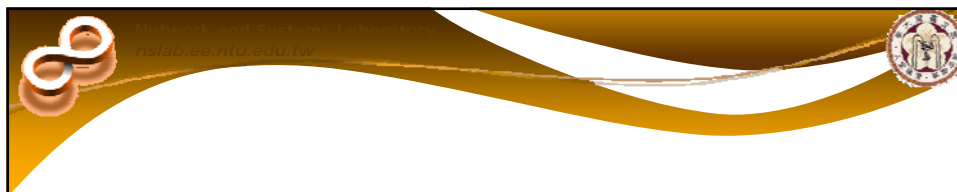


Network Simulation and Testing

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

Polly@NTU Copyright © 2008 1



Testing on Real Systems

dummynet

Polly@NTU Copyright © 2008 2

Let's Take a Good Look Back

The Internet

- For the Internet
 - Monitor the usage
 - Passive and active measurement
 - Characterize the workload
 - Traffic, topology, routing errors, access pattern modeling
 - Predict for the future
 - Scalable simulation & testing tools
 - Revise original design
 - Protocol and Infrastructure
 - Instrument the changes
 - IETF

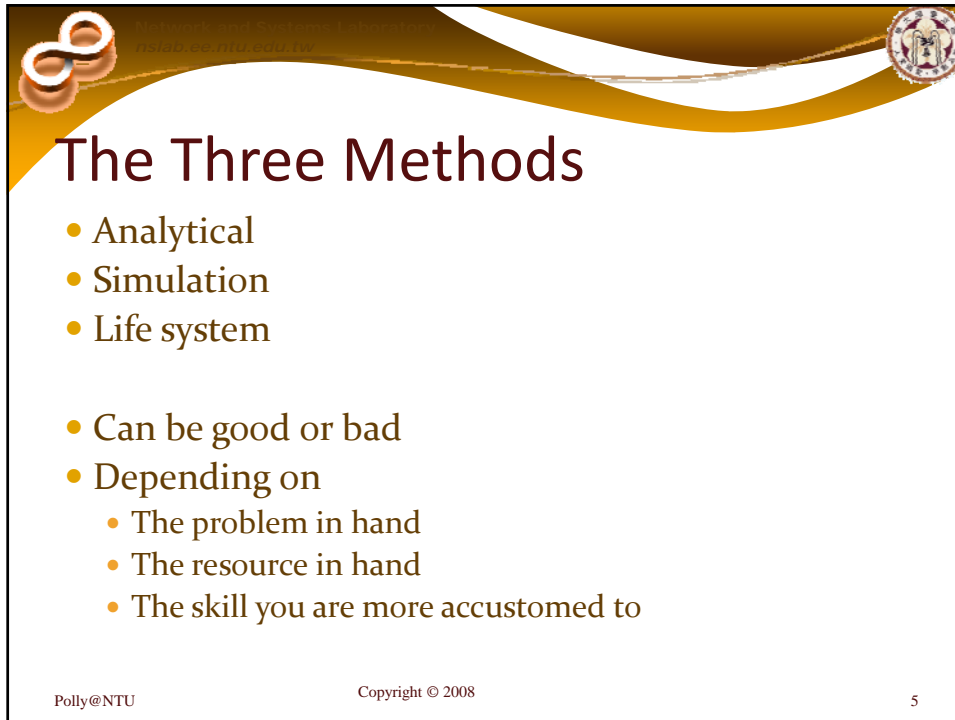
This course

Polly@NTU Copyright © 2008 3

Predict for the Future

- Nature
 - Find out whether a system is going to work!
- A.k.a
 - The **Performance Evaluation** problem

Polly@NTU Copyright © 2008 4

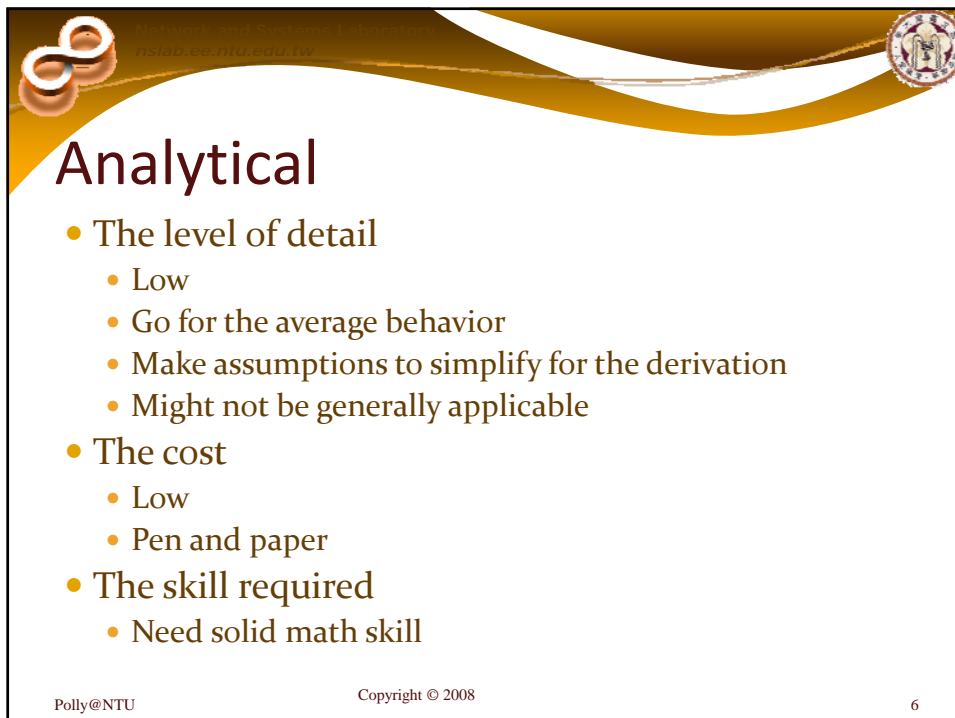


The Three Methods

- Analytical
- Simulation
- Life system

- Can be good or bad
- Depending on
 - The problem in hand
 - The resource in hand
 - The skill you are more accustomed to


Polly@NTU Copyright © 2008 5



Analytical

- The level of detail
 - Low
 - Go for the average behavior
 - Make assumptions to simplify for the derivation
 - Might not be generally applicable
- The cost
 - Low
 - Pen and paper
- The skill required
 - Need solid math skill


Polly@NTU Copyright © 2008 6



Simulation

- The level of detail
 - Medium but flexible
 - Can go with various levels of details
 - Can go large scale
- The cost
 - Medium
 - Usually 1 computer
- The required Skill
 - Need solid programming skill
 - Vivid imagination


Polly@NTU Copyright © 2008 7



Live Testing

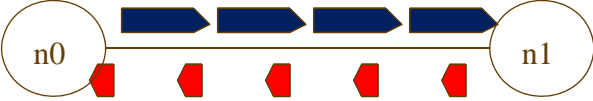
- The level of detail
 - High
 - Go with full detail
 - Difficult to go large scale
- The cost
 - High
 - Multiple computers
- The required skill
 - Need solid system skill
 - Vivid imagination

Polly@NTU Copyright © 2008 8



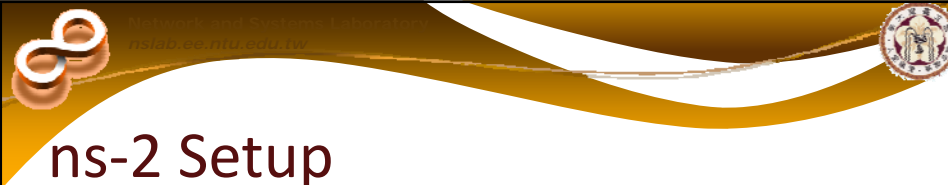
For Example

Obtaining TCP Throughput



The diagram illustrates a network link between two nodes, n0 and n1. Blue arrows represent data packets being sent from n0 to n1. Red pentagons represent packets being dropped at the destination n1.

Polly@NTU Copyright © 2008 9




ns-2 Setup

```

set ns [new Simulator]
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1.5Mb
  10ms DropTail
set tcp [$ns create-connection
  TCP $n0 TCPSink $n1 0]
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 0.2 "$ftp start"
$ns at 1.2 "exit"
$ns run

```

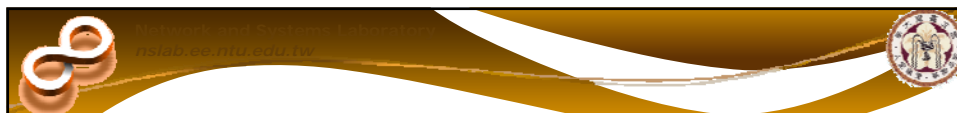
Polly@NTU Copyright © 2008 10



BW with ns-2

- duplex-link ...
- Run the script
- Obtain the output file
- Process the output file
 - Add the number of data bytes
 - Divide by the time duration


Polly@NTU Copyright © 2008 11



BW with dummynet

- We will figure this out during the coming 3 weeks

Polly@NTU Copyright © 2008 12



2 Weeks

- Week 1
 - Introduction, basic features
 - Setup and bandwidth management
- Week 2
 - Testing over a simple network

Polly@NTU Copyright © 2008 13




Today

- Introduction
- Basic features

- Largely based on Luigi's slides

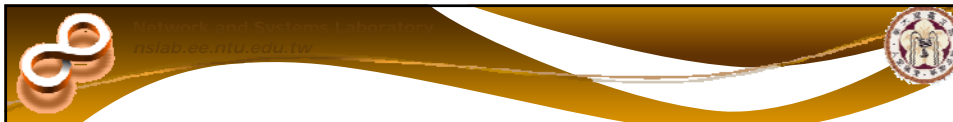
Polly@NTU Copyright © 2008 14



Introduction

- dummynet as a software
 - Part of FreeBSD kernel
- dummynet as an emulation tool
 - Can do various bandwidth, delay, loss
- dummynet as a traffic filter
 - Real-time on live traffic
 - Filter on local and transit traffic
 - Filter at levels as fine as flows
- dummynet works also as a traffic shaper
 - For example, to limit the bandwidth of certain flows


Polly@NTU Copyright © 2008 15



Motivation

- To do repeatable live code experiments
 - ns-2 simulations might ignore certain details
 - For example, the processing time
 - Live experiments not in controlled environment
 - Hard to track and re-produce
 - Expensive to build
 - Certain network components or protocols cannot be modeled
 - For example, the Microsoft Windows TCP, Skype


Polly@NTU Copyright © 2008 16



Research Applications

- Small-scale performance testing
 - Behavior in the presence of loss, delay, bandwidth limitation
 - Live user access, client-server interaction
 - Evaluation of metrics subject to user perception
 - Audio/video quality
 - Validation of simulation results


Polly@NTU Copyright © 2008 17



Production Applications

- Bandwidth limitation
 - Restricting services
 - Protecting networks
 - Probably the most widely used application of dummynet
- Product/Protocol testing
 - Slightly different goal from research applications
 - Checking whether the implementation complies with the specification
 - Checking whether the product will work in the field of heterogeneous network conditions

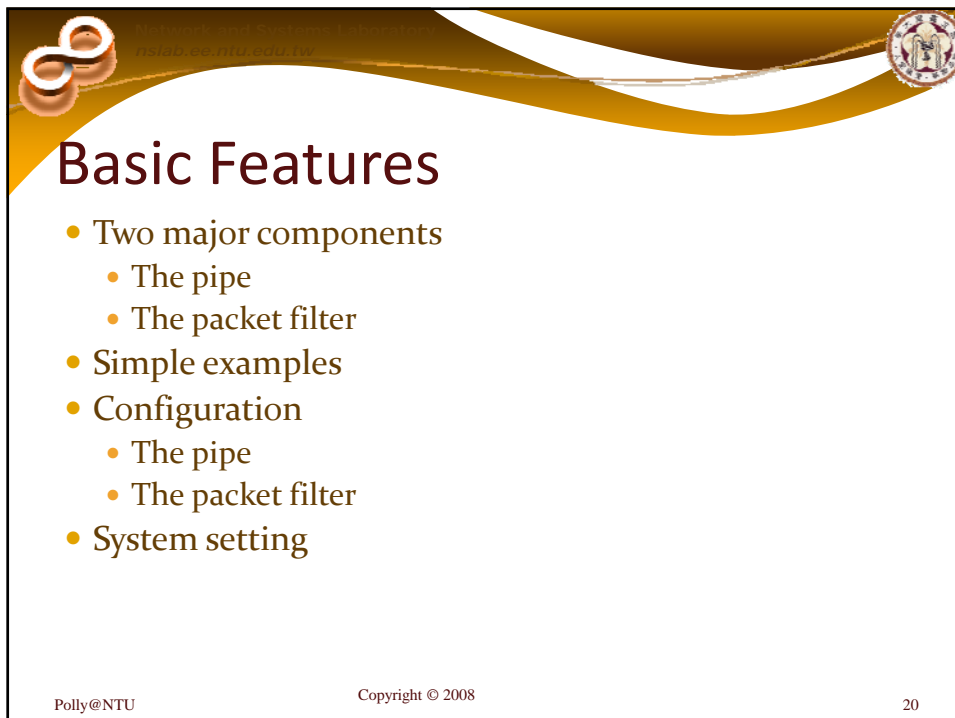
Polly@NTU Copyright © 2008 18



Design Goals

- Simplicity and ease of use
 - Will see in usage
- Be realistic
 - To see the network behavior as the results of how it is implemented, structured, and driven in the real world
- Flexibility and extendibility
 - Orthogonal features can be added over time
 - Will see in basic vs. advanced features

Polly@NTU Copyright © 2008 19




Basic Features

- Two major components
 - The pipe
 - The packet filter
- Simple examples
- Configuration
 - The pipe
 - The packet filter
- System setting

Polly@NTU Copyright © 2008 20

The Pipe




- Emulates a communication link
- Configurable with the usual pipe link parameters

pipe 10 config bw 500 Kbits/s delay 12ms plr 0.02

}
}
}
}
 pipe ID bandwidth delay loss

Polly@NTU Copyright © 2008 21

Optional Queue




- Input queue
 - Essentially a leaky bucket
 - Why it's used frequently as a traffic shaper

pipe 10 config bw 500 Kbits/s queue 30KB delay 12ms plr 0.02

}
queue size

Polly@NTU Copyright © 2008 22



The Packet Filter


- Use the FreeBSD firewall – ipfw
- Set ipfw rules to select packets that will go through the pipe

```

pipe 10 tcp from any to any 80 in
         {   {       {   {   {
         type from to port direction

```

Polly@NTU Copyright © 2008 23




A Simple Example

```

ipfw add 01100 allow pipe 10 tcp from any to any 80 in
ipfw pipe 10 config bw 500 Kbits/s delay 12ms plr 0.02

```

Polly@NTU Copyright © 2008 24

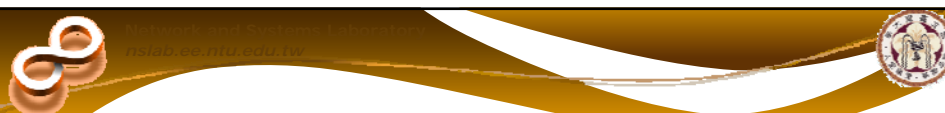


Emulating a Multi-Access Link

- One pipe
- Data going both direction content for 10Mbps bandwidth

ipfw add pipe 1 ip from any to any
ipfw pipe 1 config bw 10 Mbits/s

Polly@NTU Copyright © 2008 25

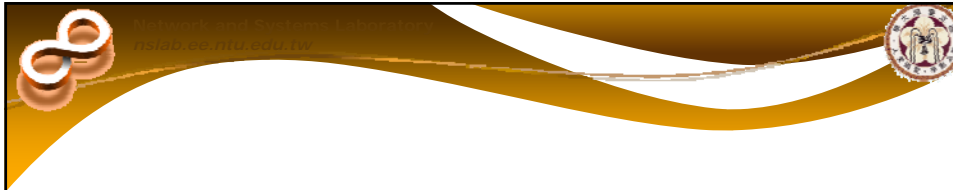


Emulating a Point-to-Point Link

- Two pipes
- One for each direction

ipfw add pipe 1 ip from any to any in
ipfw add pipe 2 ip from any to any out ← Modem downlink
ipfw pipe 1 config bw 640 Kbits/s
ipfw pipe 2 config bw 128 Kbits/s ←


Polly@NTU Copyright © 2008 26



Configuring Pipes

Bandwidth, Delay, Loss, Queue

Polly@NTU Copyright © 2008 27



1. Bandwidth

- The rate the packets are extracted from the queue
 - Unit: bit/s, Kbit/s, Mbit/s, Bytes/s, KBytes/s, MBytes/s
 - Packet size: up to IP
- Note
 - Used by production applications a lot

Polly@NTU Copyright © 2008 28

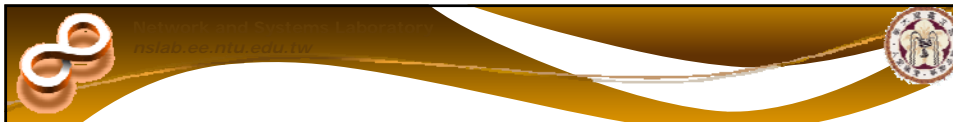


2. Delay

- Propagation delay
 - Unit: millisecond

- Note
 - Mainly used for performance evaluation

Polly@NTU Copyright © 2008 29




3. Loss

- Random packet loss
 - Range: 0 to 1
 - Granularity: 0.01

- Note
 - Used mainly for performance evaluation
 - No significant drops unless the traffic rate is extremely high

Polly@NTU Copyright © 2008 30

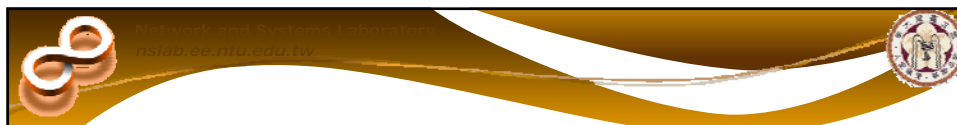


4. Queue

- Input Queue
 - Size unit: packets or Bytes, KBytes

- Note
 - Extra queuing add up to the delay
 - Typical Ethernet queues: 50 packets


Polly@NTU Copyright © 2008 31



Configuring Packet Filters

ipfw basic
Setting for types of machines


Polly@NTU Copyright © 2008 32



ipfw

- A simple firewall utility function in FreeBSD
- Specifying rules to set up packet filters

Polly@NTU Copyright © 2008 33




ipfw: Adding Filters

ipfw add [N] [prob X] [action] [PROTO] from [SRC] to [DST] [OPTIONS]

[N]: rule number (indexing)
 [prob X]: 0 to 1, default = 1
 [action]: allow, deny, **pipe N**
 N: pipe ID, 1 to 65534

[PROTO]: ip, tcp, udp, icmp etc
 [SRC] [DST]: address with netmask, port, range of ports
 [OPTIONS]: various protocol option

Polly@NTU Copyright © 2008 34




Rule Examples

```
ipfw add 01012 allow tcp from 10.0.1.0/8 to 10.0.1.1 139
ipfw add 01012 allow udp from 10.0.1.0/8 to 10.0.1.1 137,138

ipfw add 02000 allow udp from any to any 4000-65535,domain,ntalk,ntp
ipfw add 02500 allow udp from any to any frag

ipfw add 03000 allow tcp from any to any http,https
ipfw add 03000 allow tcp from any to any 4000-65535,ssh,smtp,domain,ntalk
ipfw add 03000 allow tcp from any to any auth,pop3,ftp,ftp-data
```

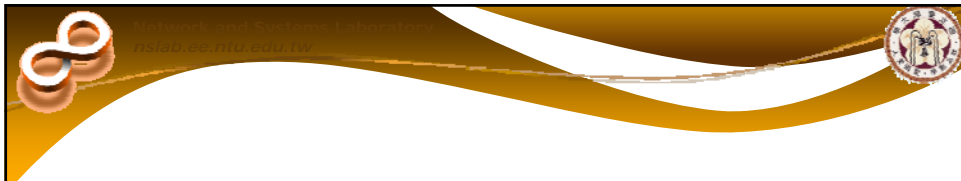
Polly@NTU Copyright © 2008 35



ipfw: Manipulating Filters

- **ipfw pipe 1 show**
 - Show parameters of pipe 1
- **ipfw pipe 1 delete**
 - Delete pipe 1
- **ipfw pipe flush**
 - Delete all pipes


Polly@NTU Copyright © 2008 36



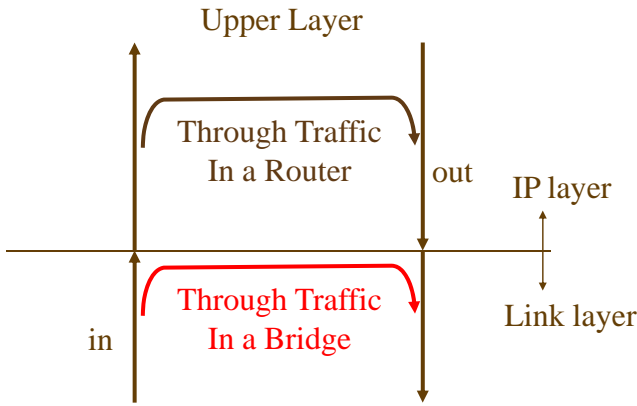
Setting for Types of Machines

Hosts
Routers
Bridges

Polly@NTU Copyright © 2008 37



Be Aware of the Data Flow



Upper Layer

Through Traffic In a Router

out

IP layer

in

Through Traffic In a Bridge

Link layer

Device Driver

Polly@NTU Copyright © 2008 38

1. Easy for the Hosts

in 1 out 2 IP layer Link layer

ipfw add pipe 1 ip from any to any in
ipfw add pipe 2 ip from any to any out

39

In Case of Routers:

in 1 out 2 IP layer Link layer

Through Traffic In a Router

Problem?

ipfw add pipe 1 ip from any to any in
ipfw add pipe 2 ip from any to any out

40

Yes, Problem

- The problem
 - Through traffic get filtered twice
- The solution
 - Deal with through traffic separately

ipfw add pipe 1 ip from any to "my IP" in
 ipfw add pipe 2 ip from "my IP" to any out
 ipfw add pipe 3 ip from any to not "my IP" in

Polly@NTU Copyright © 2008 41

2. Careful for the Routers

The diagram illustrates traffic flow in a router. A horizontal line represents the boundary between the IP layer (top) and the Link layer (bottom). Above the line, three orange boxes labeled 1, 2, and 3 represent firewall pipes. Pipe 1 is at the 'in' interface, pipe 2 is at the 'out' interface, and pipe 3 is in the middle. A curved arrow labeled 'Through Traffic In a Router' shows traffic entering at pipe 1, passing through pipe 3, and exiting at pipe 2. A vertical double-headed arrow on the right indicates the transition between the IP layer and the Link layer.

ipfw add pipe 1 ip from any to "my IP" in
 ipfw add pipe 2 ip from "my IP" to any out
 ipfw add pipe 3 ip from any to not "my IP" in

42

3. Similarly, for the Bridges

in

1

2

out

IP layer

Link layer

3

Through Traffic
In a Bridge

```

ipfw add pipe 1 ip from any to "my IP" in
ipfw add pipe 2 ip from "my IP" to any out
ipfw add pipe 3 ip from any to not "my IP" in
    
```

43

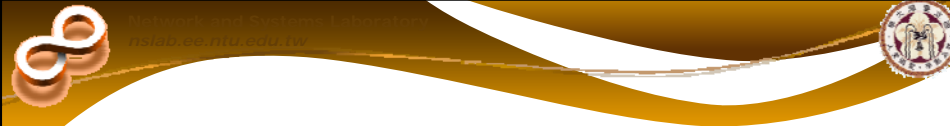
System Setting

Kernel Options
sysctl Variables

Polly@NTU

Copyright © 2008


44



Kernel Options

- IPFWALL - enable ipfirewall
- DUMMYNET - enable dummynet operation


Polly@NTU Copyright © 2008 45



Required Kernel Option

- options IPFWALL
- options DUMMYNET
- Must have the above 2 options in the kernel configuration file


Polly@NTU Copyright © 2008 46



Related Kernel Option

- options HZ
 - Sets the timer granularity
 - Default 100, meaning 10ms
 - Options HZ=1000, system tick reduced to 1ms
- options NMBCLUSTERS
 - not necessary after FreeBSD v6.2
 - Set the buffer size to store network packets
 - Product of bandwidth * delay
 - Plus queue size


Polly@NTU Copyright © 2008 47



Minor Kernel Options

- options IPFW_VERBOSE
 - Enable firewall output
- options IPFW_VERBOSE_LIMIT
 - Limit firewall output


Polly@NTU Copyright © 2008 48



ipfw Related sysctl Variables

- `net.inet.ip.fw.enable: 1`
 - Enable firewall in the IP stack
- `net.inet.ip.fw.one_pass: 1`
 - Force a single pass through the firewall.
 - If set to 0, packets coming out of a pipe will be re-injected into the firewall starting with the rule after the matching one.
 - One could create multi-hop paths with a bit of imagination
 - NOTE: there is always one pass for bridged packet


Polly@NTU Copyright © 2008 49



Bridge Related sysctl Variables

- `net.link.ether.bridge_cfg: ed2:1,r10:1`
 - Set of interfaces for which bridging is enabled, and cluster they belong to
- `net.link.ether.bridge: 0`
 - Enable bridging
- `net.link.ether.bridge_ipfw: 0`
 - Enable ipfw for bridging

Polly@NTU Copyright © 2008 50



dummynet Related sysctl Variables

- net.inet.ip.dummynet.hash_size: 64
 - Size of hash table for dynamic pipes.
- net.inet.ip.dummynet.expire: 1
 - Delete dynamic pipes when they become empty.

- Dynamic pipes will be covered in week 2

Polly@NTU

Copyright © 2008

51




Questions?

Polly@NTU

Copyright © 2008


52



The Project Presentation

- You have 40 minutes to do the following
 - Talk (30 min)
 - You'll receive a 3 min left and 1 min left warning
 - You'll be cut bluntly to stop talking after 30 mins
 - Answer questions (10 min)


53



Grading of Presentation

- Instructor evaluation 50%
- Peer evaluation 50%

54



Criteria (1-10)

- Technical merit
 - Do you understand the motivation?
 - Is the research statement clear?
 - Do you think the experimental method is appropriate to verify the statement?
 - Are the data acquired from the experiments supportive of the statement
 - Do you find the final conclusions convincing?
- Artistic merit
 - Are the slides well prepared?
 - Is the presentation complete?
 - Is the presentation comprehensive?
 - Is the presentation interesting?
 - Is the speaker sincere?

55