Routing tables and Route Summarisation

What is a routing table?
How do I create a "good" one?

Modern Routing Tables

- Each entry in a routing table has 3 main items:
- A network address (the destination)
- A netmask length
- A next hop address

```
$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.19.64.0     0.0.0.0         255.255.192.0   U     0      0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         172.19.127.254  0.0.0.0         UG    0      0        0 eth0
```


The Routing Algorithm

- For a given destination IP address
- Search the routing table for the longest prefix match for the address
- Extract the next hop address from the routing table entry
- Send the packet to the next hop address
- If no match found, report that the destination is unreachable.

Longest Prefix

- So what does "longest prefix match" mean?
- To see if the prefix matches,
  - Bitwise AND netmask with destination
  - Bitwise AND netmask with network from routing table entry
  - If the two results are equal, then the prefix matches
- If we do the same for all entries in the routing table, the match with the longest netmask wins.

Example:

- Given this routing table, where does the packet with destination 192.168.0.3 go to?

```
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.25.0    0.0.0.0         255.255.255.0   U     0      0        0 vmnet1
192.168.0.0     172.19.35.254   255.255.0.0     UG    0      0        0 ppp1
0.0.0.0         202.180.160.251 0.0.0.0         UG    0      0        0 ppp0
```

- How about 192.168.128.48?
- 192.168.25.10?
- 192.169.0.1?

The Big Emergency

- In the early 90s, it became apparent that two problems were quickly going to become overwhelming:
- We were running out of IP addresses
- The routing tables were growing too fast for the router hardware to cope

The Solution: CIDR and NAT

- Two solutions were developed:
- CIDR (Classless Internet Domain Routing), and
- NAT (Network Address Translation).
  - NAT allows a firewall or router to present one address to the outside world, but many to the inside.
  - In Linux, use iptables: module is called NAT.
  - Use private addresses:
  - 192.168.0.0/16
  - 172.12.0.0/12
  - 10.0.0.0/8

The Problems CIDR helps fix: address depletion

- Class C was too small for medium sized enterprises
- Class B was too big
- Many organisations asked for (and received) class B networks when they needed only a /22 or /21 network
- This used up the available 2^{32} addresses too fast
- Later there was a need for small Internet allocations of 1 or 2 addresses.
  - Class C was too wasteful for this.

The Problems CIDR helps fix: router table explosion

- As class B addresses became scarce, SMEs were given a number of class C network allocations
- But each class C needed a separate routing table advertisement
- Local information about the internal network structure of a company needed to be advertised world wide
- This did not scale
- By now routing would need much more CPU and RAM than is currently used, and the Internet would have slowed further.

How does CIDR solve them?

- New address allocations can be sized accurately to the need
  - When requesting addresses, the authority (www.apnic.net) will reserve some addresses for future growth if you specify you will need them
- New address allocations are made taking into account neighbouring networks
- Aim is to summarise many routes into as few routes as possible.

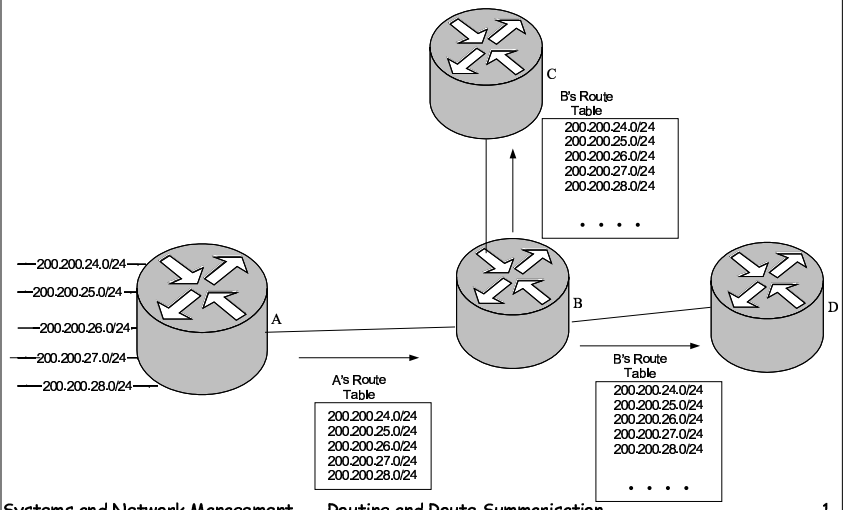
Aggregating routes

- Routers summarise routes themselves when they use classless routing protocols such as:
  - Rip2
  - OSPF
  - BGP

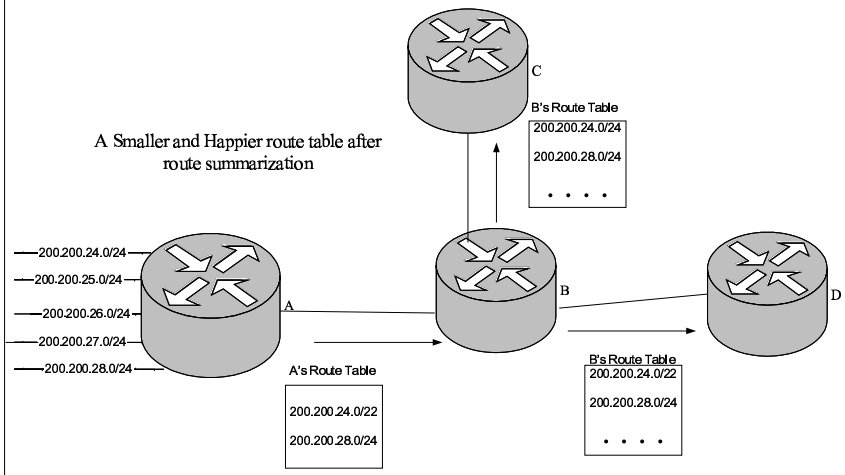
Route Aggregation

- There is a Perl module for working with IP addresses (of course):
- NetAddr::IP
- Includes the method compact(), which takes a list of networks and returns a list of summarised address blocks.

Without Route Summarisation



With Route Summarisation



Explanation

- The first diagram shows the subnets behind router A advertised everywhere
  - This is because the routes are unable to summarise the routes
- The second diagram shows the subnets behind A summarised into two routes instead of 5
  - The routers must be running a classless routing protocol such as OSPF or RIP2.

How the Routes were Summarised

- 200.200.24.0/24: $24_{10} = 00011000_2$
- 200.200.25.0/24: $25_{10} = 00011001_2$
- 200.200.26.0/24: $26_{10} = 00011010_2$
- 200.200.27.0/24: $27_{10} = 00011011_2$
  - So these can be summarised into:
  - 200.200.24.0/22
- 200.200.28.0/24: $28_{10} = 00011100_2$
  - This cannot be summarised with the other routes, so it must be advertised separately.