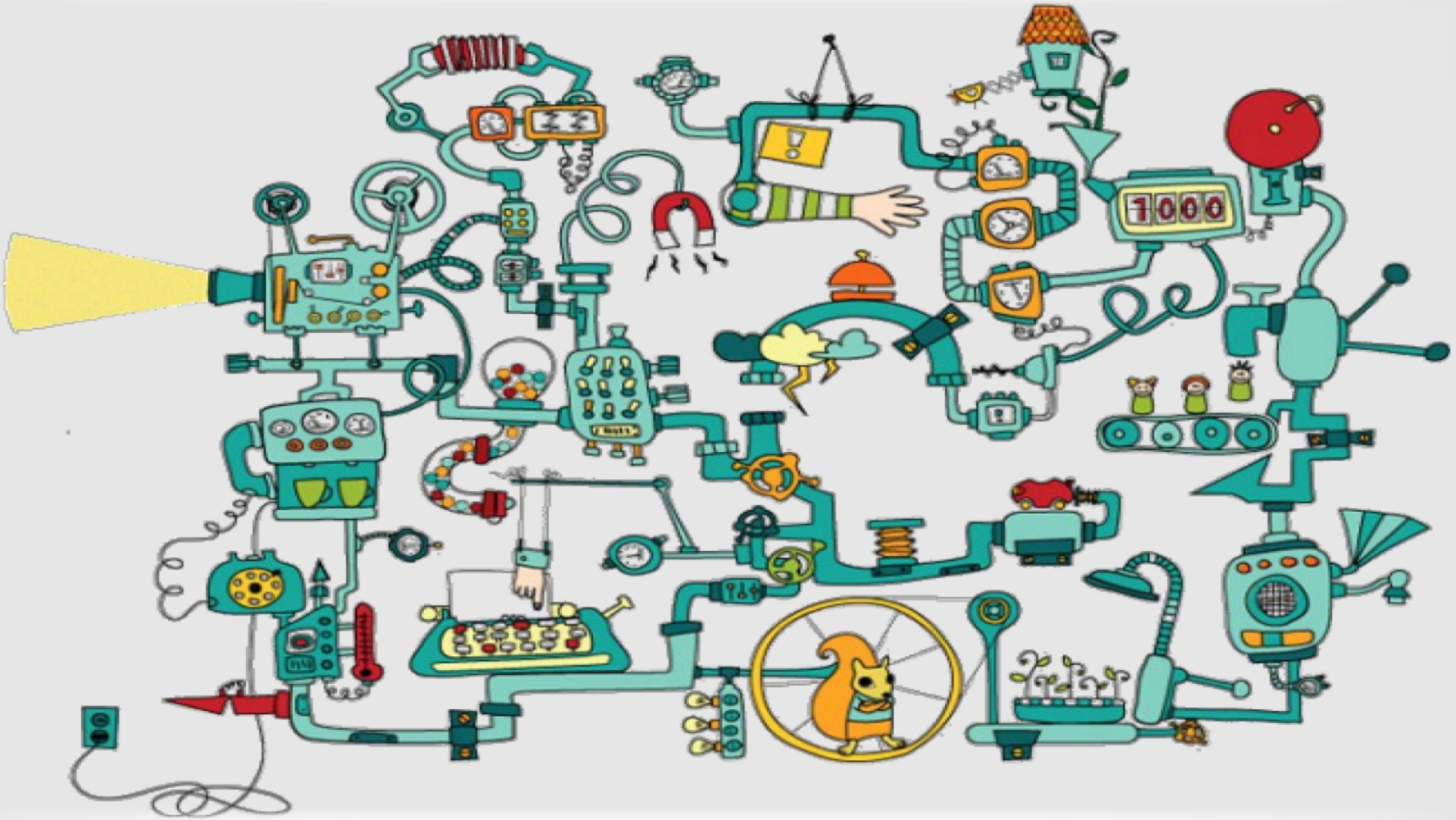
Units of software delivery (ship it!)

- run everywhere
  - regardless of kernel version
  - regardless of host distro
  - (but container and host architecture must match*)

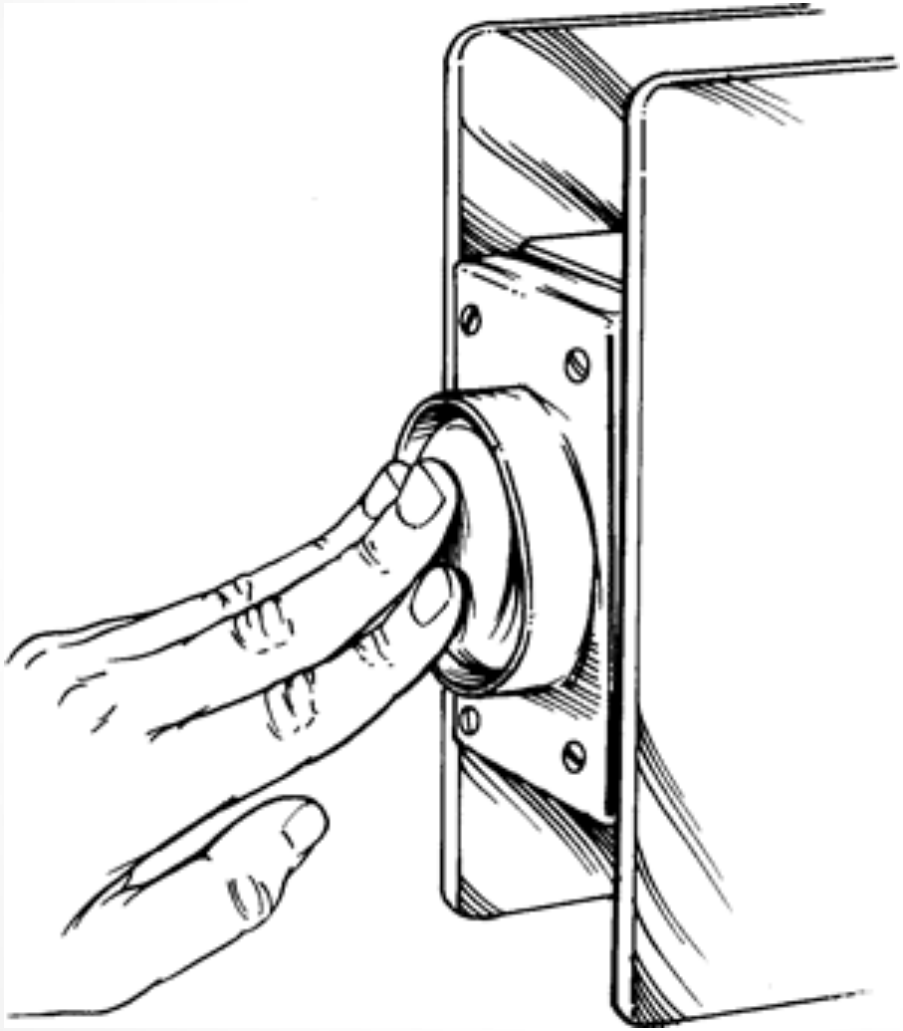
- run anything
  - if it can run on the host, it can run in the container
  - i.e., if it can run on a Linux kernel, it can run

*Unless you emulate CPU with qemu and binfmt

Containers before Docker……



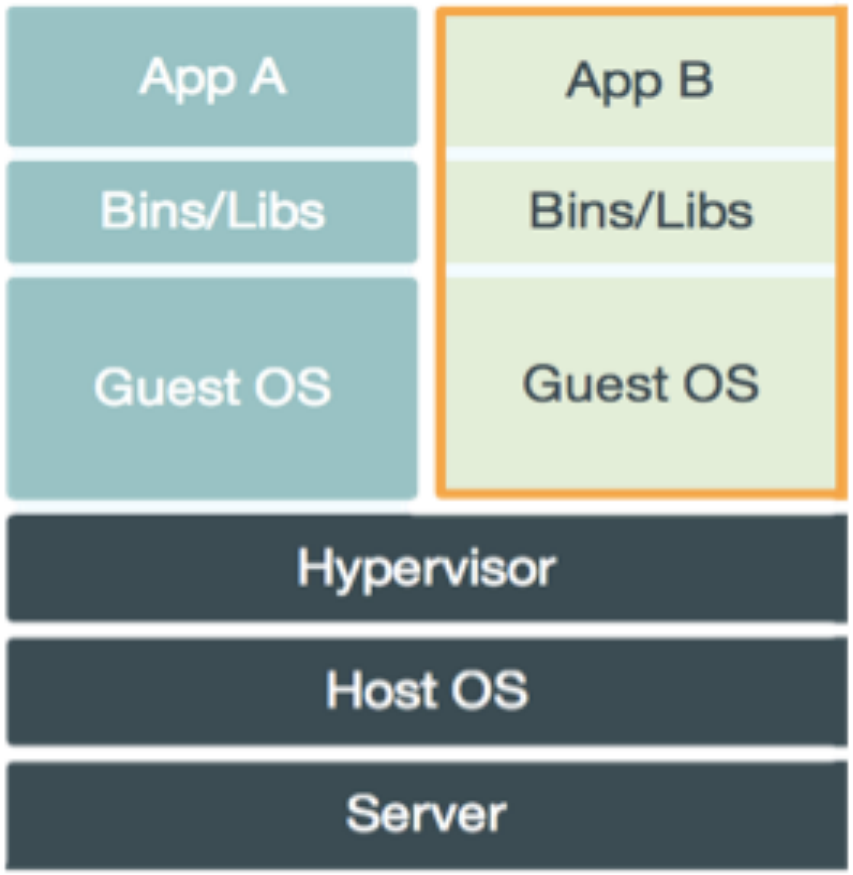
Containers after Docker ……

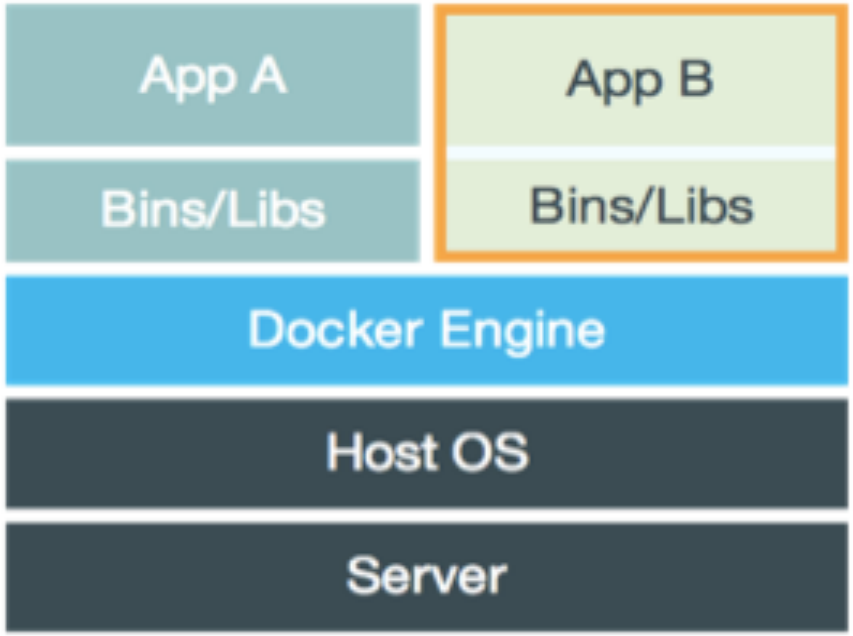


How does Docker work ?

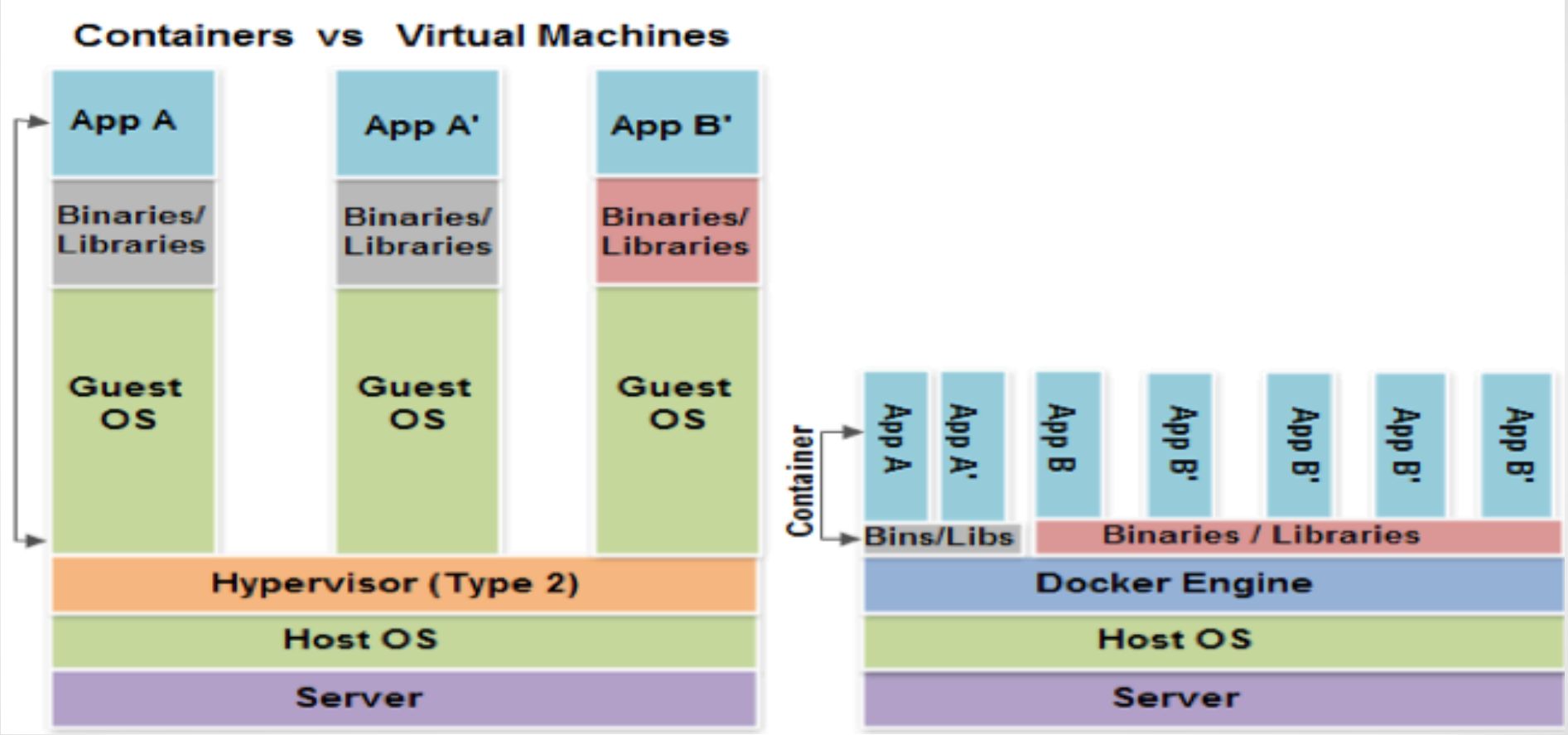
- You can build Docker images that hold your applications
- You can create Docker containers from those Docker images to run your applications.
- You can share those Docker images via Docker Hub or your own registry

Virtual Machine Versus Container……





Virtual Machine Versus Container……

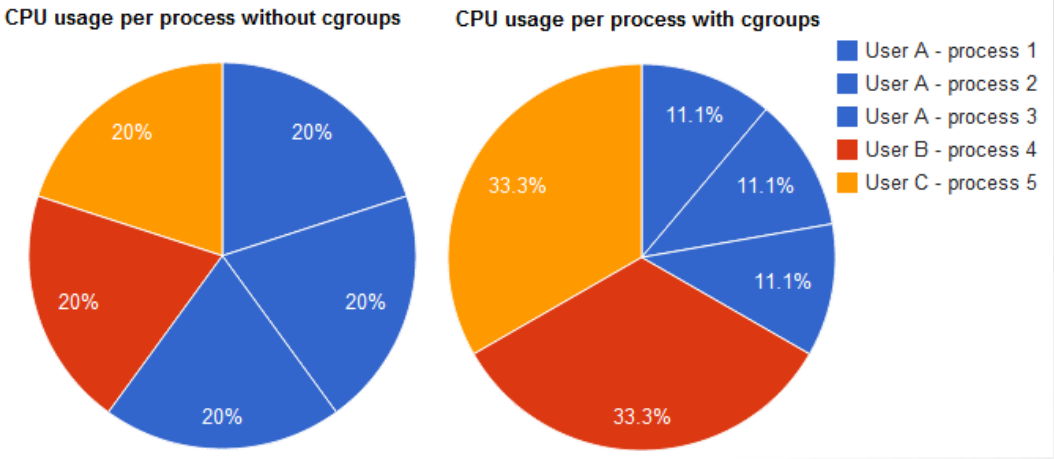


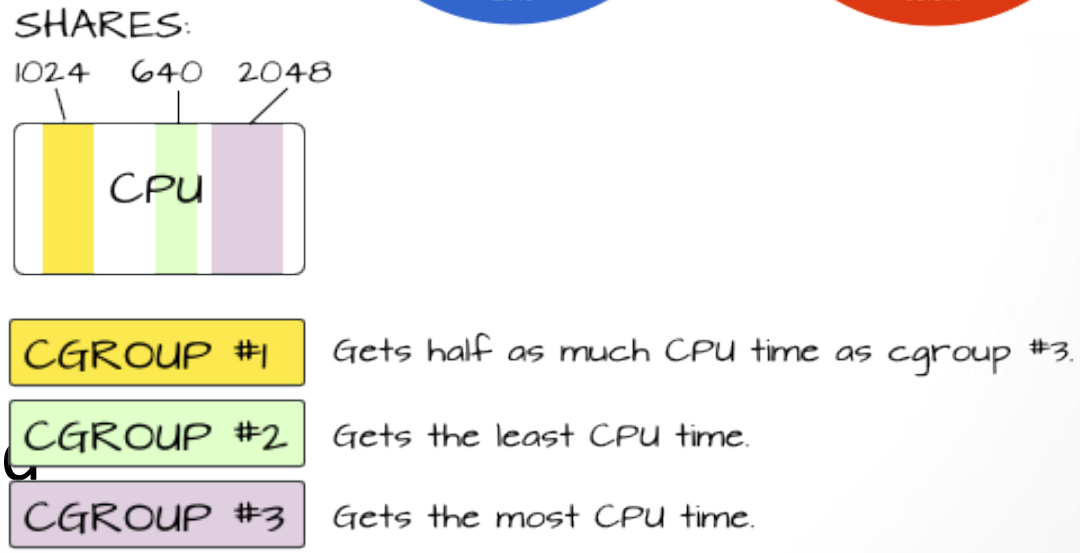
Docker Container Lifecycle ……

- The Life of a Container
  - Conception
    - **BUILD** an Image from a Dockerfile
  - Birth
    - **RUN** (create+start) a container
  - Reproduction
    - **COMMIT** (persist) a container to a new image
    - **RUN** a new container from an image
  - Sleep
    - **KILL** a running container
  - Wake
    - **START** a stopped container
  - Death
    - **RM** (delete) a stopped container
- Extinction
  - **RMI** a container image (delete image)

Linux Cgroups .....

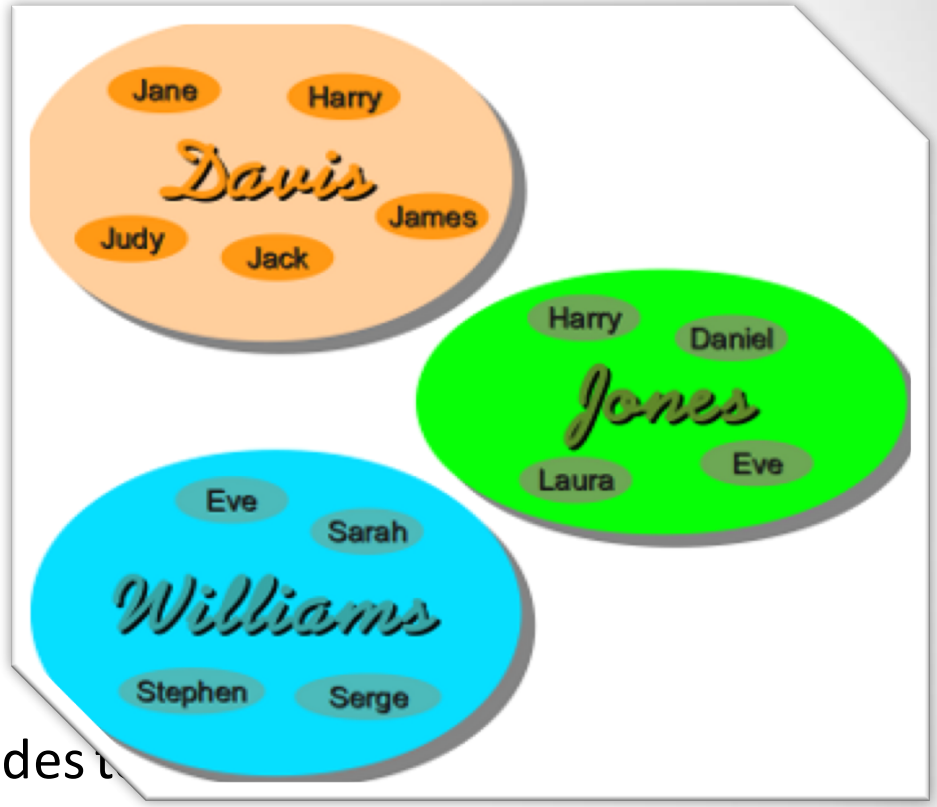
- Kernel Feature
- Groups of processes
- Control resource alloca
  - CPU
  - Memory
  - Disk
  - I/O
- May be nested



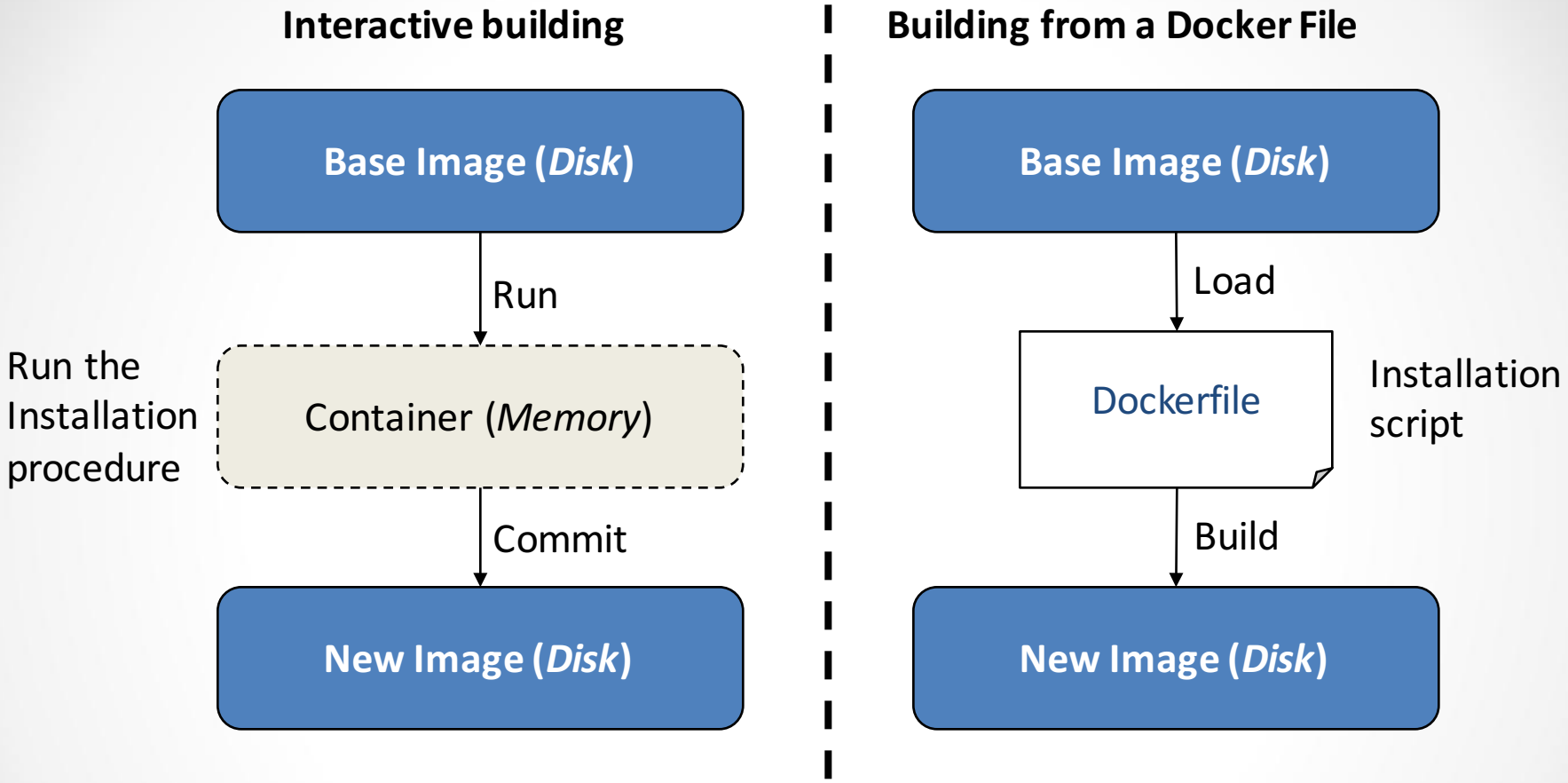


Linux Kernel Namespaces ……

- Kernel Feature
- Restrict your view of the system
  - Mounts (CLONE_NEWNS)
  - UTS (CLONE_NEWUTS)
    - uname() output
  - IPC (CLONE_NEWIPC)
  - PID (CLONE_NEWPID)
  - Networks (CLONE_NEWNET)
  - User (CLONE_NEWUSER)
    - Not supported in Docker yet
    - Has privileged/unprivileged modes t
- May be nested



Building a Docker Image



Docker Run Platforms

- Various Linux distributions (Ubuntu, Fedora, RHEL, Centos, openSUSE, ...)
- Cloud (Amazon EC2, Google Compute Engine, Rackspace)
- Windows, OSX: Boot2Docker

Installing Docker

$sudo yum -y install docker-io

$sudo yum -y update docker-io

$ sudo service docker start

Uninstalling Docker

$sudo service docker stop

$sudo rm -rf /var/lib/docker

$sudo yum erase docker-io

Terminology – Image (borrowed)

- **Persisted snapshot that can be run**
  - *images:* List all local images
  - *run*: Create a container from an image and execute a command in it
  - *pull*: Download image from repository
  - *rmi*: Delete a local image

Terminology – Container (borrowed)

- **Runnable instance of an image**
  - *ps:* List all running containers
  - *ps –a*: List all containers (incl. stopped)
  - *top*: Display processes of a container
  - *start*: Start a stopped container
  - *stop*: Stop a running container
  - *pause*: Pause all processes within a container
  - *rm*: Delete a container
  - *commit*: Create an image from a container

Daemon Container (borrowed)

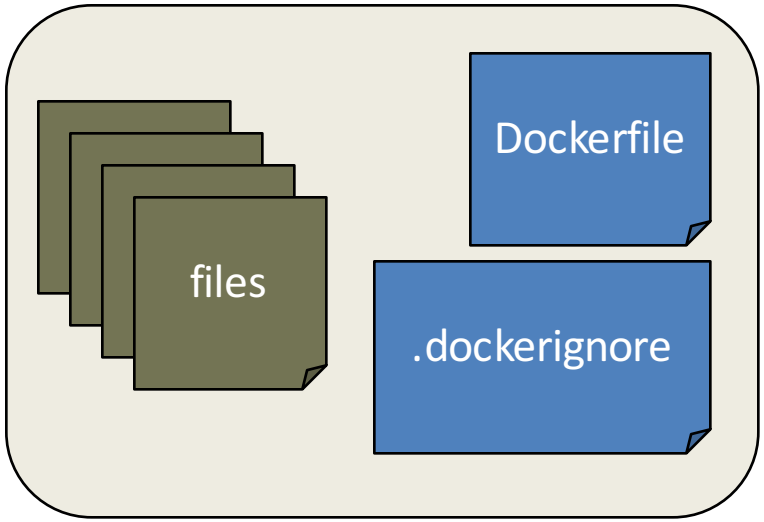
- Open Terminal in container:
  - **`docker run -it ubuntu /bin/bash`**
- Run as deamon: `docker run -d [image] command`

Dockerfile Example: Bowtie2

```
FROM ubuntu:14.04
MAINTAINER Enis Afgan <enis.afgan@jhu.edu>
RUN apt-get update -qq --fix-missing; \
    apt-get install -qq -y wget unzip;
RUN wget -q -O bowtie2.zip http://sourceforge.net/projects/bowtie-bio/files/bowtie2/2.2.4/bowtie2-2.2.4-linux-x86_64.zip/download;  \
    unzip bowtie2.zip -d /opt/; \
    ln -s /opt/bowtie2-2.2.4/ /opt/bowtie2; \
    rm bowtie2.zip
ENV PATH $PATH:/opt/bowtie2
```

Building a Docker Image from a Dockerfile

<source-directory>



$docker build -t <image-name> <source-directory>

SUMMARY……

- Easy to build, run & share containers
- Rapidly expanding ecosystem
- Better performance vs. VMs
- Layered file system gives us git-like control of images
- Reduces complexity of system builds
- Red Hat - Project Atomic Host, and certifications - containerized applications, Geard and OpenShift.
- Google is expected to tightly integrate containers with its IaaS and PaaS offerings.

TCPDump / Windump

- Low level package sniffer.
  - Good, if you see a new type of attack or try to diagnose a networking problem.
  - Bad, since you have to look at all these packages and learn how to interpret them.

TCPDump / Windump: The Good

- Provides an audit trail of network activity.
- Provides absolute fidelity.
- Universally available and cheap.

TCPDump / Windump: The Bad

- Does not collect the payload by default.
- Does not scale well.
- State / connections are hidden.
- Very Limited analysis of packages.
- Collects a given number of bytes from each package:
  - This could turn “trap and trace” monitoring into wiretaping because content might be captured.

Versions

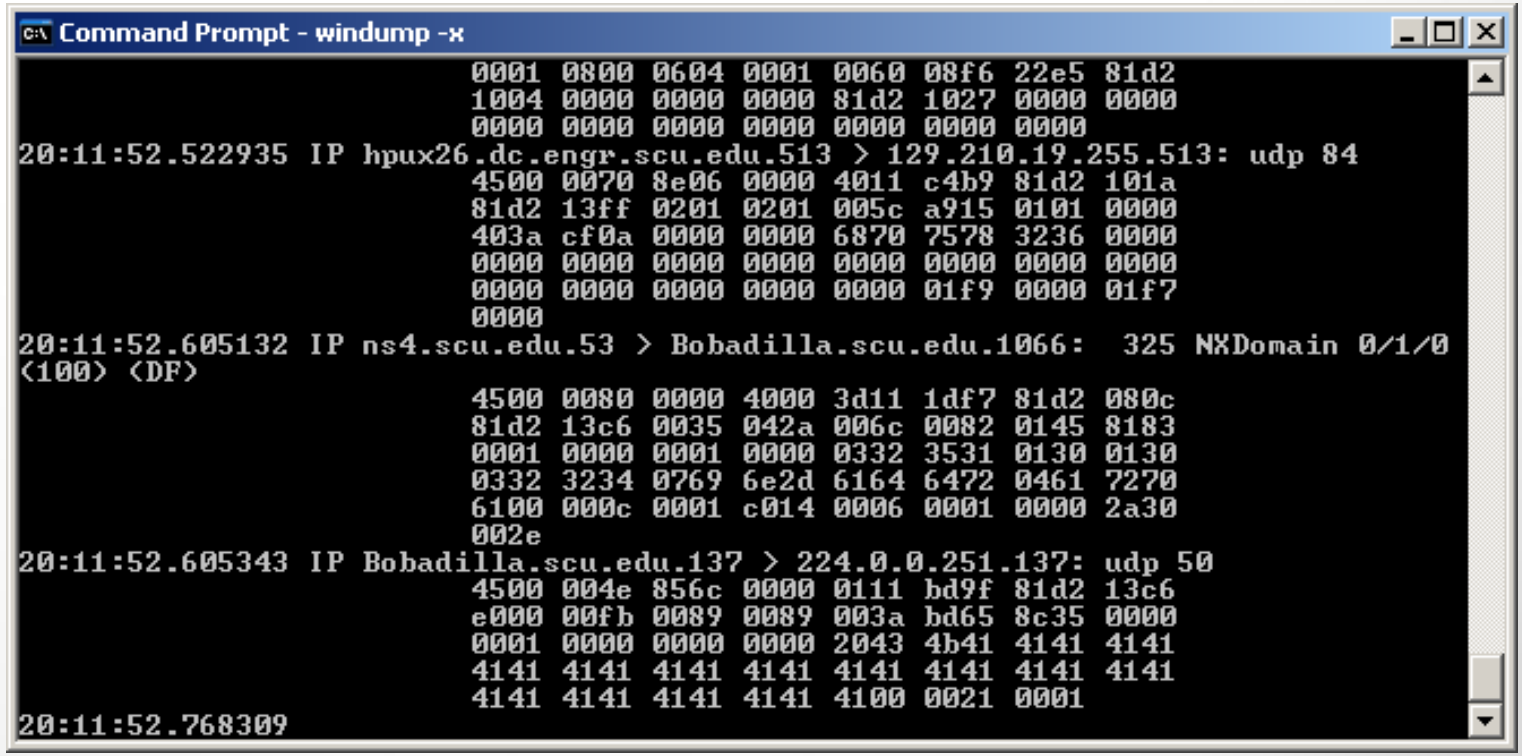
- Unix Version 3.4. ftp.ee.lbl.gov/tcpdump.tar.Z
- Windump

  http://netgroup-serv.polito.it/windump

  http://netgroup-serv.polito.it/winpcap
- www.tcpdump.org

Running TCPDump

- tcpdump –x looks at packages in hex format



Running tcpdump

- IP Header
- ICMP Header

`windump -x`

20:20:55.778140 IP dhcp-19-211.engr.scu.edu > Bobadilla.scu.edu: icmp 108: echo request seq 4864

4500 0080 0231 0000 8001 0d0f 81d2 13d3
81d2 13c6 0800 d5ee 0200 1300 6162 6364
6566 6768 696a 6b6c 6d6e 6f70 7172 7374
7576 7761 6263 6465 6667 6869 6a6b 6c6d
6e6f 7071 7273 7475 7677 6162 6364 6566
6768

tcpdump

- Use reference card to identify fields
- IP Version 4
- Header Length (Nr * 4B)

20:20:55.778140 IP dhcp-19-211.engr.scu.edu > Bobadilla.scu.edu: icmp 108: echo request seq 4864

4500 0080 0231 0000 8001 0d0f 81d2 13d3
81d2 13c6 0800 d5ee 0200 1300 6162 6364
6566 6768 696a 6b6c 6d6e 6f70 7172 7374
7576 7761 6263 6465 6667 6869 6a6b 6c6d
6e6f 7071 7273 7475 7677 6162 6364 6566
6768

tcpdump

- 20B header
- Type of Service
- Total Length: 0x80 = $128_{decimal}$

20:20:55.778140 IP dhcp-19-211.engr.scu.edu > Bobadilla.scu.edu: icmp 108: echo request seq 4864

4500 0080 0231 0000 8001 0d0f 81d2 13d3
81d2 13c6 0800 d5ee 0200 1300 6162 6364
6566 6768 696a 6b6c 6d6e 6f70 7172 7374
7576 7761 6263 6465 6667 6869 6a6b 6c6d
6e6f 7071 7273 7475 7677 6162 6364 6566
6768

tcpdump

- Length of capture: tcpdump –s 68
- Default is 68B
- We see only 54B, because the ethernet header is 14B long.
  - Remember, this could become a legal problem if you see content.

tcpdump

- tcpdump –e host bobadilla
  - Displays data link data filtered by host named bobadilla.
- Shows Source MAC
- Destination MAC
- Protocol

20:37:48.124457 0:8:74:3f:2:46 0:d:56:8:e4:db ip 142: IP dhcp-19-211.engr.scu.edu > Bobadilla.scu.edu: icmp 108: echo request seq 5376

Tcpdump
Fragmentation Total Length

- Total Length: Number of Bytes in Packet

```
20:42:07.217979 IP Bobadilla.scu.edu.137 >
  239.255.255.250.137: udp 50
4500 004e 892b 0000 0111 aae1 81d2 13c6
efff fffa 0089 0089 003a adb9 8ce2 0000
0001 0000 0000 0000 2043 4b41 4141 4141
4141 4141 4141 4141 4141 4141 4141 4141
4141 4141 4141 4141 4100 0021 0001
```

Tcpdump Fragmentation Offset Header

- Length 0x33c = 828 (-20B for header)
- Offset: 1ce8 → 0001 1100 1110 1000 = 7400
  - Leading 000 are flags.
- Multiply by 8: Offset = 59200

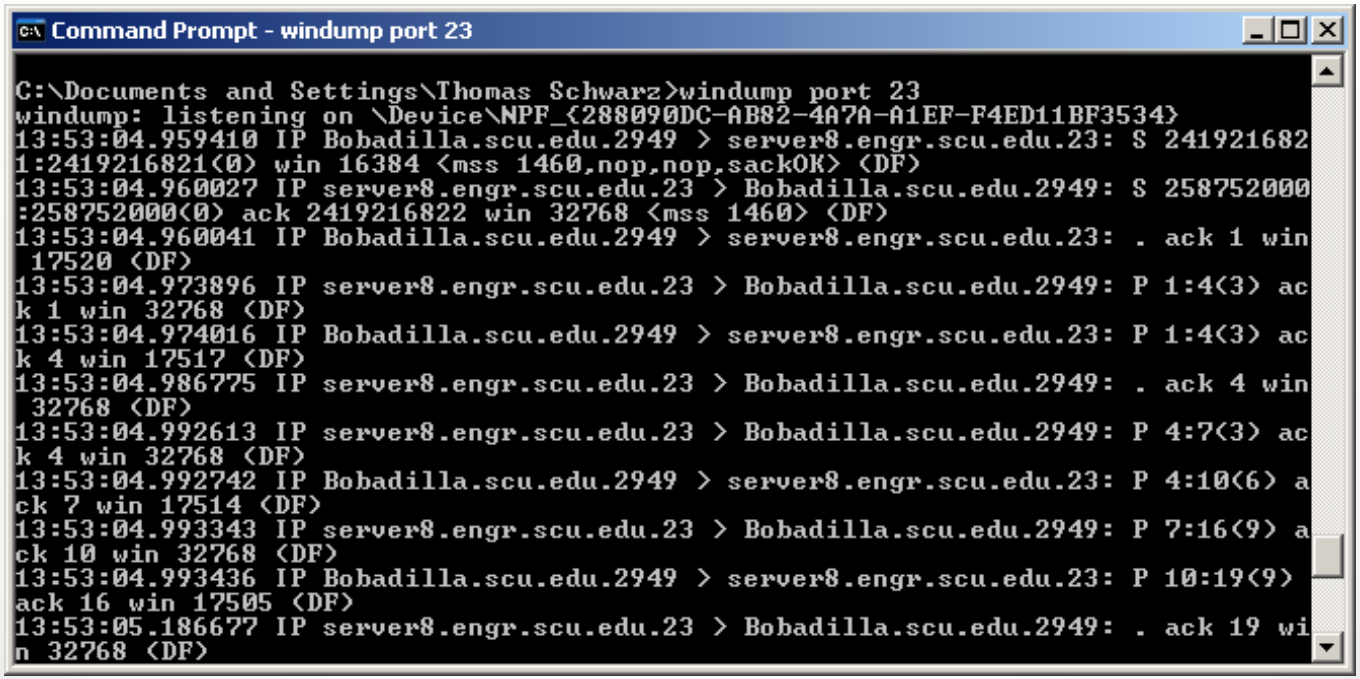
```
20:53:26.443325 IP Bobadilla.scu.edu >
  dhcp-19-211.engr.scu.edu: icmp (frag
  35188:808@59200)
4500 033c 8974 1ce8 8001 6627 81d2 13c6
81d2 13d3 6e6f 7071 7273 7475 7677 6162
6364 6566 6768 696a 6b6c 6d6e 6f70 7172
7374 7576 7761 6263 6465 6667 6869 6a6b
6c6d 6e6f 7071 7273 7475 7677 6162 6364
6566
```

TCPDump Filters

- Capture only packages that are useful.
  - Specify in the filter what items are interesting.
  - Filters use common fields such as host or port.
  - Filters also for individual bytes and bits in the datagram

TCPDump Filters

- Format 1: macro and value
- “tcpdump port 23”
  - Only displays packages going to or from port 23.



TCPDump Filters

- Format 2:
- <protocol header> [offset:length] <relation> <value>
- “ip[9] = 1”
  - Selects any record with the IP protocol of 1.
- “icmp[0] = 8”
  - Selects any record that is an ICMP echo requests.

That’s why you should learn to use the reference card.

TCPDump Filters

- Reference single bits through bit masking.
- An example is TCP flag bits
- Byte 13 in a TCP header has the 8 flag fields.
- CWR,ECE,URG,ACK,PSH,RST,SYN,FIN

TCPDump Filters

- Assume we want to mask out the PSH field.
- Translate the mask into binary.
- 0x08

cwr	ece	urg	ack	psh	rst	syn	fin
0	0	0	0	1	0	0	0

TCPDump Filters

- Set filter to

  tcp[13] & 0x80 != 0.

- Your turn:
  - Filter for packets that have the Syn or the Ack flag set.

TCPDump Filters

- Your turn:
  - Filter for packets that have the Syn or the Ack flag set.
  - tcp[13] & 0x12 != 0

cwr	ece	urg	ack	psh	rst	syn	fin
0	0	0	1	0	0	1	0

TCPDump Filters

- We can of course use exact values for filtering.
- tcp[13] = 0x20 looks only for tcp-packets that have the urg flag set.

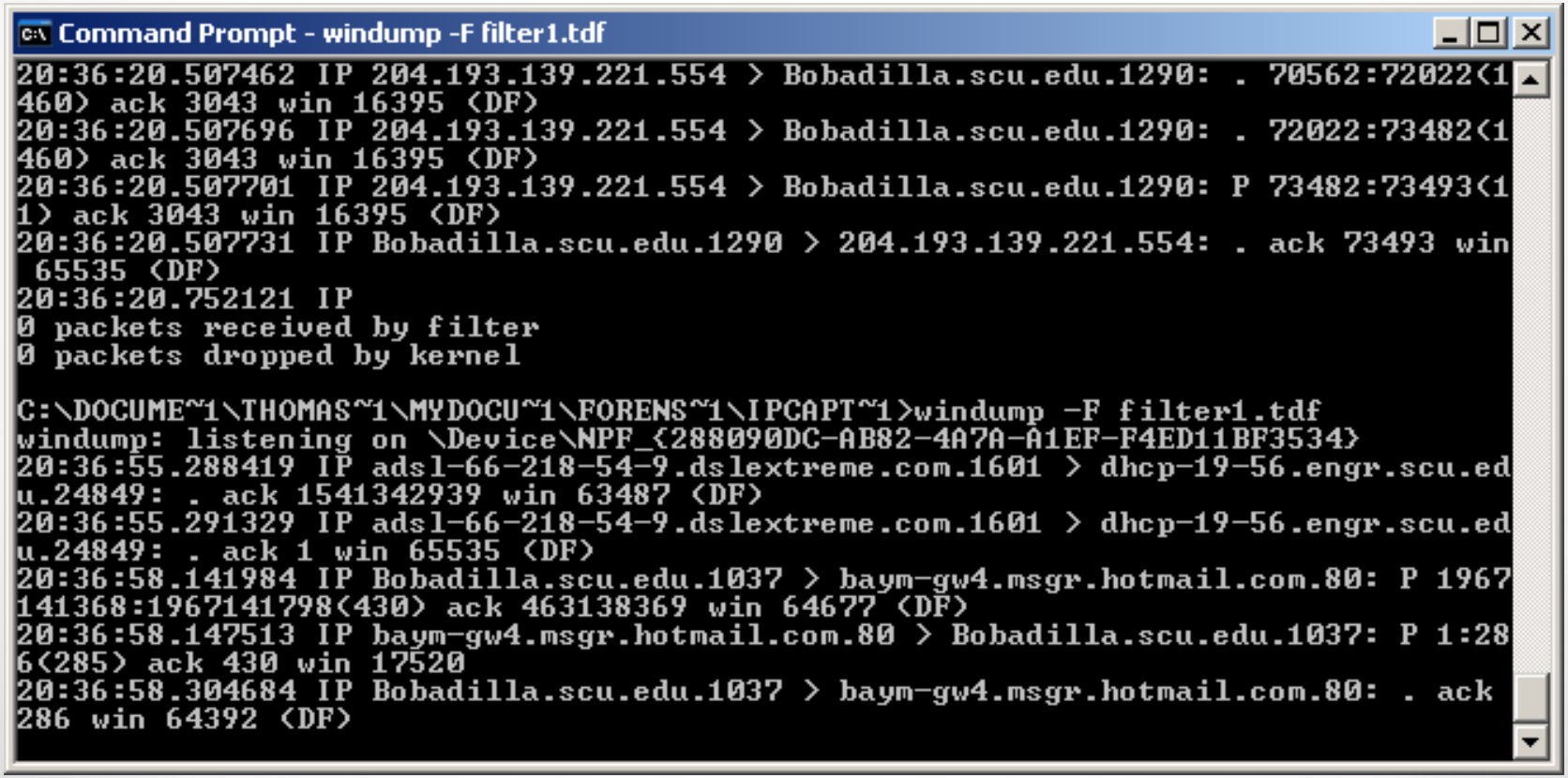
cwr	ece	urg	ack	psh	rst	syn	fin
0	0	1	0	0	0	0	0

TCPDump Filters

- Can combine filters with the and, or, not operators
- (tcp and tcp[13]&0x0f != 0 and not port 25) or port 20
- Filter can be written in file, specified with the –F flag.

TCPDump Filters

- Use –F filename to specify a file containing the filter.



TCPDump

- Use the –w extension to capture into a file.
- Use the –c extension to limit the number of packets captured.
- Use –v, -vv, -vvv for verbosity.
- Use –x for ASCI values of package contents.
- Use –tttt to display time / day stamps.
- Use –r to specify capture file.

Motivation for Network Monitoring

- Essential for Network Management
  - Router and Firewall policy
  - Detecting abnormal/error in networking
  - Access control
- Security Management
  - Detecting abnormal traffic
  - Traffic log for future forensic analysis

Demo 2

1. Capture only udp packets
   - tcpdump “udp”
2. Capture only tcp packets
   - tcpdump “tcp”

Demo 2 (contd.)

1. Capture only UDP packets with destination port 53 (DNS requests)
   - tcpdump “udp dst port 53”
2. Capture only UDP packets with source port 53 (DNS replies)
   - tcpdump “udp src port 53”
3. Capture only UDP packets with source or destination port 53 (DNS requests and replies)
   - tcpdump “udp port 53”

Demo 2 (contd.)

1. Capture only packets destined to quasar.cs.berkeley.edu
   - tcpdump “dst host quasar.cs.berkeley.edu”
2. Capture both DNS packets and TCP packets to/from quasar.cs.berkeley.edu
   - tcpdump “(tcp and host quasar.cs.berkeley.edu) or udp port 53”

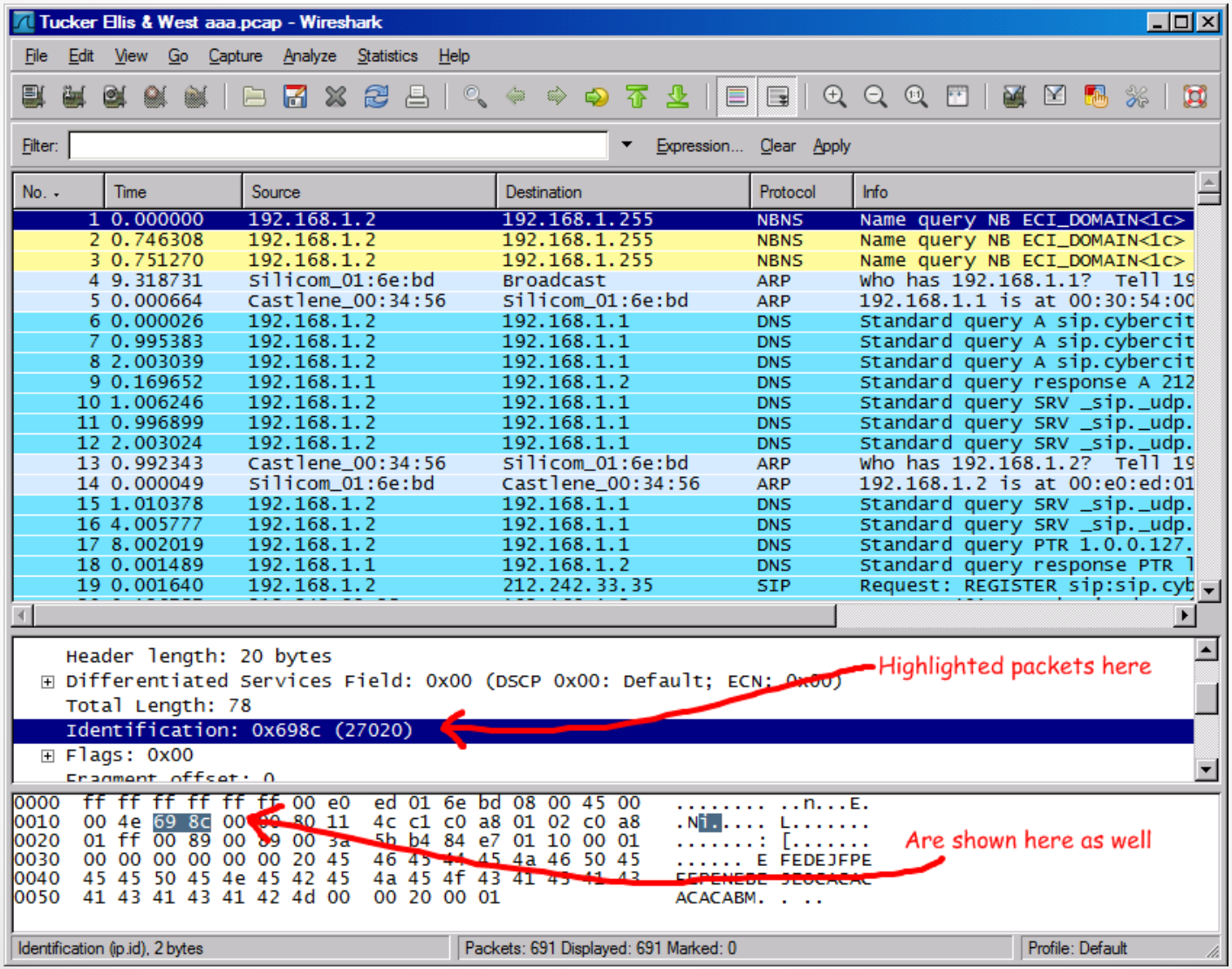
How to write filters

- Refer the tcpdump/tshark man page
- Many example webpages on the Internet

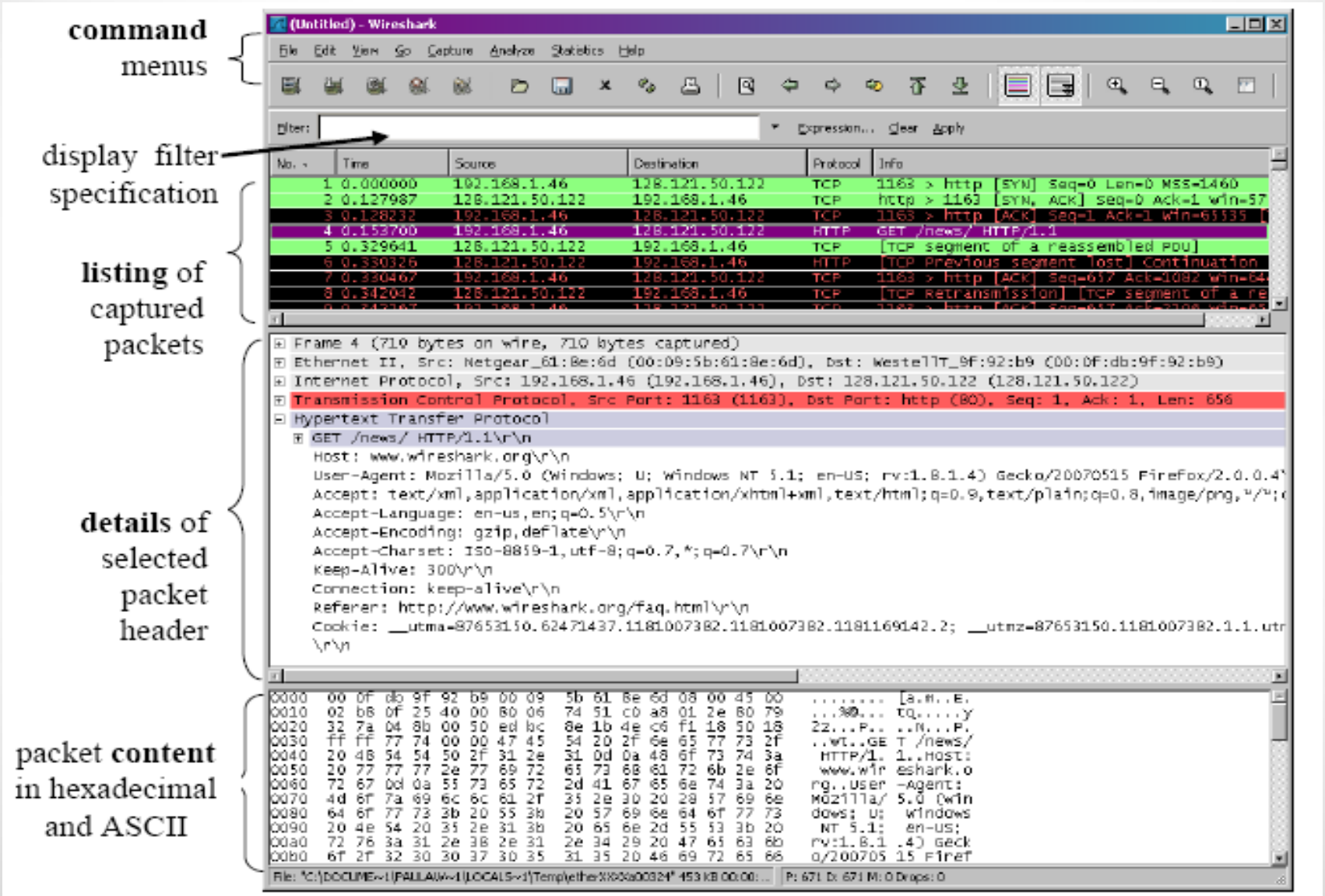
So What is WireShark?

- Packet sniffer/protocol analyzer
- Open Source Network Tool
- Latest version of the ethereal tool

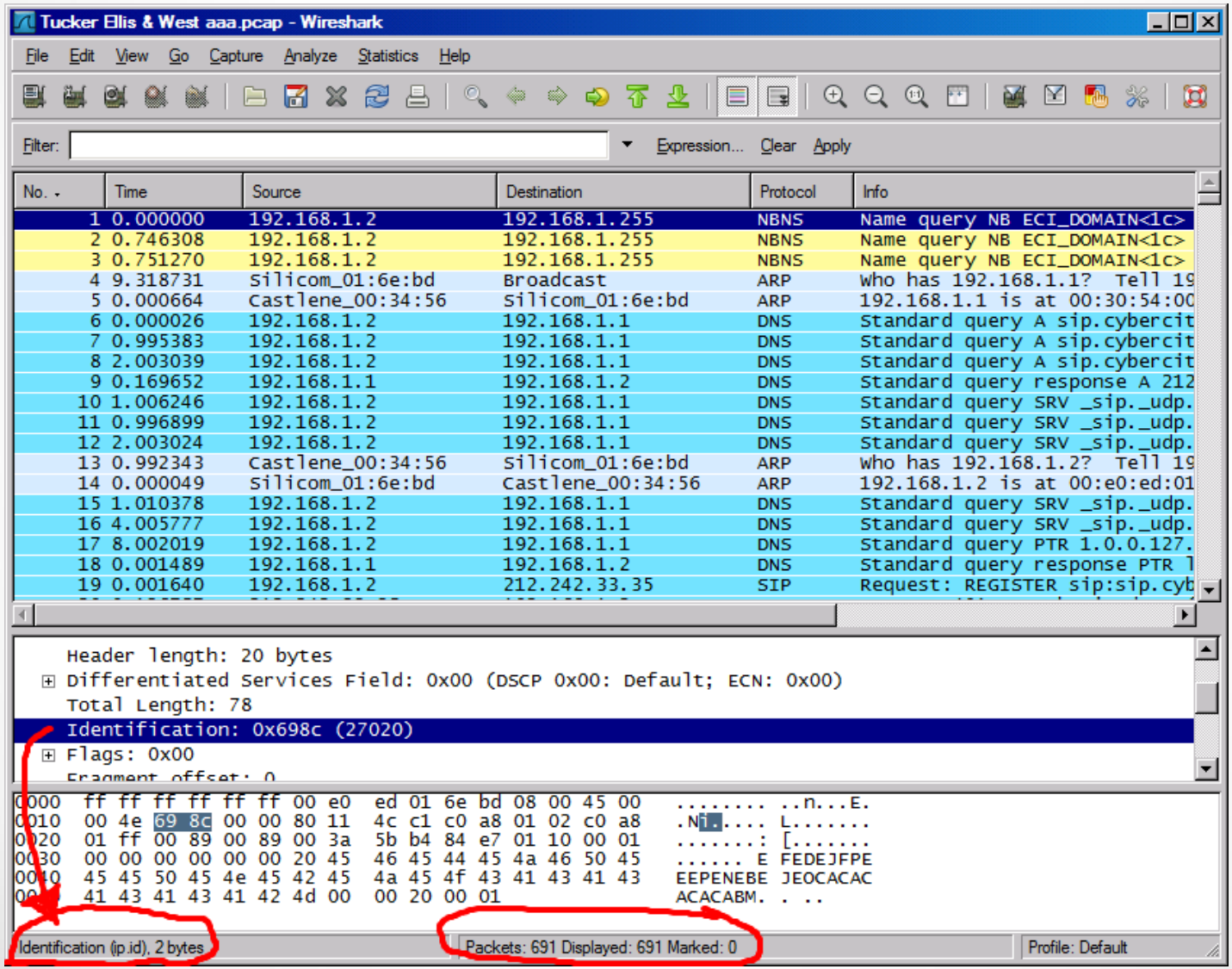
Wireshark Interface



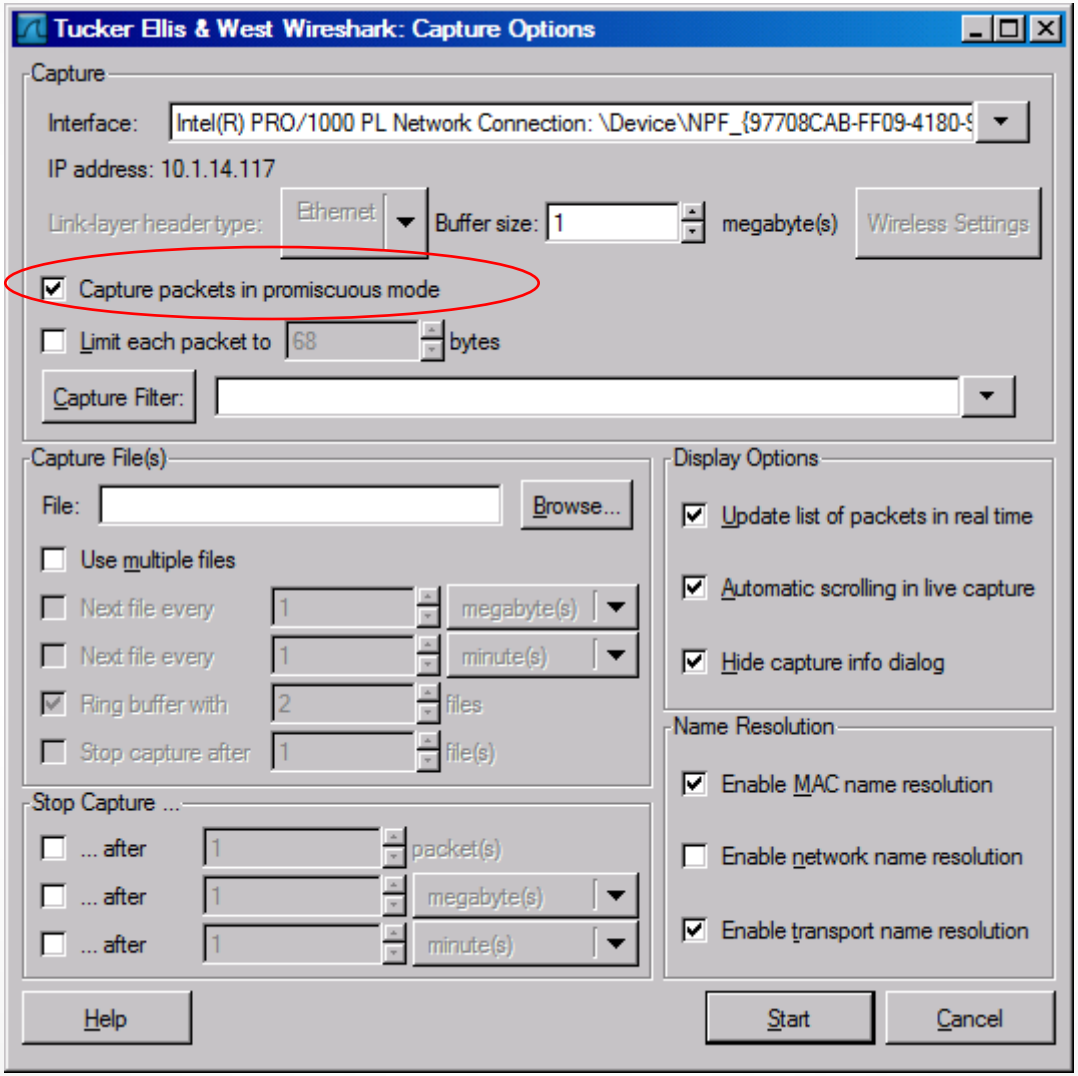
Wireshark Interface



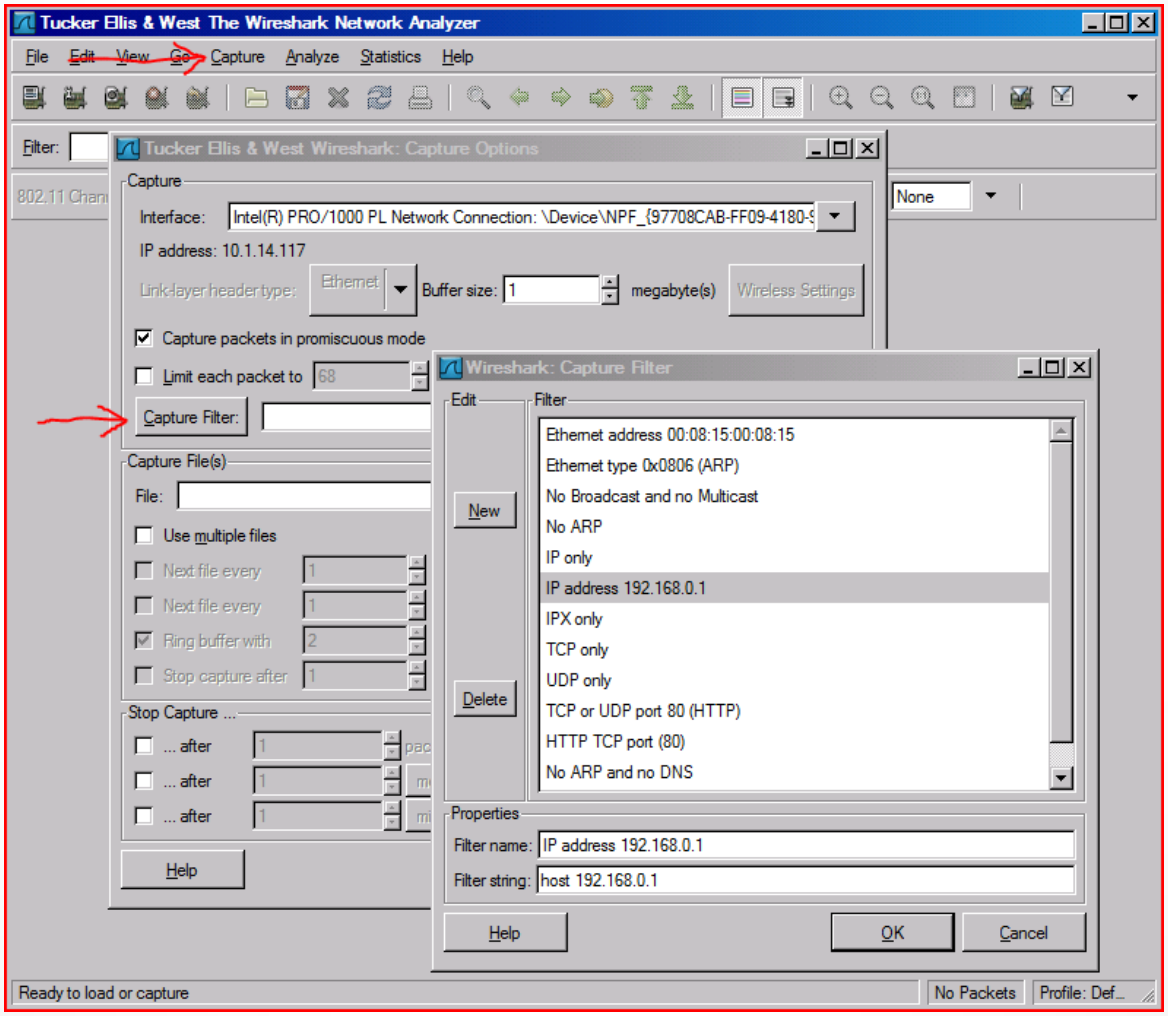
Status Bar


Tucker Ellis & West aaa.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 192.168.1.2 192.168.1.255 NBNS Name query NB ECI_DOMAIN<1c>
2 0.746308 192.168.1.2 192.168.1.255 NBNS Name query NB ECI_DOMAIN<1c>
3 0.751270 192.168.1.2 192.168.1.255 NBNS Name query NB ECI_DOMAIN<1c>
4 9.318731 Silicom_01:6e:bd Broadcast ARP Who has 192.168.1.1? Tell 19
5 0.000664 Castlene_00:34:56 Silicom_01:6e:bd ARP 192.168.1.1 is at 00:30:54:00
6 0.000026 192.168.1.2 192.168.1.1 DNS Standard query A sip.cybercit
7 0.995383 192.168.1.2 192.168.1.1 DNS Standard query A sip.cybercit
8 2.003039 192.168.1.2 192.168.1.1 DNS Standard query A sip.cybercit
9 0.169652 192.168.1.1 192.168.1.2 DNS Standard query response A 212
10 1.006246 192.168.1.2 192.168.1.1 DNS Standard query SRV _sip._udp.
11 0.996899 192.168.1.2 192.168.1.1 DNS Standard query SRV _sip._udp.
12 2.003024 192.168.1.2 192.168.1.1 DNS Standard query SRV _sip._udp.
13 0.992343 Castlene_00:34:56 Silicom_01:6e:bd ARP Who has 192.168.1.2? Tell 19
14 0.000049 Silicom_01:6e:bd Castlene_00:34:56 ARP 192.168.1.2 is at 00:e0:ed:01
15 1.010378 192.168.1.2 192.168.1.1 DNS Standard query SRV _sip._udp.
16 4.005777 192.168.1.2 192.168.1.1 DNS Standard query SRV _sip._udp.
17 8.002019 192.168.1.2 192.168.1.1 DNS Standard query PTR 1.0.0.127.
18 0.001489 192.168.1.1 192.168.1.2 DNS Standard query response PTR 1
19 0.001640 192.168.1.2 212.242.33.35 SIP Request: REGISTER sip:sip.cyb
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 78
Identification: 0x698c (27020)
Flags: 0x00
0000 ff ff ff ff ff ff 00 e0 ed 01 6e bd 08 00 45 00 ....n...E.
0010 00 4e 69 8c 00 00 80 11 4c c1 c0 a8 01 02 c0 a8 .Ni..... L.....
0020 01 ff 00 89 00 89 00 3a 5b b4 84 e7 01 10 00 01 ..... [.....
0030 00 00 00 00 00 00 20 45 46 45 44 45 4a 46 50 45 ..... E FEDEJFPE
0040 45 45 50 45 4e 45 42 45 4a 45 4f 43 41 43 41 43 EEPENEBE JEOCACAC
0050 41 43 41 43 41 42 4d 00 00 20 00 01 ACACABM. . ..
Identification (ip.id), 2 bytes
Packets: 691 Displayed: 691 Marked: 0
Profile: Default

Capture Options



Capture Filter



Capture Filter examples

host 10.1.11.24

host 192.168.0.1 and host 10.1.11.1

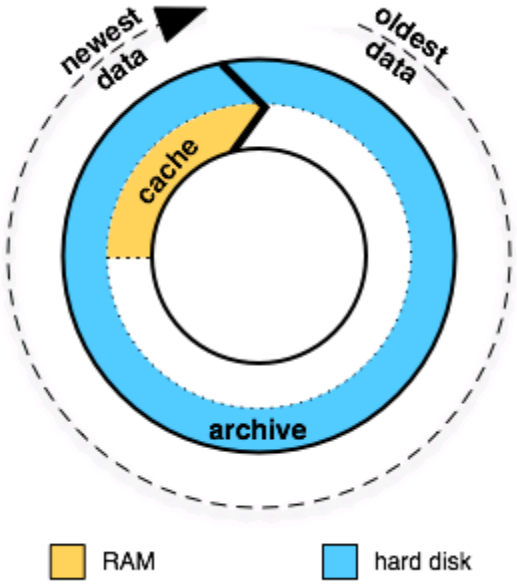
tcp port http

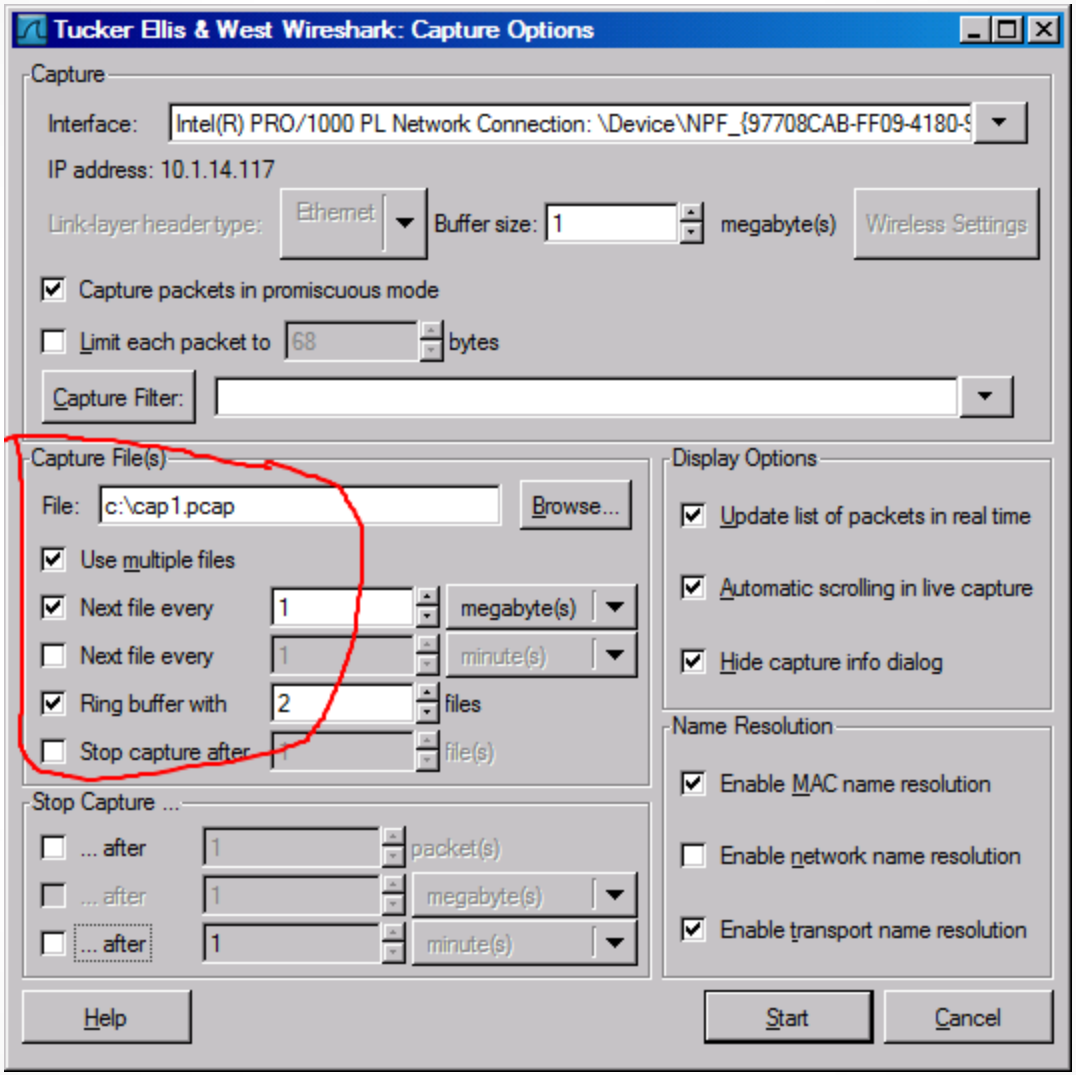
ip

not broadcast not multicast

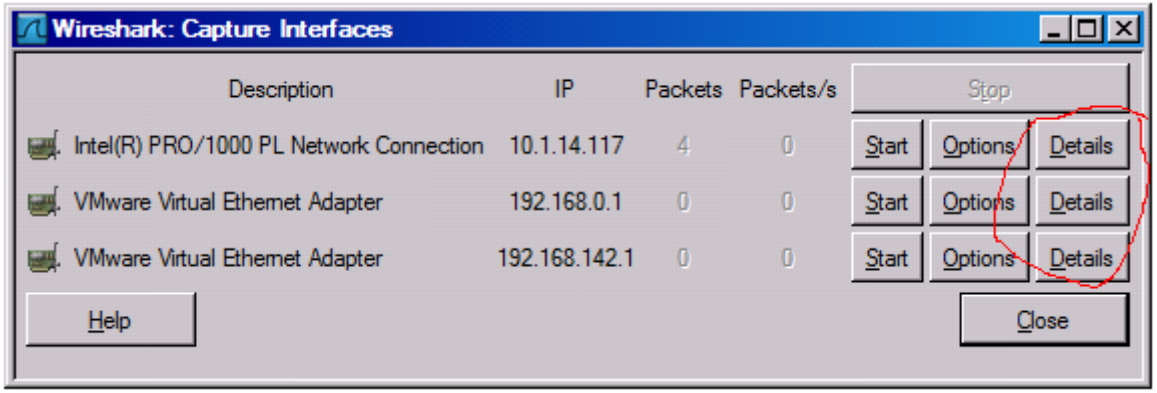
ether host 00:04:13:00:09:a3

Capture Buffer Usage

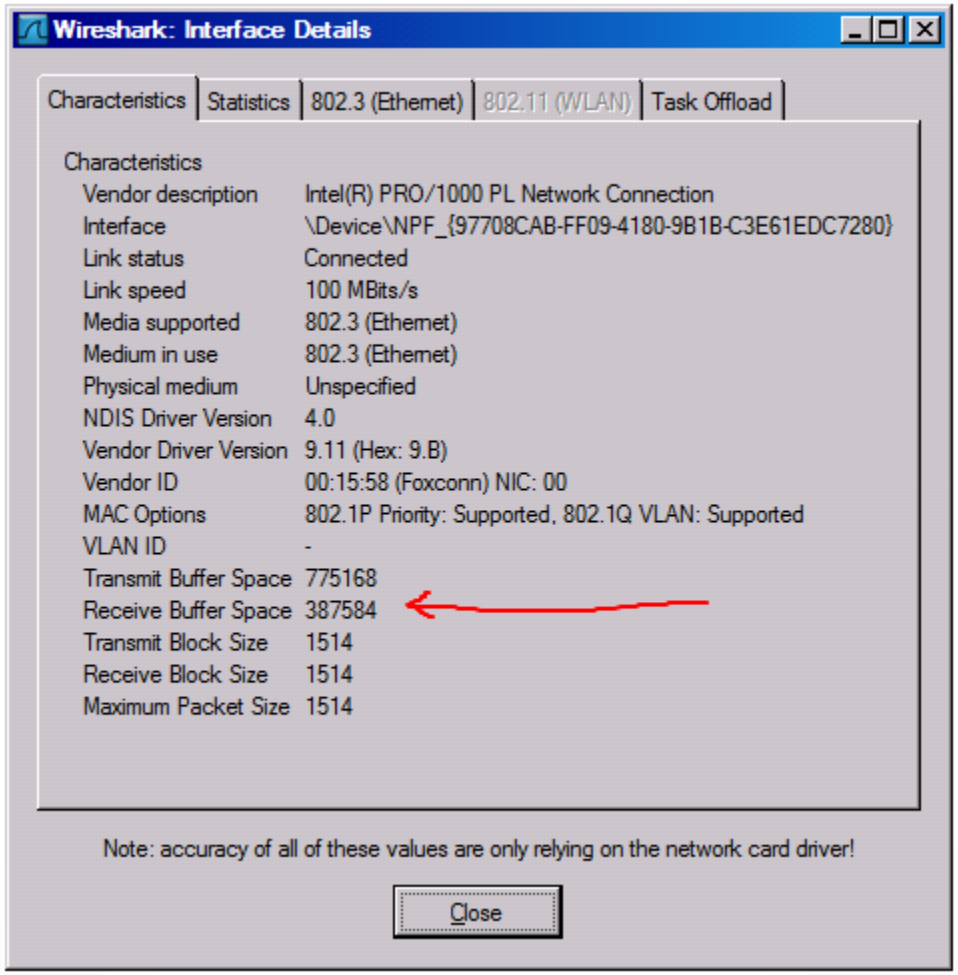



Tucker Ellis & West Wireshark: Capture Options
Capture
Interface:
Intel(R) PRO/1000 PL Network Connection: \Device\NPF_{97708CAB-FF09-4180-9
IP address: 10.1.14.117
Link-layer header type:
Ethernet
Buffer size:
1
megabyte(s)
Wireless Settings
Capture packets in promiscuous mode
Limit each packet to
68
bytes
Capture Filter:
Capture File(s)
File:
c:\cap1.pcap
Browse...
Use multiple files
Next file every
1
megabyte(s)
Next file every
1
minute(s)
Ring buffer with
2
files
Stop capture after
1
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
Help
Start
Cancel

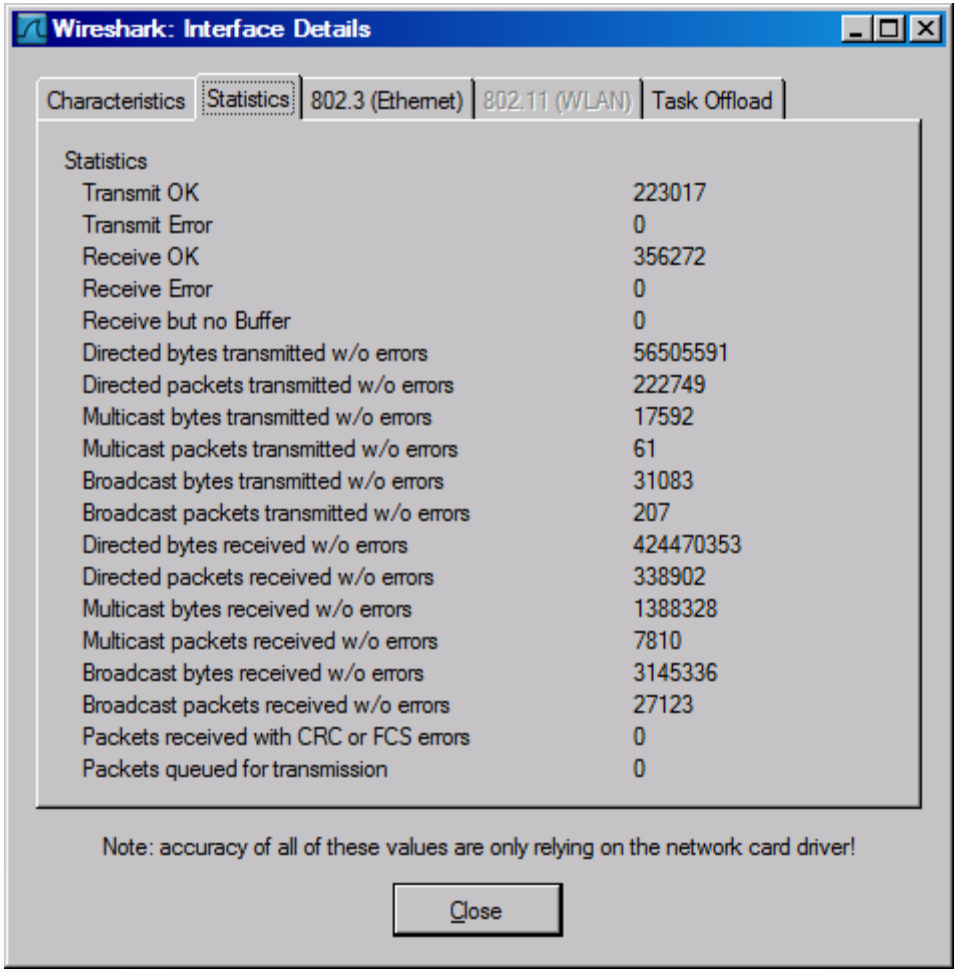
Capture Interfaces



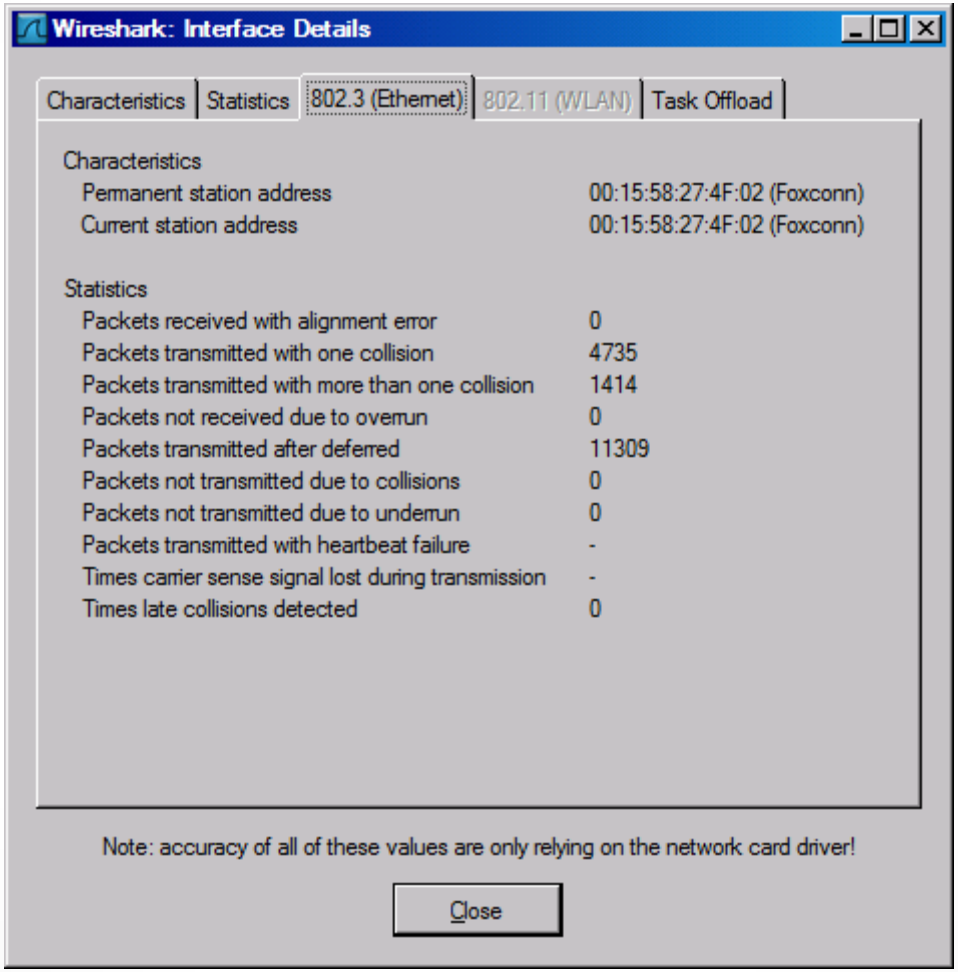
Interface Details: Characteristics



Interface Details: Statistics



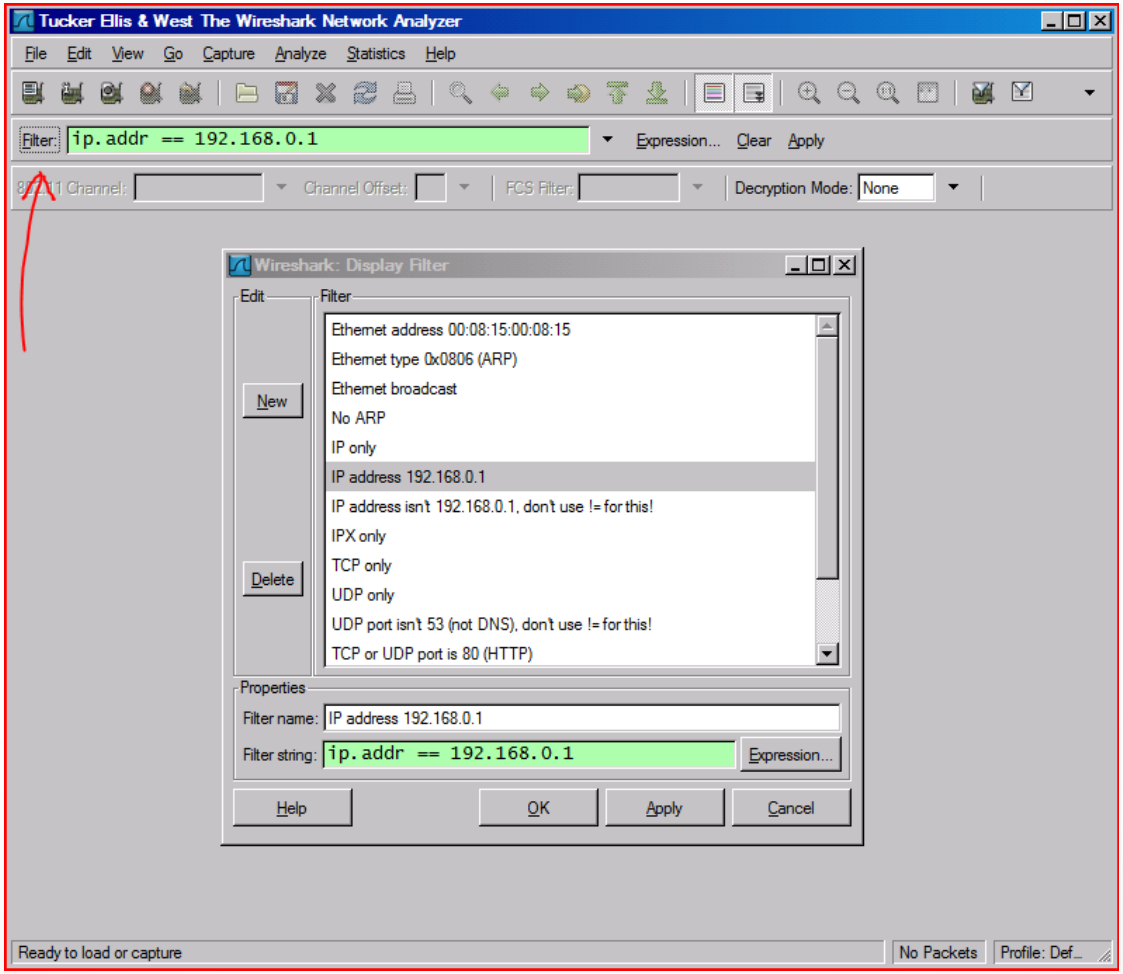
Interface Details: 802.3 (Ethernet)



Display Filters (Post-Filters)

- Display filters (also called post-filters) only filter the view of what you are seeing. All packets in the capture still exist in the trace
- Display filters use their own format and are much more powerful then capture filters

Display Filter



Display Filter Examples

ip.src==10.1.11.00/24

ip.addr==192.168.1.10 && ip.addr==192.168.1.20

tcp.port==80 || tcp.port==3389

!(ip.addr==192.168.1.10 && ip.addr==192.168.1.20)

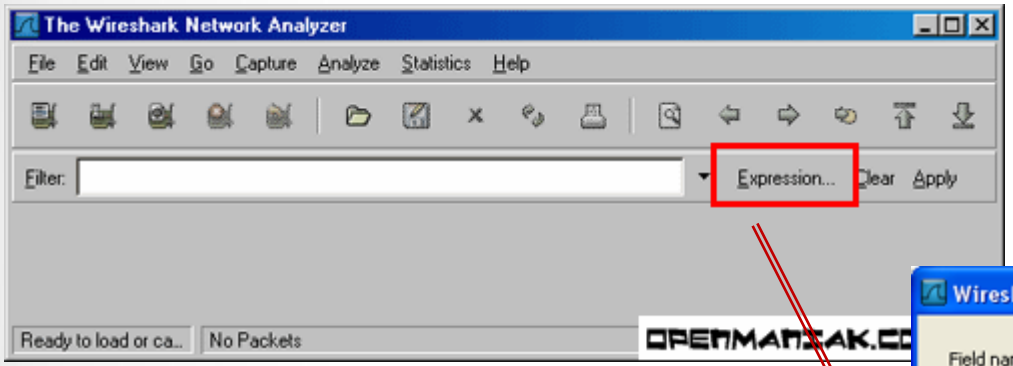
(ip.addr==192.168.1.10 && ip.addr==192.168.1.20) && (tcp.port==445 || tcp.port==139)

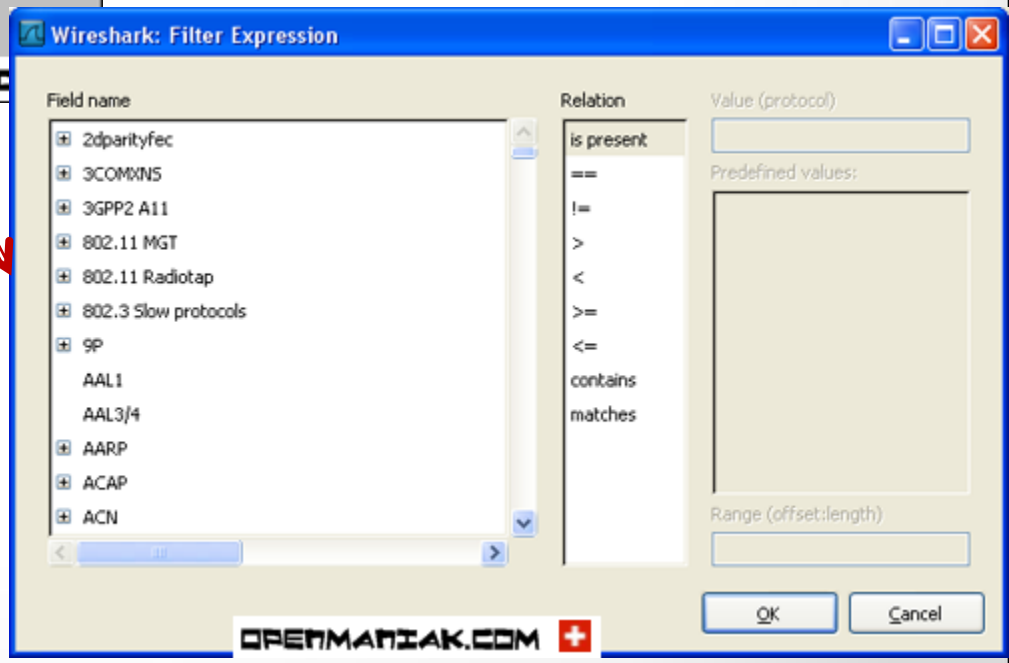
(ip.addr==192.168.1.10 && ip.addr==192.168.1.20) && (udp.port==67 || udp.port==68)

tcp.dstport == 80

Display Filter

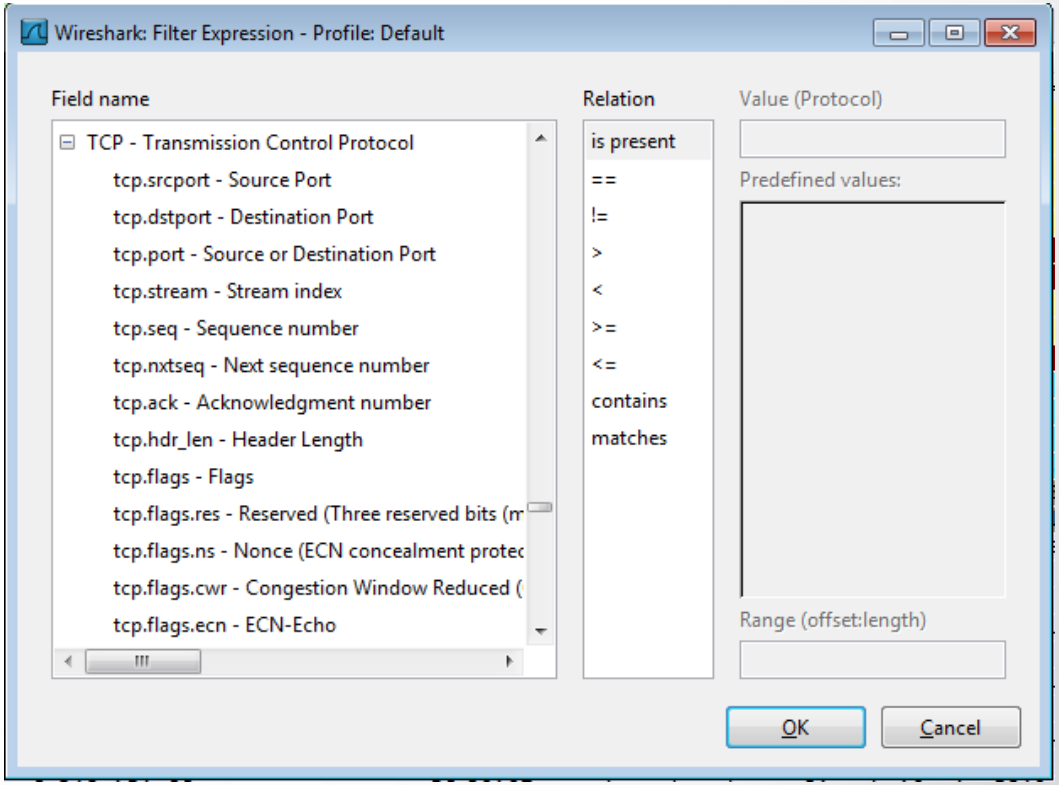
Syntax:	**Protocol**	.	**String 1**	.	**String 2**	**Comparison operator**	**Value**	**Logical Operations**	**Other expression**
Example:	ftp		passive		ip	==	10.2.3.4	xor	icmp.type





Display Filter

- String1, String2 (Optional settings):
  - Sub protocol categories inside the protocol.
  - Look for a protocol and then click on the "+" character.
  - Example:
  - **tcp.srcport == 80**
  - **tcp.flags == 2**
    - **SYN packet**
    - **Tcp.flags.syn==1**
  - **tcp.flags == 18**
    - **SYN/ACK**
  - **Note of TCP Flag field:**



U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N

Display Filter Expressions

- snmp || dns || icmp
  - Display the SNMP or DNS or ICMP traffics.
- tcp.port == 25
  - Display packets with TCP source or destination port 25.
- tcp.flags
  - Display packets having a TCP flags
- tcp.flags.syn == 0x02
  - Display packets with a TCP SYN flag.

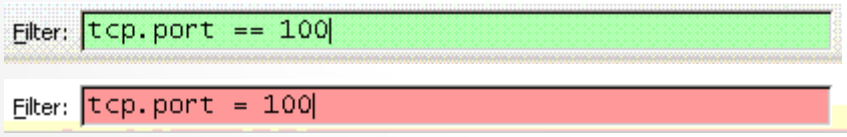
Six comparison operators are available:

English format:	C like format:	Meaning:
eq	==	Equal
ne	!=	Not equal
gt	>	Greater than
lt	<	Less than
ge	>=	Greater or equal
le	<=	Less or equal

Logical expressions:

English format:	C like format:	Meaning:
and	&&	Logical AND
or	||	Logical OR
xor	^^	Logical XOR
not	!	Logical NOT

If the filter syntax is correct, it will be highlighted in green, otherwise if there is a syntax mistake it will be highlighted in red.



Correct syntax

Wrong syntax

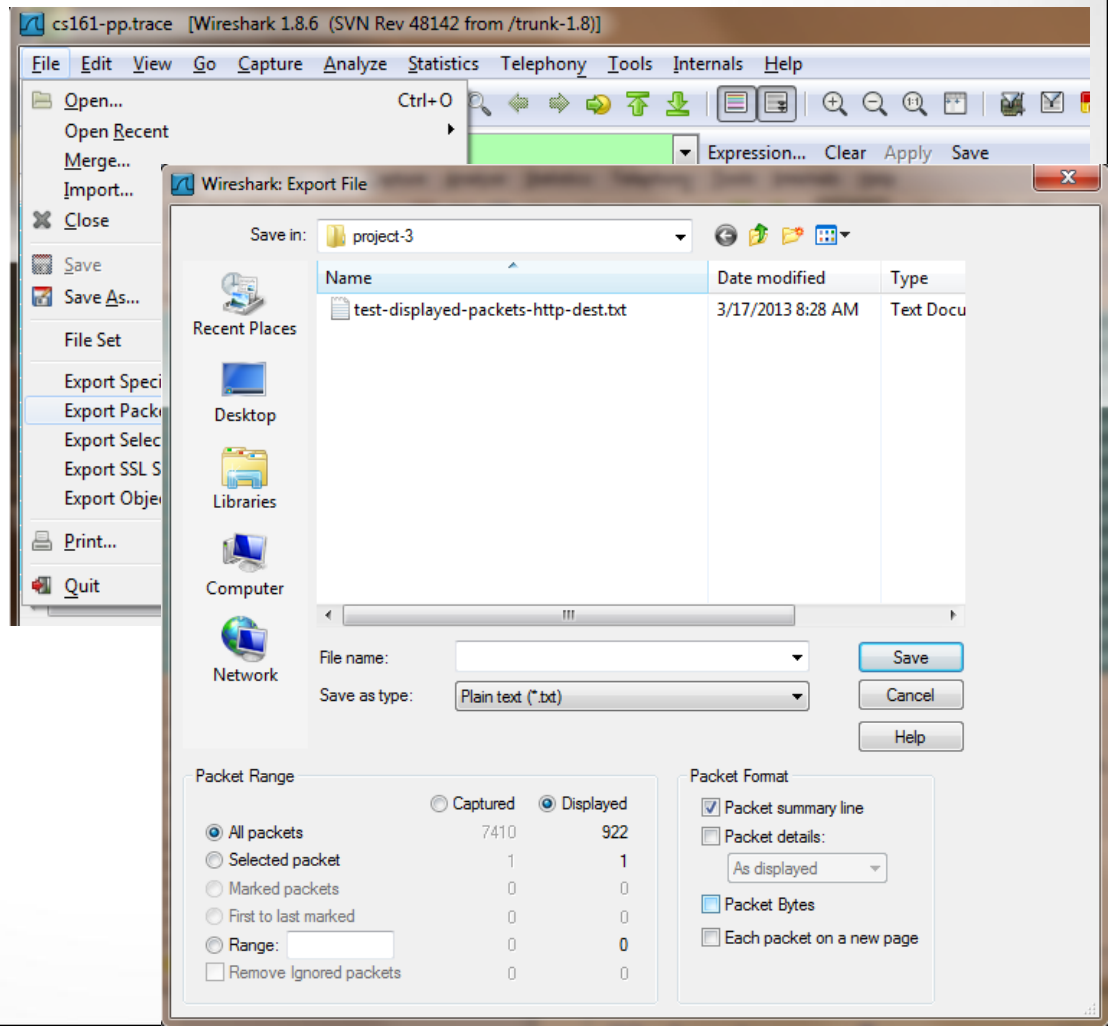
Save Filtered Packets After Using Display Filter

- We can also save all filtered packets in text file for further analysis
- Operation:

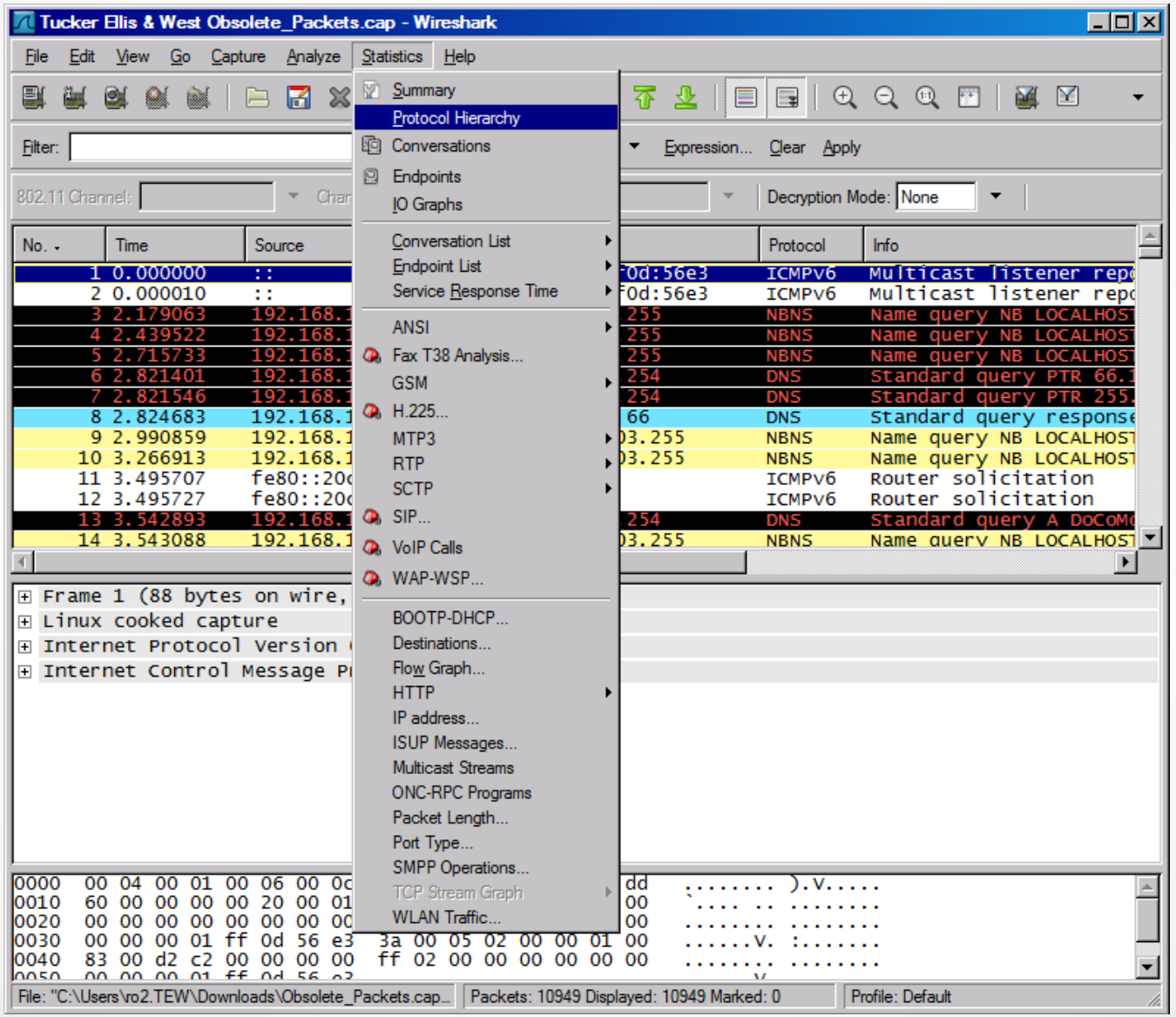
File→Export packet dissections →as "plain text" file

1). In "packet range" option, select "Displayed"

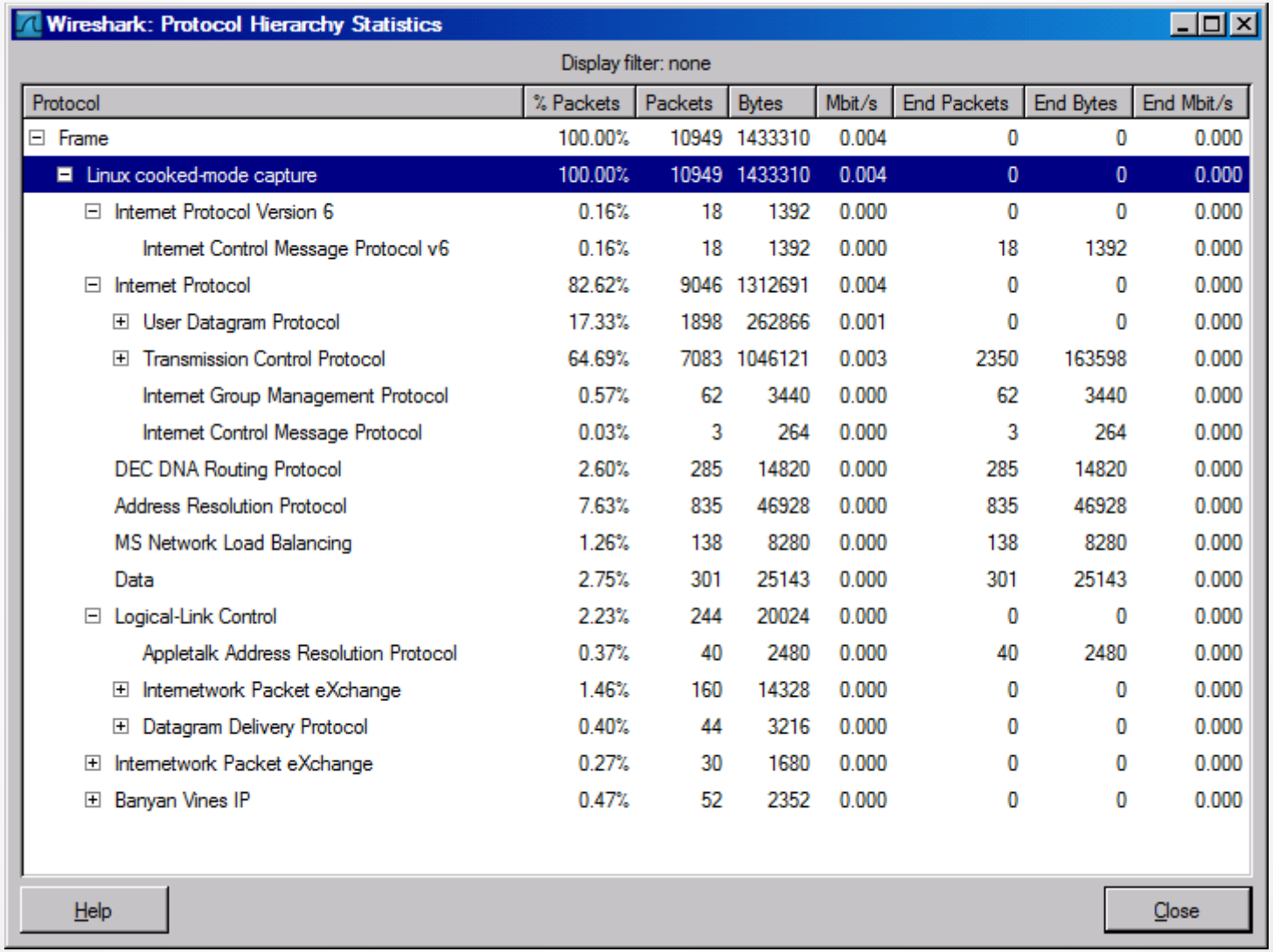
2). In choose "summary line" or "detail"



Protocol Hierarchy

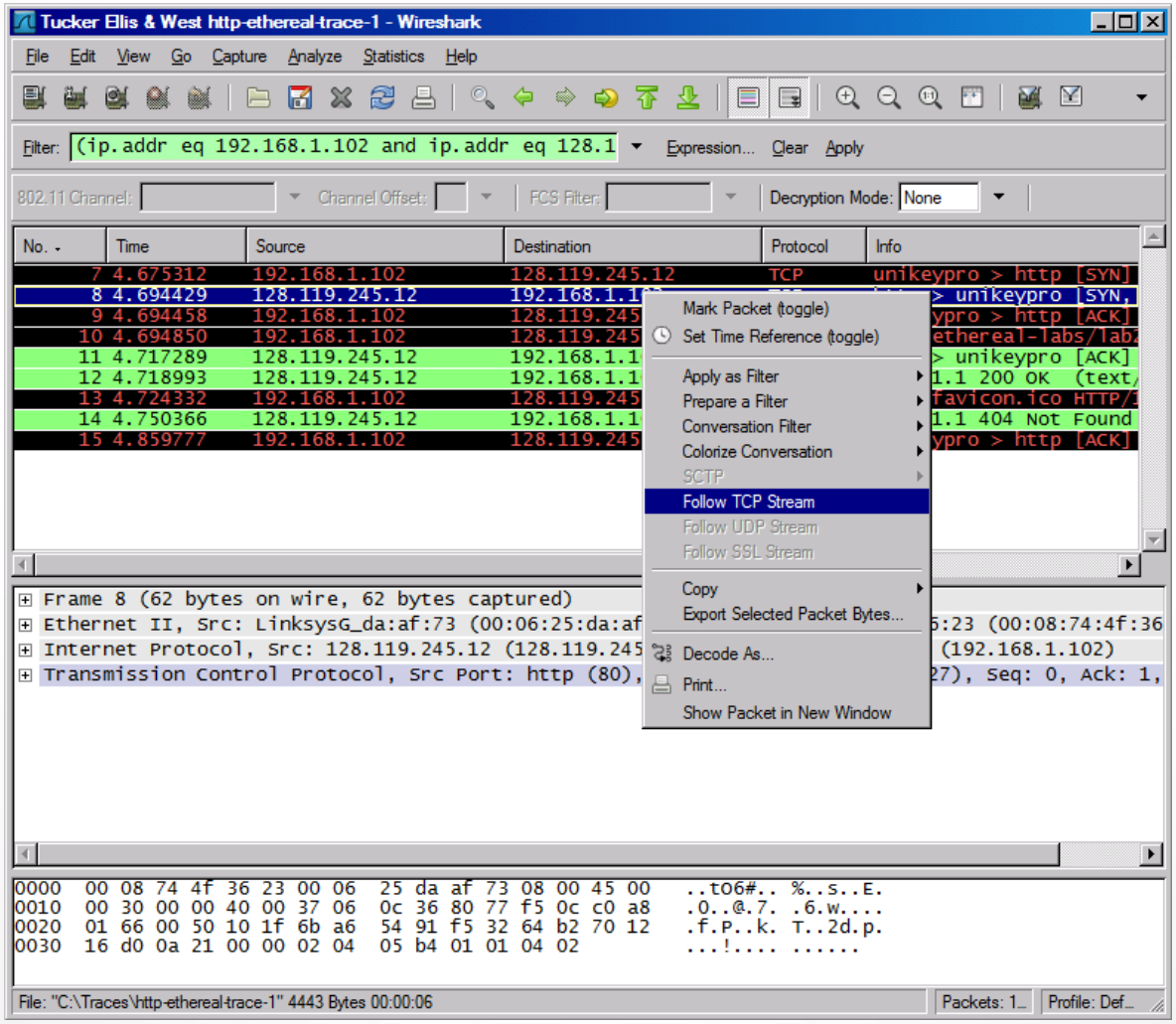


Protocol Hierarchy



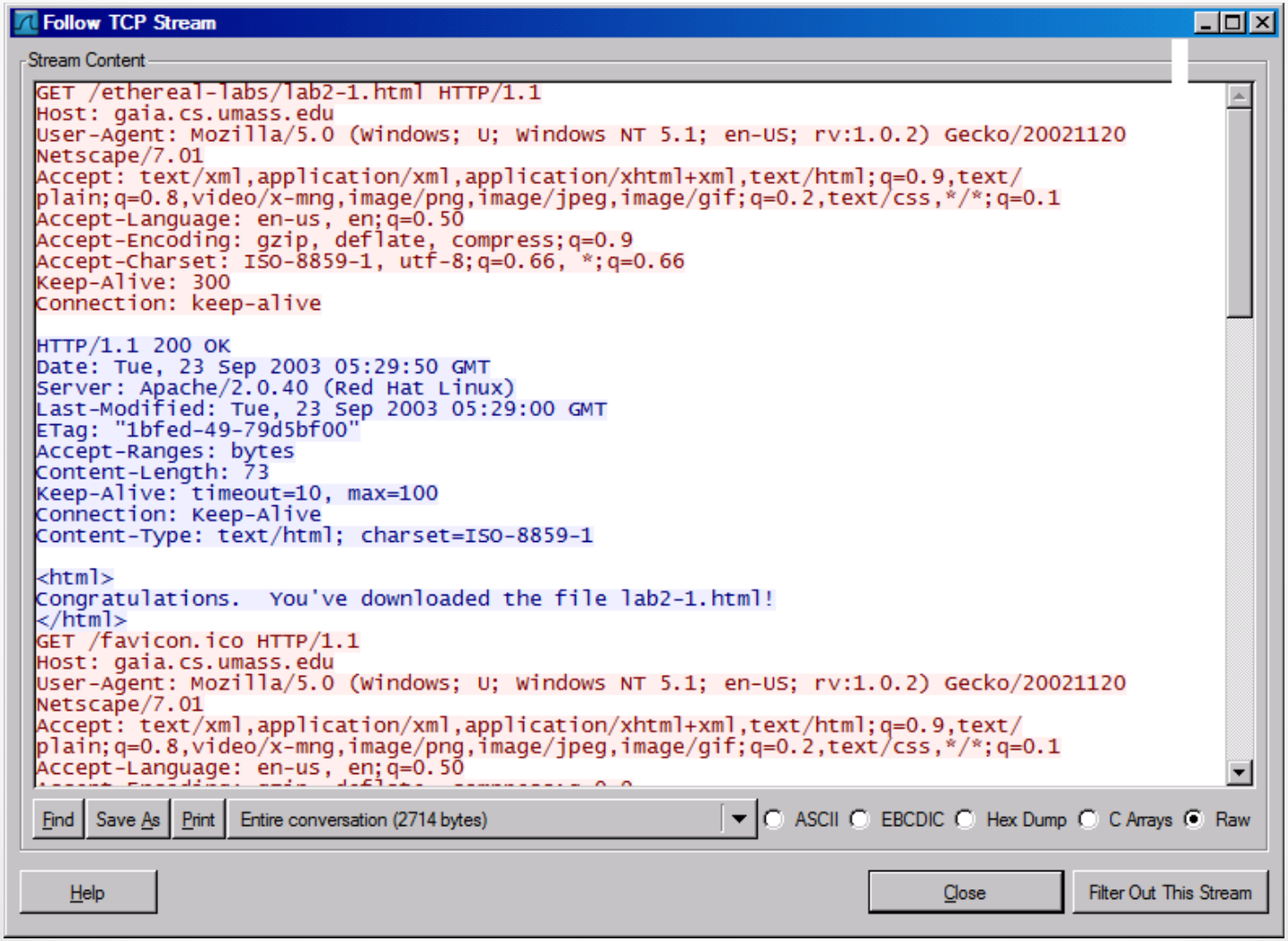
Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00%	10949	1433310	0.004	0	0	0.000
Linux cooked-mode capture	100.00%	10949	1433310	0.004	0	0	0.000
Internet Protocol Version 6	0.16%	18	1392	0.000	0	0	0.000
Internet Control Message Protocol v6	0.16%	18	1392	0.000	18	1392	0.000
Internet Protocol	82.62%	9046	1312691	0.004	0	0	0.000
User Datagram Protocol	17.33%	1898	262866	0.001	0	0	0.000
Transmission Control Protocol	64.69%	7083	1046121	0.003	2350	163598	0.000
Internet Group Management Protocol	0.57%	62	3440	0.000	62	3440	0.000
Internet Control Message Protocol	0.03%	3	264	0.000	3	264	0.000
DEC DNA Routing Protocol	2.60%	285	14820	0.000	285	14820	0.000
Address Resolution Protocol	7.63%	835	46928	0.000	835	46928	0.000
MS Network Load Balancing	1.26%	138	8280	0.000	138	8280	0.000
Data	2.75%	301	25143	0.000	301	25143	0.000
Logical-Link Control	2.23%	244	20024	0.000	0	0	0.000
Appletalk Address Resolution Protocol	0.37%	40	2480	0.000	40	2480	0.000
Internetwork Packet eXchange	1.46%	160	14328	0.000	0	0	0.000
Datagram Delivery Protocol	0.40%	44	3216	0.000	0	0	0.000
Internetwork Packet eXchange	0.27%	30	1680	0.000	0	0	0.000
Banyan Vines IP	0.47%	52	2352	0.000	0	0	0.000

Follow TCP Stream



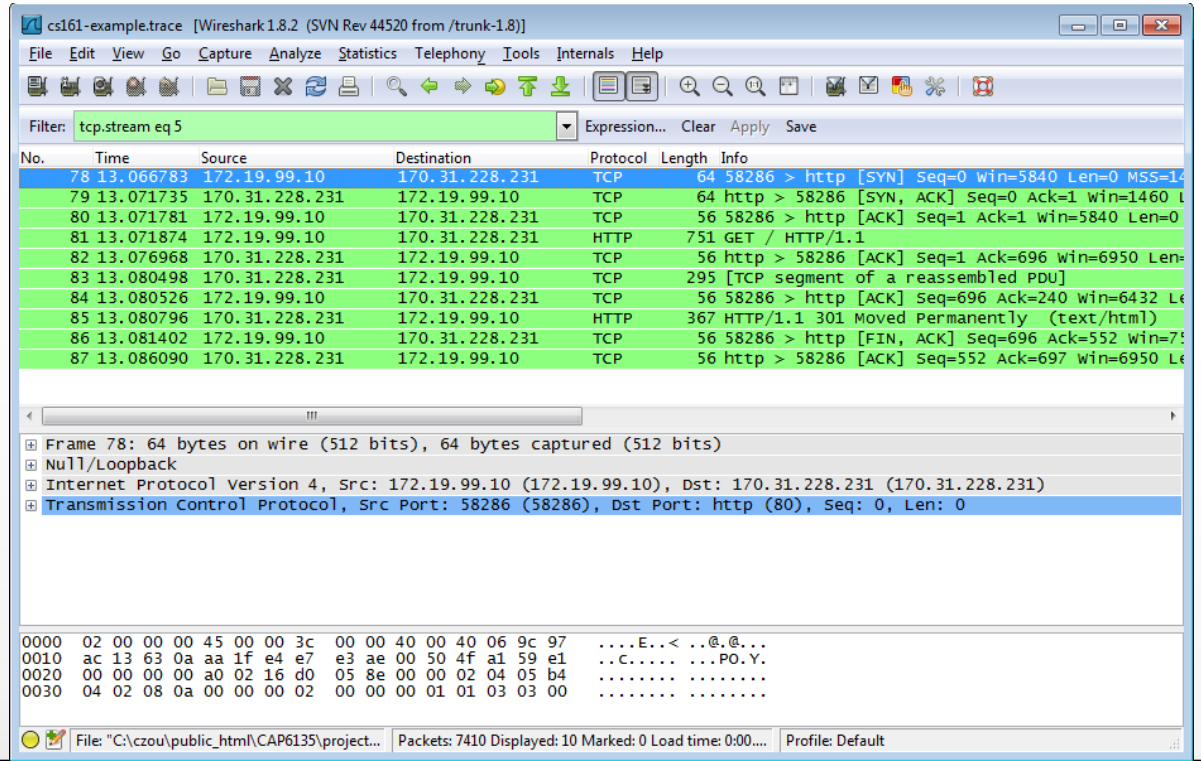
Follow TCP Stream

red - stuff you sent **blue - stuff you get**

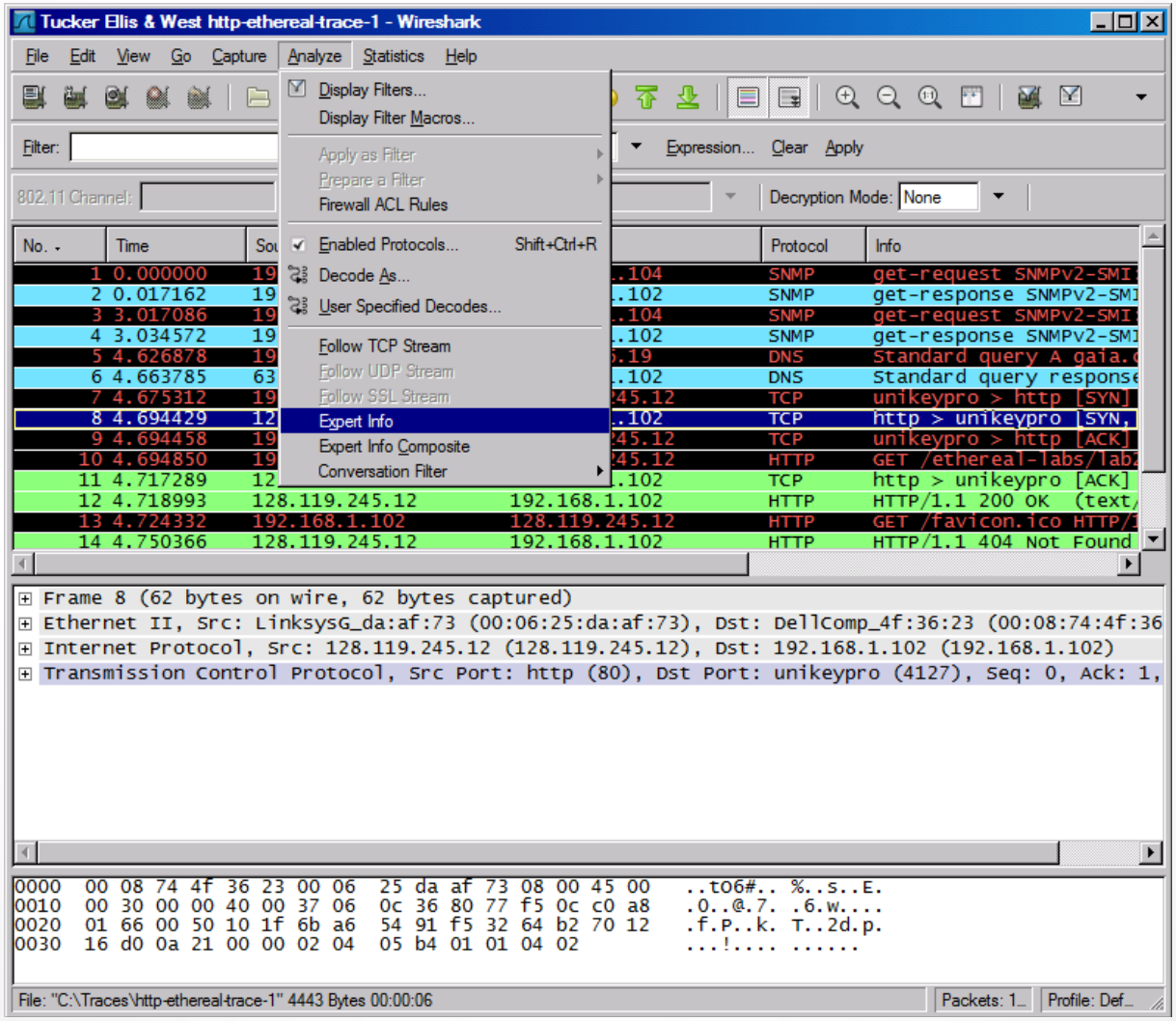


Filter out/in Single TCP Stream

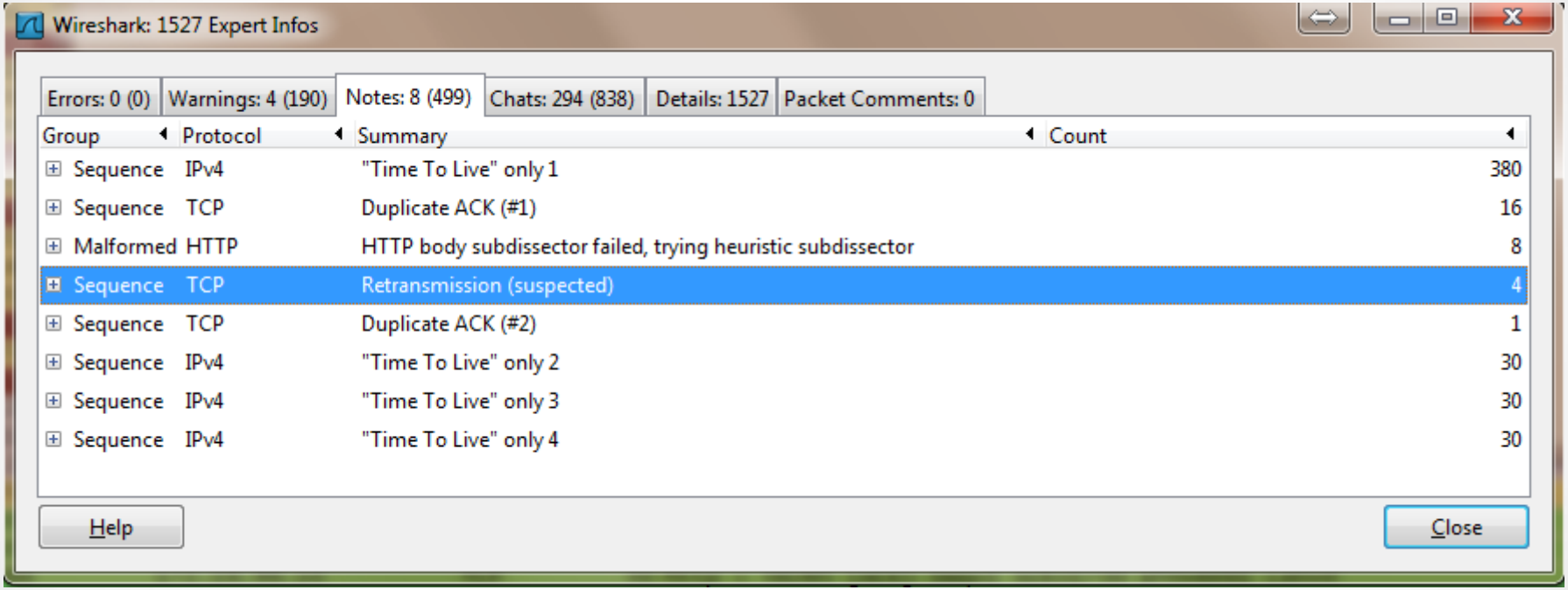
- When click “filter out this TCP stream” in previous page’s box, new filter string will contain like:
  - http and !(tcp.stream eq 5)
- So, if you use “tcp.stream eq 5” as filter string, you keep this HTTP session



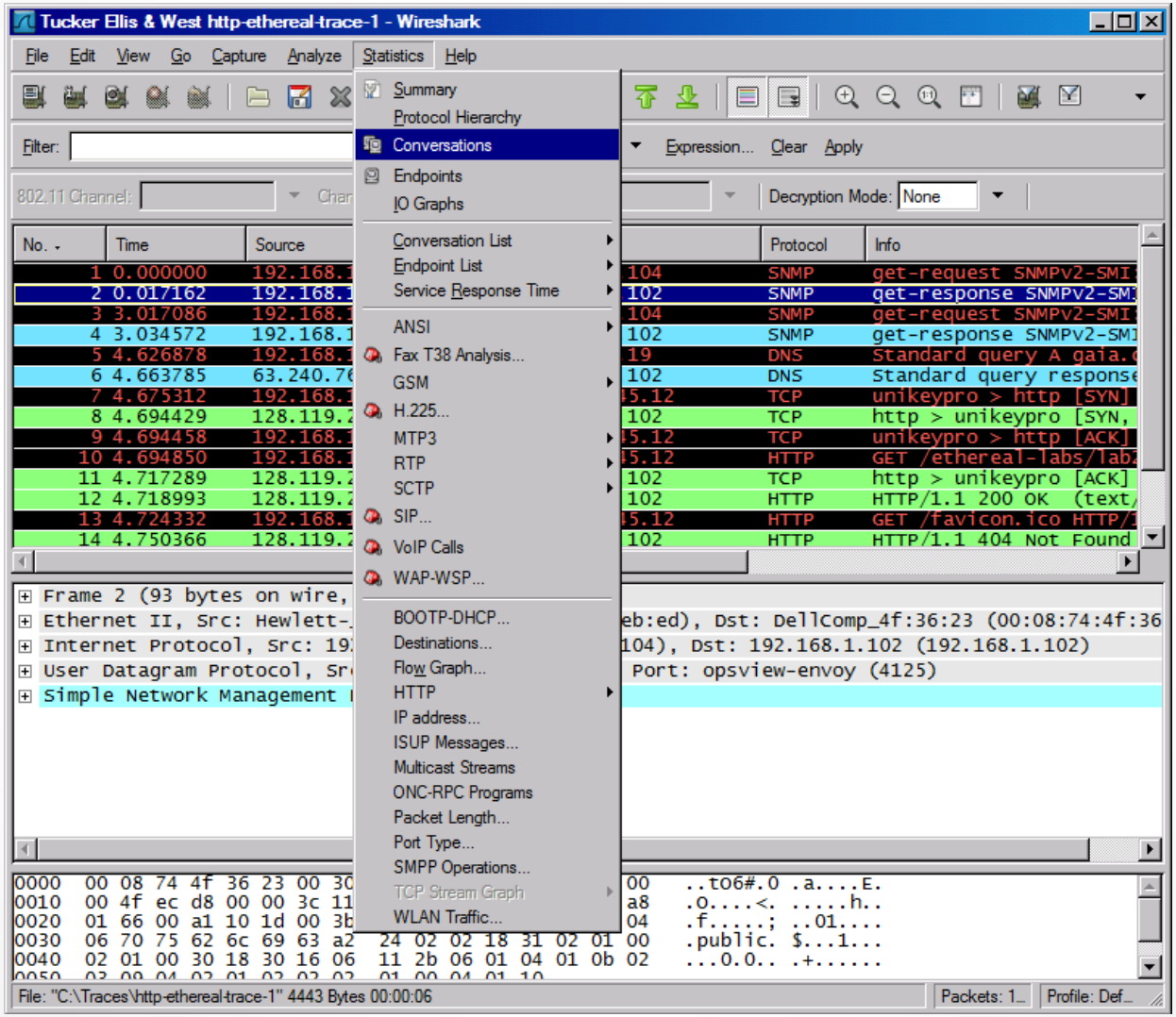
Expert Info


Tucker Ellis & West http-ethereal-trace-1 - Wireshark
File Edit View Go Capture Analyze Statistics Help
Display Filters...
Display Filter Macros...
Apply as Filter
Prepare a Filter
Firewall ACL Rules
Enabled Protocols... Shift+Ctrl+R
Decode As...
User Specified Decodes...
Follow TCP Stream
Follow UDP Stream
Follow SSL Stream
Expert Info
Expert Info Composite
Conversation Filter
Filter:
Expression... Clear Apply
802.11 Channel:
Decryption Mode: None
No. Time Protocol Info
1 0.000000 SNMP get-request SNMPv2-SMI
2 0.017162 SNMP get-response SNMPv2-SMI
3 3.017086 SNMP get-request SNMPv2-SMI
4 3.034572 SNMP get-response SNMPv2-SMI
5 4.626878 DNS Standard query A gaia.
6 4.663785 DNS Standard query response
7 4.675312 TCP unikeypro > http [SYN]
8 4.694429 TCP http > unikeypro [SYN,
9 4.694458 TCP unikeypro > http [ACK]
10 4.694850 HTTP GET /ethereal-labs/lab
11 4.717289 TCP http > unikeypro [ACK]
12 4.718993 128.119.245.12 192.168.1.102 HTTP HTTP/1.1 200 OK (text/
13 4.724332 192.168.1.102 128.119.245.12 HTTP GET /favicon.ico HTTP/1
14 4.750366 128.119.245.12 192.168.1.102 HTTP HTTP/1.1 404 Not Found
Frame 8 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: DellComp_4f:36:23 (00:08:74:4f:36
Internet Protocol, Src: 128.119.245.12 (128.119.245.12), Dst: 192.168.1.102 (192.168.1.102)
Transmission Control Protocol, Src Port: http (80), Dst Port: unikeypro (4127), Seq: 0, Ack: 1,
0000 00 08 74 4f 36 23 00 06 25 da af 73 08 00 45 00 ..tO6#.. %..s..E.
0010 00 30 00 00 40 00 37 06 0c 36 80 77 f5 0c c0 a8 .0..@.7. .6.w....
0020 01 66 00 50 10 1f 6b a6 54 91 f5 32 64 b2 70 12 .f.P..k. T..2d.p.
0030 16 d0 0a 21 00 00 02 04 05 b4 01 01 04 02 ...!.... ......
File: "C:\Traces\http-ethereal-trace-1" 4443 Bytes 00:00:06
Packets: 1...
Profile: Def...

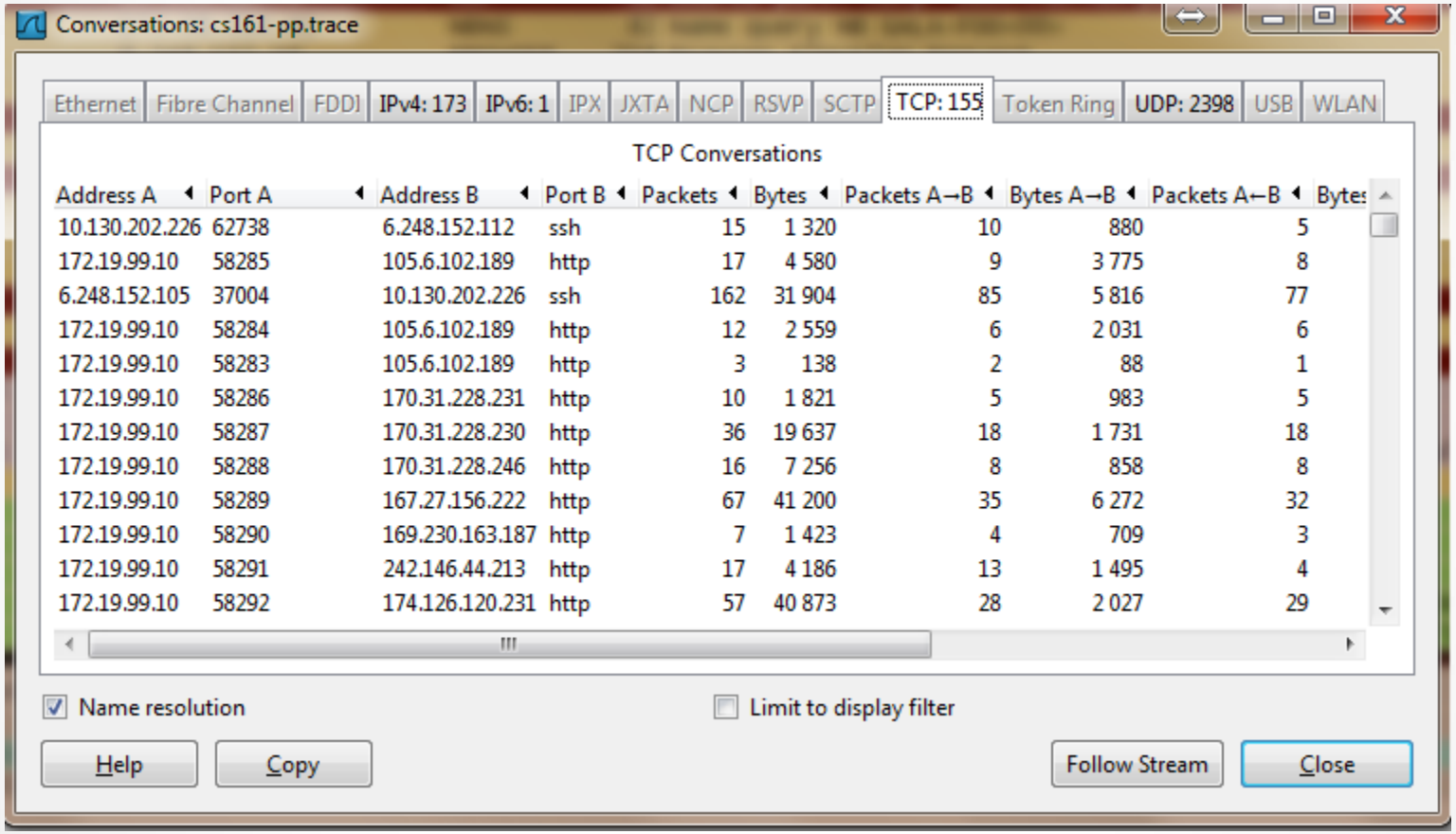
Expert Info


Wireshark: 1527 Expert Infos
Errors: 0 (0)
Warnings: 4 (190)
Notes: 8 (499)
Chats: 294 (838)
Details: 1527
Packet Comments: 0
Group
Protocol
Summary
Count
Sequence IPv4 "Time To Live" only 1 380
Sequence TCP Duplicate ACK (#1) 16
Malformed HTTP HTTP body subdissector failed, trying heuristic subdissector 8
Sequence TCP Retransmission (suspected) 4
Sequence TCP Duplicate ACK (#2) 1
Sequence IPv4 "Time To Live" only 2 30
Sequence IPv4 "Time To Live" only 3 30
Sequence IPv4 "Time To Live" only 4 30
Help
Close

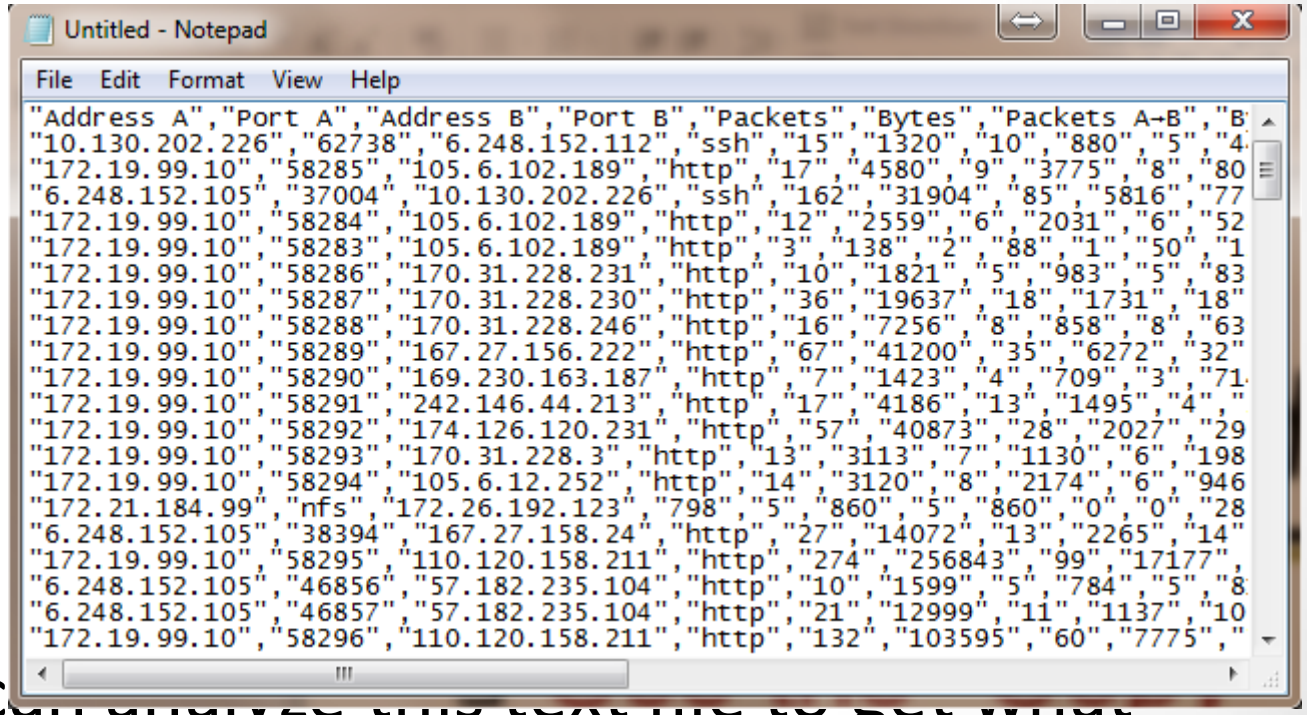
Conversations



Conversations



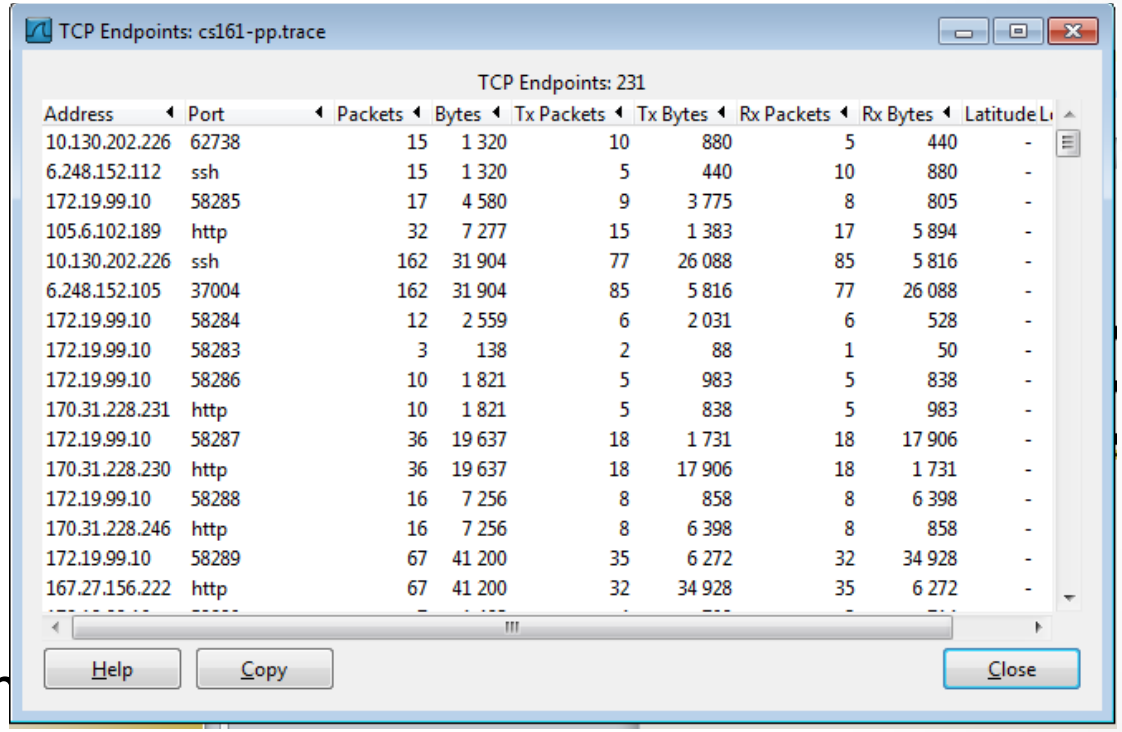
- Use the “Copy” button to copy all text into clipboard



- Then, you can analyze this text file to get what statistics you want

Find EndPoint Statistics

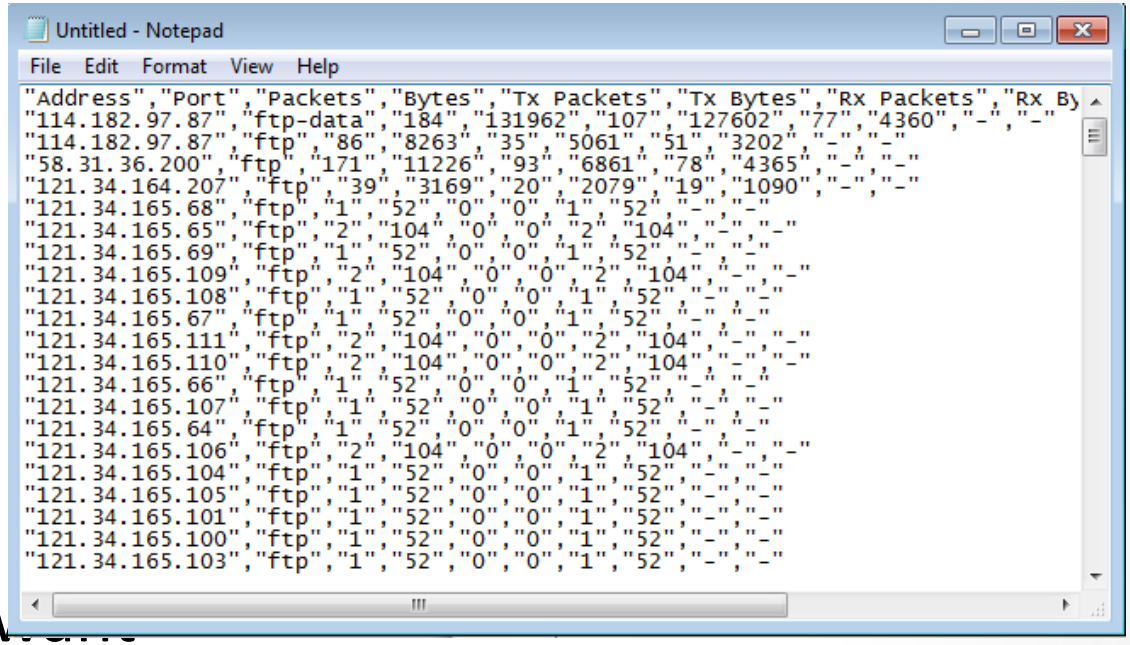
- Menu “statistics” → “endpoint list” → “TCP”



- You can sort
- “Tx” : transm

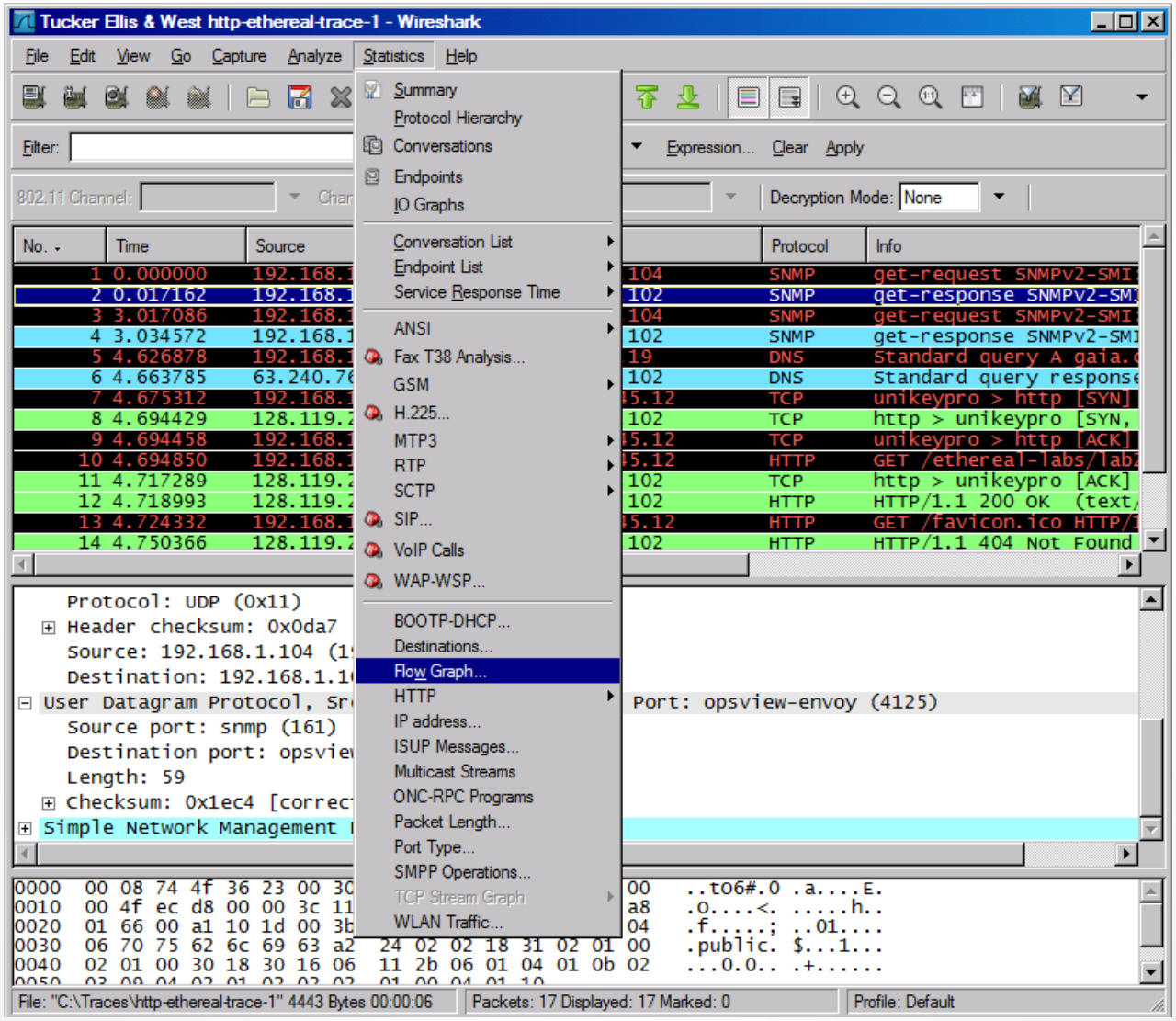
Find EndPoint Statistics

- Use the “Copy” button to copy all text into clipboard

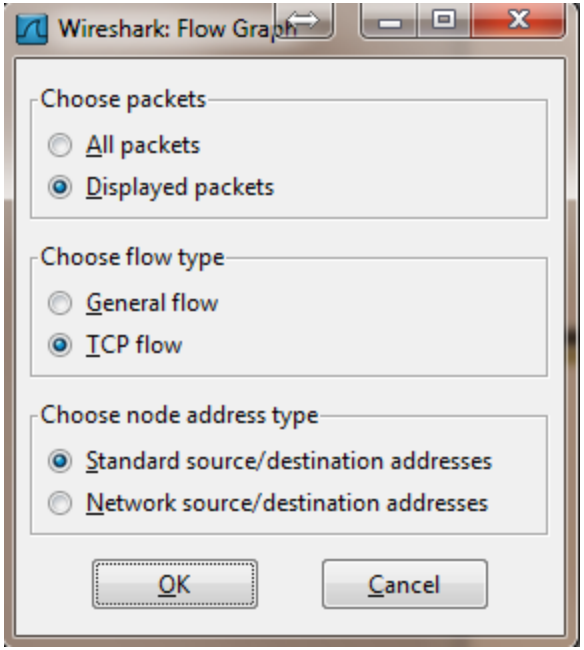


- Then, you can
  statistics you

Flow Graphs

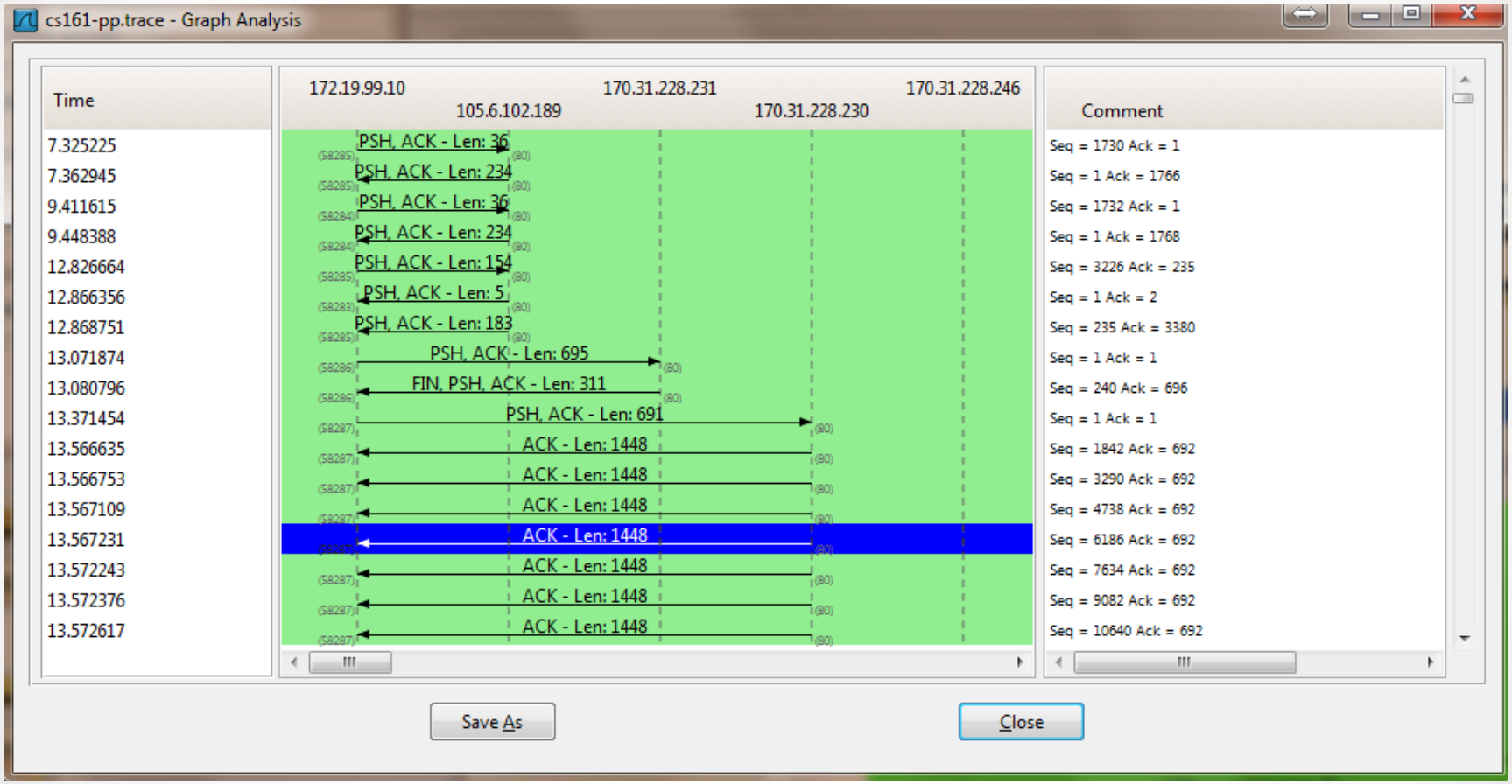

Tucker Ellis & West http-ethereal-trace-1 - Wireshark
File Edit View Go Capture Analyze Statistics Help
Summary
Protocol Hierarchy
Conversations
Endpoints
IO Graphs
Conversation List
Endpoint List
Service Response Time
ANSI
Fax T38 Analysis...
GSM
H.225...
MTP3
RTP
SCTP
SIP...
VoIP Calls
WAP-WSP...
BOOTP-DHCP...
Destinations...
Flow Graph...
HTTP
IP address...
ISUP Messages...
Multicast Streams
ONC-RPC Programs
Packet Length...
Port Type...
SMPP Operations...
TCP Stream Graph
WLAN Traffic...
Filter:
Expression... Clear Apply
Decryption Mode: None
No. Time Source Protocol Info
Protocol: UDP (0x11)
Header checksum: 0x0da7
Source: 192.168.1.104
Destination: 192.168.1.1
User Datagram Protocol, Sr
Source port: snmp (161)
Destination port: opsvie
Length: 59
Checksum: 0x1ec4 [correc
Simple Network Management
Port: opsview-envoy (4125)
File: "C:\Traces\http-ethereal-trace-1" 4443 Bytes 00:00:06
Packets: 17 Displayed: 17 Marked: 0
Profile: Default

Flow Graphs

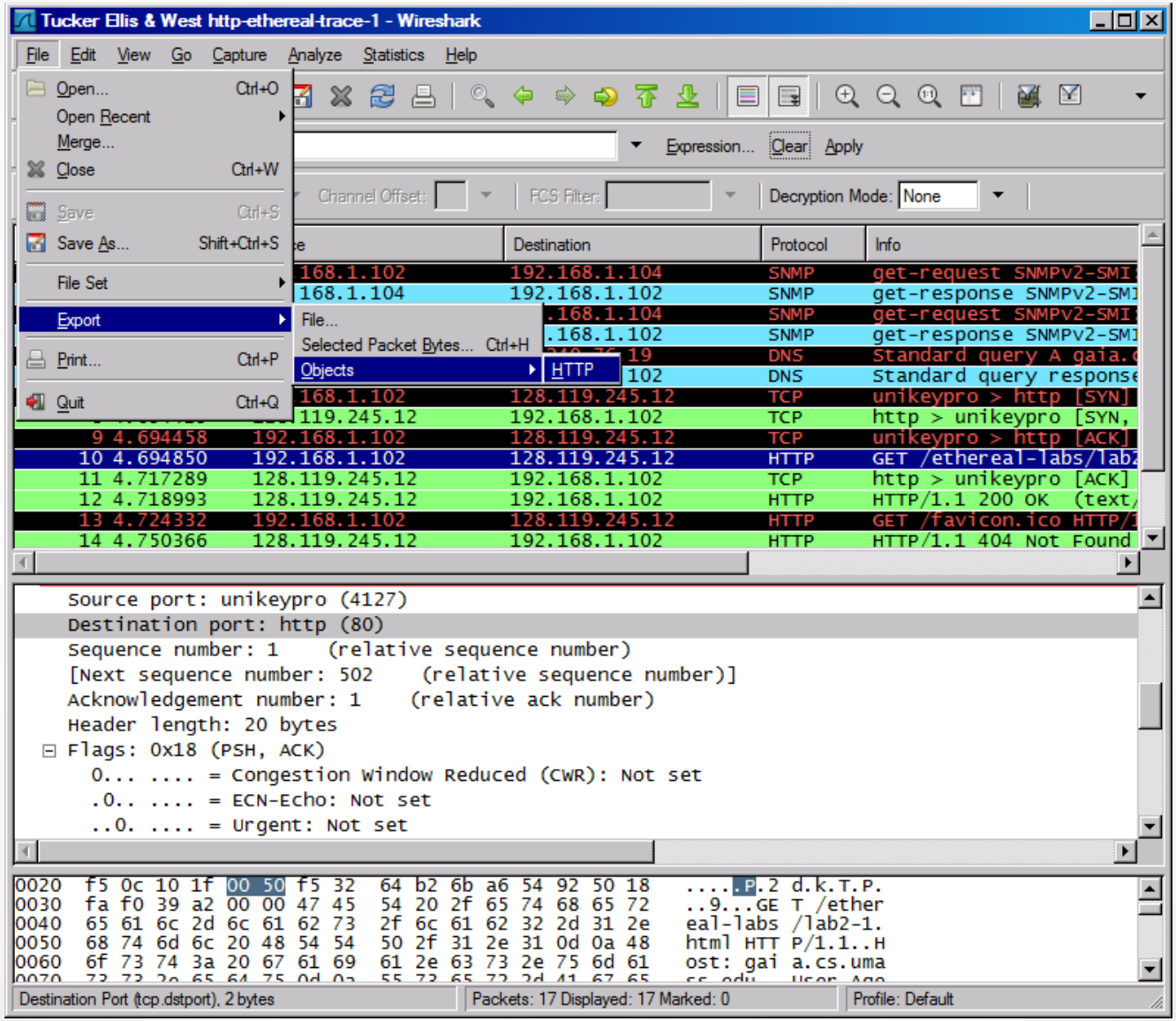


- The "displayed packet" option could let you only Show the flow of packets shown up
  for example, only display http traffic, then show The flow to analyze

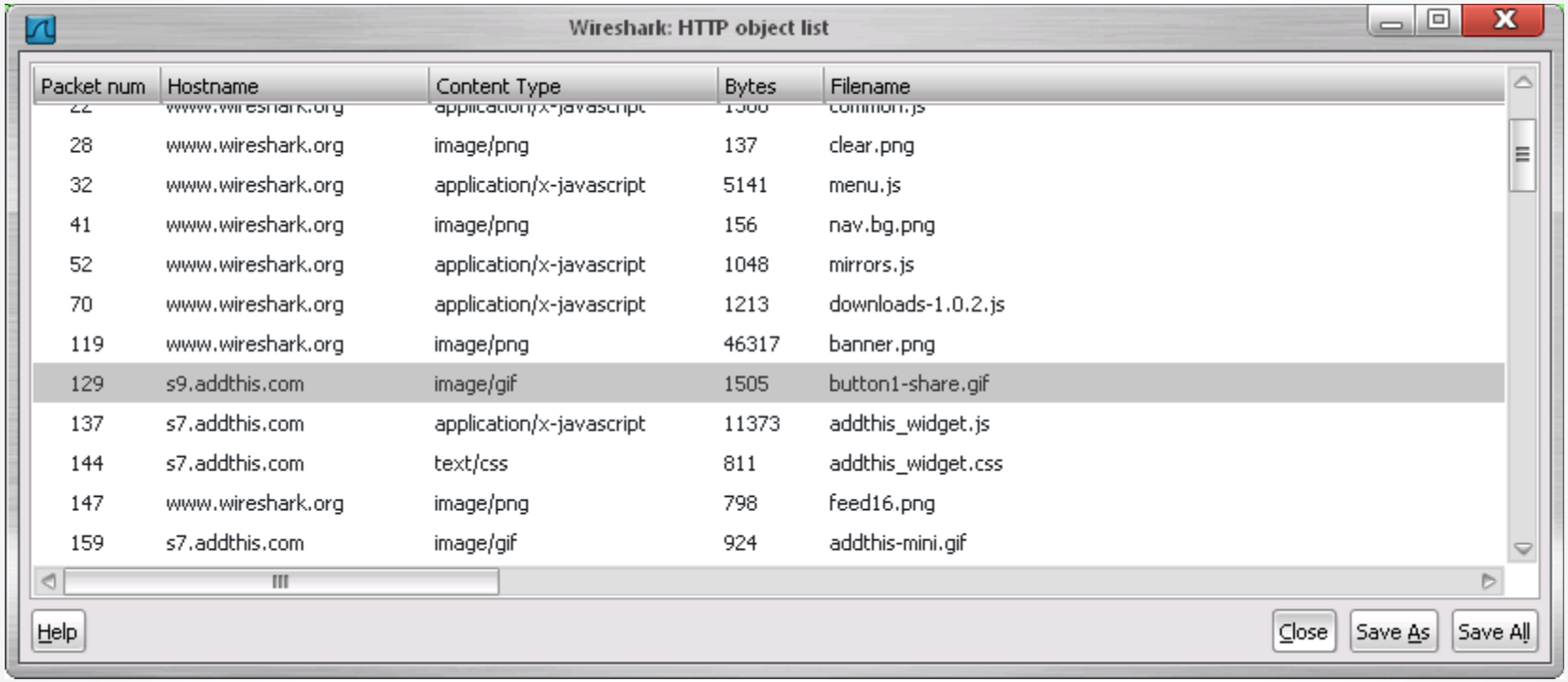
Flow Graphs


cs161-pp.trace - Graph Analysis
Time
7.325225
7.362945
9.411615
9.448388
12.826664
12.866356
12.868751
13.071874
13.080796
13.371454
13.566635
13.566753
13.567109
13.567231
13.572243
13.572376
13.572617
172.19.99.10
105.6.102.189
170.31.228.231
170.31.228.230
170.31.228.246
PSH, ACK - Len: 36
PSH, ACK - Len: 234
PSH, ACK - Len: 36
PSH, ACK - Len: 234
PSH, ACK - Len: 154
PSH, ACK - Len: 5
PSH, ACK - Len: 183
PSH, ACK - Len: 695
FIN, PSH, ACK - Len: 311
PSH, ACK - Len: 691
ACK - Len: 1448
ACK - Len: 1448
ACK - Len: 1448
ACK - Len: 1448
ACK - Len: 1448
ACK - Len: 1448
ACK - Len: 1448
Comment
Seq = 1730 Ack = 1
Seq = 1 Ack = 1766
Seq = 1732 Ack = 1
Seq = 1 Ack = 1768
Seq = 3226 Ack = 235
Seq = 1 Ack = 2
Seq = 235 Ack = 3380
Seq = 1 Ack = 1
Seq = 240 Ack = 696
Seq = 1 Ack = 1
Seq = 1842 Ack = 692
Seq = 3290 Ack = 692
Seq = 4738 Ack = 692
Seq = 6186 Ack = 692
Seq = 7634 Ack = 692
Seq = 9082 Ack = 692
Seq = 10640 Ack = 692
Save As
Close

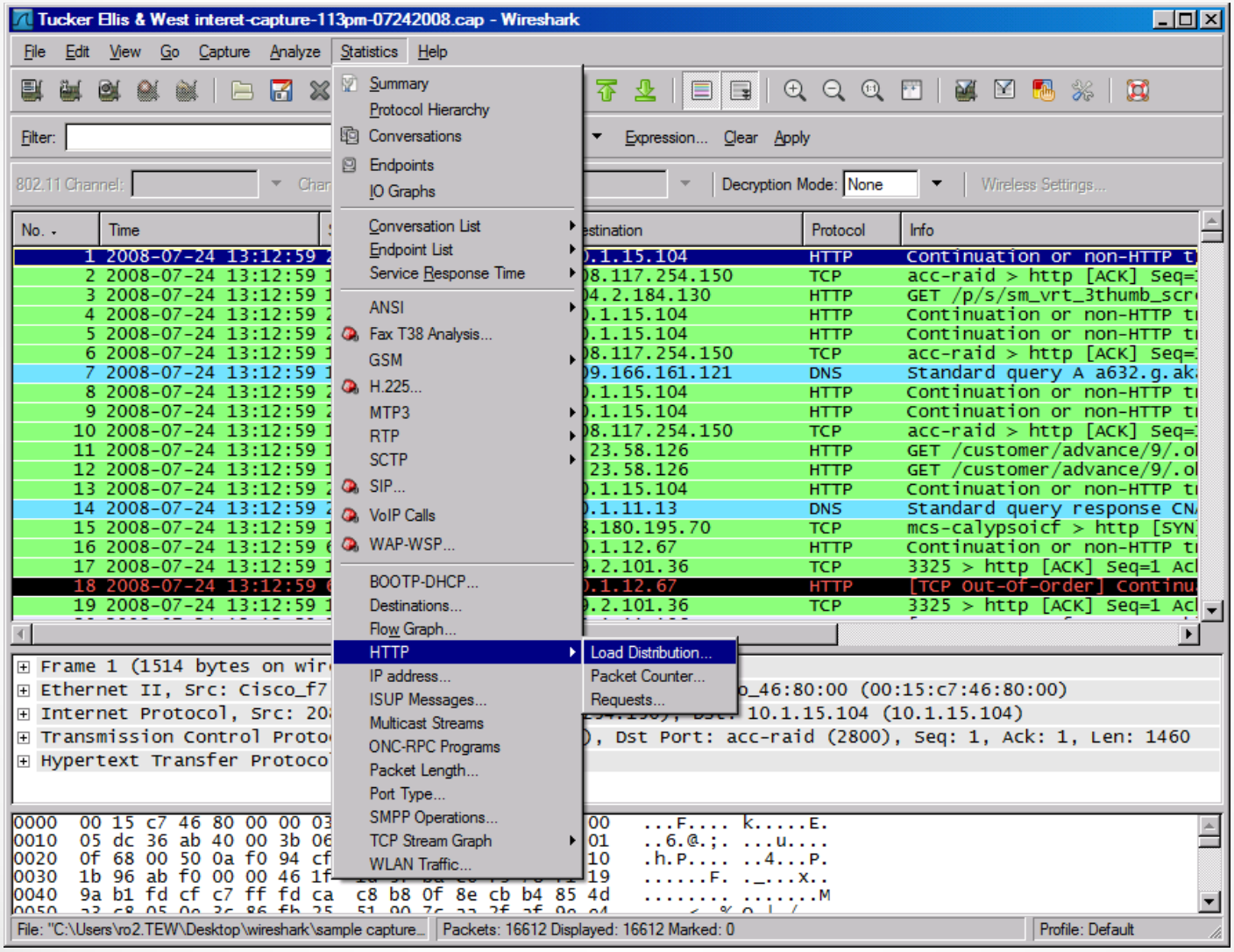
Export HTTP


Tucker Ellis & West http-ethereal-trace-1 - Wireshark
File Edit View Go Capture Analyze Statistics Help
Open... Ctrl+O
Open Recent
Merge...
Close Ctrl+W
Save Ctrl+S
Save As... Shift+Ctrl+S
File Set
Export
Print... Ctrl+P
Quit Ctrl+Q
File...
Selected Packet Bytes... Ctrl+H
Objects
HTTP
Expression... Clear Apply
Channel Offset: FCS Filter: Decryption Mode: None
Destination Protocol Info
168.1.102 192.168.1.104 SNMP get-request SNMPv2-SMI
168.1.104 192.168.1.102 SNMP get-response SNMPv2-SMI
.168.1.104 SNMP get-request SNMPv2-SMI
.168.1.102 SNMP get-response SNMPv2-SMI
19 DNS Standard query A gaia.
102 DNS Standard query response
168.1.102 128.119.245.12 TCP unikeypro > http [SYN]
119.245.12 192.168.1.102 TCP http > unikeypro [SYN,
9 4.694458 192.168.1.102 128.119.245.12 TCP unikeypro > http [ACK]
10 4.694850 192.168.1.102 128.119.245.12 HTTP GET /ethereal-labs/lab
11 4.717289 128.119.245.12 192.168.1.102 TCP http > unikeypro [ACK]
12 4.718993 128.119.245.12 192.168.1.102 HTTP HTTP/1.1 200 OK (text/
13 4.724332 192.168.1.102 128.119.245.12 HTTP GET /favicon.ico HTTP/1
14 4.750366 128.119.245.12 192.168.1.102 HTTP HTTP/1.1 404 Not Found
Source port: unikeypro (4127)
Destination port: http (80)
Sequence number: 1 (relative sequence number)
[Next sequence number: 502 (relative sequence number)]
Acknowledgement number: 1 (relative ack number)
Header length: 20 bytes
Flags: 0x18 (PSH, ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
0020 f5 0c 10 1f 00 50 f5 32 64 b2 6b a6 54 92 50 18 .....P.2 d.k.T.P.
0030 fa f0 39 a2 00 00 47 45 54 20 2f 65 74 68 65 72 ..9...GE T /ether
0040 65 61 6c 2d 6c 61 62 73 2f 6c 61 62 32 2d 31 2e eal-labs /lab2-1.
0050 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 0a 48 html HTT P/1.1..H
0060 6f 73 74 3a 20 67 61 69 61 2e 63 73 2e 75 6d 61 ost: gai a.cs.uma
Destination Port (tcp.dstport), 2 bytes
Packets: 17 Displayed: 17 Marked: 0
Profile: Default

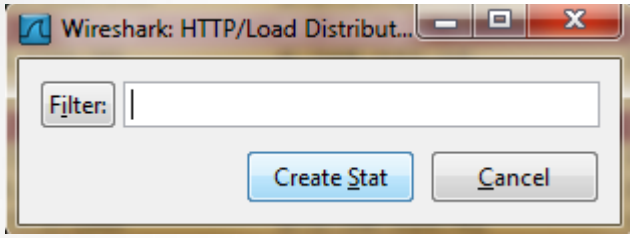
Export HTTP Objects



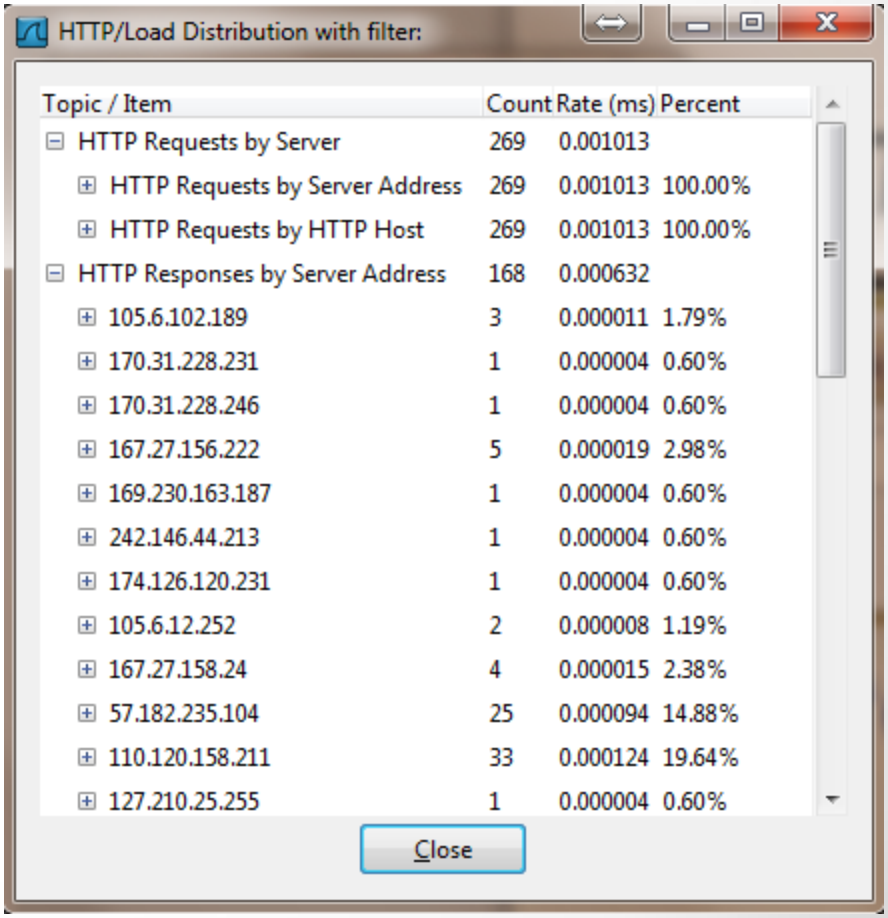
HTTP Analysis



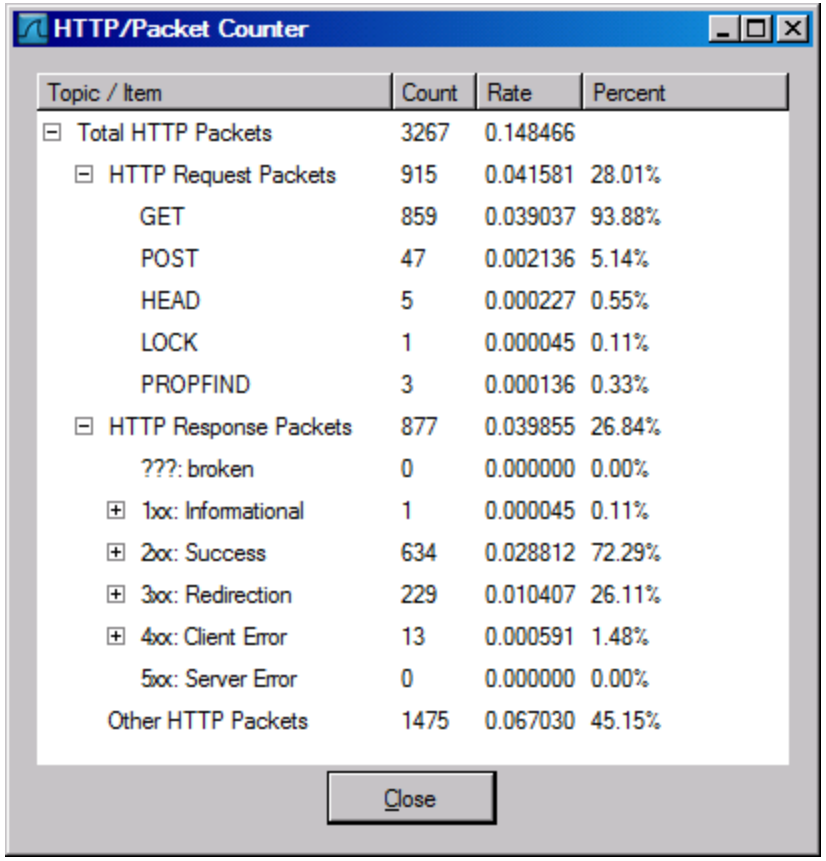
HTTP Analysis – Load Distribution



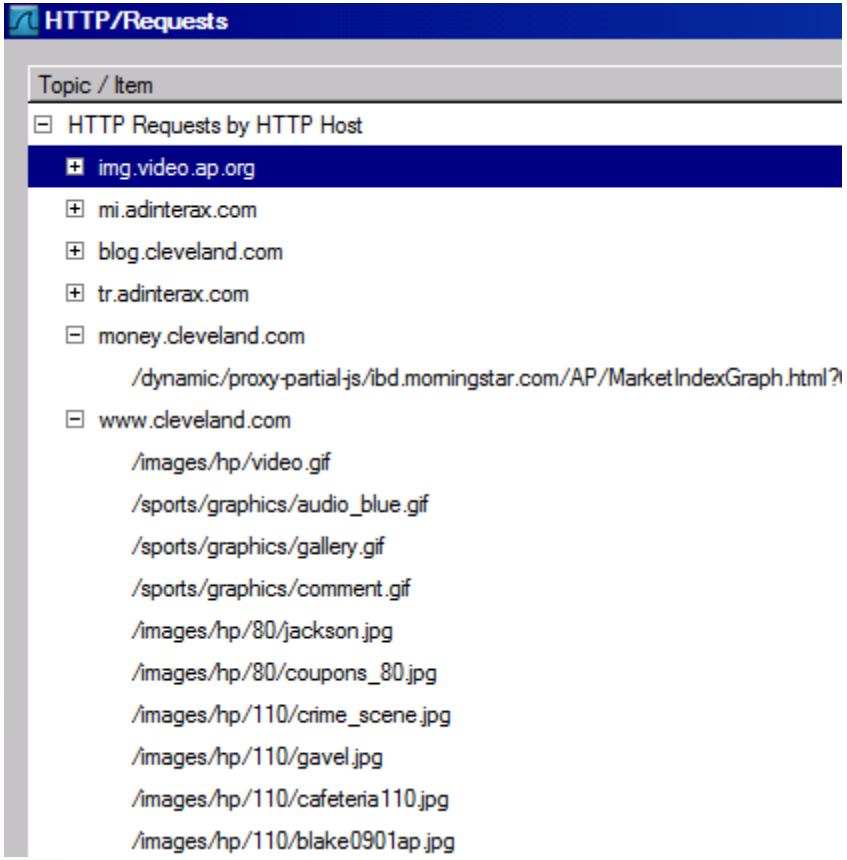
Click “Create Stat” button
You can add “filter” to only
Show selected traffic



HTTP Analysis – Packet Counter



HTTP Analysis – Requests



Basic usage of Grep

- Command-line text-search program in Linux
- Some useful usage:
  - Grep ‘word’ filename # find lines with ‘word’
  - Grep –v ‘word’ filename # find lines without ‘word’
  - Grep ‘^word’ filename # find lines beginning with ‘word’
  - Grep ‘word’ filename > file2 # output lines with ‘word’ to file2
  - ls -l | grep rwxrwxrwx # list files that have ‘rwxrwxrwx’ feature
  - grep '^[0-4]‘ filename # find lines beginning with any of the numbers from 0-4
  - Grep –c ‘word’ filename # find lines with ‘word’ and print out the number of these lines
  - Grep –i ‘word’ filename # find lines with ‘word’ regardless of case

- Many tutorials on grep online
  - http://www.cyberciti.biz/faq/howto-use-grep-command-in-linux-unix/
  - http://www.thegeekstuff.com/2009/03/15-practical-unix-grep-command-examples/

Спасибо за внимание!