

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

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 (Imctfy, 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

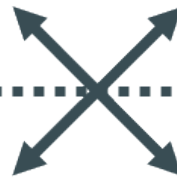
 Analytics DB
hadoop + hive + thrift + OpenJDK

 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

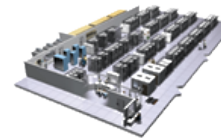
Customer Data Center



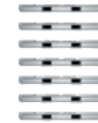
Public Cloud

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)

Multiplicity of methods for transporting/storing



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

Solution: Intermodal Shipping Container.....

Multiplicity of Goods



A standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery.

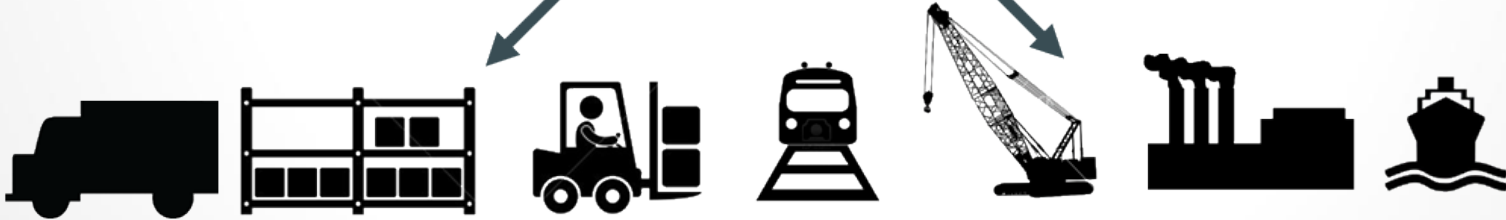
Do I worry about how goods interact (e.g. coffee beans next to spices)



...in between, can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another

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

Multiplicity of methods for transporting/storing



Docker is a Container System for Code.....

Multiplicity of Stacks

- Static website
- User DB
- Web frontend
- Queue
- Analytics DB

An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container...



Do services and apps interact appropriately?

Multiplicity of hardware environments

- Development VM
- QA server
- Customer Data Center
- Public Cloud
- Production Cluster
- Contributor's laptop

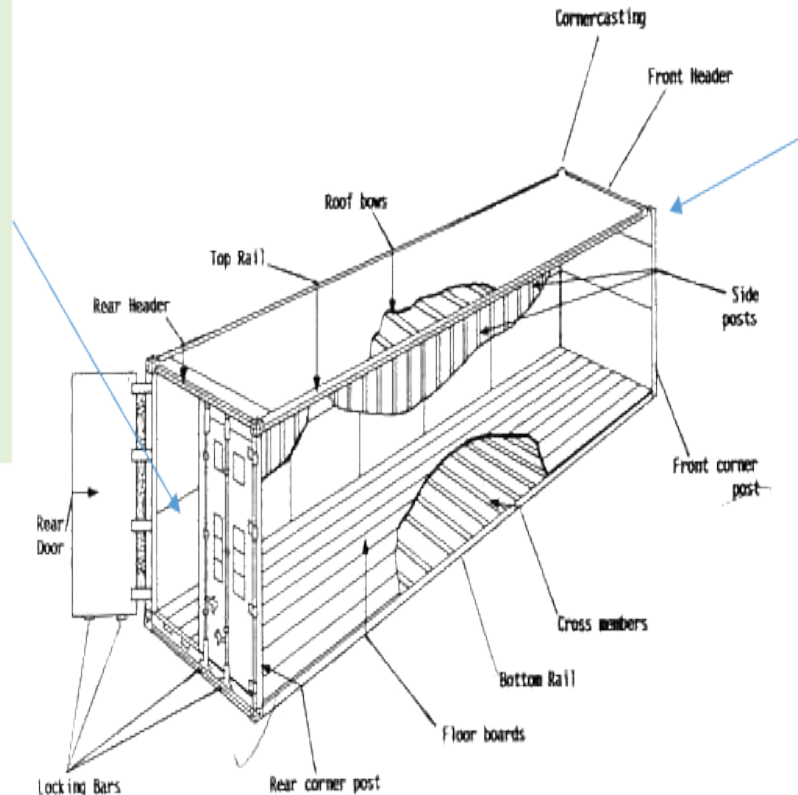
...that can be manipulated using standard operations and run consistently on virtually any hardware platform

Can I migrate smoothly and quickly

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



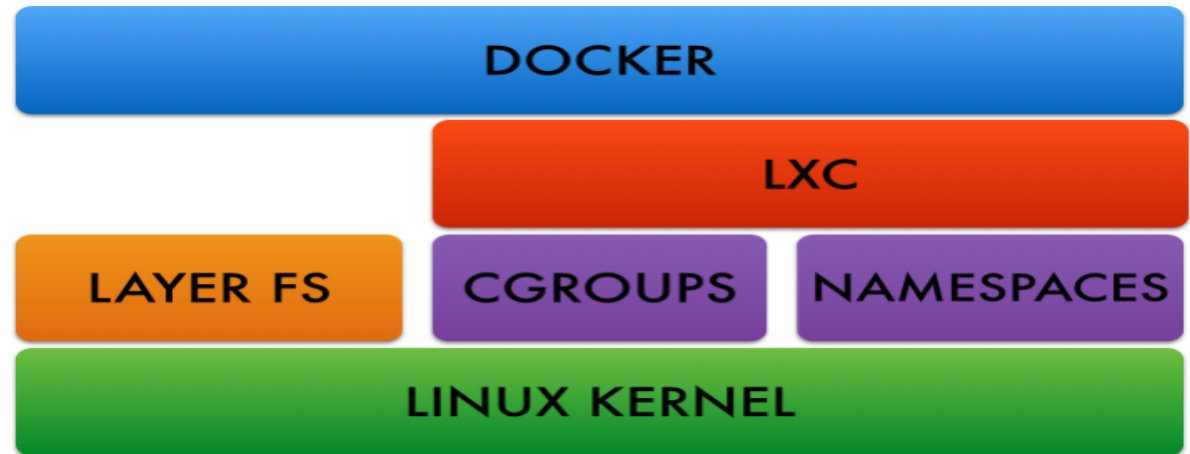
Major components of the container:

• 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

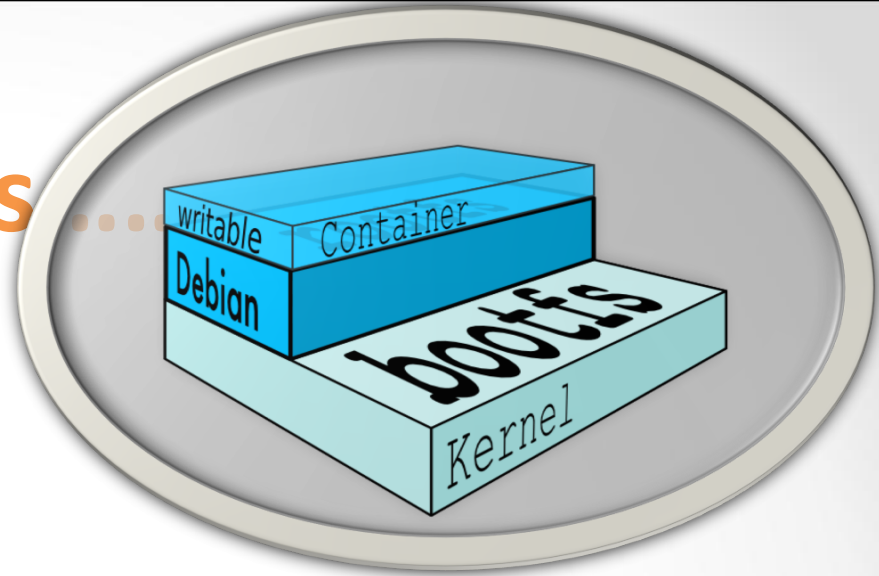
Docker Architecture.....

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

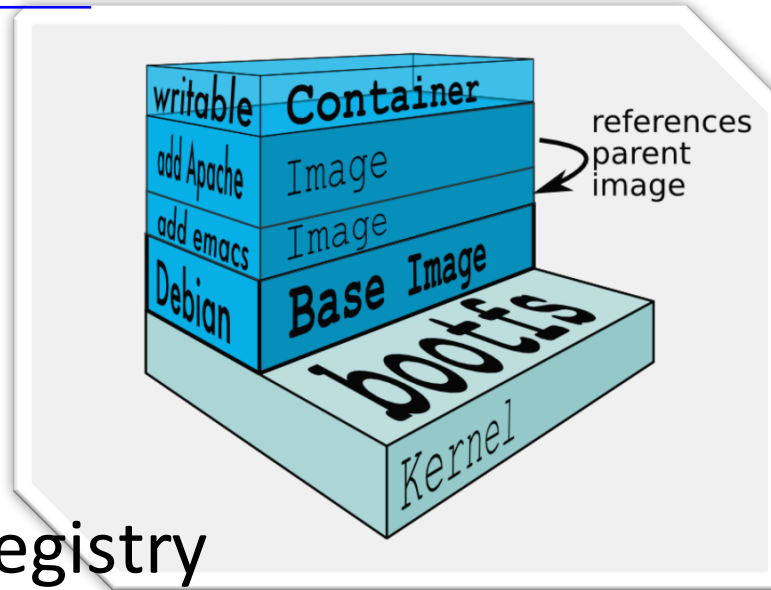


LINUX KERNEL

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

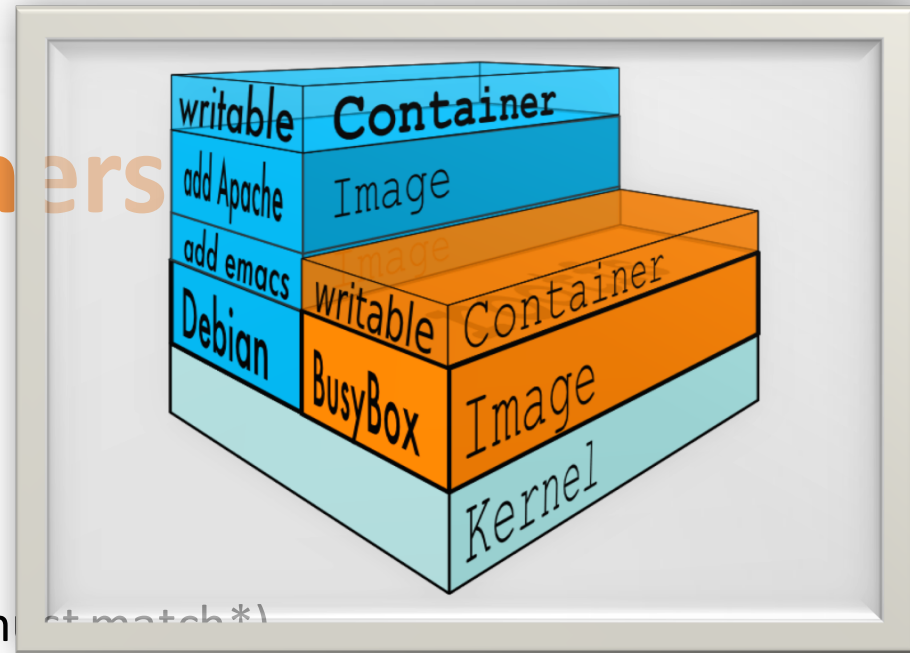


Docker Containers

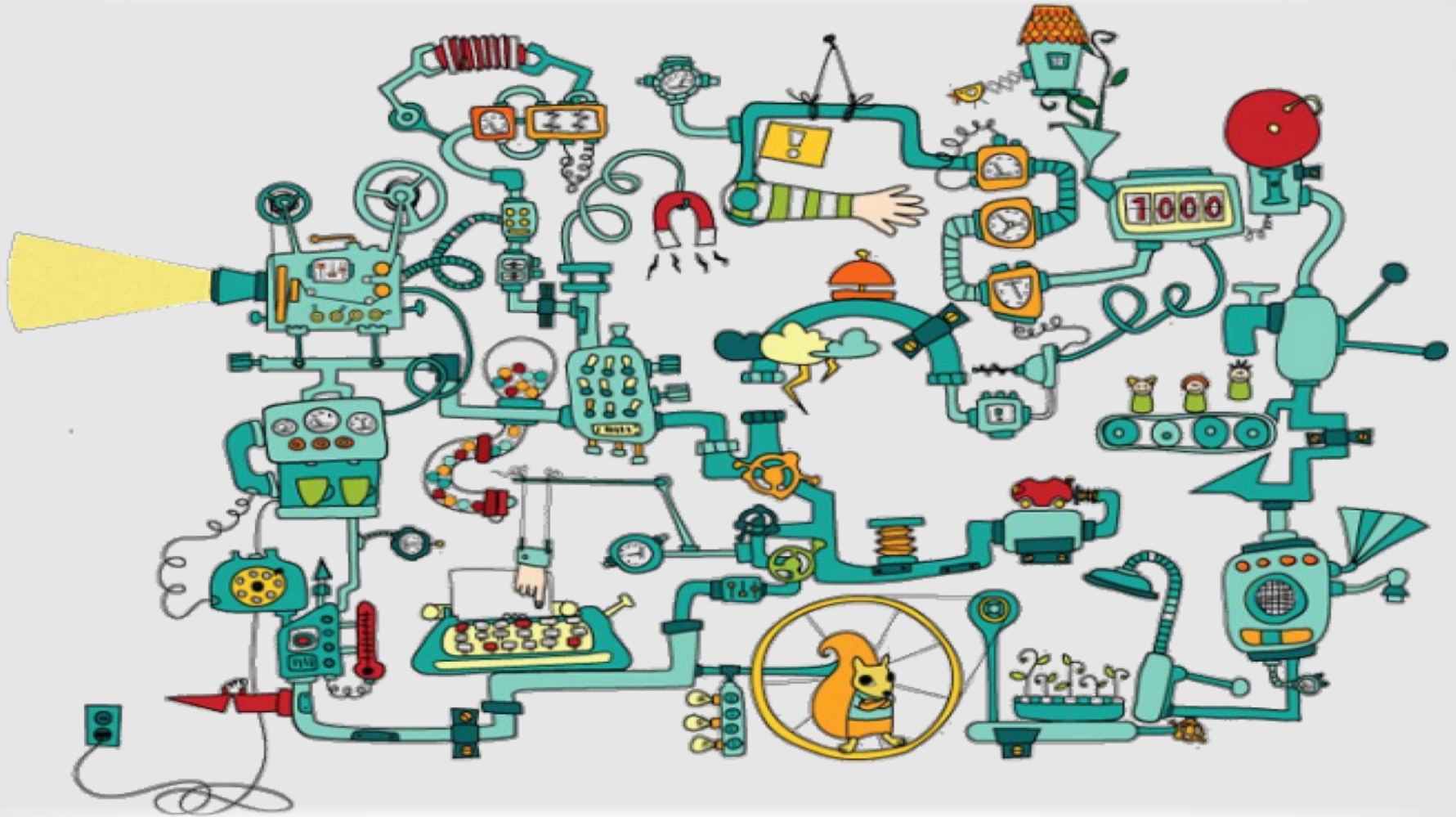
Units of software delivery (ship it!)

- 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

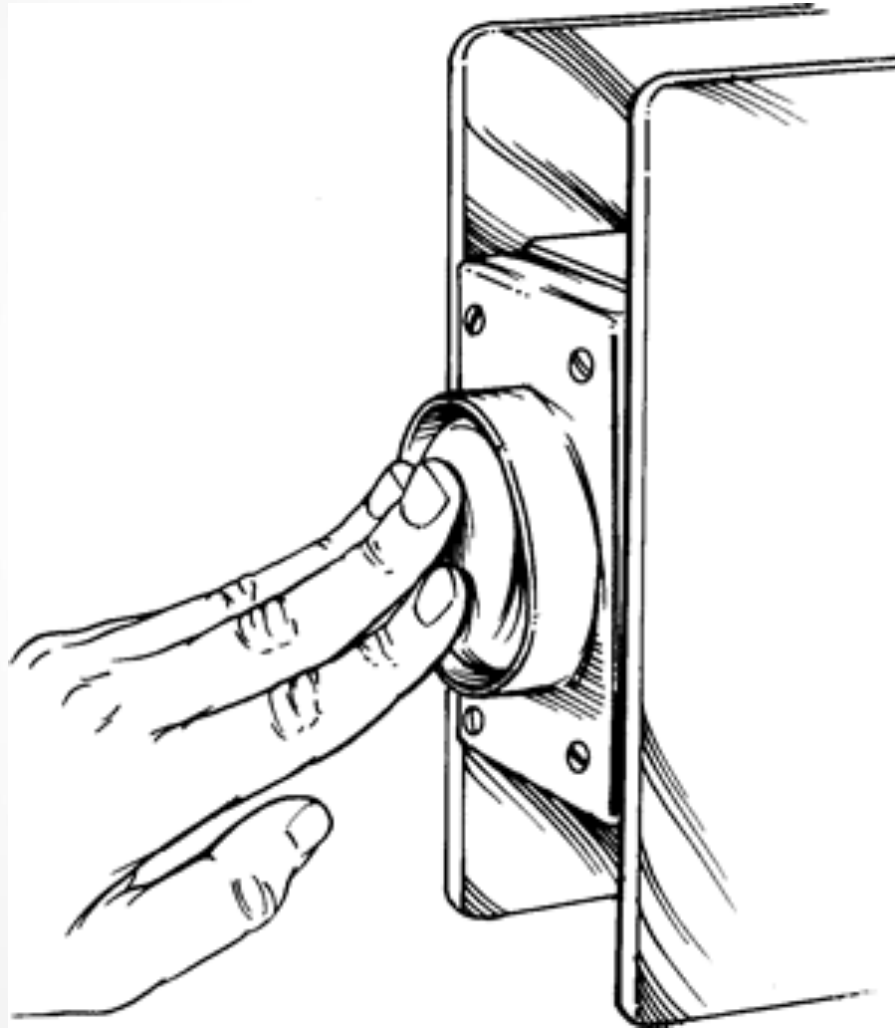
*Unless you emulate CPU with qemu and binfmt



Containers before Docker.....



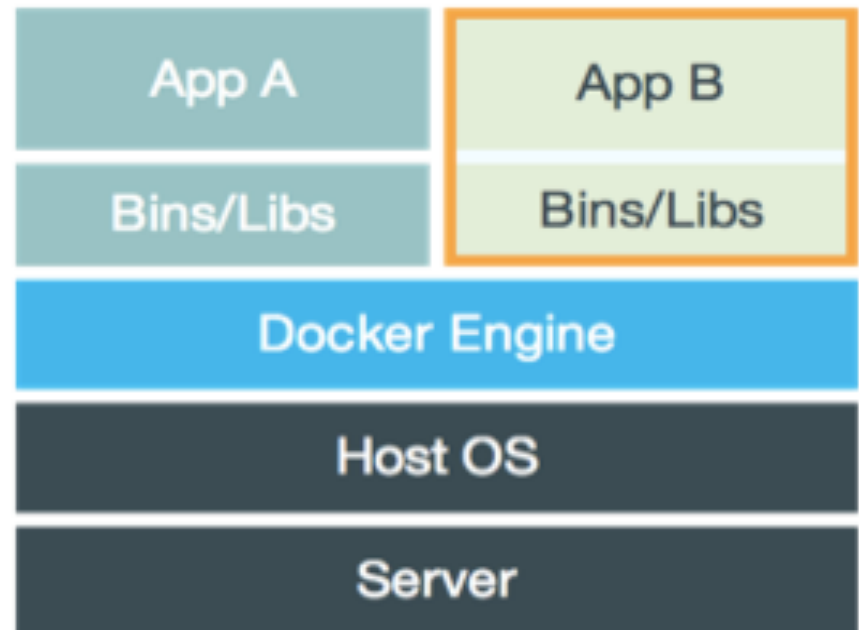
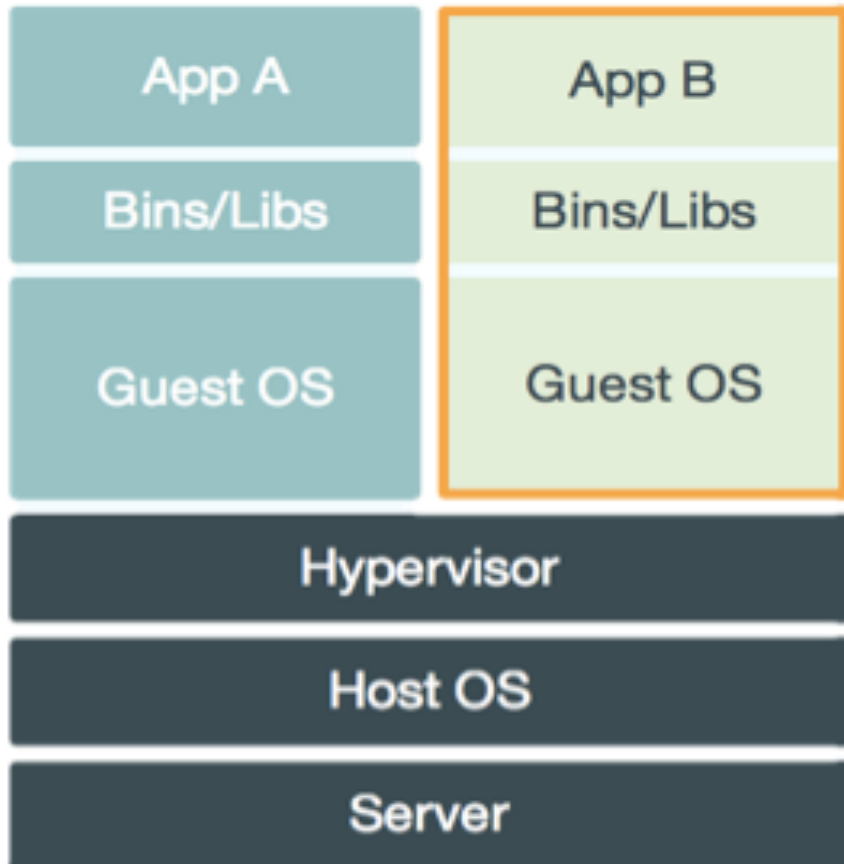
Containers after Docker



How does Docker work ?

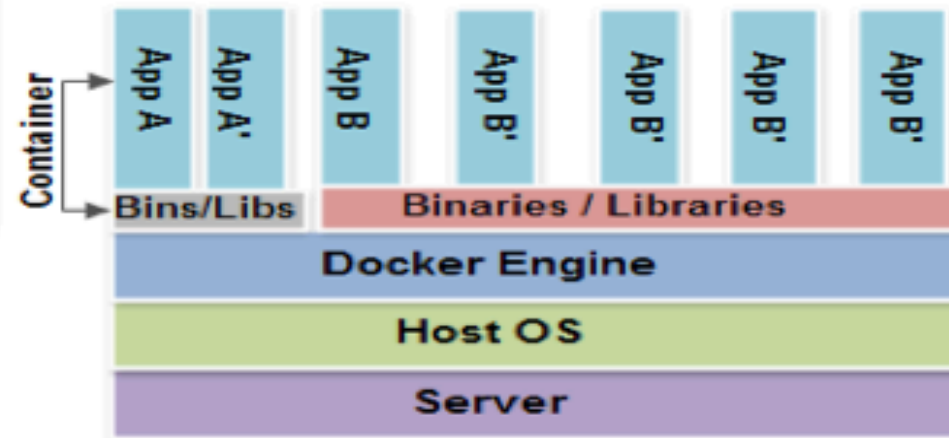
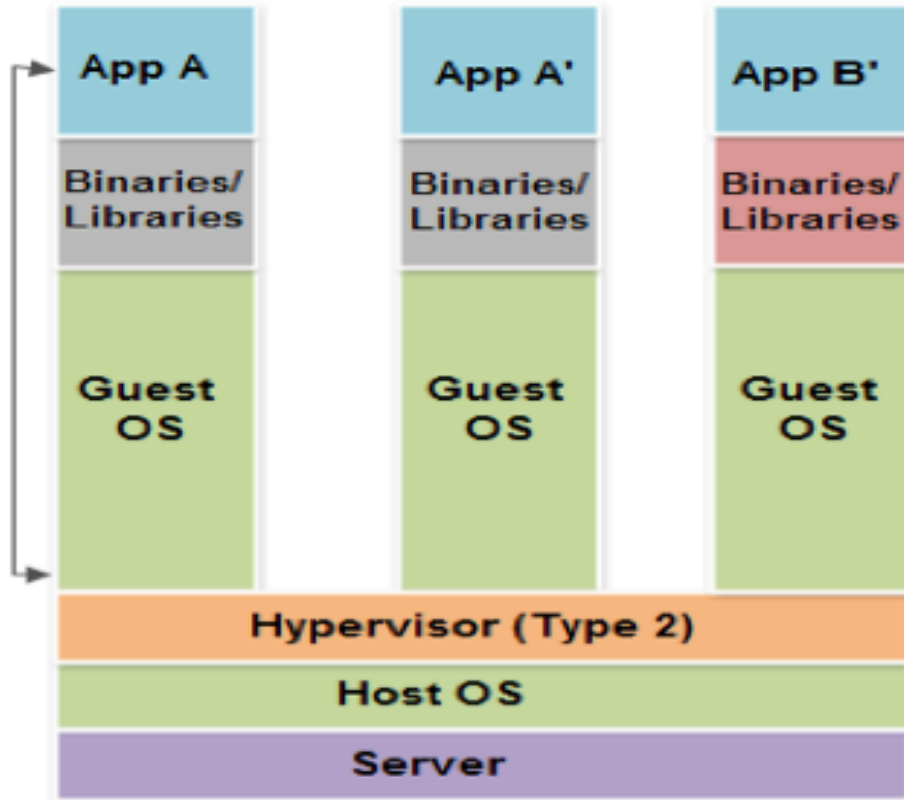
- You can build Docker images that hold your applications
- You can create Docker containers from those Docker images to run your applications.
- You can share those Docker images via Docker Hub or your own registry

Virtual Machine Versus Container.....



Virtual Machine Versus Container.....

Containers vs Virtual Machines



Docker Container Lifecycle

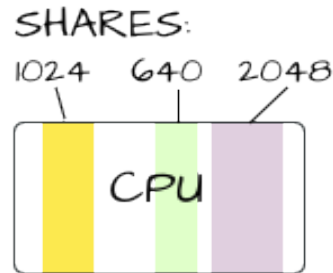
- The Life of a Container
 - Conception
 - **BUILD** an Image from a Dockerfile
 - Birth
 - **RUN** (create+start) a container
 - Reproduction
 - **COMMIT** (persist) a container to a new image
 - **RUN** a new container from an image
 - Sleep
 - **KILL** a running container
 - Wake
 - **START** a stopped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

Linux Cgroups

- Kernel Feature
- Groups of processes
- Control resource allocation

- CPU
- Memory
- Disk
- I/O

- May be nested



CGROUP #1

Gets half as much CPU time as cgroup #3.

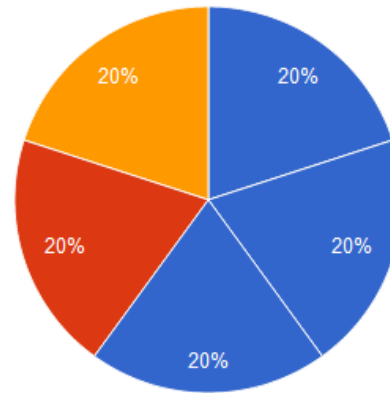
CGROUP #2

Gets the least CPU time.

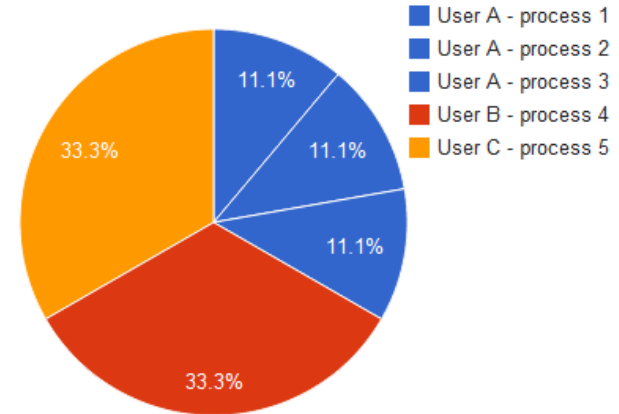
CGROUP #3

Gets the most CPU time.

CPU usage per process without cgroups

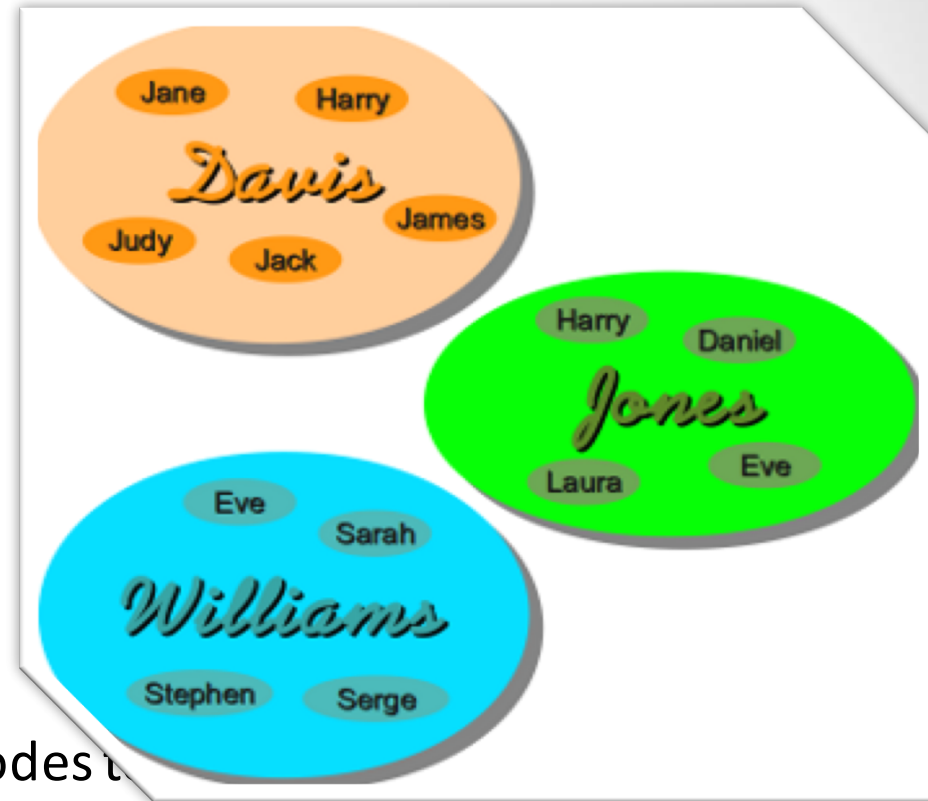


CPU usage per process with cgroups



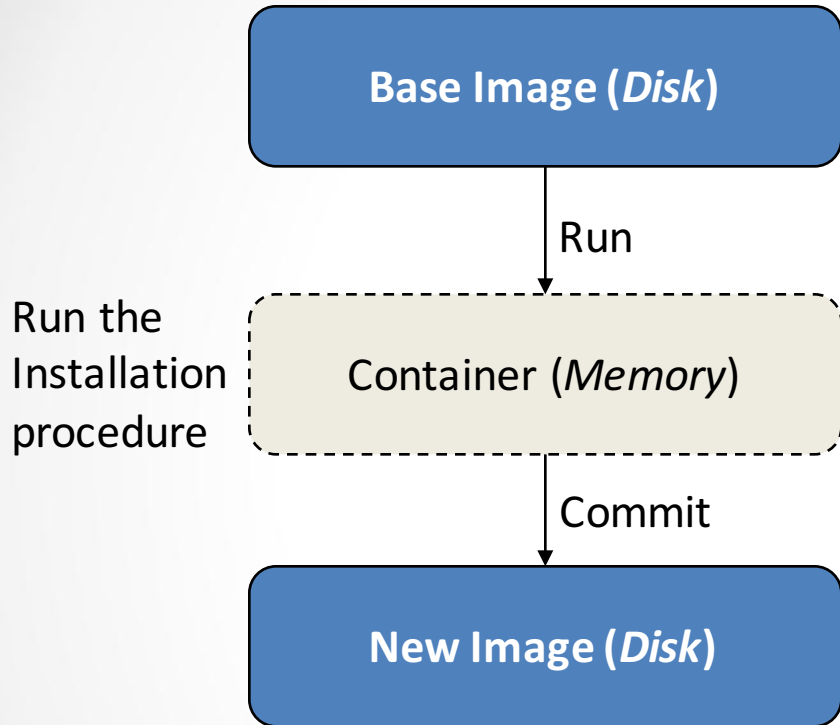
Linux Kernel Namespaces

- 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
- May be nested

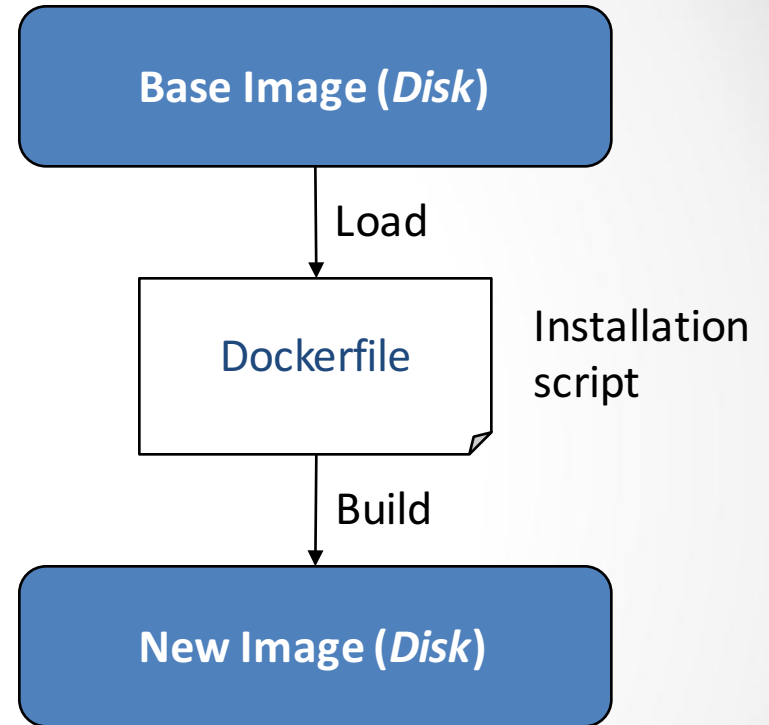


Building a Docker Image

Interactive building



Building from a Docker File



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 daemon: `docker run -d [image] command`

Dockerfile Example:

Bowtie2

FROM ubuntu:14.04

MAINTAINER Enis Afgan <enis.afgan@jhu.edu>

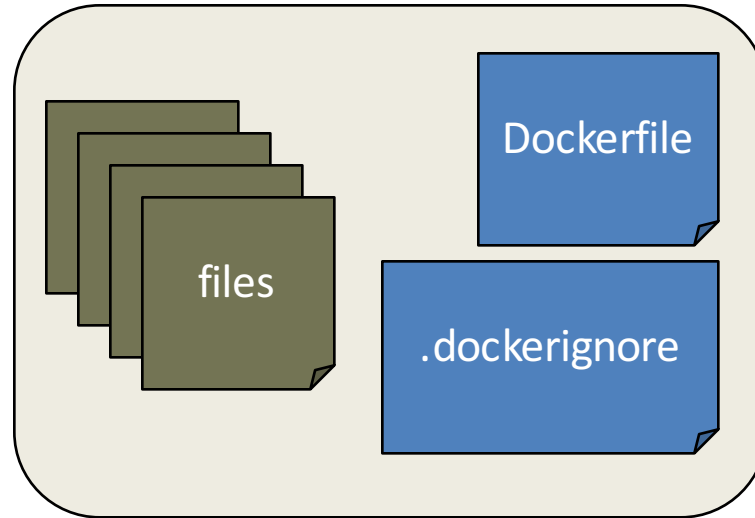
RUN apt-get update -qq --fix-missing; \
apt-get install -qq -y wget unzip;

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

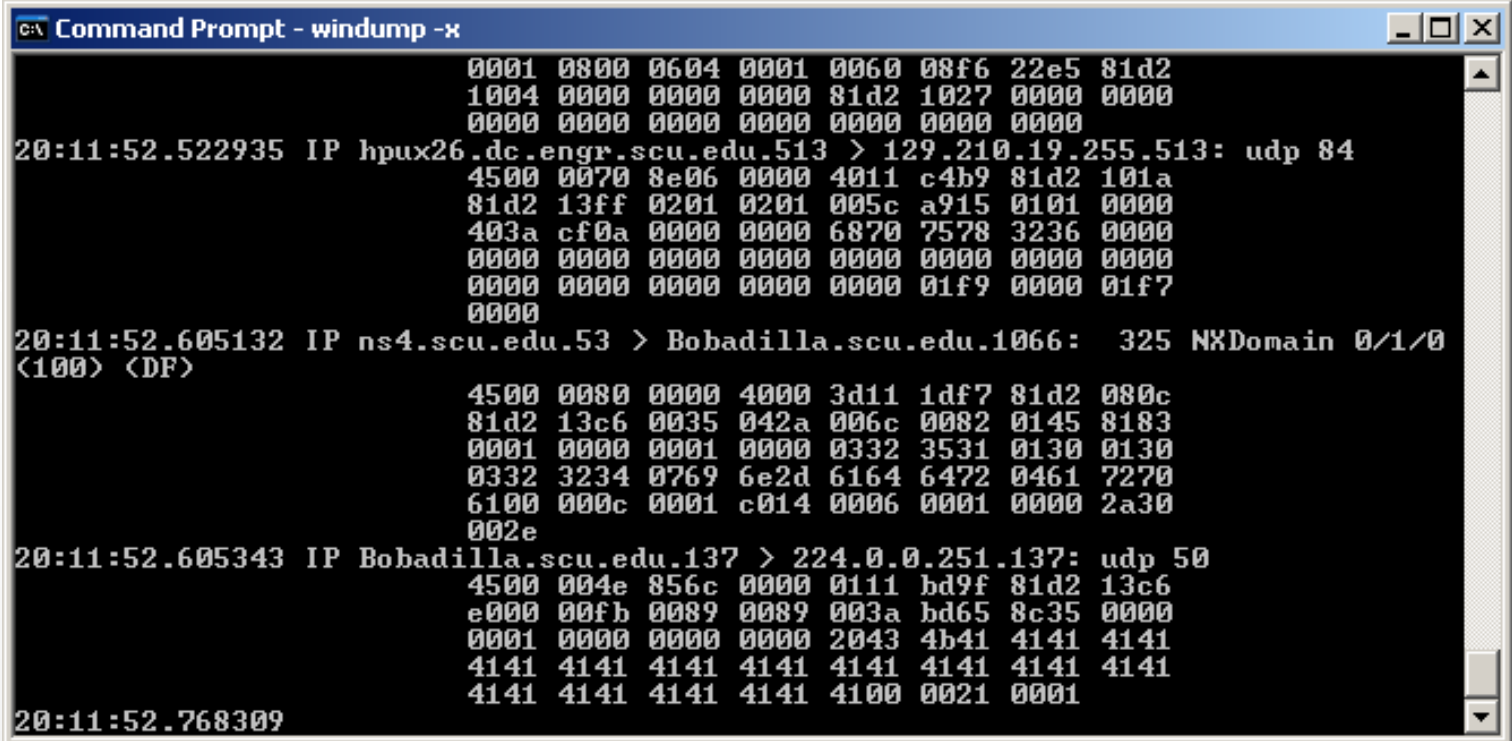
- Does not collect the payload by default.
- Does not scale well.
- State / connections are hidden.
- Very Limited analysis of packages.
- Collects a given number of bytes from each package:
 - This could turn “trap and trace” monitoring into wiretaping because content might be captured.

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



```
c:\ Command Prompt - windump -x
0001 0800 0604 0001 0060 08f6 22e5 81d2
1004 0000 0000 0000 81d2 1027 0000 0000
0000 0000 0000 0000 0000 0000 0000
20:11:52.522935 IP hpux26.dc.engr.scu.edu.513 > 129.210.19.255.513: udp 84
4500 0070 8e06 0000 4011 c4b9 81d2 101a
81d2 13ff 0201 0201 005c a915 0101 0000
403a cf0a 0000 0000 6870 7578 3236 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 01f9 0000 01f7
0000
20:11:52.605132 IP ns4.scu.edu.53 > Bobadilla.scu.edu.1066: 325 NXDomain 0/1/0
<100> <DF>
4500 0080 0000 4000 3d11 1df7 81d2 080c
81d2 13c6 0035 042a 006c 0082 0145 8183
0001 0000 0001 0000 0332 3531 0130 0130
0332 3234 0769 6e2d 6164 6472 0461 7270
6100 000c 0001 c014 0006 0001 0000 2a30
002e
20:11:52.605343 IP Bobadilla.scu.edu.137 > 224.0.0.251.137: udp 50
4500 004e 856c 0000 0111 bd9f 81d2 13c6
e000 00fb 0089 0089 003a bd65 8c35 0000
0001 0000 0000 0000 2043 4b41 4141 4141
4141 4141 4141 4141 4141 4141 4141 4141
4141 4141 4141 4141 4100 0021 0001
20:11:52.768309
```

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 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)
```

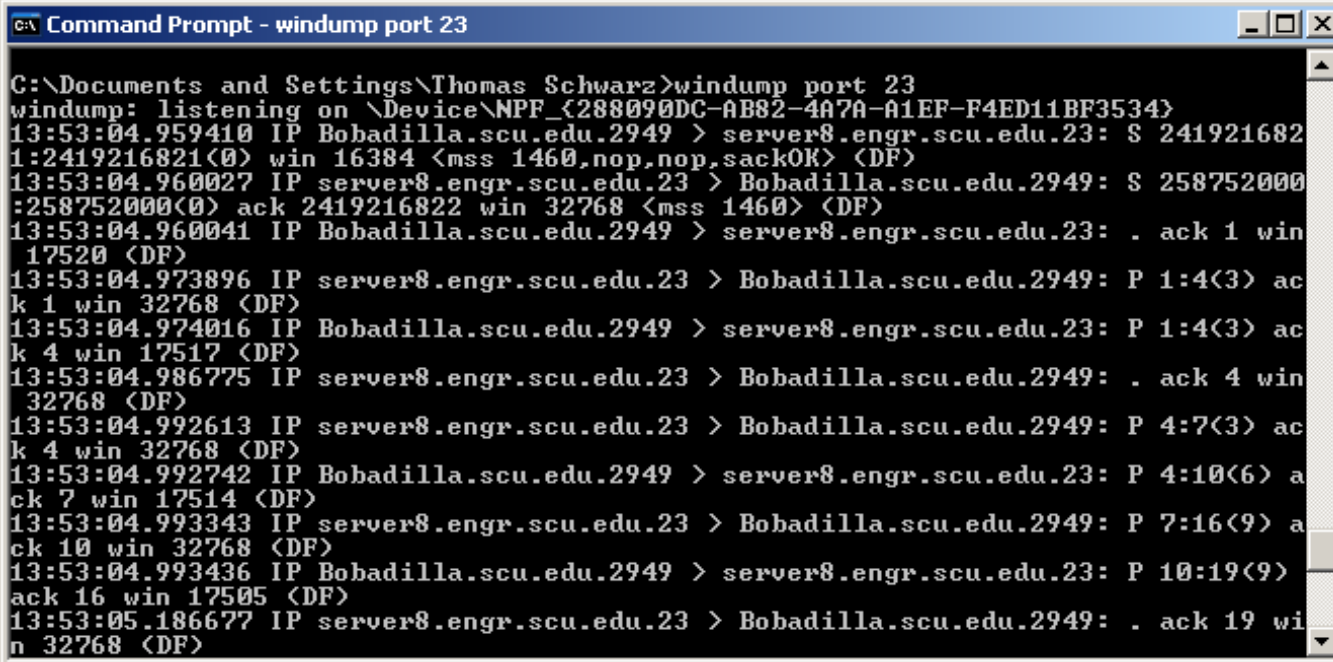
```
4500  033c  8974  1ce8  8001  6627  81d2  13c6
81d2  13d3  6e6f  7071  7273  7475  7677  6162
6364  6566  6768  696a  6b6c  6d6e  6f70  7172
7374  7576  7761  6263  6465  6667  6869  6a6b
6c6d  6e6f  7071  7273  7475  7677  6162  6364
6566
```

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.



```
C:\Documents and Settings\Thomas Schwarz>windump port 23
windump: listening on \Device\NPF_{288090DC-AB82-4A7A-A1EF-F4ED11BF3534}
13:53:04.959410 IP Bobadilla.scu.edu.2949 > server8.engr.scu.edu.23: S 241921682
1:2419216821(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
13:53:04.960027 IP server8.engr.scu.edu.23 > Bobadilla.scu.edu.2949: S 258752000
:258752000(0) ack 2419216822 win 32768 <mss 1460> (DF)
13:53:04.960041 IP Bobadilla.scu.edu.2949 > server8.engr.scu.edu.23: . ack 1 win
17520 (DF)
13:53:04.973896 IP server8.engr.scu.edu.23 > Bobadilla.scu.edu.2949: P 1:4(3) ac
k 1 win 32768 (DF)
13:53:04.974016 IP Bobadilla.scu.edu.2949 > server8.engr.scu.edu.23: P 1:4(3) ac
k 4 win 17517 (DF)
13:53:04.986775 IP server8.engr.scu.edu.23 > Bobadilla.scu.edu.2949: . ack 4 win
32768 (DF)
13:53:04.992613 IP server8.engr.scu.edu.23 > Bobadilla.scu.edu.2949: P 4:7(3) ac
k 4 win 32768 (DF)
13:53:04.992742 IP Bobadilla.scu.edu.2949 > server8.engr.scu.edu.23: P 4:10(6) a
ck 7 win 17514 (DF)
13:53:04.993343 IP server8.engr.scu.edu.23 > Bobadilla.scu.edu.2949: P 7:16(9) a
ck 10 win 32768 (DF)
13:53:04.993436 IP Bobadilla.scu.edu.2949 > server8.engr.scu.edu.23: P 10:19(9)
ack 16 win 17505 (DF)
13:53:05.186677 IP server8.engr.scu.edu.23 > Bobadilla.scu.edu.2949: . ack 19 wi
n 32768 (DF)
```

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`

<code>cwr</code>	<code>ece</code>	<code>urg</code>	<code>ack</code>	<code>psh</code>	<code>rst</code>	<code>syn</code>	<code>fin</code>
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.

<code>cwr</code>	<code>ece</code>	<code>urg</code>	<code>ack</code>	<code>psh</code>	<code>rst</code>	<code>syn</code>	<code>fin</code>
<code>0</code>	<code>0</code>	<code>1</code>	<code>0</code>	<code>0</code>	<code>0</code>	<code>0</code>	<code>0</code>

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.

```
C:\ Command Prompt - windump -F filter1.tdf
20:36:20.507462 IP 204.193.139.221.554 > Bobadilla.scu.edu.1290: . 70562:72022<1
460> ack 3043 win 16395 <DF>
20:36:20.507696 IP 204.193.139.221.554 > Bobadilla.scu.edu.1290: . 72022:73482<1
460> ack 3043 win 16395 <DF>
20:36:20.507701 IP 204.193.139.221.554 > Bobadilla.scu.edu.1290: P 73482:73493<1
1> ack 3043 win 16395 <DF>
20:36:20.507731 IP Bobadilla.scu.edu.1290 > 204.193.139.221.554: . ack 73493 win
65535 <DF>
20:36:20.752121 IP
0 packets received by filter
0 packets dropped by kernel

C:\DOCUMENTS\THOMAS\MYDOCUMENTS\FORENS\IPCAPT>windump -F filter1.tdf
windump: listening on \Device\NPF_{288090DC-AB82-4A7A-A1EF-F4ED11BF3534}
20:36:55.288419 IP adsl-66-218-54-9.dslextrême.com.1601 > dhcp-19-56.engr.scu.ed
u.24849: . ack 1541342939 win 63487 <DF>
20:36:55.291329 IP adsl-66-218-54-9.dslextrême.com.1601 > dhcp-19-56.engr.scu.ed
u.24849: . ack 1 win 65535 <DF>
20:36:58.141984 IP Bobadilla.scu.edu.1037 > haym-gw4.msgr.hotmail.com.80: P 1967
141368:1967141798(430) ack 463138369 win 64677 <DF>
20:36:58.147513 IP haym-gw4.msgr.hotmail.com.80 > Bobadilla.scu.edu.1037: P 1:28
6(285) ack 430 win 17520
20:36:58.304684 IP Bobadilla.scu.edu.1037 > haym-gw4.msgr.hotmail.com.80: . ack
286 win 64392 <DF>
```

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 ASCII 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

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
Fragment offset: 0

```
0000 ff ff ff ff ff 00 e0 ed 01 6e bd 08 00 45 00 ..... ..n...E.  
0010 00 4e 69 8c 00 80 80 11 4c c1 c0 a8 01 02 c0 a8 .N[. .... L.....  
0020 01 ff 00 89 00 89 00 3a 5b b4 84 e7 01 10 00 01 ..... : [.....  
0030 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 43 FEPENEDE JECCACAL  
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

Wireshark Interface

command
menus

display filter
specification

listing of
captured
packets

details of
selected
packet
header

packet content
in hexadecimal
and ASCII

The screenshot displays the Wireshark interface with the following components:

- Command Menus:** A menu bar at the top with options: File, Edit, View, Go, Capture, Analyze, Statistics, Help.
- Display Filter Specification:** A text box labeled "Filter:" with a dropdown menu and buttons for "Expression...", "Clear", and "Apply".
- Listing of Captured Packets:** A table with columns: No., Time, Source, Destination, Protocol, and Info. The table contains several rows of network traffic data.
- Details of Selected Packet Header:** A tree view showing the structure of a selected packet (Frame 4). The selected item is "Hypertext Transfer Protocol" with a sub-item "GET /news/ HTTP/1.1". The details pane shows various header fields such as Host, User-Agent, Accept, and Cookie.
- Packet Content in Hexadecimal and ASCII:** A pane at the bottom showing the raw data of the selected packet in hexadecimal and ASCII format.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.46	128.121.50.122	TCP	1163 > http [SYN] Seq=0 Len=0 MSS=1460
2	0.127987	128.121.50.122	192.168.1.46	TCP	http > 1163 [SYN, ACK] Seq=0 Ack=1 win=57
3	0.128232	192.168.1.46	128.121.50.122	TCP	1163 > http [ACK] Seq=1 Ack=1 win=65535
4	0.153700	192.168.1.46	128.121.50.122	HTTP	GET /news/ HTTP/1.1
5	0.329641	128.121.50.122	192.168.1.46	TCP	[TCP segment of a reassembled PDU]
6	0.330326	128.121.50.122	192.168.1.46	HTTP	[TCP previous segment lost] Continuation
7	0.330467	192.168.1.46	128.121.50.122	TCP	1163 > http [ACK] Seq=617 Ack=1082 win=64
8	0.342042	128.121.50.122	192.168.1.46	TCP	[TCP Retransmission] [TCP segment of a re

```
Frame 4 (710 bytes on wire (568 bytes captured) on interface 0:
Ethernet II, Src: Netgear_61:8e:6d (00:09:5b:61:8e:6d), Dst: WestellT_9f:92:b9 (00:0f:db:9f:92:b9)
Internet Protocol, Src: 192.168.1.46 (192.168.1.46), Dst: 128.121.50.122 (128.121.50.122)
Transmission Control Protocol, Src Port: 1163 (1163), Dst Port: http (80), Seq: 1, Ack: 1, Len: 656
Hypertext Transfer Protocol
  GET /news/ HTTP/1.1\r\n
  Host: www.wireshark.org\r\n
  User-Agent: Mozilla/5.0 (Windows; U; windows NT 5.1; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4\r\n
  Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5\r\n
  Accept-Language: en-us,en;q=0.5\r\n
  Accept-Encoding: gzip,deflate\r\n
  Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
  Keep-Alive: 300\r\n
  Connection: keep-alive\r\n
  Referer: http://www.wireshark.org/faq.html\r\n
  Cookie: __utma=87653150.62471437.1181007382.1181007382.1181169142.2; __utnz=87653150.1181007382.1.1.1.utr\r\n
```

Hex	ASCII
0000 00 0f 0b 9f 92 b8 0b 09 5b 61 8e 6d 08 00 45 00[a..E.
0010 02 b8 0f 25 40 00 80 06 74 51 c0 a8 01 2e 80 79	...80...tq....y
0020 32 7a 04 8b 00 50 ed bc 8e 1b 4e c6 f1 18 50 18	22...P...R...P.
0030 ff ff 77 74 00 00 47 45 54 20 2f 6e 65 77 73 2f	..wt...GE t /news/
0040 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a	HTTP/1. 1..Host:
0050 20 77 77 77 2e 77 69 72 65 73 68 61 72 6b 2e 6f	www.wir eshark.o
0060 72 67 0d 0a 53 73 65 72 2d 41 67 65 6e 74 3a 20	rg..User -agent:
0070 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 29 57 69 6e	Mozilla/ 5.0 (win
0080 64 6f 77 73 3b 20 55 3b 20 57 69 6e 64 6f 77 73	ows; U; windows
0090 20 4e 54 20 35 2e 31 3b 20 65 6e 2d 55 53 3b 20	NT 5.1; en-US;
00a0 72 76 3a 31 2e 38 2e 31 2e 34 29 20 47 65 63 6b	rv:1.8.1 .4) Geck
00b0 6f 2f 32 30 30 37 30 35 31 35 20 46 69 72 65 66	o/200705 15 Firef

Status Bar

Tucker Ellis & West aaa.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

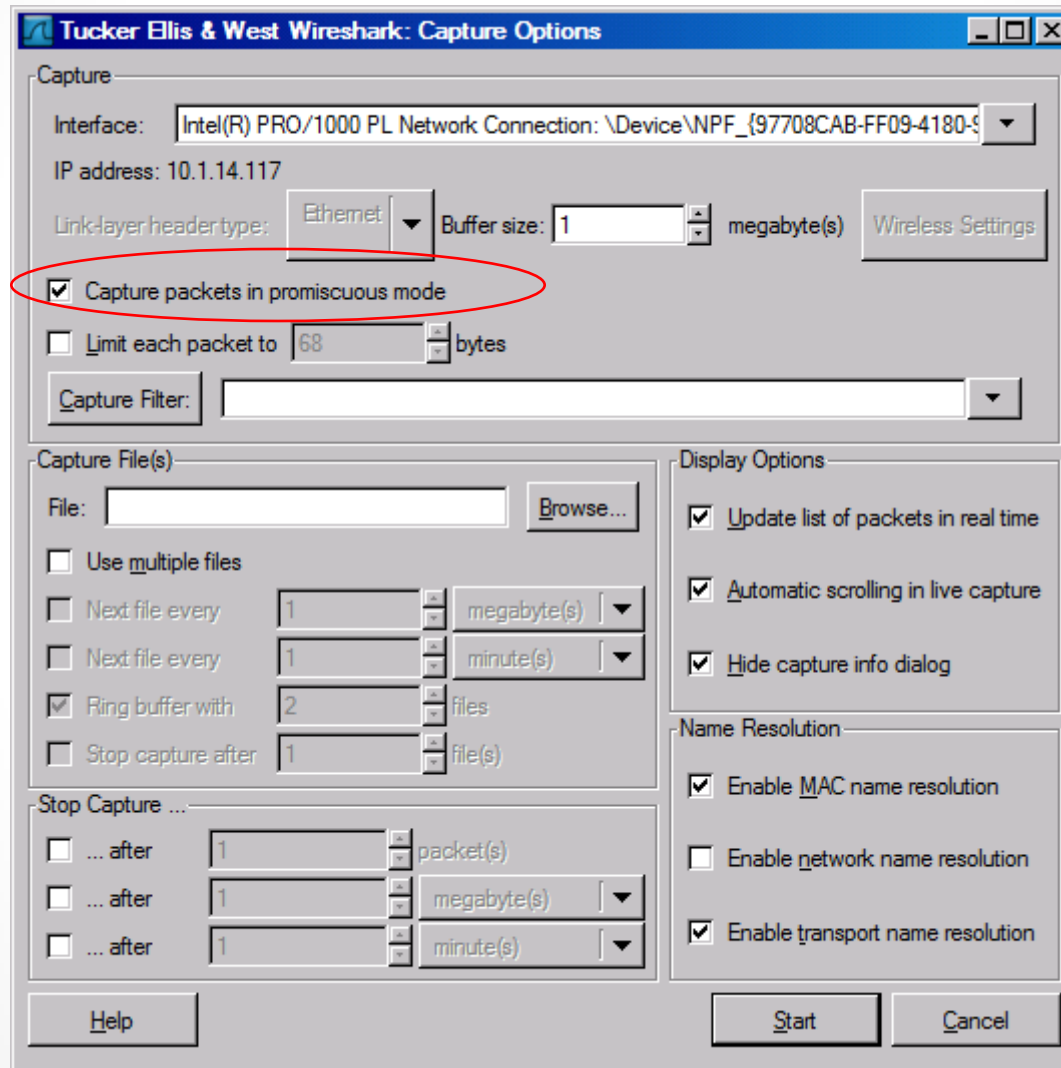
No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.2	192.168.1.255	NBNS	Name query NB ECI_DOMAIN<1c>
2	0.746308	192.168.1.2	192.168.1.255	NBNS	Name query NB ECI_DOMAIN<1c>
3	0.751270	192.168.1.2	192.168.1.255	NBNS	Name query NB ECI_DOMAIN<1c>
4	9.318731	silicom_01:6e:bd	Broadcast	ARP	who 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
Fragment offset: 0

```
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 .N.....L.....  
0020 01 ff 00 89 00 89 00 3a 5b b4 84 e7 01 10 00 01 .....: [.....  
0030 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 EEPENEJE EOCACAC  
0050 41 43 41 43 41 42 4d 00 00 20 00 01 ACACABM. ...
```

Identification (p.id), 2 bytes Packets: 691 Displayed: 691 Marked: 0 Profile: Default

Capture Options



The image shows the 'Capture Options' dialog box in Wireshark. The window title is 'Tucker Ellis & West Wireshark: Capture Options'. The 'Capture' section is at the top, with 'Interface' set to 'Intel(R) PRO/1000 PL Network Connection: \Device\NPF_{97708CAB-FF09-4180-9...}' and 'IP address' set to '10.1.14.117'. The 'Linklayer headertype' is 'Ethernet' and 'Buffer size' is '1 megabyte(s)'. A red circle highlights the checked checkbox 'Capture packets in promiscuous mode'. Below it is 'Limit each packet to 68 bytes' and an empty 'Capture Filter' field. The 'Capture File(s)' section includes a 'File' field with a 'Browse...' button, and checkboxes for 'Use multiple files', 'Next file every 1 megabyte(s)', 'Next file every 1 minute(s)', 'Ring buffer with 2 files' (checked), and 'Stop capture after 1 file(s)'. The 'Stop Capture ...' section has three options: '... after 1 packet(s)', '... after 1 megabyte(s)', and '... after 1 minute(s)'. The 'Display Options' section has three checked checkboxes: 'Update list of packets in real time', 'Automatic scrolling in live capture', and 'Hide capture info dialog'. The 'Name Resolution' section has three checkboxes: 'Enable MAC name resolution' (checked), 'Enable network name resolution' (unchecked), and 'Enable transport name resolution' (checked). At the bottom are 'Help', 'Start', and 'Cancel' buttons.

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

Linklayer headertype: 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: ▾ 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

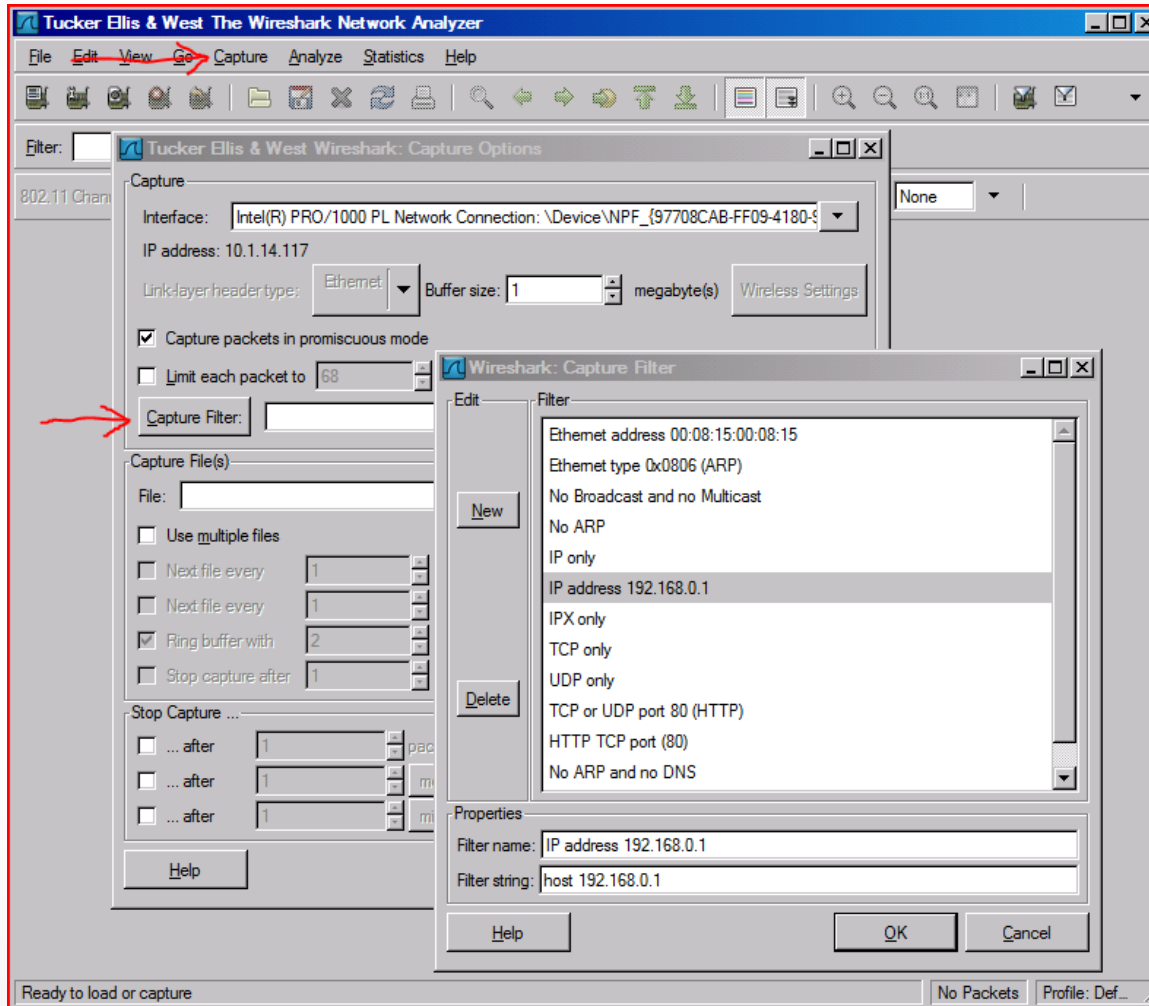
Enable MAC name resolution

Enable network name resolution

Enable transport name resolution

Help Start Cancel

Capture Filter



Capture Filter examples

host 10.1.11.24

host 192.168.0.1 and host 10.1.11.1

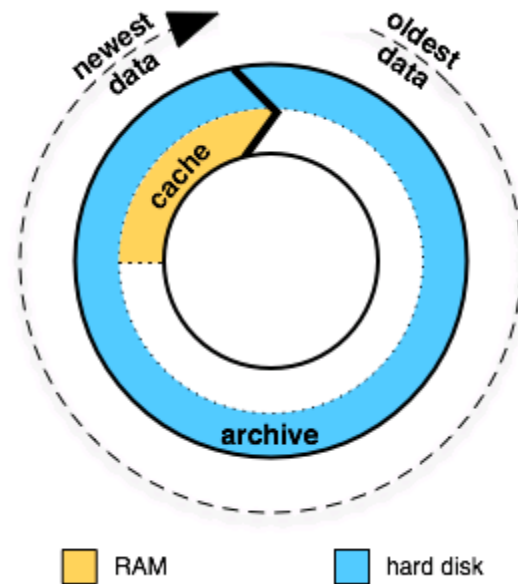
tcp port http

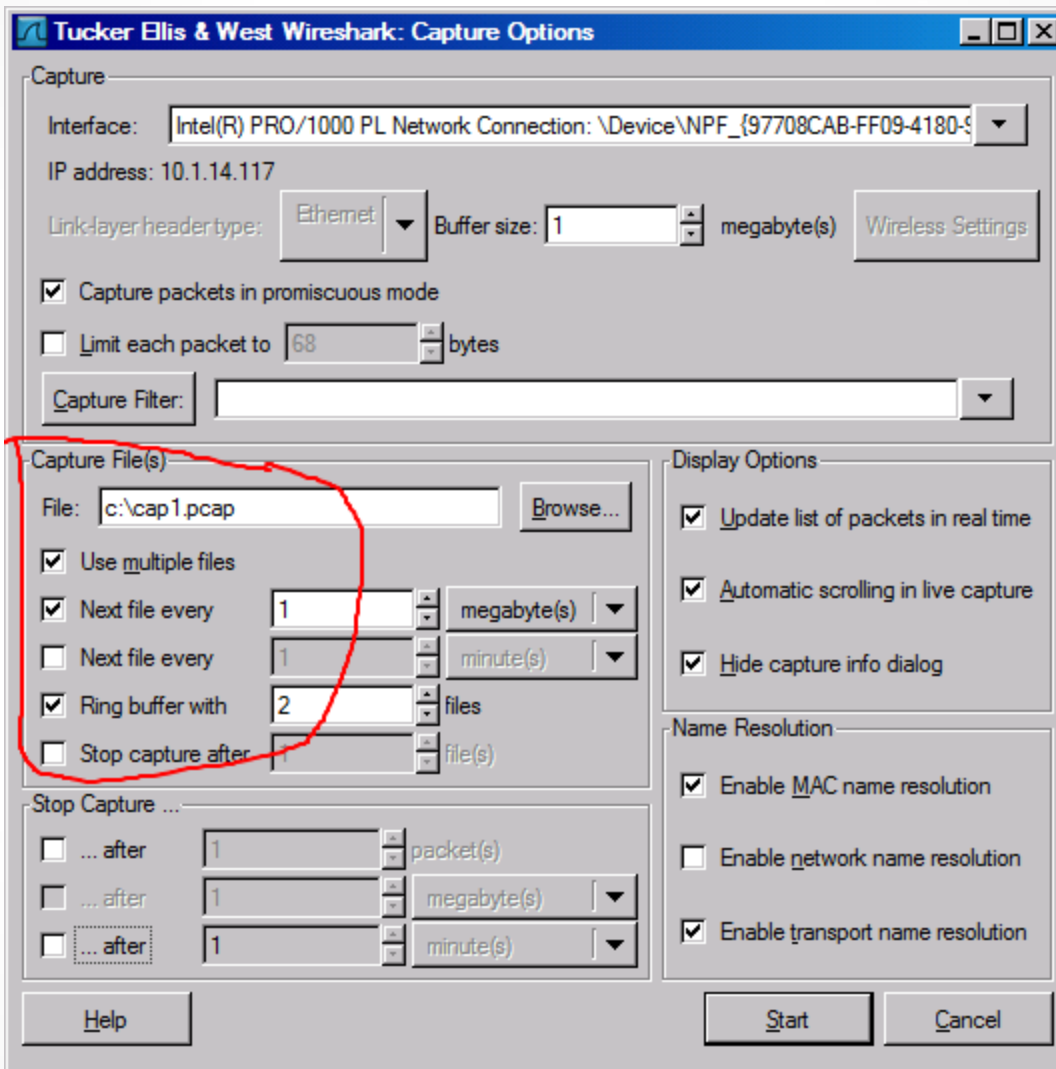
ip

not broadcast not multicast

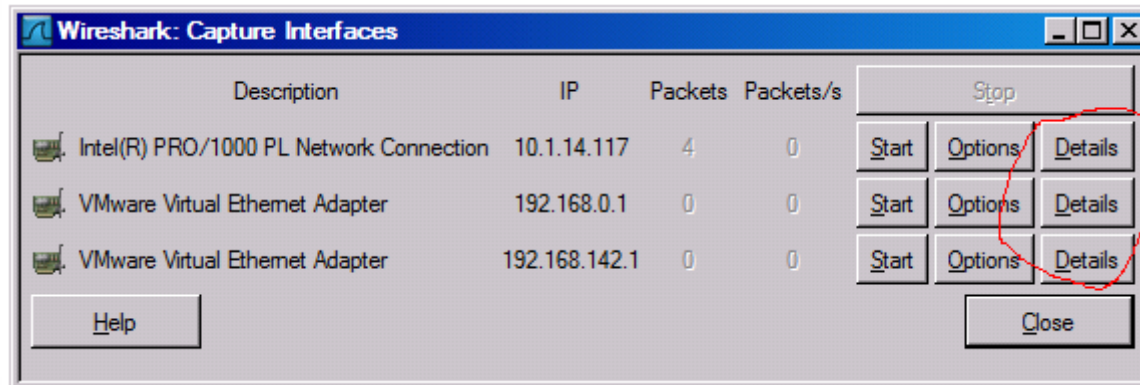
ether host 00:04:13:00:09:a3

Capture Buffer Usage

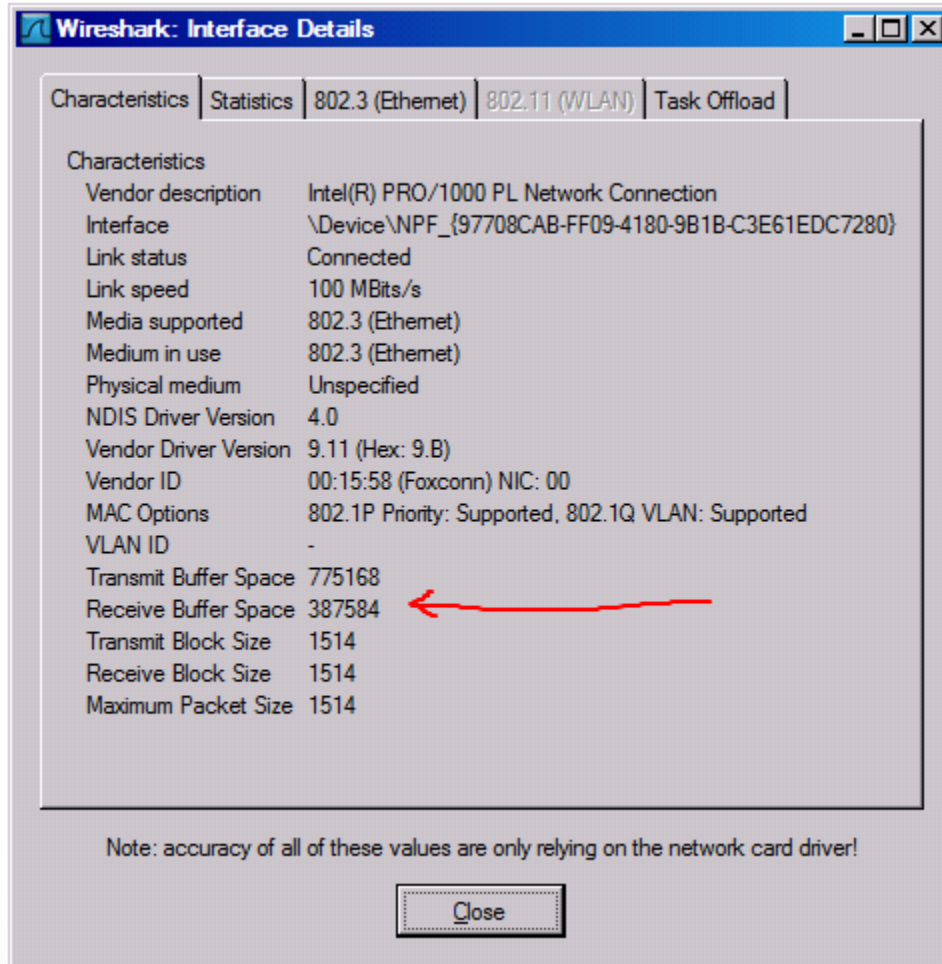




Capture Interfaces



Interface Details: Characteristics



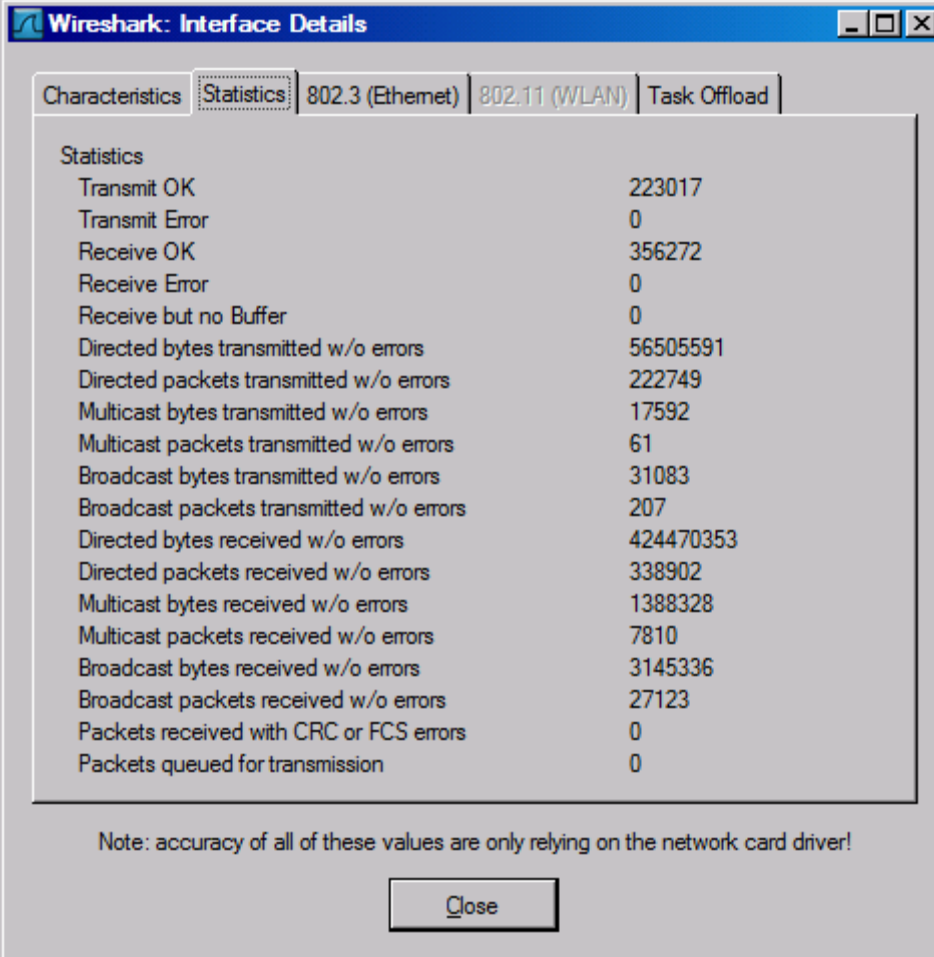
The image shows a screenshot of the Wireshark 'Interface Details' window. The window title is 'Wireshark: Interface Details'. It has a tabbed interface with 'Characteristics' selected. The '802.11 (WLAN)' tab is highlighted in the tab bar. The main content area displays a list of interface characteristics. A red arrow points to the 'Receive Buffer Space' value of 387584.

Characteristics	
Vendor description	Intel(R) PRO/1000 PL Network Connection
Interface	\Device\NPF_{97708CAB-FF09-4180-9B1B-C3E61EDC7280}
Link status	Connected
Link speed	100 MBits/s
Media supported	802.3 (Ethernet)
Medium in use	802.3 (Ethernet)
Physical medium	Unspecified
NDIS Driver Version	4.0
Vendor Driver Version	9.11 (Hex: 9.B)
Vendor ID	00:15:58 (Foxconn) NIC: 00
MAC Options	802.1P Priority: Supported, 802.1Q VLAN: Supported
VLAN ID	-
Transmit Buffer Space	775168
Receive Buffer Space	387584
Transmit Block Size	1514
Receive Block Size	1514
Maximum Packet Size	1514

Note: accuracy of all of these values are only relying on the network card driver!

Close

Interface Details: Statistics



Wireshark: Interface Details

Characteristics **Statistics** 802.3 (Ethernet) 802.11 (WLAN) Task Offload

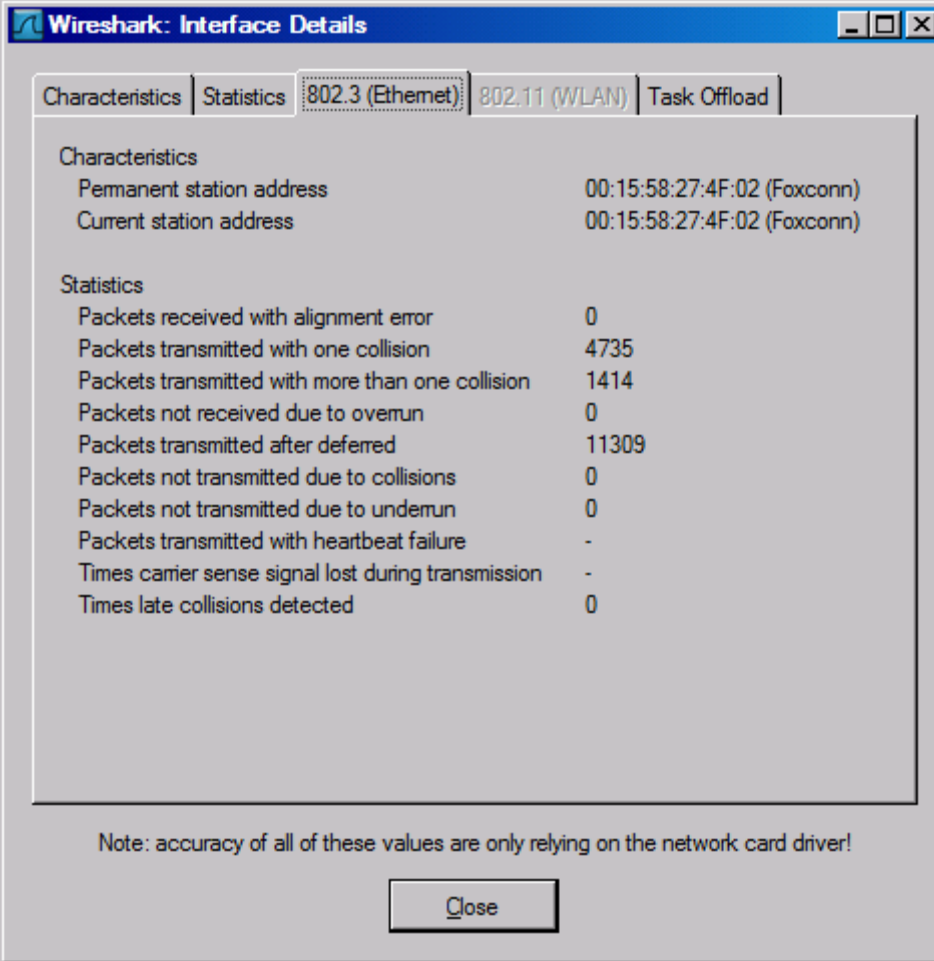
Statistics

Transmit OK	223017
Transmit Error	0
Receive OK	356272
Receive Error	0
Receive but no Buffer	0
Directed bytes transmitted w/o errors	56505591
Directed packets transmitted w/o errors	222749
Multicast bytes transmitted w/o errors	17592
Multicast packets transmitted w/o errors	61
Broadcast bytes transmitted w/o errors	31083
Broadcast packets transmitted w/o errors	207
Directed bytes received w/o errors	424470353
Directed packets received w/o errors	338902
Multicast bytes received w/o errors	1388328
Multicast packets received w/o errors	7810
Broadcast bytes received w/o errors	3145336
Broadcast packets received w/o errors	27123
Packets received with CRC or FCS errors	0
Packets queued for transmission	0

Note: accuracy of all of these values are only relying on the network card driver!

Close

Interface Details: 802.3 (Ethernet)



The screenshot shows the 'Wireshark: Interface Details' window. It has a blue title bar and a tabbed interface with three tabs: 'Characteristics', 'Statistics', and '802.3 (Ethernet)'. The '802.3 (Ethernet)' tab is selected. Below the tabs, there are two sections: 'Characteristics' and 'Statistics'. The 'Characteristics' section shows 'Permanent station address' and 'Current station address', both with the value '00:15:58:27:4F:02 (Foxconn)'. The 'Statistics' section shows various network metrics with their corresponding values. At the bottom of the window, there is a note and a 'Close' button.

Characteristics	
Permanent station address	00:15:58:27:4F:02 (Foxconn)
Current station address	00:15:58:27:4F:02 (Foxconn)

Statistics	
Packets received with alignment error	0
Packets transmitted with one collision	4735
Packets transmitted with more than one collision	1414
Packets not received due to overrun	0
Packets transmitted after deferred	11309
Packets not transmitted due to collisions	0
Packets not transmitted due to underrun	0
Packets transmitted with heartbeat failure	-
Times carrier sense signal lost during transmission	-
Times late collisions detected	0

Note: accuracy of all of these values are only relying on the network card driver!

Close

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 than capture filters

Display Filter

The screenshot displays the Wireshark Network Analyzer interface. The main window title is "Tucker Ellis & West The Wireshark Network Analyzer". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, and Help. The toolbar contains various icons for file operations, capture, and analysis. The Filter field at the top shows the expression `ip.addr == 192.168.0.1`. Below this, the interface shows the selected interface (802.11 Channel), Channel Offset, FCS Filter, and Decryption Mode (None). A red arrow points to the Filter field.

A dialog box titled "Wireshark: Display Filter" is open, showing a list of filter expressions. The "Filter" list includes:

- Ethernet address 00:08:15:00:08:15
- Ethernet type 0x0806 (ARP)
- Ethernet broadcast
- No ARP
- IP only
- IP address 192.168.0.1
- IP address isn't 192.168.0.1, don't use != for this!
- IPX only
- TCP only
- UDP only
- UDP port isn't 53 (not DNS), don't use != for this!
- TCP or UDP port is 80 (HTTP)

The "Properties" section of the dialog shows:

- Filter name: IP address 192.168.0.1
- Filter string: `ip.addr == 192.168.0.1`

The dialog has buttons for "New", "Delete", "Help", "OK", "Apply", and "Cancel".

At the bottom of the main window, the status bar shows "Ready to load or capture", "No Packets", and "Profile: Def..."

Display Filter Examples

ip.src==10.1.11.0/24

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)

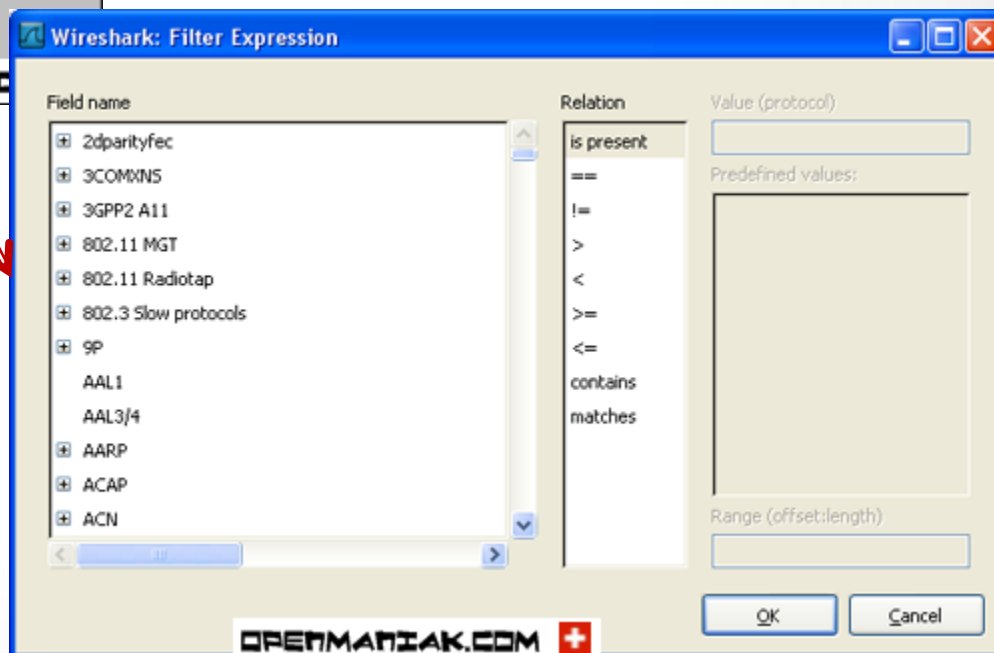
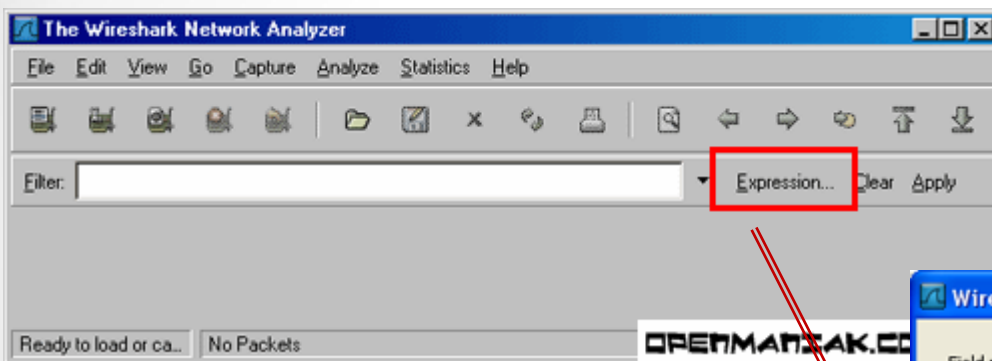
**(ip.addr==192.168.1.10 && ip.addr==192.168.1.20) && (tcp.port==445 ||
tcp.port==139)**

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

tcp.dstport == 80

Display Filter

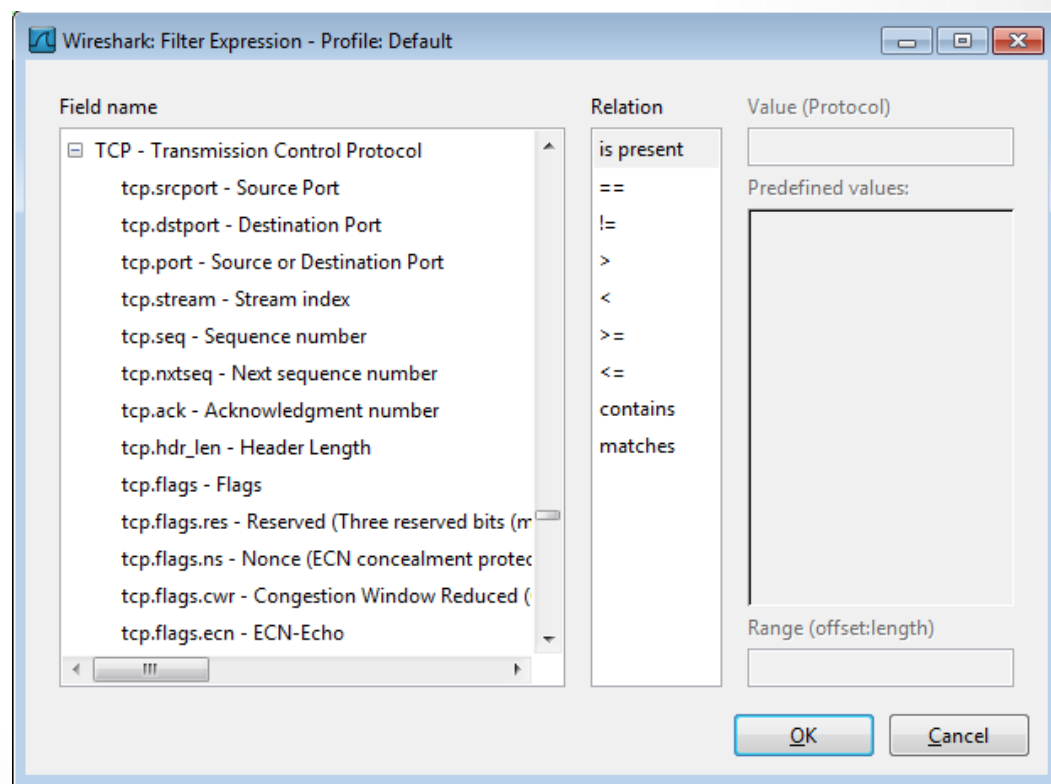
Syntax:	Protocol	String 1	String 2	Comparison operator	Value	Logical Operations	Other expression
Example:	ftp	passive	ip	==	10.2.3.4	xor	icmp.type



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:
<code>eq</code>	<code>==</code>	Equal
<code>ne</code>	<code>!=</code>	Not equal
<code>gt</code>	<code>></code>	Greater than
<code>lt</code>	<code><</code>	Less than
<code>ge</code>	<code>>=</code>	Greater or equal
<code>le</code>	<code><=</code>	Less or equal

→ Logical expressions:

English format:	C like format:	Meaning:
<code>and</code>	<code>&&</code>	Logical AND
<code>or</code>	<code> </code>	Logical OR
<code>xor</code>	<code>^^</code>	Logical XOR
<code>not</code>	<code>!</code>	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.

Filter: `tcp.port == 100`

Filter: `tcp.port = 100`

Correct syntax

Wrong syntax

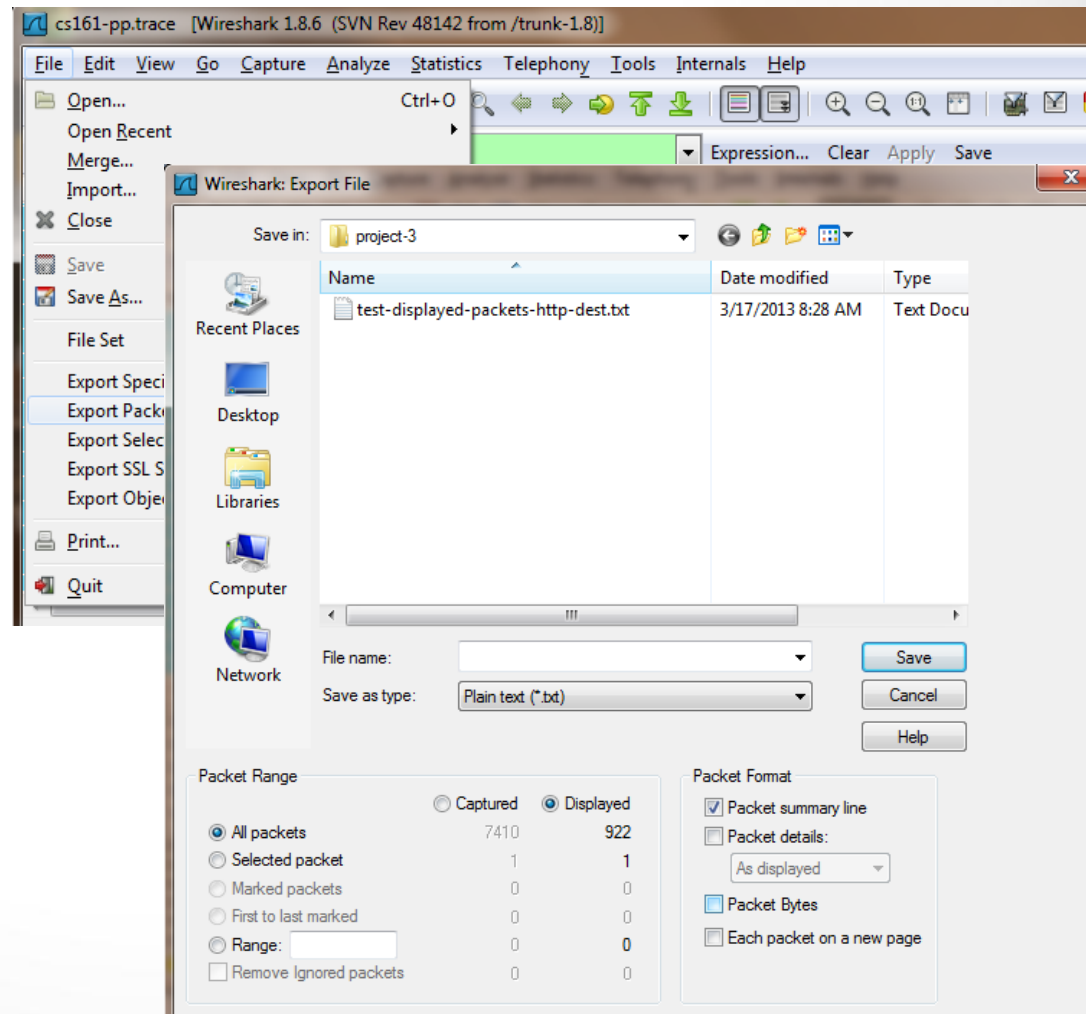
Save Filtered Packets After Using Display Filter

- 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

The screenshot displays the Wireshark interface with the following components:

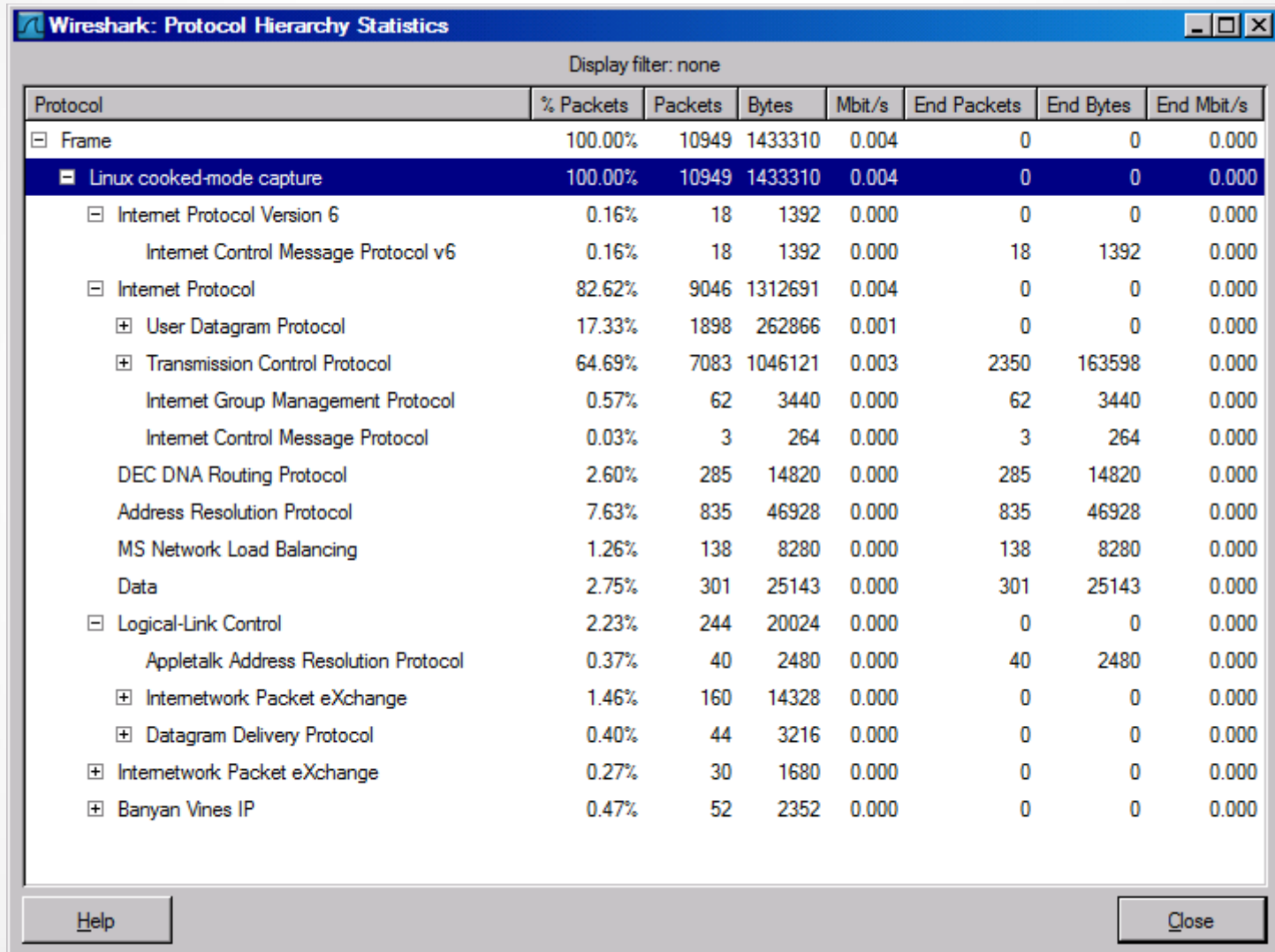
- File:** Tucker Ellis & West Obsolete_Packets.cap - Wireshark
- Menu Bar:** File, Edit, View, Go, Capture, Analyze, Statistics, Help
- Toolbar:** Summary, Protocol Hierarchy, Conversations, Endpoints, IO Graphs, Expression..., Decryption Mode: None
- Filter:** 802.11 Channel: [dropdown]
- Packet List:**

No.	Time	Source
1	0.000000	::
2	0.000010	::
3	2.179063	192.168.1.1
4	2.439522	192.168.1.1
5	2.715733	192.168.1.1
6	2.821401	192.168.1.1
7	2.821546	192.168.1.1
8	2.824683	192.168.1.1
9	2.990859	192.168.1.1
10	3.266913	192.168.1.1
11	3.495707	fe80::20c
12	3.495727	fe80::20c
13	3.542893	192.168.1.1
14	3.543088	192.168.1.1
- Protocol Hierarchy:**
 - 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...
- Packet Details:**
 - Frame 1 (88 bytes on wire)
 - Linux cooked capture
 - Internet Protocol Version 4
 - Internet Control Message Protocol
- Packet Bytes:**

```

0000  00 04 00 01 00 06 00 0c
0010  60 00 00 00 00 20 00 01
0020  00 00 00 00 00 00 00 00
0030  00 00 00 01 ff 0d 56 e3
0040  83 00 d2 c2 00 00 00 00
0050  00 00 01 ff 0d 56 e3
  
```
- Status Bar:** File: "C:\Users\vo2.TEW\Downloads\Obsolete_Packets.cap... Packets: 10949 Displayed: 10949 Marked: 0 Profile: Default

Protocol Hierarchy



Wireshark: Protocol Hierarchy Statistics

Display filter: none

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

Help Close

Follow TCP Stream

Tucker Ellis & West http-ethereal-trace-1 - Wireshark

Filter: `(ip.addr eq 192.168.1.102 and ip.addr eq 128.1` Expression... Clear Apply

802.11 Channel: Channel Offset: FCS Filter: Decryption Mode: None

No.	Time	Source	Destination	Protocol	Info
7	4.675312	192.168.1.102	128.119.245.12	TCP	unikeypro > http [SYN]
8	4.694429	128.119.245.12	192.168.1.102	TCP	> unikeypro [SYN, Seq: 1281192451, Window=0, Len=52]
9	4.694458	192.168.1.102	128.119.245.12	TCP	ypro > http [ACK, Seq=1281192451, Win=0, Len=0]
10	4.694850	192.168.1.102	128.119.245.12	TCP	ethereal-labs/lab... > unikeypro [ACK, Seq=1281192451, Win=0, Len=0]
11	4.717289	128.119.245.12	192.168.1.102	TCP	> unikeypro [ACK, Seq=1281192451, Win=0, Len=0]
12	4.718993	128.119.245.12	192.168.1.102	TCP	1.1 200 OK (text/css)
13	4.724332	192.168.1.102	128.119.245.12	TCP	favicon.ico HTTP/1.1
14	4.750366	128.119.245.12	192.168.1.102	TCP	1.1 404 Not Found
15	4.859777	192.168.1.102	128.119.245.12	TCP	ypro > http [ACK, Seq=1281192451, Win=0, Len=0]

Frame 8 (62 bytes on wire, 62 bytes captured)

- Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: 192.168.1.102 (08:00:27:00:00:02)
- 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: http (80), Seq: 1281192451, Win: 0, Len: 52

```
0000 00 08 74 4f 36 23 00 06 25 da af 73 08 00 45 00  ..t06#..%.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...

Follow TCP Stream

red - stuff you sent

blue - stuff you get

```
Follow TCP Stream
Stream Content
GET /ethereal-labs/lab2-1.html HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (windows; U; windows NT 5.1; en-US; rv:1.0.2) Gecko/20021120
Netscape/7.01
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,text/css,*/*;q=0.1
Accept-Language: en-us, en;q=0.50
Accept-Encoding: gzip, deflate, compress;q=0.9
Accept-Charset: ISO-8859-1, utf-8;q=0.66, */*;q=0.66
Keep-Alive: 300
Connection: keep-alive

HTTP/1.1 200 OK
Date: Tue, 23 Sep 2003 05:29:50 GMT
Server: Apache/2.0.40 (Red Hat Linux)
Last-Modified: Tue, 23 Sep 2003 05:29:00 GMT
ETag: "1bfed-49-79d5bf00"
Accept-Ranges: bytes
Content-Length: 73
Keep-Alive: timeout=10, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1

<html>
Congratulations. You've downloaded the file lab2-1.html!
</html>

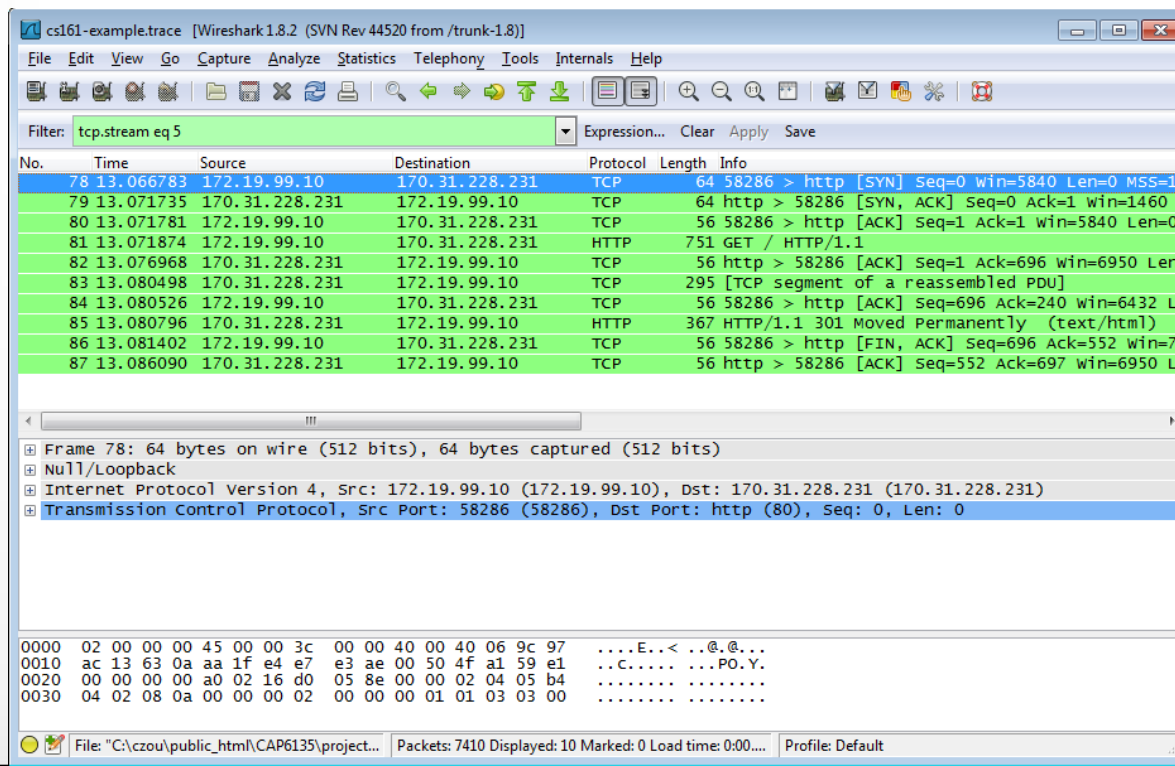
GET /favicon.ico HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (windows; U; windows NT 5.1; en-US; rv:1.0.2) Gecko/20021120
Netscape/7.01
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,text/css,*/*;q=0.1
Accept-Language: en-us, en;q=0.50
Accept-Encoding: gzip, deflate, compress;q=0.9
Accept-Charset: ISO-8859-1, utf-8;q=0.66, */*;q=0.66
Keep-Alive: 300
Connection: keep-alive

Find Save As Print Entire conversation (2714 bytes)
 ASCII  EBCDIC  Hex Dump  C Arrays  Raw

Help Close Filter Out This Stream
```

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

The screenshot shows the Wireshark interface with the 'Expert Info' pane open for frame 8. The main packet list shows a sequence of protocols: SNMP, DNS, TCP, and HTTP. The selected frame 8 is an HTTP GET request from 128.119.245.12 to 192.168.1.102.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	19	19	SNMP	get-request SNMPv2-SMI
2	0.017162	19	19	SNMP	get-response SNMPv2-SMI
3	3.017086	19	19	SNMP	get-request SNMPv2-SMI
4	3.034572	19	19	SNMP	get-response SNMPv2-SMI
5	4.626878	19	19	DNS	Standard query A gaia.c
6	4.663785	63	63	DNS	Standard query response
7	4.675312	19	19	TCP	unikeypro > http [SYN]
8	4.694429	12	12	TCP	http > unikeypro [SYN]
9	4.694458	19	19	TCP	unikeypro > http [ACK]
10	4.694850	19	19	HTTP	GET /ethereal-labs/lab
11	4.717289	12	12	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/
14	4.750366	128.119.245.12	192.168.1.102	HTTP	HTTP/1.1 404 Not Found

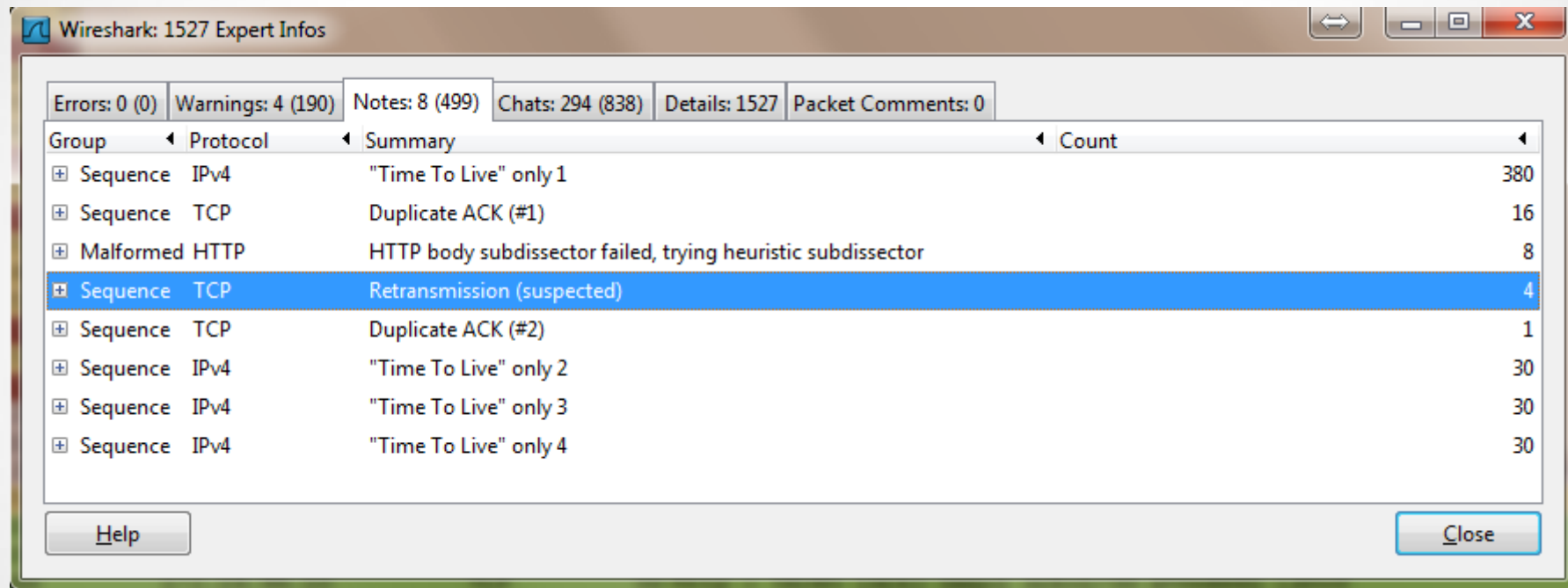
Expert Info

- 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:36:23)
- 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, Len: 0

```
0000  00 08 74 4f 36 23 00 06 25 da af 73 08 00 45 00  ..t06#..%.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

The screenshot shows the Wireshark interface with the 'Conversations' pane open. The pane displays a list of protocols and their corresponding packet numbers. The protocols listed are SNMP, DNS, TCP, and HTTP. The packet numbers are 104, 102, 104, 102, 19, 102, 15.12, 102, 15.12, 15.12, 102, 102, 15.12, and 102. The details pane shows the structure of frame 2, including Ethernet II, Internet Protocol, and Simple Network Management Protocol (SNMP).

No.	Time	Source
1	0.000000	192.168.1.1
2	0.017162	192.168.1.1
3	3.017086	192.168.1.1
4	3.034572	192.168.1.1
5	4.626878	192.168.1.1
6	4.663785	63.240.76.1
7	4.675312	192.168.1.1
8	4.694429	128.119.2.1
9	4.694458	192.168.1.1
10	4.694850	192.168.1.1
11	4.717289	128.119.2.1
12	4.718993	128.119.2.1
13	4.724332	192.168.1.1
14	4.750366	128.119.2.1

No.	Time	Source	Destination	Protocol	Info
104				SNMP	get-request SNMPv2-SMI
102				SNMP	get-response SNMPv2-SMI
104				SNMP	get-request SNMPv2-SMI
102				SNMP	get-response SNMPv2-SMI
19				DNS	Standard query A gaia.d
102				DNS	Standard query response
15.12				TCP	unikeypro > http [SYN]
102				TCP	http > unikeypro [SYN]
15.12				TCP	unikeypro > http [ACK]
15.12				HTTP	GET /ethereal-labs/lab
102				TCP	http > unikeypro [ACK]
102				HTTP	HTTP/1.1 200 OK (text
15.12				HTTP	GET /favicon.ico HTTP/1
102				HTTP	HTTP/1.1 404 Not Found

Frame 2 (93 bytes on wire) (Captured on interface 802.11 Channel):

- Ethernet II, Src: Hewlett-Packard (08:00:0c:2c:3e:00), Dst: DellComp_4f:36:23 (00:08:74:4f:36:23)
- Internet Protocol, Src: 192.168.1.1, Dst: 192.168.1.102
- User Datagram Protocol, Src Port: 161, Dst Port: 161
- Simple Network Management Protocol (SNMP)

File: "C:\Traces\http-ethereal-trace-1" 4443 Bytes 00:00:06

Packets: 1... Profile: Def...

Conversations

Conversations: cs161-pp.trace

Ethernet Fibre Channel FDDI IPv4: 173 IPv6: 1 IPX JXTA NCP RSVP SCTP **TCP: 155** Token Ring UDP: 2398 USB WLAN

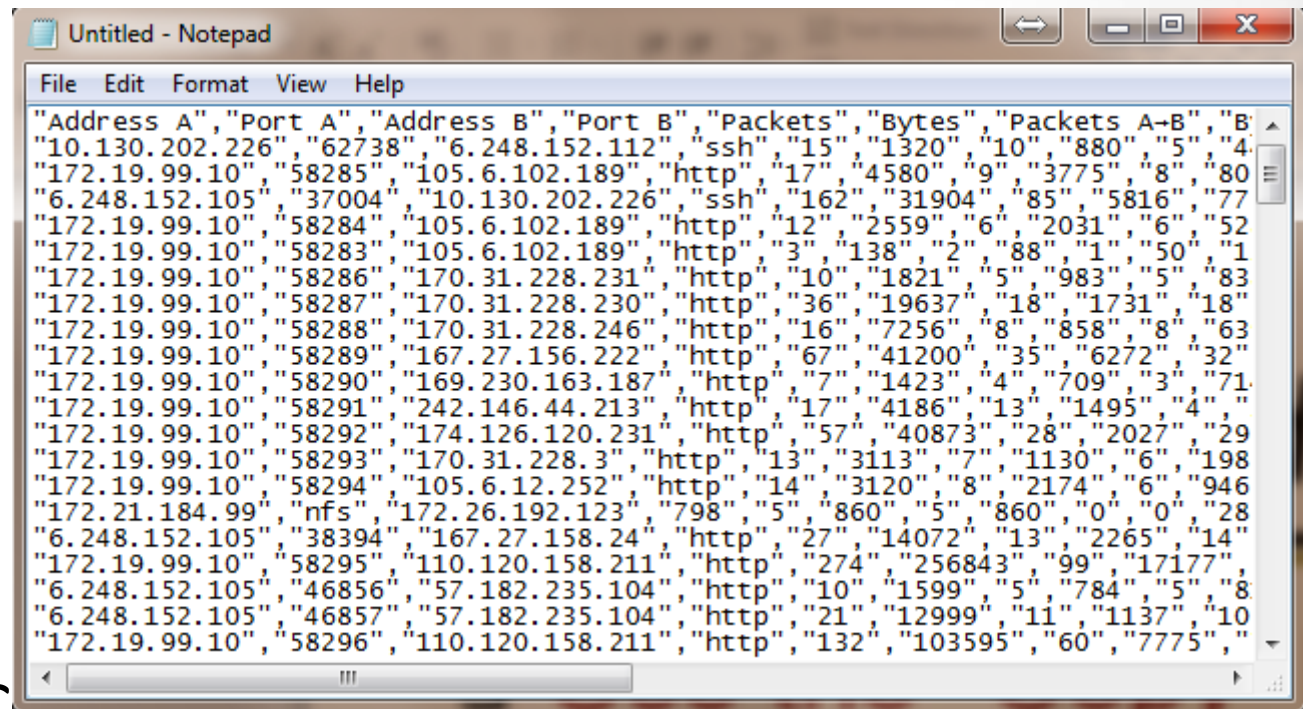
TCP Conversations

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B	Packets A-B	Bytes
10.130.202.226	62738	6.248.152.112	ssh	15	1 320	10	880	5	
172.19.99.10	58285	105.6.102.189	http	17	4 580	9	3 775	8	
6.248.152.105	37004	10.130.202.226	ssh	162	31 904	85	5 816	77	
172.19.99.10	58284	105.6.102.189	http	12	2 559	6	2 031	6	
172.19.99.10	58283	105.6.102.189	http	3	138	2	88	1	
172.19.99.10	58286	170.31.228.231	http	10	1 821	5	983	5	
172.19.99.10	58287	170.31.228.230	http	36	19 637	18	1 731	18	
172.19.99.10	58288	170.31.228.246	http	16	7 256	8	858	8	
172.19.99.10	58289	167.27.156.222	http	67	41 200	35	6 272	32	
172.19.99.10	58290	169.230.163.187	http	7	1 423	4	709	3	
172.19.99.10	58291	242.146.44.213	http	17	4 186	13	1 495	4	
172.19.99.10	58292	174.126.120.231	http	57	40 873	28	2 027	29	

Name resolution Limit to display filter

Help Copy Follow Stream Close

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



```
File Edit Format View Help
"Address A", "Port A", "Address B", "Port B", "Packets", "Bytes", "Packets A-B", "Bytes A-B"
"10.130.202.226", "62738", "6.248.152.112", "ssh", "15", "1320", "10", "880", "5", "40"
"172.19.99.10", "58285", "105.6.102.189", "http", "17", "4580", "9", "3775", "8", "80"
"6.248.152.105", "37004", "10.130.202.226", "ssh", "162", "31904", "85", "5816", "77"
"172.19.99.10", "58284", "105.6.102.189", "http", "12", "2559", "6", "2031", "6", "52"
"172.19.99.10", "58283", "105.6.102.189", "http", "3", "138", "2", "88", "1", "50", "1"
"172.19.99.10", "58286", "170.31.228.231", "http", "10", "1821", "5", "983", "5", "83"
"172.19.99.10", "58287", "170.31.228.230", "http", "36", "19637", "18", "1731", "18"
"172.19.99.10", "58288", "170.31.228.246", "http", "16", "7256", "8", "858", "8", "63"
"172.19.99.10", "58289", "167.27.156.222", "http", "67", "41200", "35", "6272", "32"
"172.19.99.10", "58290", "169.230.163.187", "http", "7", "1423", "4", "709", "3", "71"
"172.19.99.10", "58291", "242.146.44.213", "http", "17", "4186", "13", "1495", "4", "28"
"172.19.99.10", "58292", "174.126.120.231", "http", "57", "40873", "28", "2027", "29"
"172.19.99.10", "58293", "170.31.228.3", "http", "13", "3113", "7", "1130", "6", "198"
"172.19.99.10", "58294", "105.6.12.252", "http", "14", "3120", "8", "2174", "6", "946"
"172.21.184.99", "nfs", "172.26.192.123", "798", "5", "860", "5", "860", "0", "0", "28"
"6.248.152.105", "38394", "167.27.158.24", "http", "27", "14072", "13", "2265", "14"
"172.19.99.10", "58295", "110.120.158.211", "http", "274", "256843", "99", "17177", "8"
"6.248.152.105", "46856", "57.182.235.104", "http", "10", "1599", "5", "784", "5", "8"
"6.248.152.105", "46857", "57.182.235.104", "http", "21", "12999", "11", "1137", "10"
"172.19.99.10", "58296", "110.120.158.211", "http", "132", "103595", "60", "7775", "8"
```

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

Find EndPoint Statistics

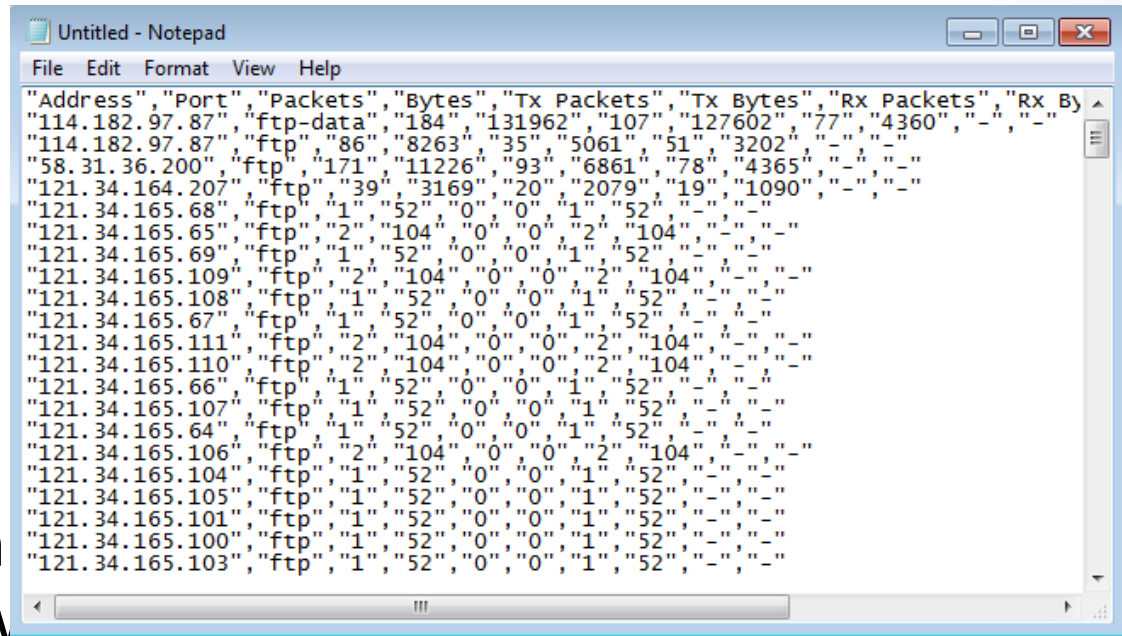
- Menu “statistics” → “endpoint list” → “TCP”

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude
10.130.202.226	62738	15	1 320	10	880	5	440	-
6.248.152.112	ssh	15	1 320	5	440	10	880	-
172.19.99.10	58285	17	4 580	9	3 775	8	805	-
105.6.102.189	http	32	7 277	15	1 383	17	5 894	-
10.130.202.226	ssh	162	31 904	77	26 088	85	5 816	-
6.248.152.105	37004	162	31 904	85	5 816	77	26 088	-
172.19.99.10	58284	12	2 559	6	2 031	6	528	-
172.19.99.10	58283	3	138	2	88	1	50	-
172.19.99.10	58286	10	1 821	5	983	5	838	-
170.31.228.231	http	10	1 821	5	838	5	983	-
172.19.99.10	58287	36	19 637	18	1 731	18	17 906	-
170.31.228.230	http	36	19 637	18	17 906	18	1 731	-
172.19.99.10	58288	16	7 256	8	858	8	6 398	-
170.31.228.246	http	16	7 256	8	6 398	8	858	-
172.19.99.10	58289	67	41 200	35	6 272	32	34 928	-
167.27.156.222	http	67	41 200	32	34 928	35	6 272	-

- You can sort
- “Tx” : transm

Find EndPoint Statistics

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



The screenshot shows a Notepad window titled "Untitled - Notepad" with a menu bar (File, Edit, Format, View, Help). The text content is a list of endpoint statistics, each line representing a different IP address and port. The columns are: Address, Port, Packets, Bytes, Tx Packets, Tx Bytes, Rx Packets, and Rx Bytes. The data is as follows:

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
"114.182.97.87"	"ftp-data"	"184"	"131962"	"107"	"127602"	"77"	"4360"
"114.182.97.87"	"ftp"	"86"	"8263"	"35"	"5061"	"51"	"3202"
"58.31.36.200"	"ftp"	"171"	"11226"	"93"	"6861"	"78"	"4365"
"121.34.164.207"	"ftp"	"39"	"3169"	"20"	"2079"	"19"	"1090"
"121.34.165.68"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.65"	"ftp"	"2"	"104"	"0"	"0"	"2"	"104"
"121.34.165.69"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.109"	"ftp"	"2"	"104"	"0"	"0"	"2"	"104"
"121.34.165.108"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.67"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.111"	"ftp"	"2"	"104"	"0"	"0"	"2"	"104"
"121.34.165.110"	"ftp"	"2"	"104"	"0"	"0"	"2"	"104"
"121.34.165.66"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.107"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.64"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.106"	"ftp"	"2"	"104"	"0"	"0"	"2"	"104"
"121.34.165.104"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.105"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.101"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.100"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"
"121.34.165.103"	"ftp"	"1"	"52"	"0"	"0"	"1"	"52"

- Then, you can statistics you want

Flow Graphs

The screenshot shows the Wireshark interface with a packet capture from 'Tucker Ellis & West http-ethereal-trace-1'. The main packet list shows several packets, with packet 14 selected. The packet details pane shows the structure of a UDP packet containing a Simple Network Management Protocol (SNMP) message. The 'Flow Graphs' menu option is highlighted in the 'Statistics' menu.

Statistics Menu:

- 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 Graphs...**
- HTTP
- IP address...
- ISUP Messages...
- Multicast Streams
- ONC-RPC Programs
- Packet Length...
- Port Type...
- SMPP Operations...
- TCP Stream Graph
- WLAN Traffic...

Packet List:

No.	Time	Source
1	0.000000	192.168.1.104
2	0.017162	192.168.1.104
3	3.017086	192.168.1.104
4	3.034572	192.168.1.104
5	4.626878	192.168.1.104
6	4.663785	63.240.70.104
7	4.675312	192.168.1.104
8	4.694429	128.119.2.104
9	4.694458	192.168.1.104
10	4.694850	192.168.1.104
11	4.717289	128.119.2.104
12	4.718993	128.119.2.104
13	4.724332	192.168.1.104
14	4.750366	128.119.2.104

Packet Details (Packet 14):

- Protocol: UDP (0x11)
 - Header checksum: 0x0da7
 - Source: 192.168.1.104 (192.168.1.104)
 - Destination: 192.168.1.104 (192.168.1.104)
- User Datagram Protocol, Src Port: snmp (161), Destination Port: opsvie...
 - Source port: snmp (161)
 - Destination port: opsvie...
 - Length: 59
 - Checksum: 0x1ec4 [corrected]
- Simple Network Management Protocol (SNMP) (0x00000000)

Packet Bytes:

Offset	Hex	ASCII
0000	00 08 74 4f 36 23 00 30	
0010	00 4f ec d8 00 00 3c 11	
0020	01 66 00 a1 10 1d 00 3b	
0030	06 70 75 62 6c 69 63 a2	
0040	02 01 00 30 18 30 16 06	
0050	02 00 04 02 01 02 02 02	

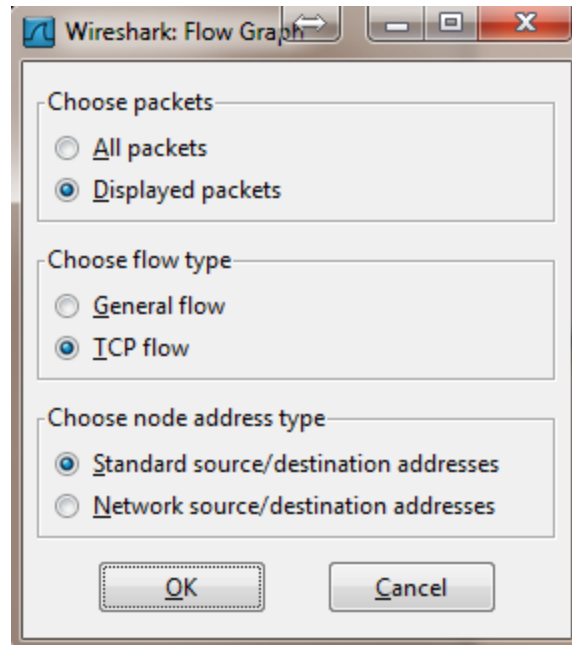
Packet Bytes (continued):

Offset	Hex	ASCII
00	..t06#.0 .a....E.	
a8	.0....<.h..	
04	.f.....; .01.WL.N	
	.public. \$....1...	
	...0.0.. .+.....	

Packet Info: Port: opsvie-envoy (4125)

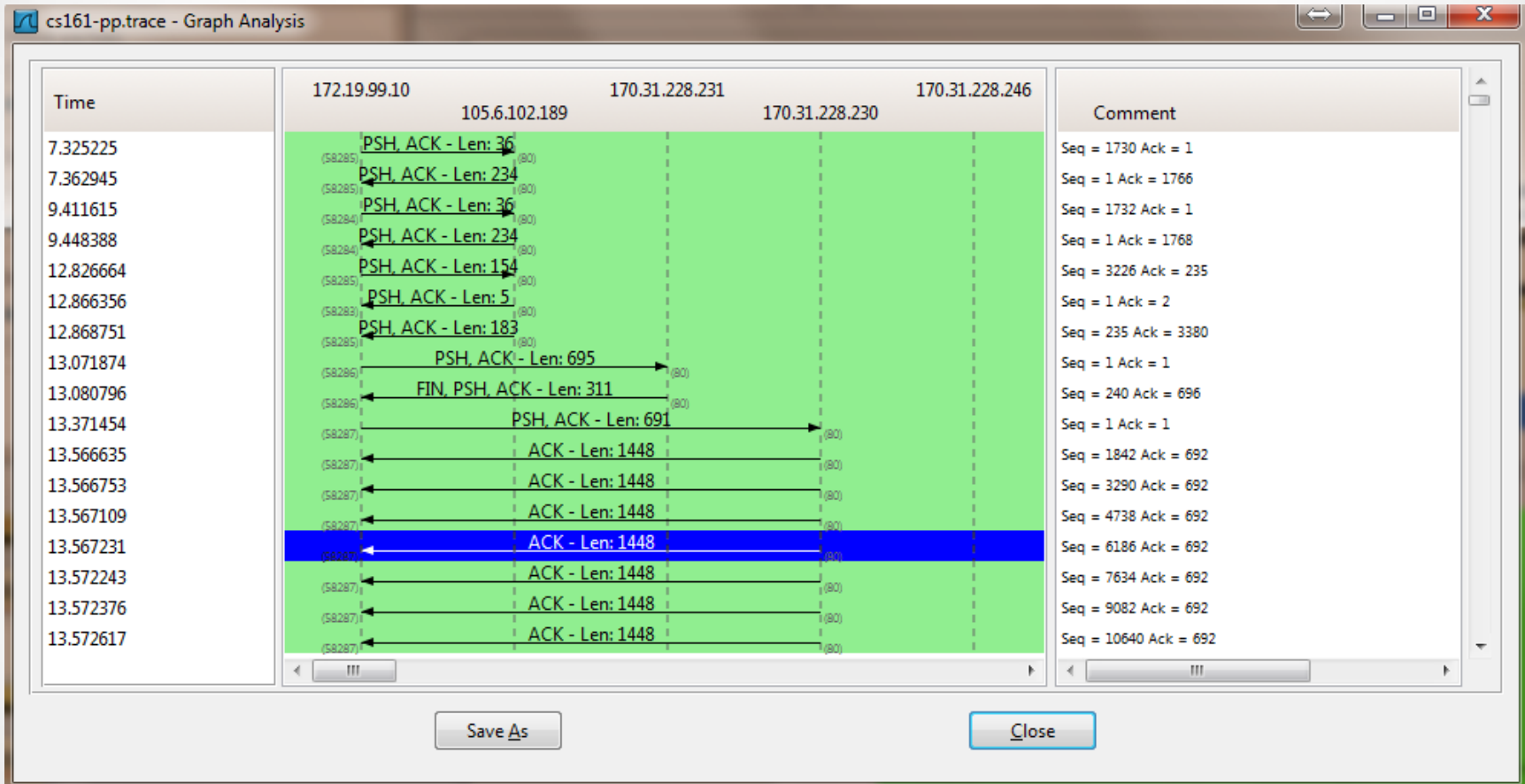
Status Bar: File: "C:\Traces\http-ethereal-trace-1" 4443 Bytes 00:00:06 Packets: 17 Displayed: 17 Marked: 0 Profile: Default

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



Export HTTP

The screenshot shows the Wireshark interface with the 'Export' menu open. The 'HTTP' object is selected under 'Objects'. The packet list shows the following details:

No.	Time	Source	Destination	Protocol	Info
9	4.694458	192.168.1.102	128.119.245.12	TCP	unikeypro > http [SYN]
10	4.694850	192.168.1.102	128.119.245.12	HTTP	GET /ethereal-labs/lab2-1.html HTTP/1.1
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/html)
13	4.724332	192.168.1.102	128.119.245.12	HTTP	GET /favicon.ico HTTP/1.1
14	4.750366	128.119.245.12	192.168.1.102	HTTP	HTTP/1.1 404 Not Found

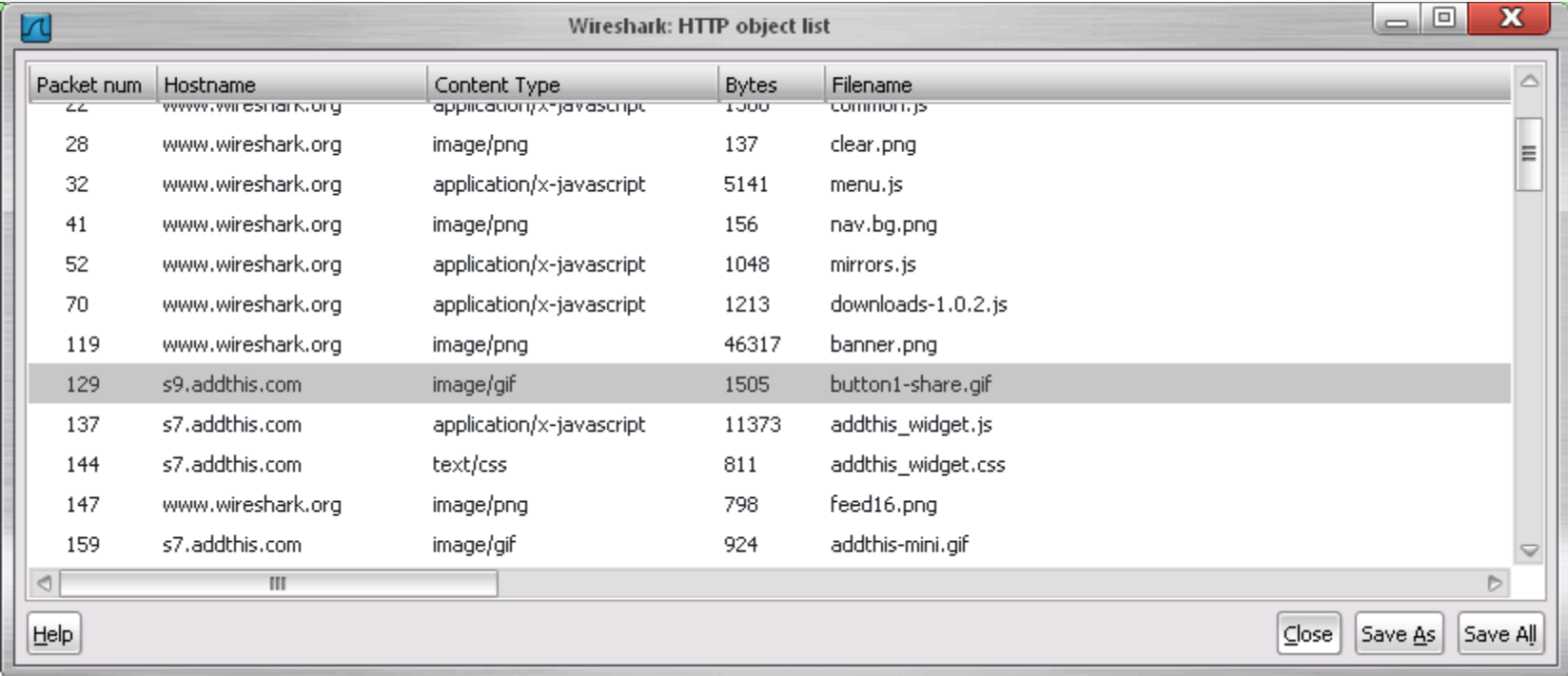
The packet details pane shows the following information for the selected packet:

- 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

The packet bytes pane shows the following hex and ASCII data:

```
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 HTTP/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
0070 72 72 2e 65 64 75 0d 02 55 72 65 72 2d 41 67 65 ss od User Age
```

Export HTTP Objects



The image shows a screenshot of the 'Wireshark: HTTP object list' window. The window contains a table with the following data:

Packet num	Hostname	Content Type	Bytes	Filename
22	www.wireshark.org	application/x-javascript	1300	common.js
28	www.wireshark.org	image/png	137	clear.png
32	www.wireshark.org	application/x-javascript	5141	menu.js
41	www.wireshark.org	image/png	156	nav.bg.png
52	www.wireshark.org	application/x-javascript	1048	mirrors.js
70	www.wireshark.org	application/x-javascript	1213	downloads-1.0.2.js
119	www.wireshark.org	image/png	46317	banner.png
129	s9.addthis.com	image/gif	1505	button1-share.gif
137	s7.addthis.com	application/x-javascript	11373	addthis_widget.js
144	s7.addthis.com	text/css	811	addthis_widget.css
147	www.wireshark.org	image/png	798	feed16.png
159	s7.addthis.com	image/gif	924	addthis-mini.gif

At the bottom of the window, there are buttons for 'Help', 'Close', 'Save As', and 'Save All'.

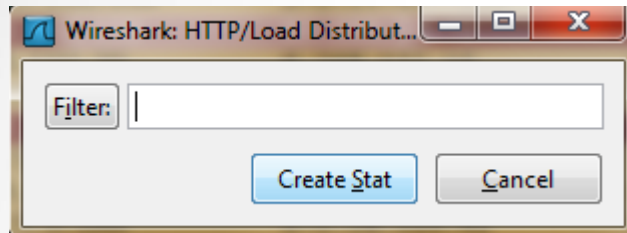
HTTP Analysis

The screenshot shows the Wireshark interface with the following components:

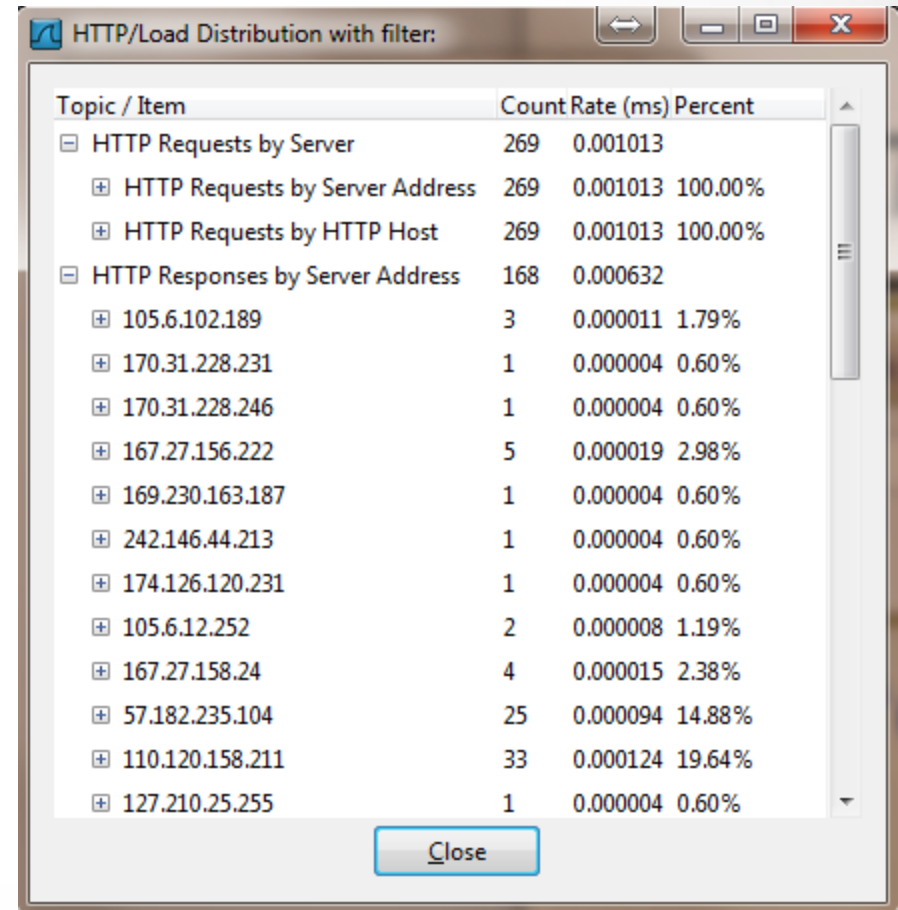
- Filter:** 802.11 Channel: [dropdown]
- Packet List:**

No.	Time	Destination	Protocol	Info
1	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP t
2	2008-07-24 13:12:59.1	108.117.254.150	TCP	acc-raid > http [ACK] Seq=
3	2008-07-24 13:12:59.1	104.2.184.130	HTTP	GET /p/s/sm_vrt_3thumb_scr
4	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP ti
5	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP ti
6	2008-07-24 13:12:59.1	108.117.254.150	TCP	acc-raid > http [ACK] Seq=
7	2008-07-24 13:12:59.1	199.166.161.121	DNS	Standard query A a632.g.ak
8	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP ti
9	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP ti
10	2008-07-24 13:12:59.1	108.117.254.150	TCP	acc-raid > http [ACK] Seq=
11	2008-07-24 13:12:59.1	23.58.126	HTTP	GET /customer/advance/9/.o
12	2008-07-24 13:12:59.1	23.58.126	HTTP	GET /customer/advance/9/.o
13	2008-07-24 13:12:59.2	10.1.15.104	HTTP	Continuation or non-HTTP ti
14	2008-07-24 13:12:59.2	10.1.11.13	DNS	Standard query response CN
15	2008-07-24 13:12:59.1	188.180.195.70	TCP	mcs-calyptoicf > http [SYN
16	2008-07-24 13:12:59.6	10.1.12.67	HTTP	Continuation or non-HTTP ti
17	2008-07-24 13:12:59.1	10.2.101.36	TCP	3325 > http [ACK] Seq=1 Ac
18	2008-07-24 13:12:59.6	10.1.12.67	HTTP	[TCP out-of-order] Continu
19	2008-07-24 13:12:59.1	10.2.101.36	TCP	3325 > http [ACK] Seq=1 Ac
- Context Menu (over packet 18):**
 - 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**
 - Load Distribution...
 - Packet Counter...
 - Requests...
 - IP address...
 - ISUP Messages...
 - Multicast Streams
 - ONC-RPC Programs
 - Packet Length...
 - Port Type...
 - SMPP Operations...
 - TCP Stream Graph
 - WLAN Traffic...
- Status Bar:** File: "C:\Users\vo2.TEW\Desktop\wireshark\sample capture..." Packets: 16612 Displayed: 16612 Marked: 0 Profile: Default

HTTP Analysis – Load Distribution



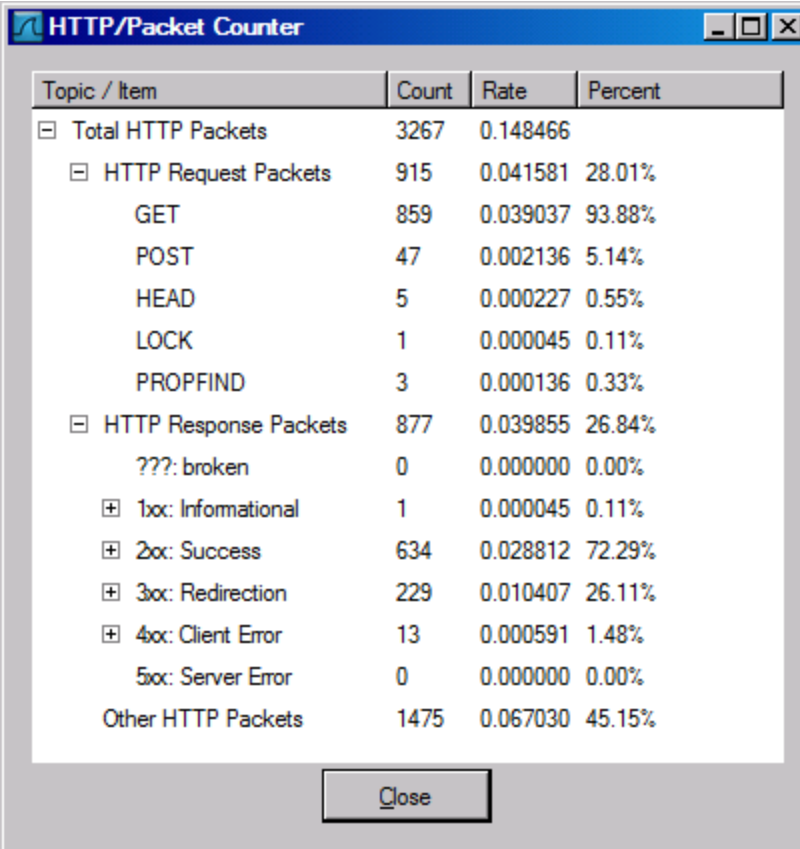
Click “Create Stat” button
You can add “filter” to only
Show selected traffic



A screenshot of the "HTTP/Load Distribution with filter:" window. It displays a tree view of statistics with columns for "Topic / Item", "Count", "Rate (ms)", and "Percent".

Topic / Item	Count	Rate (ms)	Percent
[-] HTTP Requests by Server	269	0.001013	
[+] HTTP Requests by Server Address	269	0.001013	100.00%
[+] HTTP Requests by HTTP Host	269	0.001013	100.00%
[-] HTTP Responses by Server Address	168	0.000632	
[+] 105.6.102.189	3	0.000011	1.79%
[+] 170.31.228.231	1	0.000004	0.60%
[+] 170.31.228.246	1	0.000004	0.60%
[+] 167.27.156.222	5	0.000019	2.98%
[+] 169.230.163.187	1	0.000004	0.60%
[+] 242.146.44.213	1	0.000004	0.60%
[+] 174.126.120.231	1	0.000004	0.60%
[+] 105.6.12.252	2	0.000008	1.19%
[+] 167.27.158.24	4	0.000015	2.38%
[+] 57.182.235.104	25	0.000094	14.88%
[+] 110.120.158.211	33	0.000124	19.64%
[+] 127.210.25.255	1	0.000004	0.60%

HTTP Analysis – Packet Counter



The screenshot shows a window titled "HTTP/Packet Counter" with a table of statistics. The table has four columns: "Topic / Item", "Count", "Rate", and "Percent". The data is organized into a tree structure with expandable/collapsible icons. A "Close" button is located at the bottom center of the window.

Topic / Item	Count	Rate	Percent
[-] Total HTTP Packets	3267	0.148466	
[-] HTTP Request Packets	915	0.041581	28.01%
GET	859	0.039037	93.88%
POST	47	0.002136	5.14%
HEAD	5	0.000227	0.55%
LOCK	1	0.000045	0.11%
PROPFIND	3	0.000136	0.33%
[-] HTTP Response Packets	877	0.039855	26.84%
???: broken	0	0.000000	0.00%
[+] 1xx: Informational	1	0.000045	0.11%
[+] 2xx: Success	634	0.028812	72.29%
[+] 3xx: Redirection	229	0.010407	26.11%
[+] 4xx: Client Error	13	0.000591	1.48%
5xx: Server Error	0	0.000000	0.00%
Other HTTP Packets	1475	0.067030	45.15%

HTTP Analysis – Requests

HTTP/Requests	
Topic / Item	
[-]	HTTP Requests by HTTP Host
[+]	img.video.ap.org
[+]	mi.adinterax.com
[+]	blog.cleveland.com
[+]	tr.adinterax.com
[-]	money.cleveland.com
	/dynamic/proxy-partial-js/ibd.momingstar.com/AP/MarketIndexGraph.html?
[-]	www.cleveland.com
	/images/hp/video.gif
	/sports/graphics/audio_blue.gif
	/sports/graphics/gallery.gif
	/sports/graphics/comment.gif
	/images/hp/80/jackson.jpg
	/images/hp/80/coupons_80.jpg
	/images/hp/110/crime_scene.jpg
	/images/hp/110/gavel.jpg
	/images/hp/110/cafeteria110.jpg
	/images/hp/110/blake0901ap.jpg

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!