Edinburgh Napier
UNIVERSITY

CSN09101
Networked Services
Week 5 : Networking

Module Leader: Dr Gordon Russell

Lecturers: G. Russell

This lecture

- Linux networking for end systems
- Linux as a router
- Linux as a switch
- Debugging a network
- Discussions

Edinburgh Napier
UNIVERSITY

End System Networking

Linux Networking

- Linux is a capable networking platform
- It runs many server applications, so is often seen as a prime platform for server applications.
- It has extensive level 2 and 3 networking support.
- It supports multiple network connections.

Default Networking

- Linux is a system which needs networking in order to work correctly.
- Even a system with no network has networking.
- The basic network is the loopback network.
- Every computer has an IP on the loopback network named *localhost*.

```
> telnet localhost
> telnet 127.0.0.1
> ping localhost
```

localhost

- The IP of localhost is 127.0.0.1
- It operates as a true network, and anything which can be done on a network in linux can operate on the localhost network.
- Linux operates a *priority* networking system, and localhost has the highest priority. If a packet can be delivered using localhost then it will always be delivered with localhost.

The localhost network device

- “lo” is often thought of as the localhost network device.
- It is rarely actually implemented as a /dev device.
- However, all the commands which expect a network device will take lo as a device name.
- It is handled internally in the kernel.

> /sbin/ifconfig lo

```
lo  Link encap:Local Loopback
    inet addr:127.0.0.1  Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING  MTU:16436  Metric:1
    RX packets:10 errors:0 dropped:0 overruns:0 frame:0
    TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:700 (700.0 b)  TX bytes:700 (700.0 b)
```

The Network Device

- In many systems /dev/eth* is the ethernet network device.
- In such systems with only one network connection, /dev/eth0 is the standard device name.
- Some distributions are renaming eth0 to reflect the hardware bus number of the device
  - This makes the name the same no matter how many hardware devices are plugged in later.
- Where hardware related slot indentities are used, the ethernet device could look like /dev/em1 for the embedded ethernet device in slot 1.
- A basic network needs
  - IP number of the host
  - Netmask for the network
  - Gateway IP for the gateway
  - Broadcast address

- The modern way to specify an IPv4 is the normal IP number and a /n value informing you of the netmask.

  10.0.1.20/24

- This indicates:
  - An IP of 10.0.1.20
  - A netmask of the first 24 bits (255.255.255.0)
  - Sensibly a broadcast of 10.0.1.255
  - Sensibly a gateway of 10.0.1.254

> ifconfig eth0

```
eth0      Link encap:Ethernet  HWaddr FE:FD:0A:00:02:02
          inet addr:10.0.2.2  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::fcfd:aff:fe00:202/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2008 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1181 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:106268 (103.7 Kb)  TX bytes:166284 (162.3 Kb)
          Interrupt:5
```

Ethernet Errors

- Difficult to find out exact meanings, but it is likely that:
  - Errors – CRC Error in packet
  - Dropped – Kernel buffers overflowed
  - Overruns – Card buffer overflowed
  - Frame – Frame length not a multiple of 8 bits
  - Carrier – Probably a fault in the card
  - Collisions – tx collided with another frame

ifconfig

- The ifconfig command takes a number of parameters:
  - Device
  - Ip
  - Broadcast address
  - Netmask

```
ifconfig eth0 10.0.50.10 broadcast 10.0.50.255
          netmask 255.255.255.0
```

ip command

- Ifconfig is being replaced with the more generic “ip” command.
- ip address show

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
   inet6 ::1/128 scope host
     valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
   link/ether 00:a0:24:e1:29:4e brd ff:ff:ff:ff:ff:ff
   inet 146.176.162.6/24 brd 146.176.162.255 scope global eth0
   inet6 fe80::2a0:24ff:fee1:294e/64 scope link
     valid_lft forever preferred_lft forever
```

The route

- Configuring the network device is only part of the process.
- It does not really manage the configuration of how to use the network.
- The old command to do this management is “route”. This has lately been replaced with the “ip” command.
- For this lecture we will mostly use “ip”.

> ip rule show

```
0:      from all lookup local
32766:  from all lookup main
32767:  from all lookup default
```

- Rules point to tables, which are like subroutines in a program.
- The number is the priority.
- In this case table local is first, then main, then default.
- If the network packet is handled in a particular table, it is not passed on to any other tables.

> ip route show table local

```
broadcast 127.255.255.255 dev lo  proto kernel  scope link  src 127.0.0.1
broadcast 10.0.2.0 dev eth0  proto kernel  scope link  src 10.0.2.2
local 10.0.2.2 dev eth0  proto kernel  scope host  src 10.0.2.2
broadcast 10.0.2.255 dev eth0  proto kernel  scope link  src 10.0.2.2
broadcast 127.0.0.0 dev lo  proto kernel  scope link  src 127.0.0.1
local 127.0.0.1 dev lo  proto kernel  scope host  src 127.0.0.1
local 127.0.0.0/8 dev lo  proto kernel  scope host  src 127.0.0.1
```

> ip route show table main

```
10.0.2.0/29 dev eth0  scope link src 10.0.2.1
default via 10.0.2.7 dev eth0
```

```
> route -n
Destination     Gateway     Genmask          Flags Metric Ref  Use Iface
10.0.2.0        0.0.0.0     255.255.255.248  UH     0       0       0 eth0
0.0.0.0         10.0.2.7    0.0.0.0          UG     0       0       0 eth0
```

Route

- For our simple example of: 10.0.50.10/24:

```
$ ifconfig eth0 10.0.50.10 broadcast 10.0.50.255
            netmask 255.255.255.0
$ ip route append 10.0.50.10 dev eth0 table main
$ ip route append default via 10.0.50.254
```

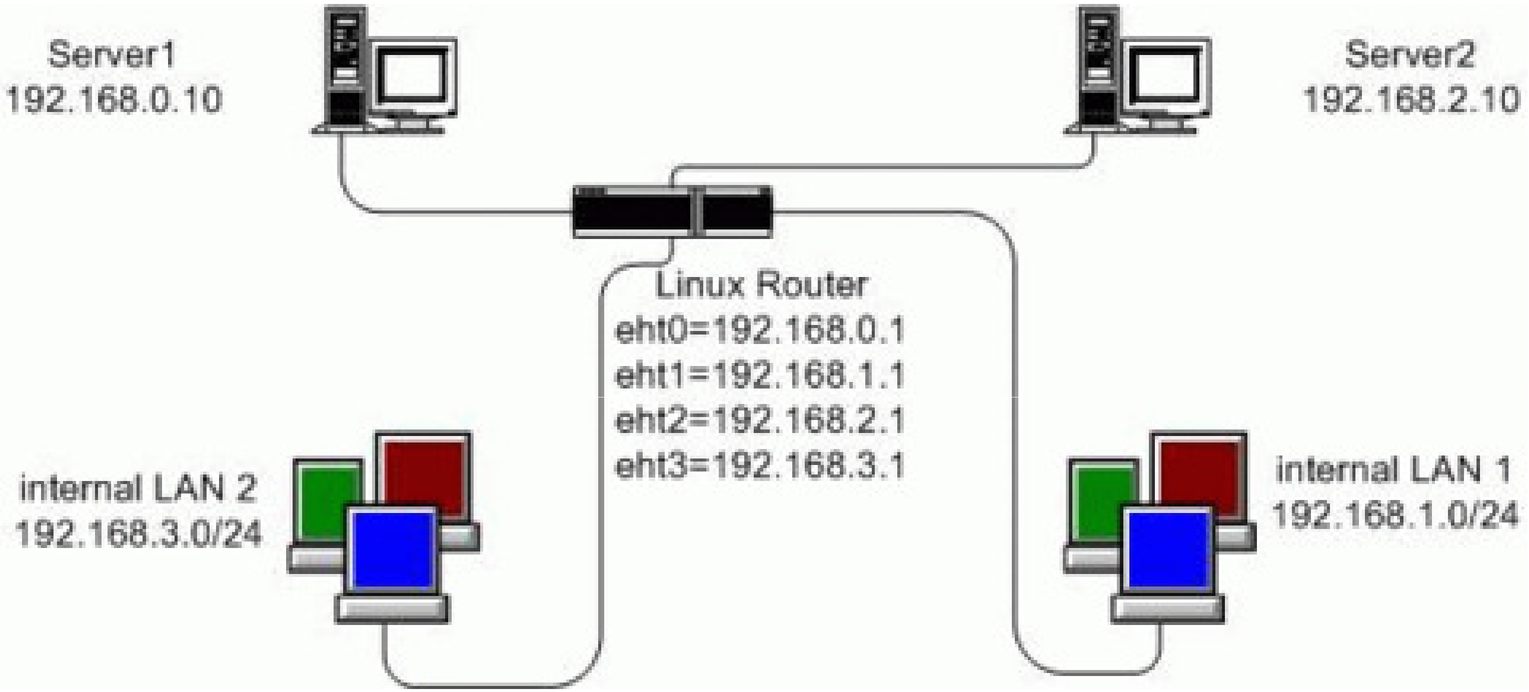
- Table main is the default, so can be left out of ip route.

ROUTING

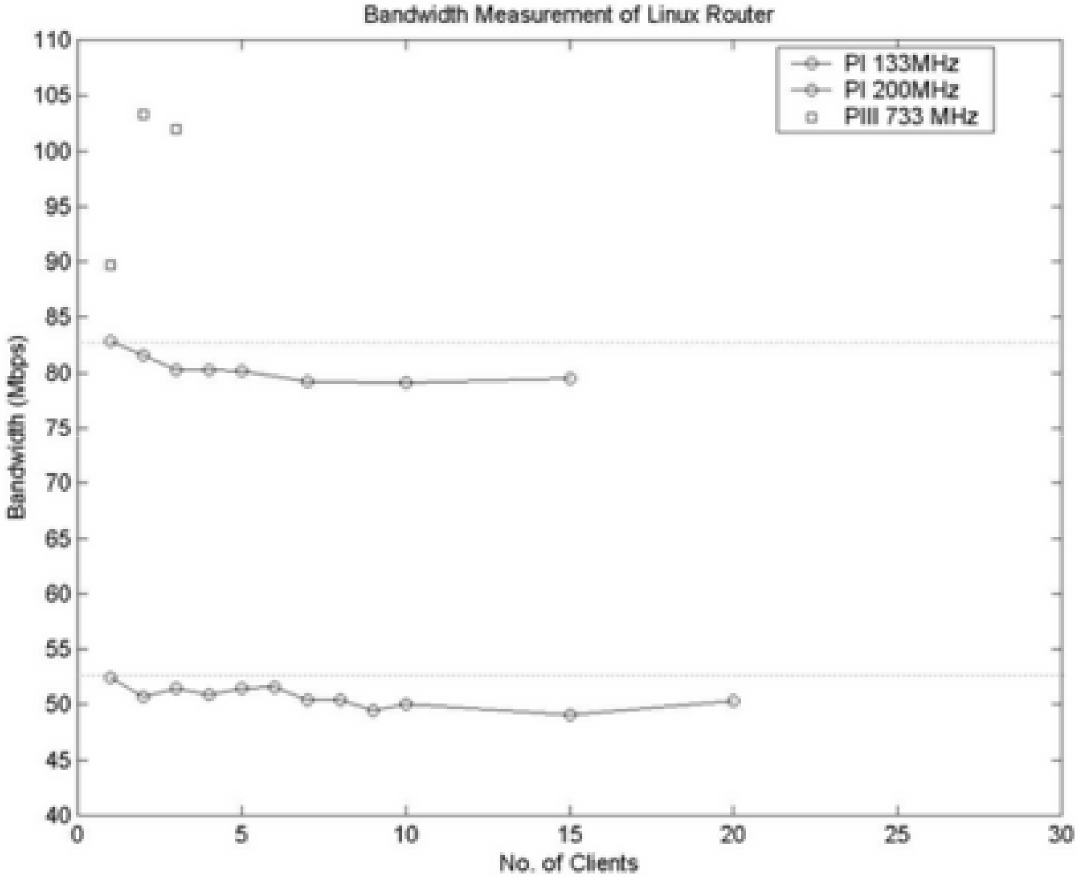
Linux for routing

- If Linux has more than 1 network connection, it can perform layer 3 routing, just like a Cisco router.
- Cisco routers often have only 2 or 3 network connections, and it is easy to build a PC to replicate this.
- Cisco argue that their routers are far superior…

Linux Routing Performance



Ref: http://www.linuxjournal.com/node/5826/print

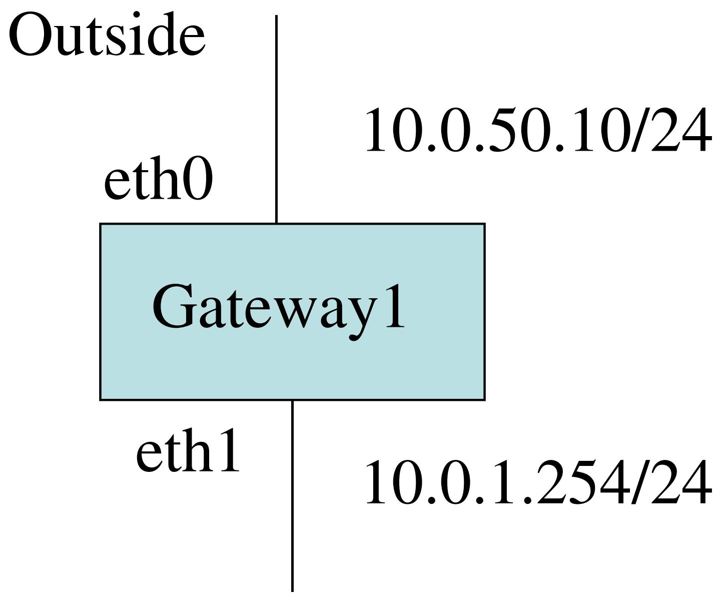


Ref:
http://www.linuxjournal.com/node/5826/print

Configuration

- Multiple networks is no different from single network configurations.
- You need ifconfig/ip address for each interface.
- You need a route for each interface
  - Ifconfig adds this route automatically… but you should still be able to do it manually for exam purposes.
  - ip address does not appear to do it automatically.
- You need 1 default route.

Example: Simple Gateway



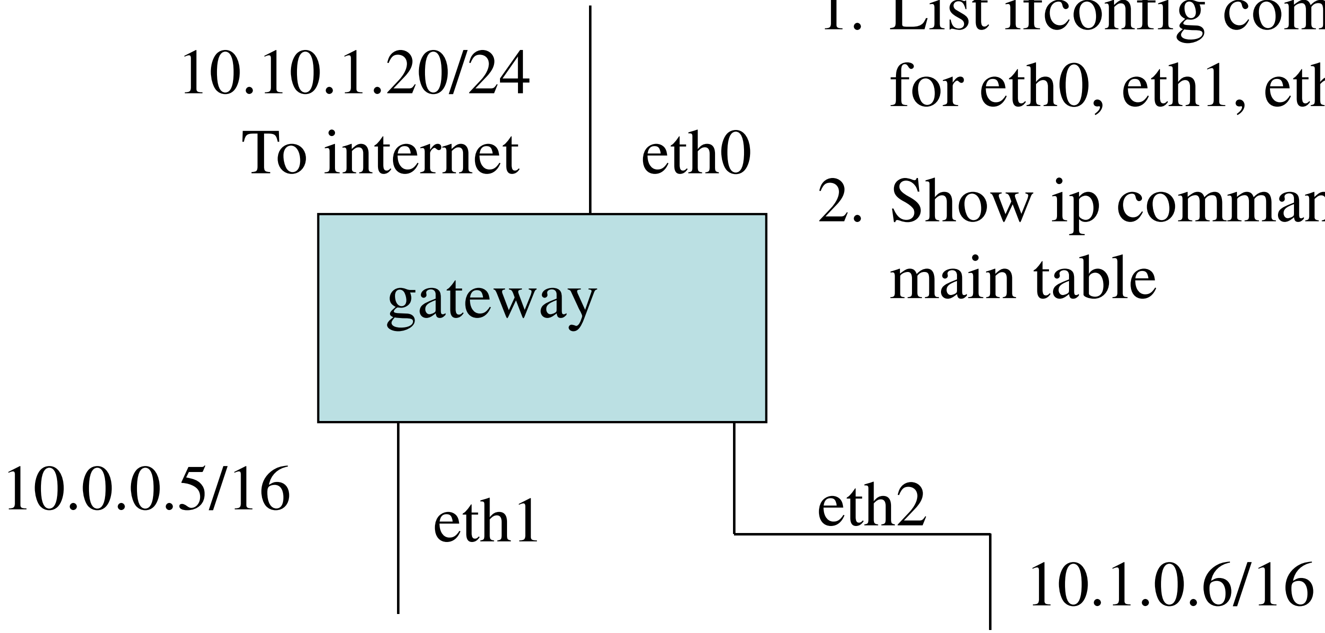
Add this example

```
> ifconfig eth0 10.0.50.10 broadcast 10.0.50.255 netmask 255.255.255.0
> ip route append 10.0.50.0/24 dev eth0
> ifconfig eth1 10.0.1.254 broadcast 10.0.1.255 netmask 255.255.255.0
> ip route append 10.0.1.0/24 dev eth1
> ip route append default via 10.0.50.254
```

> ip route show

```
10.0.50.0/24 dev eth0  scope link
10.0.1.0/24 dev eth1  scope link
default via 10.0.50.254 dev eth0
```

Class Exercise:



1. List ifconfig commands for eth0, eth1, eth2
2. Show ip commands in the main table

The netmask

- The netmask can be any size from /0 to /32.
- Perhaps you considered only /8, /16, /24 masks.
- These are fixed-length masks, matching the IP type (like Class A, B, etc).
- Complex networks use variable-length subnet masks.

VLSM

- Variable length subnet masks:
- Subdivide the host part of the network mask into smaller pieces.
- Each subdivision has its own network
- So if you need to run 2 networks, but only have 10.1.1.0/24, you can create 2 networks as:
  - 10.1.1.0/25
  - 10.1.1.128/25
- Remember that first and last host is reserved for “network” and “broadcast”. Thus you cannot use 10.1.1.0 or 10.1.1.127 or 10.1.1.128 for host addresses.

VLSM is “borrowing bits”

- Problem: You need 5 networks, but you only have 10.10.10.0/24.
- You cannot split into an number of networks which is not a power of 2 (2,4,8,16,etc), so split into 8.
- 8 needs 3 bits in binary (000-111 is 8 combinations)
- So borrow 3 bits from /24, making it /27.
- The new network numbers are:
  - 10.10.10.0/27 10.10.10.32/27
  - 10.10.10.64/27 10.10.10.96/27
  - 10.10.10.128/27 10.10.10.160/27
  - 10.10.10.192/27 10.10.10.224/27

VLSM for minimum hosts

- Sometimes you have a problem which states that you need n hosts per network.
- Consider the example of 10.1.1.0/24, where you need to divide your network into as many subnets as possible, where each subnet can hold at least 10 hosts.
- Increase “10” by 2, then increase to the next power of 2 (i.e. 16).
- 16 needs 4 bits (0000-1111 is 16 combinations).
- Take 32-4, giving 28. Network is 10.1.1.0/28, or:
  - 10.1.1.0/28, 10.1.1.16/28, 10.1.1.32/28, etc.

Class Exercise

- You have 10.20.1.0/24. Split the network into subnets so that each net can support at least 31 hosts.

Broken VLSM

- Some legacy systems don’t understand VLSM (e.g. RIP)
- Sometimes called the “subnet zero” problem
- This leads to 2 points of confusion, concerning the first and last network:
  - With 10.10.10.0/24 split into /27, networks 10.10.10.0/27 and 10.10.10.224/27 cause problems.
- For 10.10.10.0/27, 10.10.10.0 is the network number, and 10.10.10.255 is the broadcast address. But in VLSM, it’s the network number for network 1, and the broadcast for network 8.
- Take care with legacy systems!!!

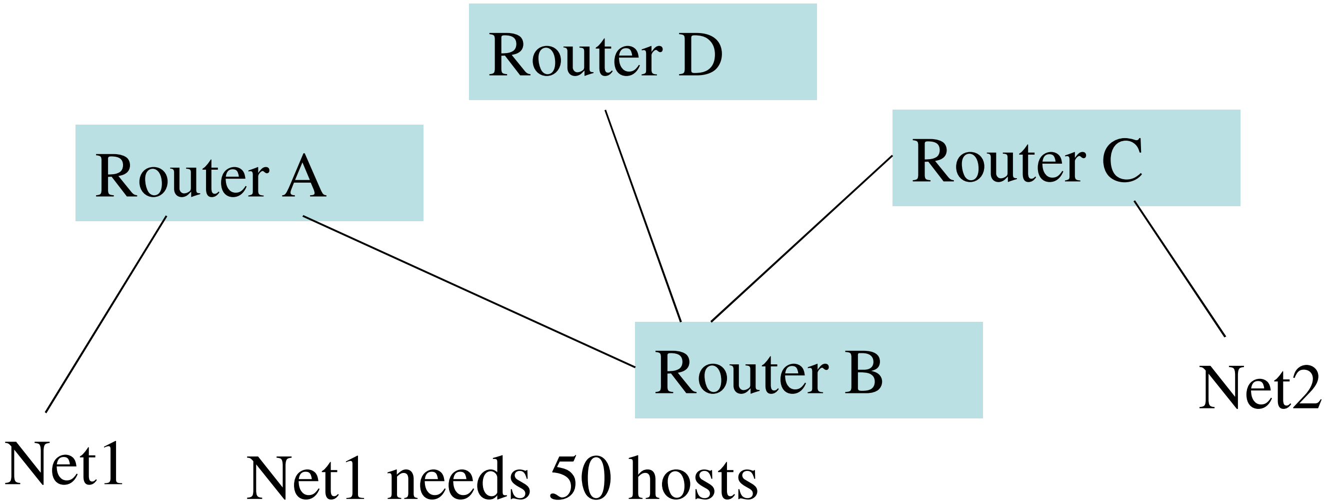
P2P netmask

- A point to point network is a little weird…
  - 10.0.0.14/32
  - Netmask 255.255.255.255
  - Broadcast 10.0.0.255
  - Gateway is likely to still be 10.0.0.254
- The gateway IP can be reused multiple times on each p2p link without difficulties.

- Really small netmasks > 1 IP …
- 10.0.0.5/30
  - 2 bits unset thus only 4 IPs in this net
  - IPs are 10.0.0.4,10.0.0.5,10.0.0.6,10.0.0.7
  - Broadcast will be highest ip, 10.0.0.7
  - The network has its own address (all bits zero) which reserves 10.0.0.4 for the network.
  - Max-1 is often the gateway, 10.0.0.6
  - Only 1 IP for host, 10.0.0.5
- Other than p2p, biggest netmask must be /30.

VLSM with mixed networks

- Consider the topology shown. You only have 10.1.1.0/24 to play with:



Router D

Router A

Router C

Router B

Net1

Net2

Net1 needs 50 hosts

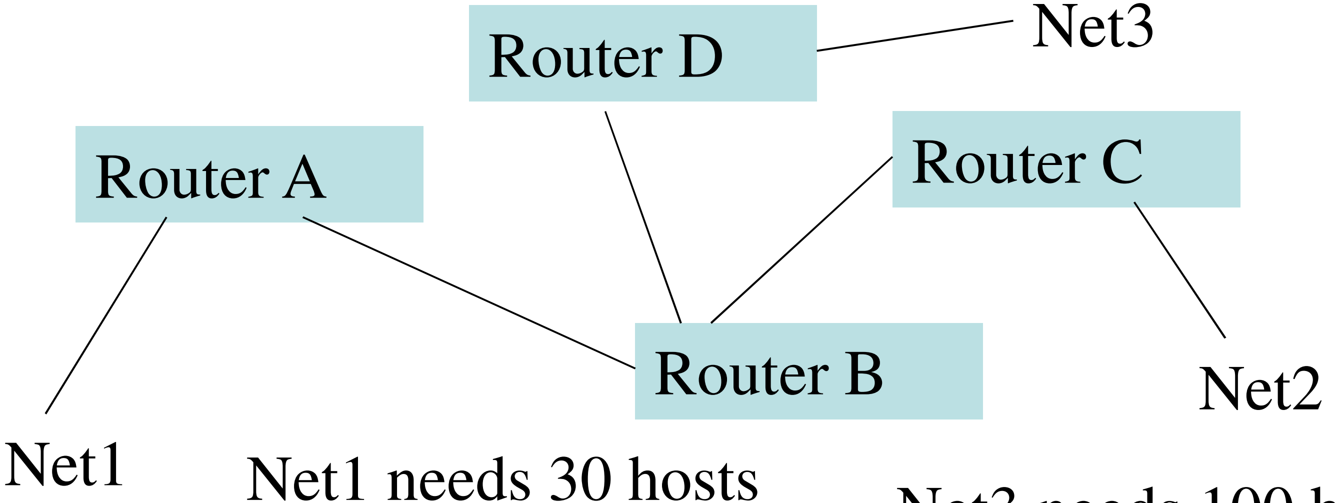
Net2 needs 50 hosts

- 50 hosts suggests 6 bits, leaving 2 bits, /26
- This provides 4 networks.
- However there are 5 networks:
  - Net1
  - Net2
  - Router A-B
  - Router B-C
  - Router B-D

- Solution is to divide up one /26, and use that for router-router links.
- For Net1+2 50 hosts suggests 6 bits, leaving 2 bits, /26
- For routers, 2 hosts suggests 2 bits or /30.
- Use 10.1.1.0/26 for Net1, 10.1.1.64/26 for net2.
- Split 10.1.1.128/26 into multiple /30 links:
  - Net1 – 10.1.1.0/26
  - Net2 – 10.1.1.64/26
  - Router A-B – 10.1.1.128/30
  - Router B-C – 10.1.1.132/30
  - Router B-D – 10.1.1.136/30

Exercise

- Consider the topology shown. You only have 10.1.1.0/24 to play with:



Net1 needs 30 hosts

Net2 needs 30 hosts

Net3 needs 100 hosts

Solution:

- 10.1.1.0/24 gets split into:
  - Net3 – 10.1.1.0/25
  - Net1 – 10.1.1.128/27
  - Net2 – 10.1.1.160/27
  - Router A-B – 10.1.1.192/30
  - Router B-C – 10.1.1.196/30
  - Router B-D – 10.1.1.200/30

Edinburgh Napier UNIVERSITY

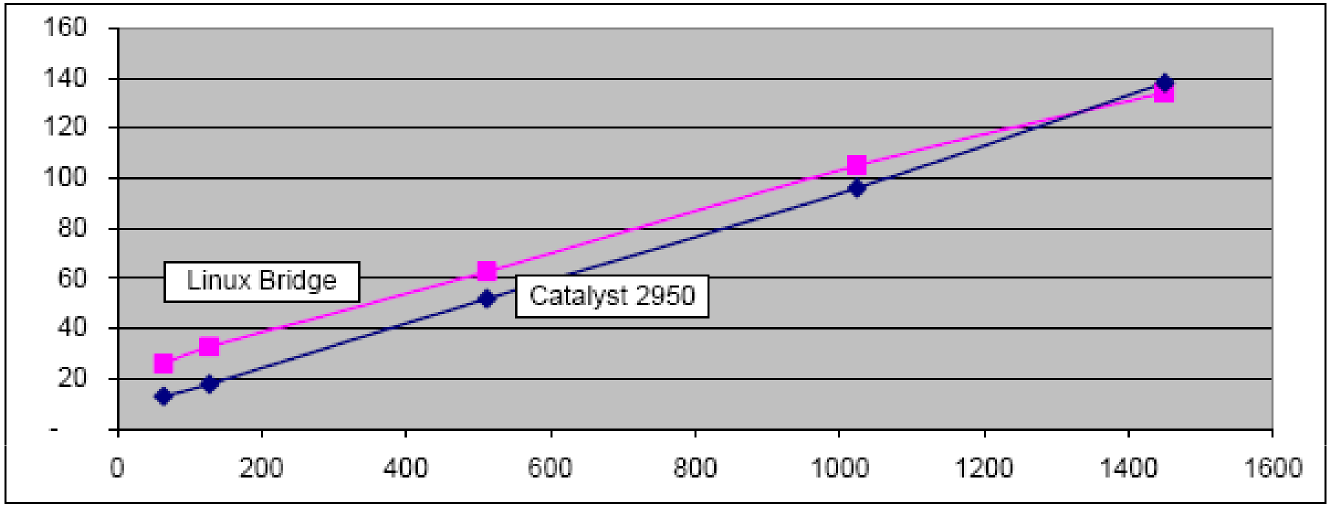
Linux Switch

Linux Switch

- A Linux box can also operate as a layer 2/3 device
- Here multiple ethernet cards are configured as layer 2 devices (mac address but no ip).
- They are then bridged together to form an intelligent switch.
- Hardware switches have custom logic to perform switching, and Linux boxes do this all in software…
- An excellent paper on its performance is: http://facweb.cti.depaul.edu/jyu/Publications/Yu-Linux-TSM2004.pdf

```
# ************ Create a bridge interface and it is called br1
brctl addbr br1
# ************ Add physical interfaces to the bridge interface
brctl addif br1 eth0
brctl addif br1 eth1
# ************ Reset IP interface
ifconfig eth0 0.0.0.0
ifconfig eth1 0.0.0.0
#Bring up the bridge
ifconfig br1 up
# ********** Set IP address of the bridge
ifconfig br1 192.168.1.10 netmask 255.255.255.0 up
# ********** Set IP default gateway
route add default gw 192.168.10.1
```

Latency vs frame size



- It is a small study, with a relatively low frame rate.
- High frame rates incur high delay (ms)
- They only used 2 network connections…

Discussion

- Is it a good idea to use:
  - Linux as a router?
  - Linux as a switch?

Edinburgh Napier
UNIVERSITY

Network Troubleshooting

Layered Approach

- Check layer 1
  - Is it wired up
- Check layer 2
  - Ethernet framing ok?
  - Layer 2 addressing?
- Check layer 3
  - Ip addresses and routes
  - Ping, traceroute
- Check layer 4
  - nmap

Linux ARP cache

- With an ethernet device the kernel must perform an ARP lookup.
- ARP is expensive, so the result is cached.
- /proc/net/arp is the arp cache record.

```
> cat /proc/net/arp
IP address …          HW address .. Device
146.176.166.254       00:08:7c:6e:90:00   eth0
146.176.166.2         00:e0:81:26:31:06   eth0
> ping 146.176.166.6
> cat /proc/net/arp
IP address …          HW address .. Device
146.176.166.254       00:08:7c:6e:90:00   eth0
146.176.166.2         00:e0:81:26:31:06   eth0
146.176.166.6         00:e0:81:25:c7:35   eth0
```

Questions:

- You ping 10.0.0.1, no response, and there is an entry for it in the arp cache. What does this tell you?
- You ping 10.0.0.1, no response, and there is no entry for it in the arp cache. What does this tell you?
- You see the following in the arp cache. What does this mean?

IP address …	HW address ..	Device
146.176.166.254	00:08:7c:6e:90:00	eth0
146.176.166.2	00:e0:81:26:31:06	eth0
146.176.166.3	00:e0:81:26:31:06	eth0

nmap

```
$ nmap linuxzoo.net
PORT        STATE SERVICE
22/tcp      open        ssh
23/tcp      open        telnet
53/tcp      open        domain
80/tcp      open        http
81/tcp      open        host2-ns
123/tcp     closed      ntp
5900/tcp    closed      vnc
5901/tcp    closed      vnc-1
5902/tcp    closed      vnc-2
5903/tcp    closed      vnc-3
```

netstat

- Netstat is another great monitoring tool
- Again it has lots of options.

```
$ netstat -al | grep LISTEN | grep tcp
tcp      0      0 *:http            *:*               LISTEN
tcp      0      0 *:ssh             *:*               LISTEN
tcp      0      0 *:https           *:*               LISTEN
```

```
$ netstat -n | head -4
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp        1      0 127.0.0.1:64359          127.0.0.1:631        CLOSE_WAIT
tcp        0      0 146.176.162.6:22         146.176.16:59160  ESTABLISHED
```

Not sure about port “:22”?

```
$ grep '22/tcp' /etc/services
ssh             22/tcp                 # SSH Remote Login Protocol
bpjava-msvc     13722/tcp              # BP Java MSVC Protocol
```

Discussion

- You cannot get ntp to work from a client machine. All other services are working normally. Nmap reports:

```
123/tcp        closed        ntp
```

What is your opinion of the problem?

Discussion

- Here are some past exam questions you should now be able to answer:

Question 1

Consider the topology shown

The Ethernet devices shown are from the point of view of M1.

Assume MGW is the gateway machine for this cluster of machines.

Also from the viewpoint of M1, the following is known:

Eth0 : 162.2.1.20/16

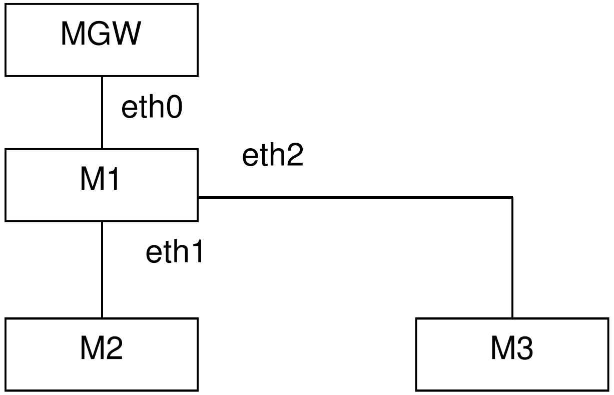
Eth1 : 162.1.1.3/24
Eth2 : 162.1.2.5/24

MGW is 162.2.1.1

M2 is 162.1.1.4

M3 is 162.1.2.10

Supply ifconfig lines for this scenario for use on M1.



Question 2

- Continuing from the previous question, supply ip route commands for M2

  The Ethernet devices shown are from the point of view of M1.

  Assume MGW is the gateway machine for this cluster of machines.

  Also from the viewpoint of M1, the following is known:

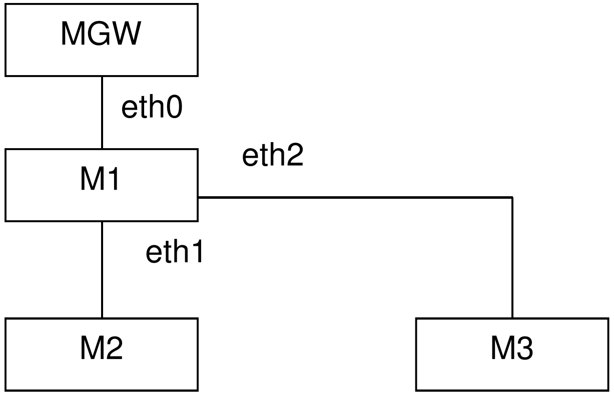
  Eth0 : 162.2.1.20/16

  Eth1 : 162.1.1.3/24
  Eth2 : 162.1.2.5/24

  MGW is 162.2.1.1

  M2 is 162.1.1.4

  M3 is 162.1.2.10


MGW
eth0
M1
eth2
eth1
M2
M3