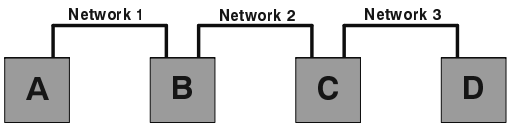
Configuring IP forwarding

Brian Candler
NSRC

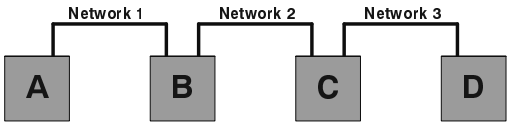
Review of IP forwarding (internetworking)

- B and C are routers: they connect to multiple networks and forward traffic between them
- Why do we need routers? Why not just build one big ethernet LAN?



From the viewpoint of "A"

- "A" can communicate directly with "B"; they are on the same network.
- But how can it send data to "D"?

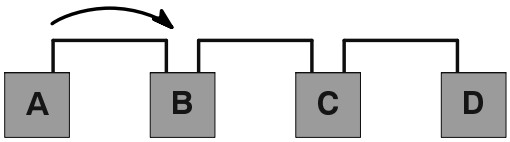


Answer: it must send to "B"

- This gets the packet one hop closer to the destination
- "Hop by hop" forwarding
- Must know which next hop to use
- For each packet, looks up destination in a *forwarding table*

A's forwarding table

Destination	Next-Hop
B	direct
C	B
D	B

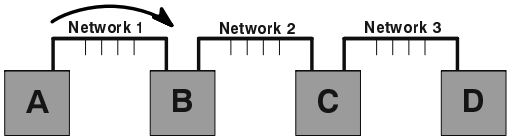


We don't list individual hosts in a forwarding table

- The Internet has hundreds of millions of hosts; the table would be too big
- However, all the hosts on one network share the same prefix
  - Example: network 200.10.194.0/24
  - The first 24 bits of each IP address is the same
- So we can match all the hosts on a remote network with a single forwarding table entry
- A forwarding table for the entire Internet is "only" about 125,000 entries, because each entry represents a whole network

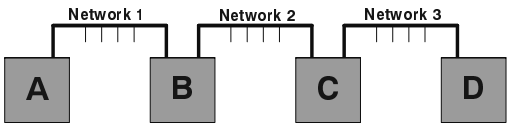
A's forwarding table (new)

Destination	Next-Hop
Network 1	direct
Network 2	B
Network 3	B



What is B's forwarding table?

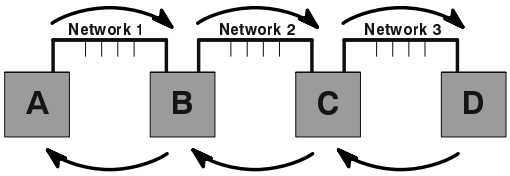
Destination	Next-Hop
Network 1	
Network 2	
Network 3	



7

In general, all the forwarding tables are *different*

- And they must all be correct, for A to be able to send traffic to D
- Also, forwarding from D to A must be correct so that responses can be received



What happens if a forwarding table entry is wrong?

- "Black holes"
- Forwarding loops
  - Why are they not forwarded forever?

9

Static routing

- Means that the forwarding table entries are built by hand
- Perfectly OK for small networks
- Error-prone for large networks
- If network topology changes, tables must be rebuilt

Dynamic routing

- Routers communicate with each other to discover the network topology
  - Examples: OSPF, IS-IS, RIP (bad)
- Forwarding tables built automatically
- Automatically responds to changes in topology, e.g. link failures
- Traffic can take alternate paths for resilience

11

IP forwarding in Linux

- Linux machines can make adequate routers
- Why do we prefer to buy expensive hardware routers (e.g. Cisco?)

Configuring interfaces

- ifconfig eth0 x.x.x.x netmask y.y.y.y
  - x.x.x.x is YOUR IP address
  - y.y.y.y is the netmask for the network
- You will immediately be able to ping other machines which are directly connected to eth0
  - On the same network as you
  - Have the same IP prefix as you

Adding static routes to reach remote networks

- route add -net x.x.x.x netmask y.y.y.y gw z.z.z.z
  - x.x.x.x/y.y.y.y are the network address and netmask of a remote network
  - z.z.z.z is the IP address of the next hop router
- z.z.z.z must be on the same network as you
- You can now ping any machine on that network (but you won't get a response until *all* intermediate hops are configured)

Reviewing your configuration

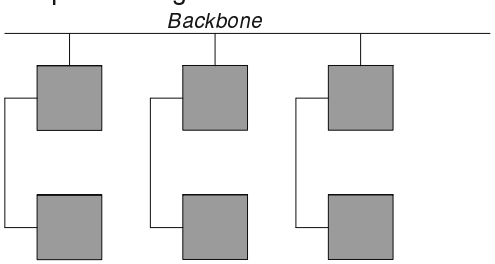
- ifconfig
  - Shows the interface configuration
- route -n
  - Shows the contents of the forwarding table

Testing the network

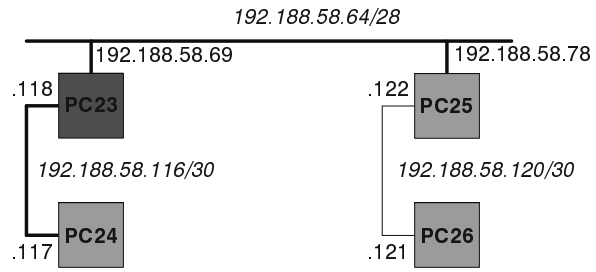
- ping x.x.x.x
  - Check you can get packets to and from that remote machine
- traceroute -n x.x.x.x
  - Shows the route traffic takes towards x.x.x.x
  - If it stops at a certain point, that may indicate where the error is
  - "-n" prevents DNS lookups
    - This is VERY important; if your network is broken then probably your DNS servers are not reachable, and attempting to do so will introduce long delays

Classroom exercise

- Break the class into a "backbone" network with separate edge networks



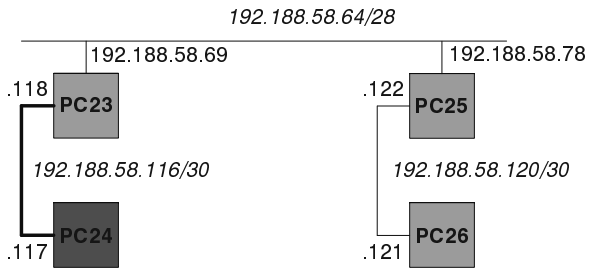
Configuration for PC23




```
# ifconfig eth0 192.188.58.69 netmask 255.255.255.240
# ping 192.188.58.78
# ifconfig eth1 192.188.58.118 netmask 255.255.255.252
# ping 192.188.58.117
# route add -net 192.188.58.120 netmask 255.255.255.252 gw 192.168.58.78
# ping 192.188.58.121
```

Repeat for all other networks in the classroom!

Configuration for PC24




```
# ifconfig eth0 192.188.58.117 netmask 255.255.255.252
# ping 192.188.58.118
# route add -net 192.188.58.64 netmask 255.255.255.240 gw 192.168.58.118
# ping 192.188.58.78
# route add -net 192.188.58.120 netmask 255.255.255.252 gw 192.168.58.118
# ping 192.188.58.121
```

Repeat for all other networks in the classroom!

Hints

- Test your network one step at a time
  - Make sure you can ping your next hop before you try to route traffic through it
- Remember that pinging won't work unless you have routes to their network AND they have routes to your network
- Pick another desk and work with them until you are able to ping both their machines and they can ping both of yours
- Review your configuration frequently. Make sure netmasks are correct!

How to get full connectivity to the Internet?

- We prefer not to add 125,000 routes!