

# TNE30019/TNE80014 – Unix for Telecommunications

## Building a FreeBSD Bridge/Switch

Dr. Jason But

Swinburne University

## Outline

- What is a bridge/switch
- Why configure Unix system as a bridge/switch
- How to configure a FreeBSD bridge/switch

## Ethernet Bridges and Switches

- Bridge = two port switch
- Keeps track of which port each MAC address is connected to

### For each Ethernet frame received

- 1 Check destination MAC
- 2 Determine correct output port
- 3 Switch Ethernet frame to output network

## How does Bridge/Switch Learn Output Port ?

### Address Resolution Protocol (ARP)

- 1 Sender wants to send IP packet to destination in subnet
- 2 Sender checks if destination IP in it's ARP table
- 3 If not, sends broadcast ARP request (who has 192.168.0.1)
- 4 Destination will reply with ARP response (I am 192.168.0.1)

- Switch initially floods packet to all but incoming port
- Switch gradually learns who is connected to what port from ARP
- If switch re-booted, it will learn via regular Ethernet frames

## Ethernet Bridges and Switches

Ethernet switches are cheap and plentiful

- ~ \$20 8 Port Gigabit Switch
- ~ \$100 Switch with VLAN capabilities

Why would we use computer to do this?

## Building FreeBSD Bridge

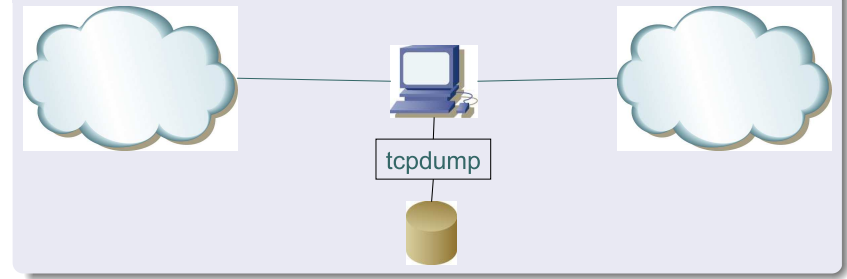
- Kernel must be built with task switching rate  $\geq 1000$ 
  - Default with current FreeBSD versions
  - Packets are processed and switched more quickly within kernel
  - Greater timing accuracy when simulating different network conditions
- Enable bridging support in kernel

### Kernel Options

```
options HZ = 1000
options BRIDGE
```

## Why Unix Bridges (and Switches)

Can monitor and manipulate passing network traffic



- Simple means of providing monitoring point in network
  - If we don't have electrical/optical splitter
- Can simulate different network conditions
  - Delays
  - Rate limiting
  - Packet loss

## FreeBSD – Enabling the Bridge

- Kernel supports bridging, it is not yet activated
- Enabled by setting sysctl variables (*kernel variables*)  
`man sysctl`
- Variables:
  - `net.link.ether.bridge.enable=1`
  - `net.link.ether.bridge.config=<if1>,<if2>`

### /etc/sysctl.conf

Ensures variables set at system boot  
`man sysctl.conf`

## Bridge/Switch NIC Configuration

- Typically NICs will not be configured with IP address
  - Possible to assign IP address to **one** NIC if you must remotely access bridge/switch
- Can specify more than two NIC devices in `net.link.ether.bridge.config` to create switch
- Can put interfaces into one of several clusters (like VLANs)
- **No** Spanning Tree protocol (SPT) implemented – can't connect bridges in circle
- Newer alternative is device `if_bridge` which supports STP  
See `man if_bridge`