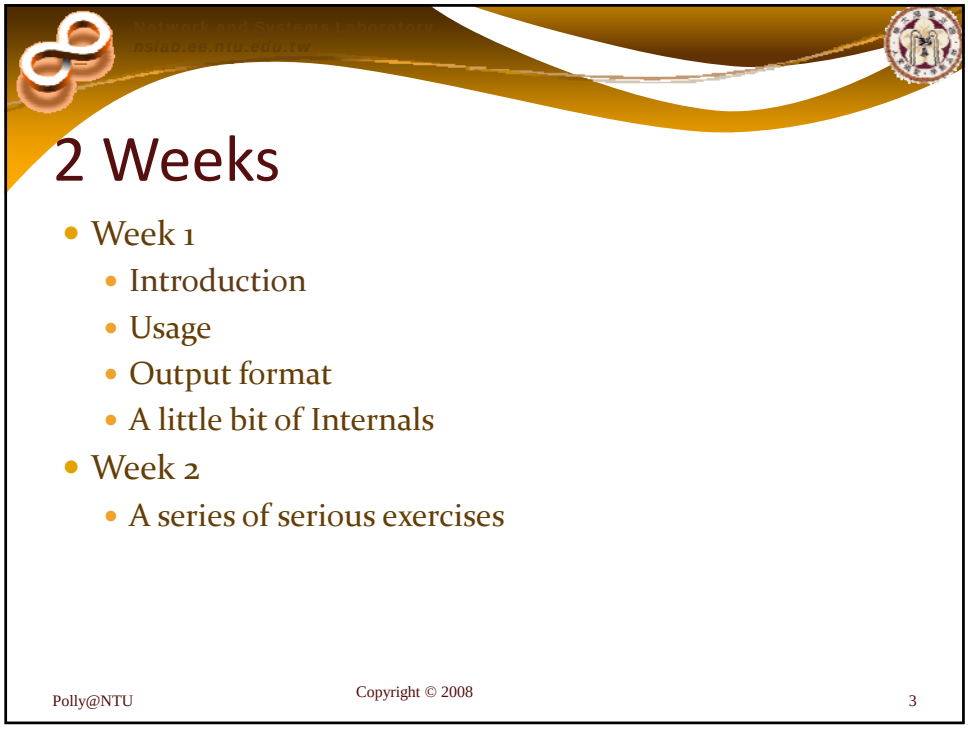
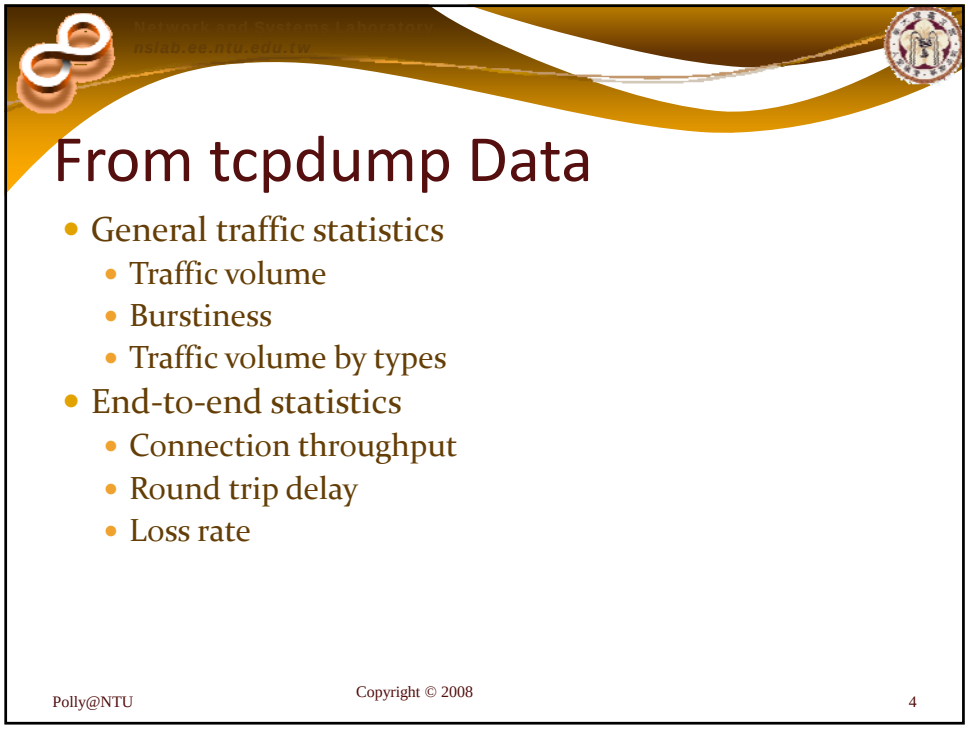
Network Simulation and Testing

Polly Huang
Department of Electrical Engineering
National Taiwan University
http://cc.ee.ntu.edu.tw/~phuang
phuang@cc.ee.ntu.edu.tw

Getting Real Network Data

tcpdump


Network and Systems Laboratory
nslab.ee.ntu.edu.tw
2 Weeks
Week 1
Introduction
Usage
Output format
A little bit of Internals
Week 2
A series of serious exercises
Polly@NTU
Copyright © 2008
3


Network and Systems Laboratory
nslab.ee.ntu.edu.tw
From tcpdump Data
General traffic statistics
Traffic volume
Burstiness
Traffic volume by types
End-to-end statistics
Connection throughput
Round trip delay
Loss rate
Polly@NTU
Copyright © 2008
4

tcpdump

- A packet tracing tool
  - Work on various host platforms
  - Capture packets going through a certain network interface
  - Show packet header information

Platforms & Access

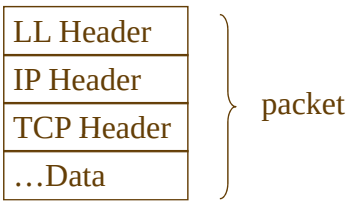
- Unix
  - BSD: read access to /dev/bpf*
  - SunOS: read access to /dev/bpf*
  - Linux: root
  - Solaris: read/write access to /dev/le (root)
  - etc
- Windows
  - WinDump

Network Interfaces

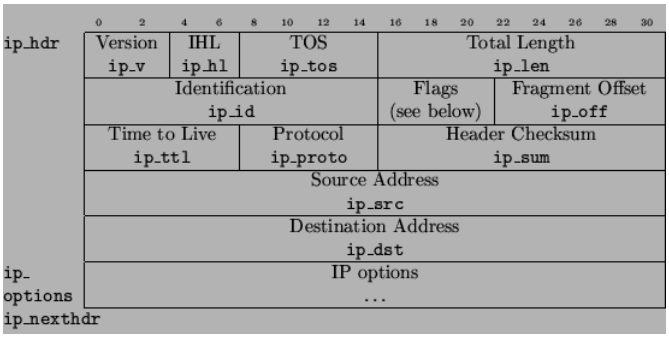
```
phuang@NSLabServer:~$ /sbin/ifconfig
eth0   Link encap:Ethernet  HWaddr 00:50:FC:35:07:52
       inet addr:140.112.154.170  Bcast:140.112.154.255  Mask:255.255.255.0
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:34519670 errors:0 dropped:0 overruns:0 frame:0
       TX packets:5817625 errors:11 dropped:0 overruns:6 carrier:9
       collisions:499737 txqueuelen:100
       RX bytes:3467682476 (3.2 GiB)  TX bytes:3249195405 (3.0 GiB)
       Interrupt:11 Base address:0xcc00

lo     Link encap:Local Loopback
       inet addr:127.0.0.1  Mask:255.0.0.0
       UP LOOPBACK RUNNING  MTU:3924  Metric:1
       RX packets:1269 errors:0 dropped:0 overruns:0 frame:0
       TX packets:1269 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:0
       RX bytes:80314 (78.4 KiB)  TX bytes:80314 (78.4 KiB)
```


Packet Headers

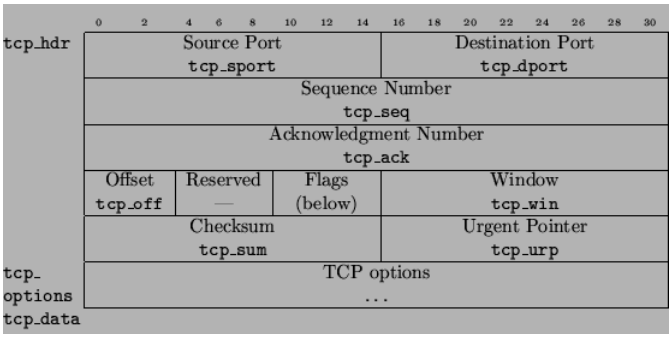
- Link layer headers vary
- IPv4, IPv6 headers
- TCP, UDP headers

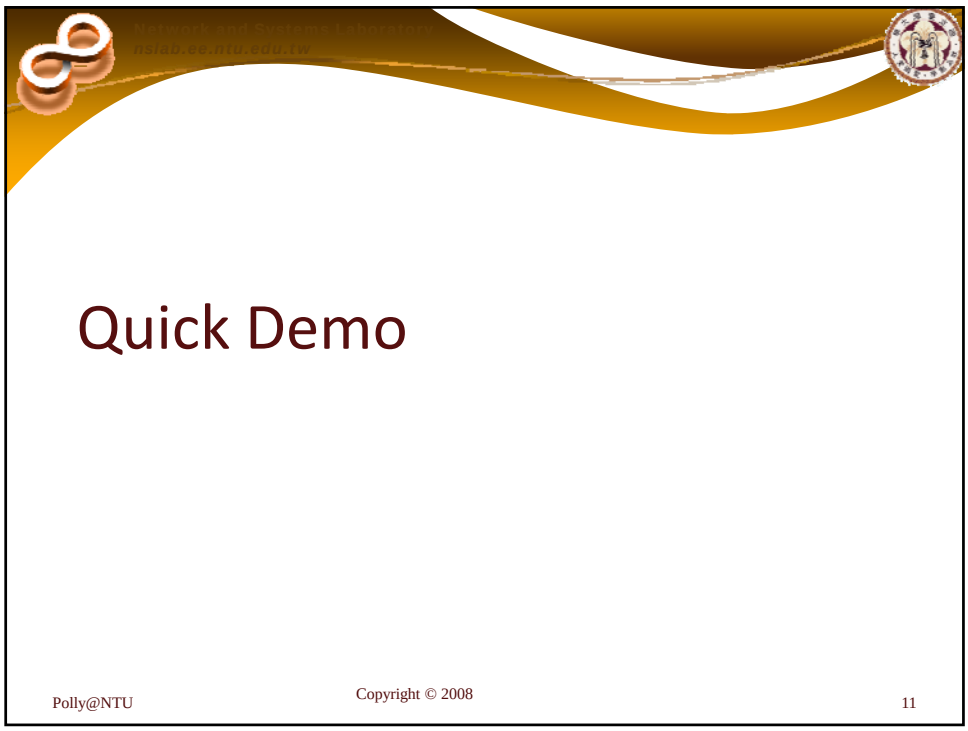


IPv4 Header

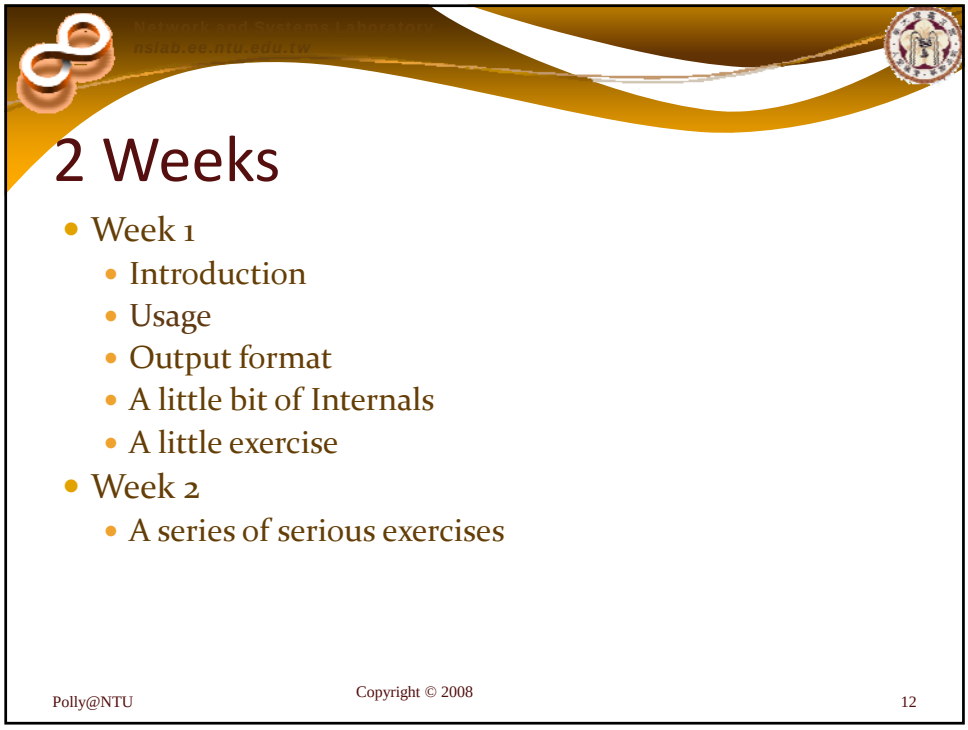


TCP Header





Quick Demo



2 Weeks

- Week 1
  - Introduction
  - Usage
  - Output format
  - A little bit of Internals
  - A little exercise
- Week 2
  - A series of serious exercises

Usage

tcpdump [**-AdDeflLnNOpqRStuUvxX**]
[**-i** *interface*][**-c** *count*]
[**-w** *file*] [**-C** *file_size*]
[**-r** *file*]
[**-T** *type*] [**-s** *snaplen*]
[**-m** *module*] [**-E** *algo:secret*] [**-y** datalinktype]
[**-F** *file*] [*expression*]

[**-i** *interface*]

- To read packets from a certain network interface

tcpdump –i eth0

[-c *count*]

- To read up to *count* number of packets

tcpdump –i eth0 –c 5

[-w *file*]

- To write the output to a file
- Instead of printing to the screen the packet header information

tcpdump –i eth0 –c 5 –w tmp.tr

[**-C** *file_size*]

- To output to files up to *file_size* million bytes
- When tmp.tr exceeds *file_size* MB, tmp.tr2 is opened to continue tracing

tcpdump –i eth0 –c 5 –w tmp.tr –C 1

[**-r** *file*]

- To read packets from a file
- Generated from [**-w** *file*]

tcpdump –r tmp.tr

[*expression*]

- To select packets to be read
- Types, directions, protocols
  - [*protocol*][*direction*][*type*]

**tcpdump –i eth0 –c 5 –w tmp.tr –C 100 **
[*expression*]

Expression: Type

- Selecting packets of a particular host, particular network, particular port
- {**host, net, port**} [{*name, number*}]

host 140.112.42.162
net 140.112.42
port 22

Expression: Direction

- Selecting packets of a particular direction, inbound or outbound
- {**src, dst, src or dst, src and dst**}[*type*]

src or dst host 140.112.42.162
dst net 140.112.42
dst port 22

Expression: Protocol

- Selecting packets of a particular protocol
- **{ether**, **ip**, **ip6**, **arp**, **rarp**, **tcp**, **udp**, …}{**multicast**, **broadcast**}

ip src or dst host 140.112.42.162
arp dst net 140.112.42
tcp dst port 22

Expression: Others

- Selecting packets of particular sizes in bytes
- {**greater, less**}[*size*]
- **len** {**>=**, **<=**}[*size*]

Expression: Operands

- **!** or **not**
- **&&** or **and**
- **||** or **or**

ip host nslab and \(140.112.42.162 or cc.ee.ntu.edu.tw \)

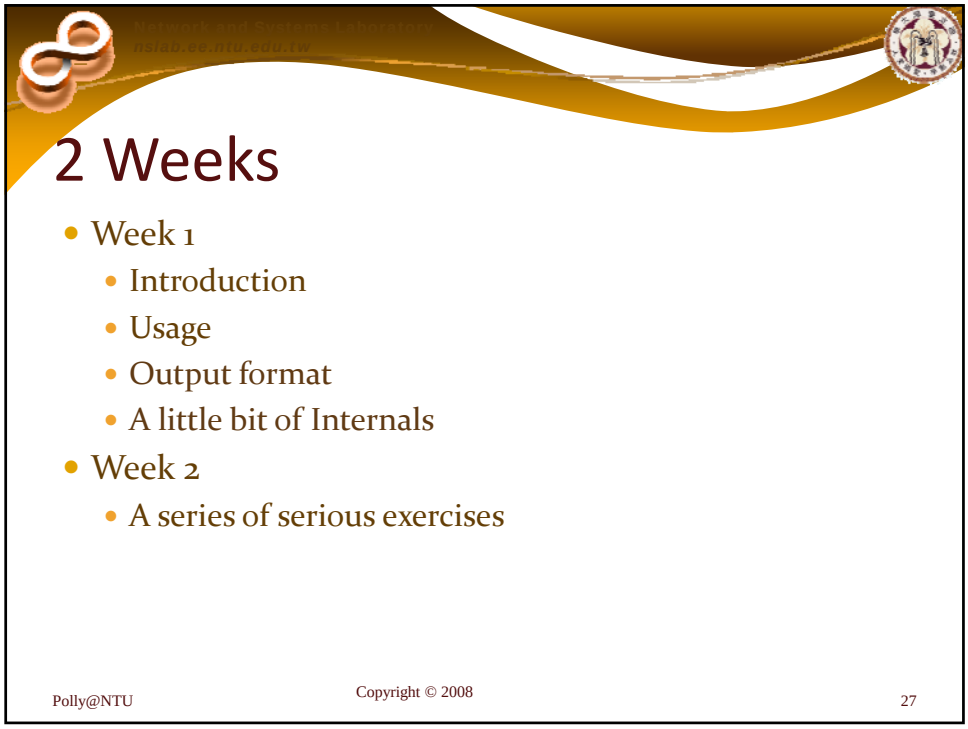
[-F *file*]

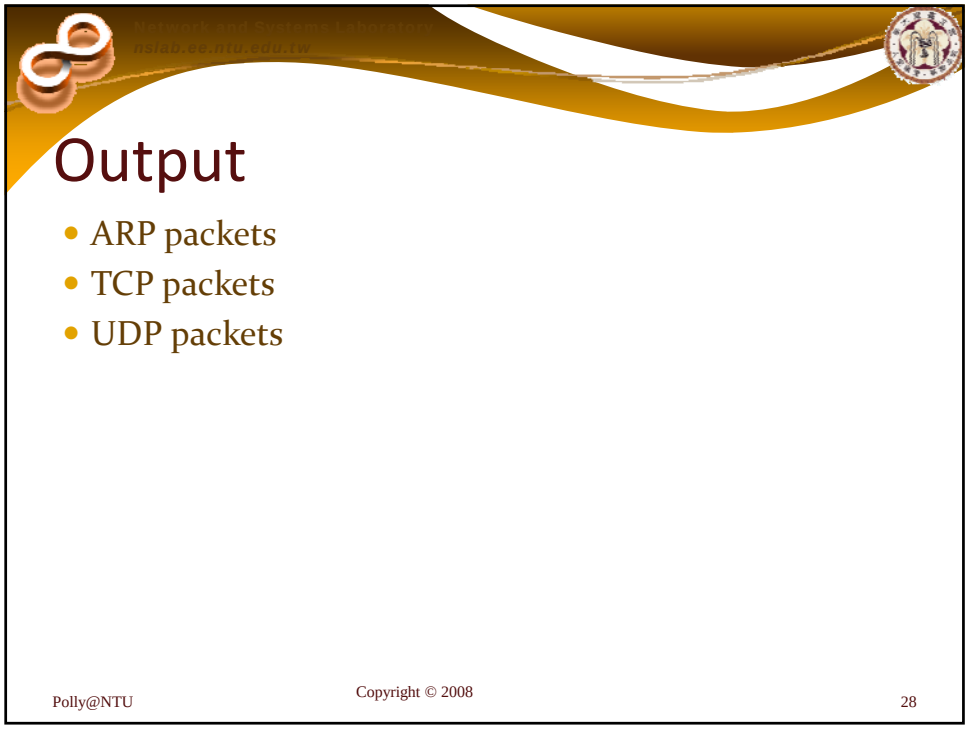
- To load [*expression*] from a file

tcpdump –i eth0 –c 5 –w tmp.tr –C 100 –F test.exp

Additional Flags

- **-n**: no converting IP to hostname
- **-N**: no domain
- **-e**: ethernet details
- **-q**, **-v**, **-vv**, **-vvv**: compact to verbose output
- **-t**, **-tt**, **-ttt**: no time, unformatted, delta
- **-S**: absolute sequence number for TCP
- **-l**: buffer output to pipeline
- **-p**: no promiscuous mode


Network and Systems Laboratory
nslab.ee.ntu.edu.tw
2 Weeks
Week 1
Introduction
Usage
Output format
A little bit of Internals
Week 2
A series of serious exercises
Polly@NTU
Copyright © 2008
27


Network and Systems Laboratory
nslab.ee.ntu.edu.tw
Output
ARP packets
TCP packets
UDP packets
Polly@NTU
Copyright © 2008
28

ARP Packets

arp who-has 128.3.254.6 tell 128.3.254.68
arp reply 128.3.254.6 is-at 02:07:01:00:01:c4

who-has: requests
reply: replies

TCP Packets

src > dst: <flags> <data-seqno> <ack> <window>
<urgent> <options>

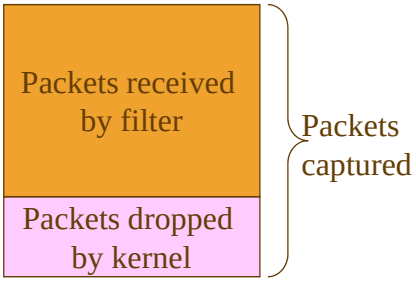
Flags:
S (SYN), F (FIN), P (PUSH), R (RST)
W (ECN CWR) , E (ECN-Echo),
. (no flags)

UDP Packets

src > dst: UDP length:<size>
src > dst: RIPv1 <packet type> length:<size>
src > dst: NBT UDP PACKET(<packet #>): <type>

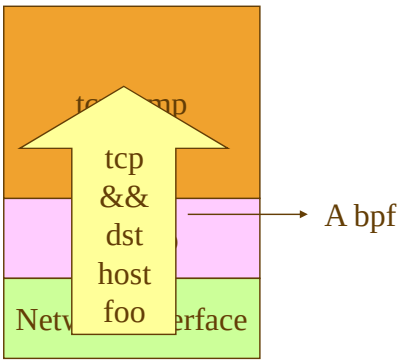
Final Output

- # packet captured
  - All packets going thru the interface
- # packet received by filter
  - Packets in tcpdump output
- # packet dropped by kernel
  - Packets not in tcpdump output



Internals

- Filter?
  - bpf
  - Berkeley packet filter
- Kernel?
  - libpcap
  - Packet capturing library



Questions?