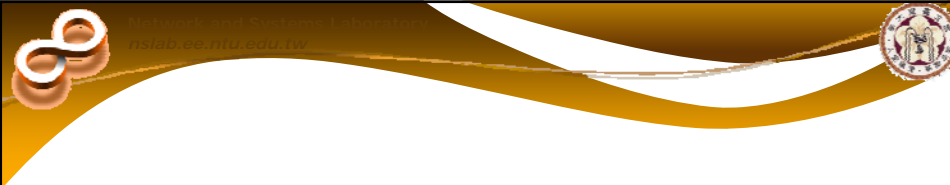


# Network Simulation and Testing

Polly Huang  
Department of Electrical Engineering  
National Taiwan University  
<http://cc.ee.ntu.edu.tw/~phuang>  
[phuang@cc.ee.ntu.edu.tw](mailto:phuang@cc.ee.ntu.edu.tw)


Polly@NTU Copyright © 2008 1



# Getting Real Network Data

`tcpdump`

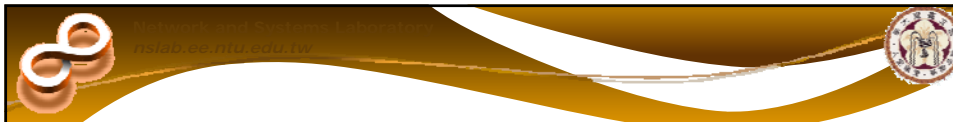
Polly@NTU Copyright © 2008 2



## 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



## From tcpdump Data

- General traffic statistics
  - Traffic volume
  - Burstiness
  - Traffic volume by types
- End-to-end statistics
  - Connection throughput
  - Round trip delay
  - Loss rate

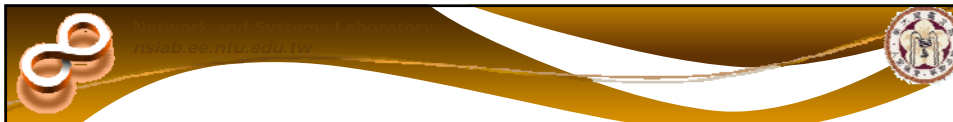
Polly@NTU Copyright © 2008 4



## tcpdump

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


Polly@NTU Copyright © 2008 5



## 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

Polly@NTU Copyright © 2008 6




# 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)
    
```

Polly@NTU Copyright © 2008 7



# Packet Headers

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

LL Header
IP Header
TCP Header
...Data

} packet

Polly@NTU Copyright © 2008 8

# IPv4 Header

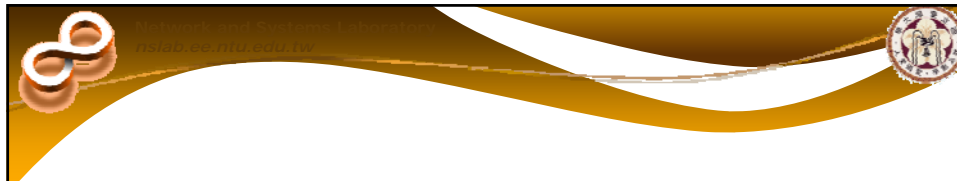
	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
ip_hdr	Version	IHL		TOS		Total Length										
	ip_v	ip_hl		ip_tos		ip_len										
	Identification						Flags			Fragment Offset						
	ip_id						(see below)			ip_off						
	Time to Live			Protocol			Header Checksum									
	ip_ttl			ip_proto			ip_sum									
	Source Address															
	ip_src															
	Destination Address															
	ip_dst															
ip- options	IP options															
ip_nexthdr	...															

Polly@NTU
Copyright © 2008
9

# TCP Header


	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
tcp_hdr	Source Port								Destination Port							
	tcp_sport								tcp_dport							
	Sequence Number															
	tcp_seq															
	Acknowledgment Number															
	tcp_ack															
	Offset		Reserved		Flags				Window							
	tcp_off		—		(below)				tcp_win							
	Checksum								Urgent Pointer							
	tcp_sum								tcp_urp							
tcp- options	TCP options															
tcp_data	...															

Polly@NTU
Copyright © 2008
10



# Quick Demo


Polly@NTU Copyright © 2008 11



# 2 Weeks

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

Polly@NTU Copyright © 2008 12



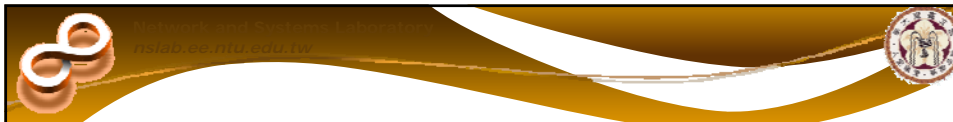
## Usage

```

tcpdump [ -AdDefILnNOpqRStuUvxX ]
[ -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 ]

```

Polly@NTU Copyright © 2008 13



## [ **-i** *interface* ]


- To read packets from a certain network interface

```

tcpdump -i eth0

```

Polly@NTU Copyright © 2008 14




**[ -c *count* ]**

- To read up to *count* number of packets

```
tcpdump -i eth0 -c 5
```

Polly@NTU Copyright © 2008 15



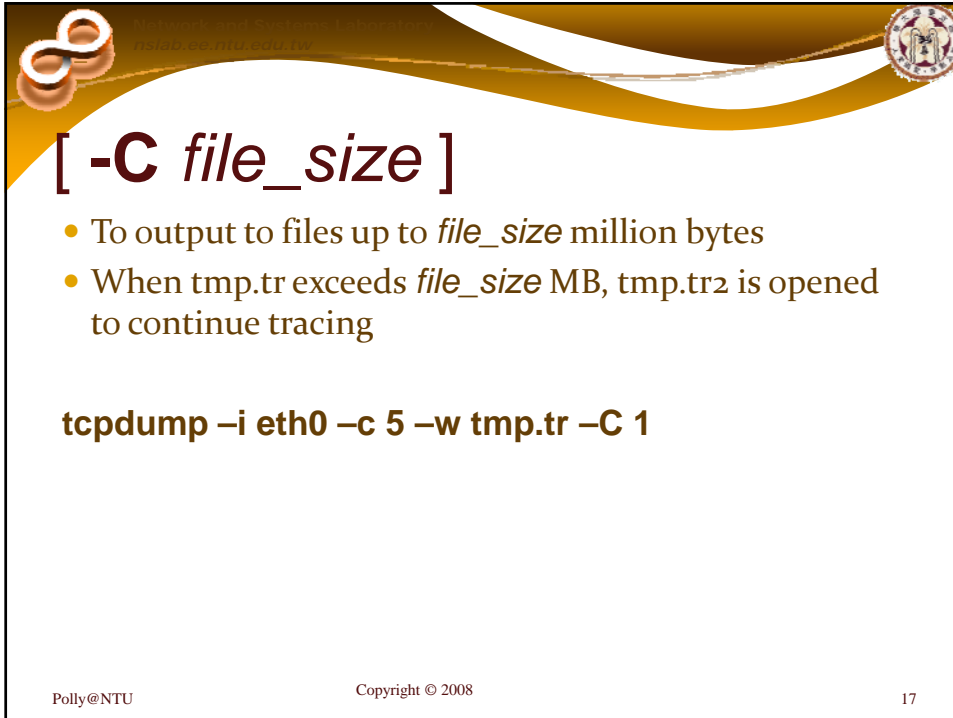
**[ -w *file* ]**

- 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
```

Polly@NTU Copyright © 2008 16



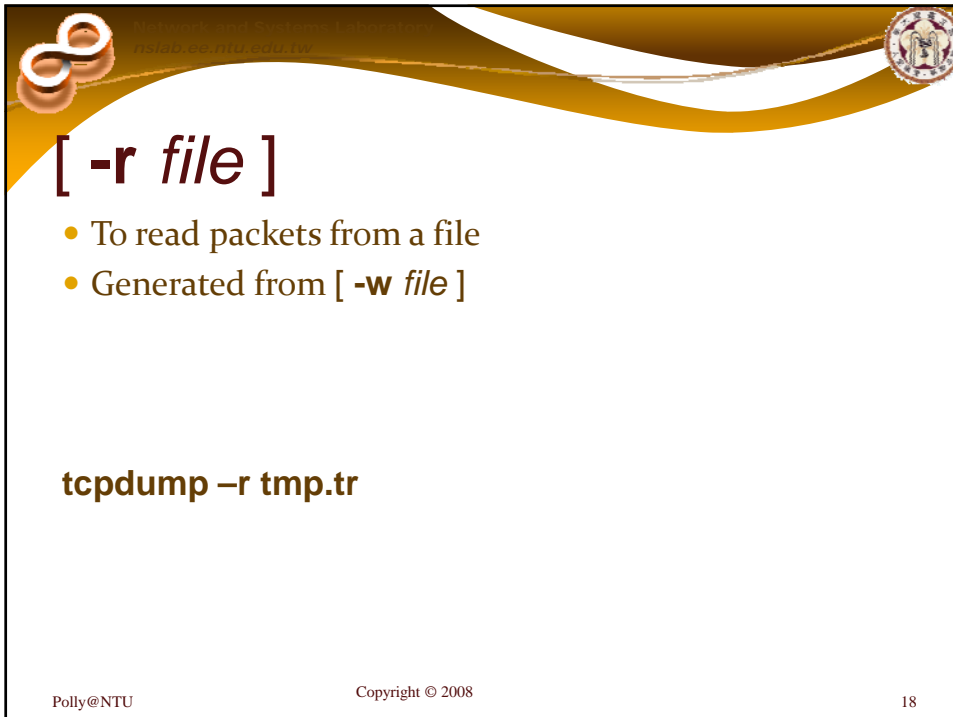


## [ **-C** *file\_size* ]

- 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
```

Polly@NTU Copyright © 2008 17




## [ **-r** *file* ]

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

```
tcpdump -r tmp.tr
```

Polly@NTU Copyright © 2008 18

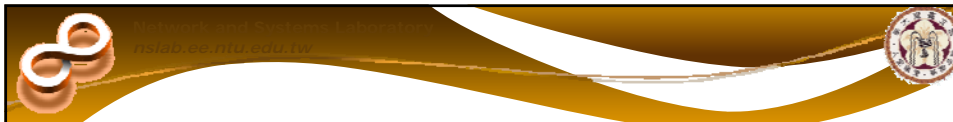


## [ *expression* ]

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

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

Polly@NTU Copyright © 2008 19




## 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**

Polly@NTU Copyright © 2008 20

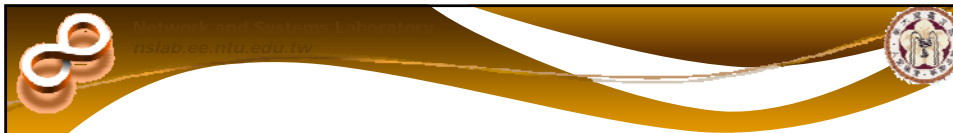


## Expression: Direction

- 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**

Polly@NTU Copyright © 2008 21




## 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**

Polly@NTU Copyright © 2008 22



## Expression: Others

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

Polly@NTU Copyright © 2008 23




## Expression: Operands

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

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

Polly@NTU Copyright © 2008 24

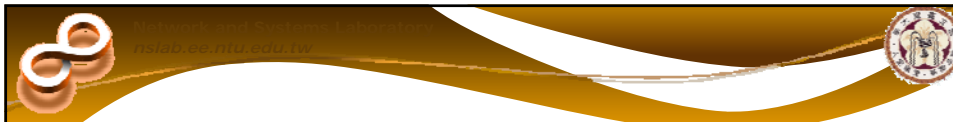


## [ **-F** *file* ]

- To load [*expression*] from a file

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


Polly@NTU Copyright © 2008 25



## 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

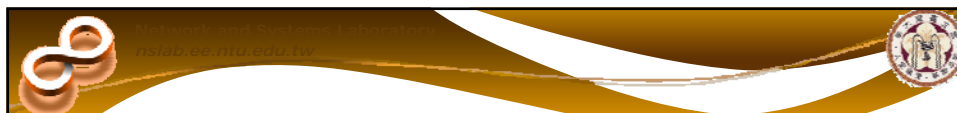
Polly@NTU Copyright © 2008 26



## 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



## 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

Polly@NTU Copyright © 2008 29




## 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)


Polly@NTU Copyright © 2008 30



# UDP Packets

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

Polly@NTU Copyright © 2008 31



# 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

Packets received by filter

Packets dropped by kernel

}

Packets captured

Polly@NTU Copyright © 2008 32



# Internals

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

The diagram illustrates a network stack with four layers. From bottom to top: a green layer labeled 'Network interface', a purple layer labeled 'host', a yellow layer labeled 'tcp & dst', and an orange layer labeled 'tcp & src'. A yellow arrow points upwards from the 'host' layer to the 'tcp & dst' layer, with the text 'A bpf' to its right.

Polly@NTU Copyright © 2008 33

# Questions?

Polly@NTU Copyright © 2008 34