Routing IP

We now look at one of the fundamental aspects of IP: routing

- A packet does not know how to get to its destination
- Must rely on the routers to send it in the right direction
- So how do the routers do that?

Routing IP

- A router can't possibly know where everything in the world is: it is only connected to a handful of neighbour routers
- How can a router in England know that to send a packet to Australia it might have to forward it to America first?

Routing IP

- Two types of routing
  - Local routing within an organisation (requiring an *interior gateway protocol*)
  - Non-local routing between organisations (requiring an *exterior gateway protocol*)
- Very different requirements, with exterior protocols mostly driven by politics and economics

Routing IP

Interior Gateway Protocols

- We have seen small routing tables

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface    213.121.147.69  *
255.255.255.255 UH    0      0        0 ppp0
172.18.0.0      *               255.255.0.0     U     0      0        0 eth0
172.17.0.0      *               255.255.0.0     U     0      0        0 eth1
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         213.121.147.69  0.0.0.0         UG    0      0        0 ppp0
```

- The address on a packet is ANDed with the mask: if the result is equal to the destination, route via the given interface. Use first match from top to bottom in the table

Routing IP

Interior Gateway Protocols

- We have seen small routing tables

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface    213.121.147.69  *
255.255.255.255 UH    0      0        0 ppp0
172.18.0.0      *               255.255.0.0     U     0      0        0 eth0
172.17.0.0      *               255.255.0.0     U     0      0        0 eth1
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         213.121.147.69  0.0.0.0         UG    0      0        0 ppp0
```

- Flags
  - U: the interface is up (i.e., working)

Routing IP

Interior Gateway Protocols

- We have seen small routing tables

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface    213.121.147.69  *
255.255.255.255 UH    0      0        0 ppp0
172.18.0.0      *               255.255.0.0     U     0      0        0 eth0
172.17.0.0      *               255.255.0.0     U     0      0        0 eth1
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         213.121.147.69  0.0.0.0         UG    0      0        0 ppp0
```

- Flags
  - G: the route is to a gateway/router. Otherwise the destination is on the local network

Routing IP

Interior Gateway Protocols

- We have seen small routing tables

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface    213.121.147.69  *
255.255.255.255 UH    0      0        0 ppp0
172.18.0.0      *               255.255.0.0     U     0      0        0 eth0
172.17.0.0      *               255.255.0.0     U     0      0        0 eth1
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         213.121.147.69  0.0.0.0         UG    0      0        0 ppp0
```

- Flags
  - H: the route is to a host. The destination address is a single host, not a network

Routing IP

Interior Gateway Protocols

- We have seen small routing tables

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface    213.121.147.69  *
255.255.255.255 UH    0      0        0 ppp0
172.18.0.0      *               255.255.0.0     U     0      0        0 eth0
172.17.0.0      *               255.255.0.0     U     0      0        0 eth1
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         213.121.147.69  0.0.0.0         UG    0      0        0 ppp0
```

- Flags
  - D: this entry was created by an ICMP *redirect*
  - M: this entry was modified by an ICMP *redirect*

Routing IP

Interior Gateway Protocols

- A *static* route is one added by hand, e.g., the route command

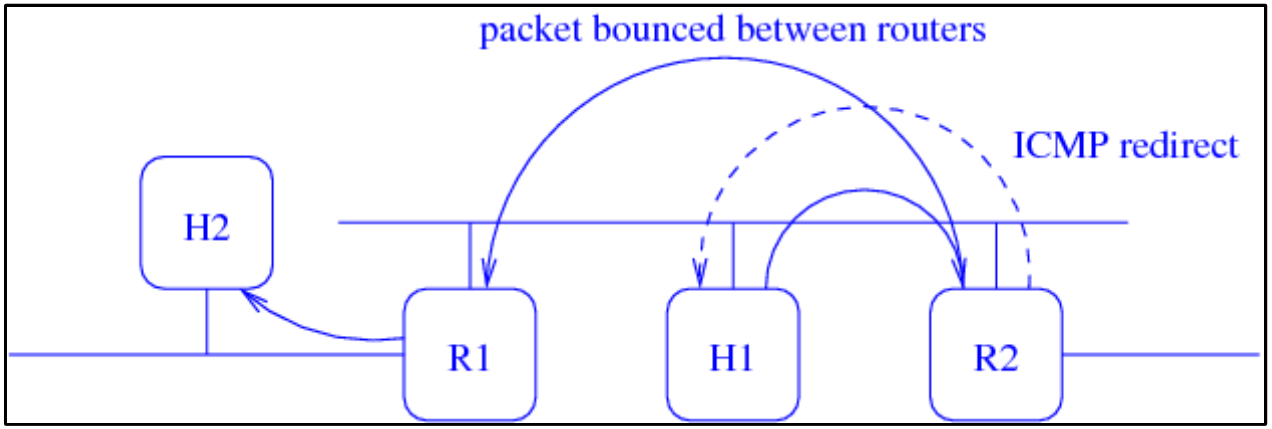
  route add default gw 213.121.147.69

adds a default route to a gateway

- Routing tables on most non-routers are trivial and set up manually by the operating system at boot time

Routing IP

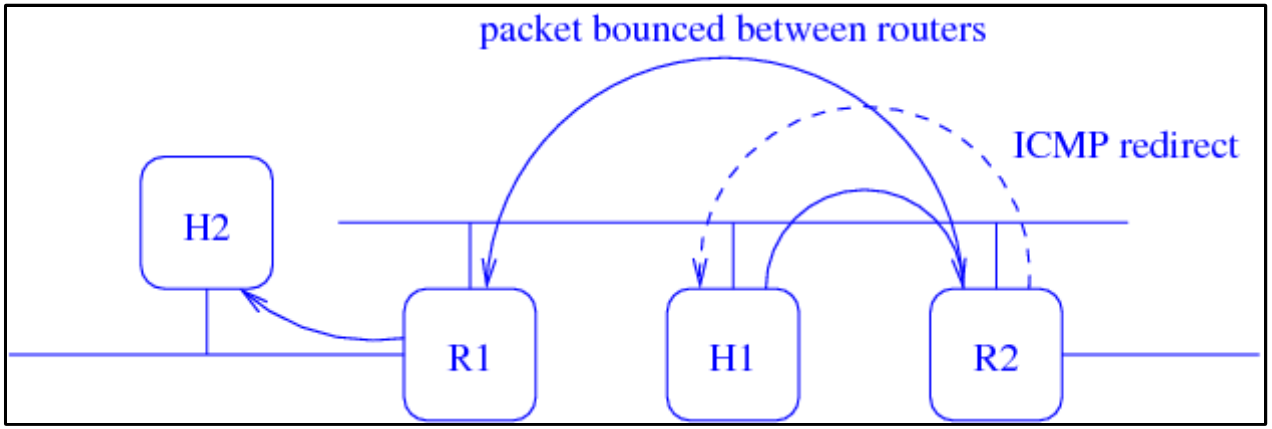
ICMP Redirect



- Sometimes routing tables are not perfectly set up
- H1 wants to send to H2 but H1's routing table tells it to route via R2

Routing IP

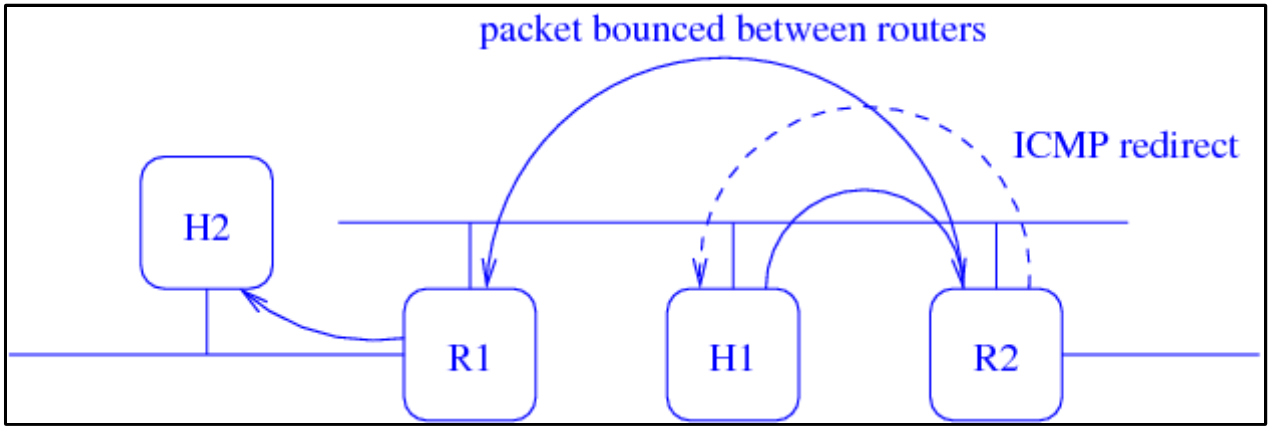
ICMP Redirect



- When the packet reaches R2 it sees it should be routed out on the interface it came in on: so R2 knows H1's table need improving
- R2 forwards the packet and send an ICMP redirect to H1

Routing IP

ICMP Redirect



- H1 gets the redirect and uses it to update its routing table. The route will be marked D or M
- Next time H1 will be able to route directly to R1

Routing IP

Dynamic Routing Protocols

- ICMP redirects are OK for small fixes, but in general routers need something more substantial
- Could get administrators to set up the tables
- Better to get the routers to do it themselves
- *Dynamic routing* is the passing of routing information between routers

Routing IP

Dynamic Routing Protocols

- Several protocols are used
  - Routing Information Protocol (RIP)
  - Open Shortest Path First (OSPF)
  - Border Gateway Protocol (BGP)
  - Many more
- Each protocol is suited to a certain purpose, no single protocol fits all

Routing IP

Dynamic Routing Protocols

- Internet is managed as a collection of *Autonomous Systems* (AS), each administered by a single entity, e.g., a University or company
- Each AS chooses a routing protocol to direct packets within the AS: these are *interior gateway protocols* (IGP), e.g., RIP or OSPF
- Between ASs run *exterior gateway protocols* (EGP), e.g., BGP

Routing IP

Dynamic Routing Protocols

- Typically a router runs a program whose sole purpose is to manage routing information and update the routing table. This program may understand several routing protocols
  - routed talks RIP
  - gated talks RIP, OSPF and BGP

Routing IP

Distance-Vector and Link-State

- Routers can collect all kinds of information about who is connected to whom and in what manner: this must be condensed down to something that can be put in a routing table
- Two main ways:
  - distance-vector protocols
  - link-state protocols

Routing IP

Distance-Vector and Link-State

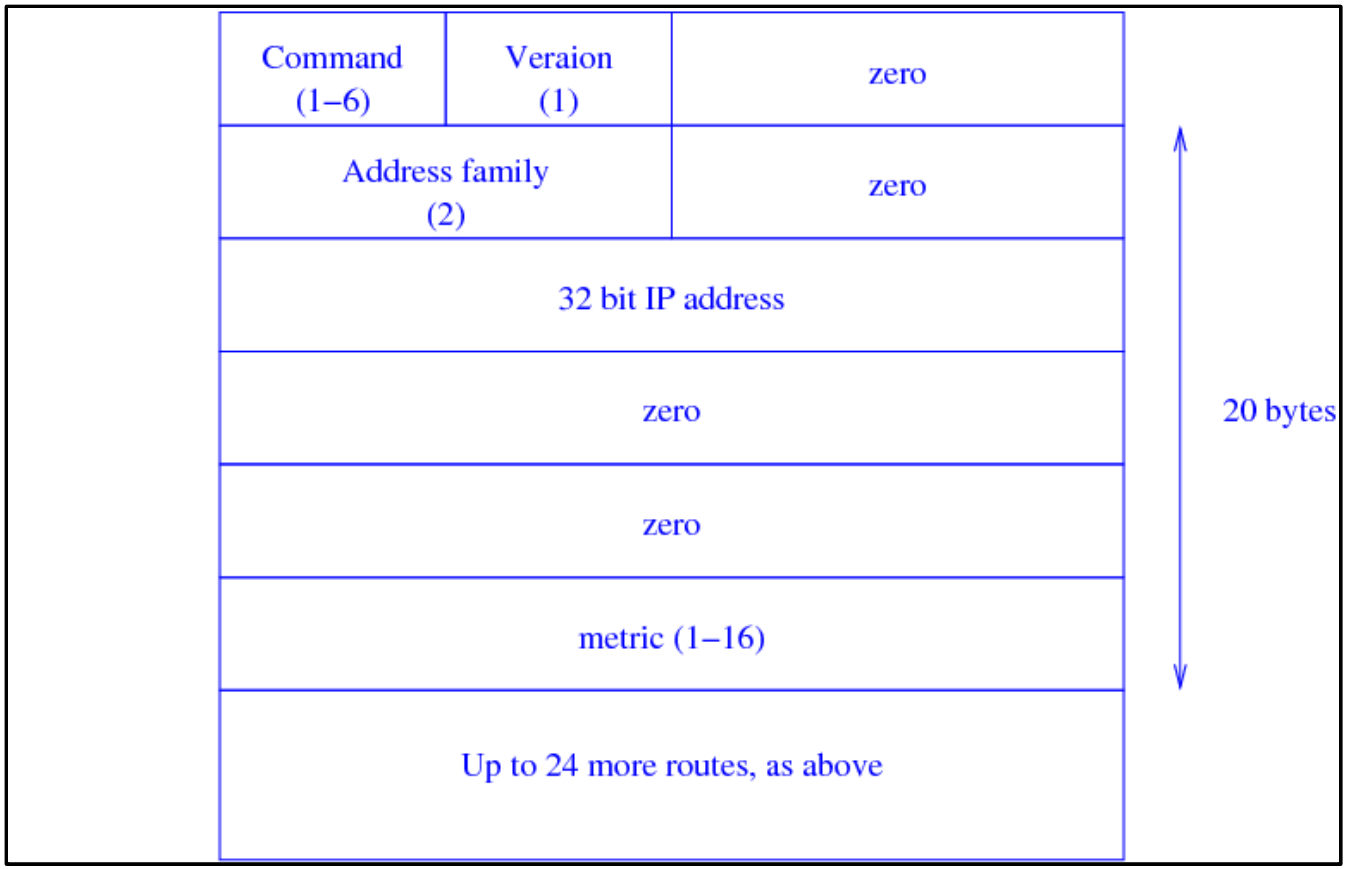
- Distance-vector gathers collections (vectors) of hop counts (distances) from its neighbouring routers to selected destinations. From this is computes its own vector of distances. RIP is an example
- Link-state gathers maps of connectivity from all the routers in the AS (or some subset) and uses this to compute its own map. OSPF is an example

Routing IP

Distance-Vector and Link-State

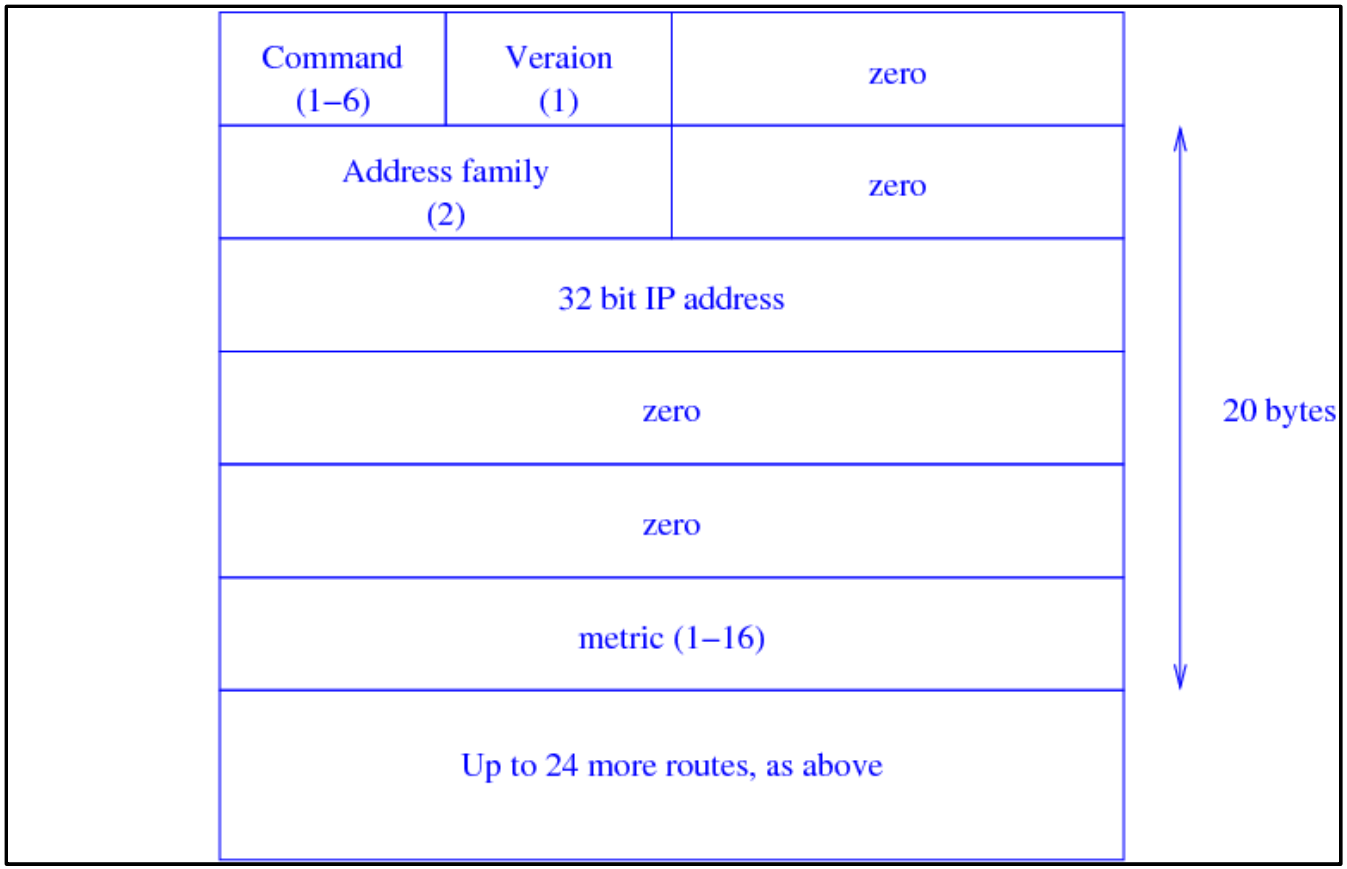
- Distance-vector is simple, but has problems
- Link-state is more complex, but has advantages

Routing IP



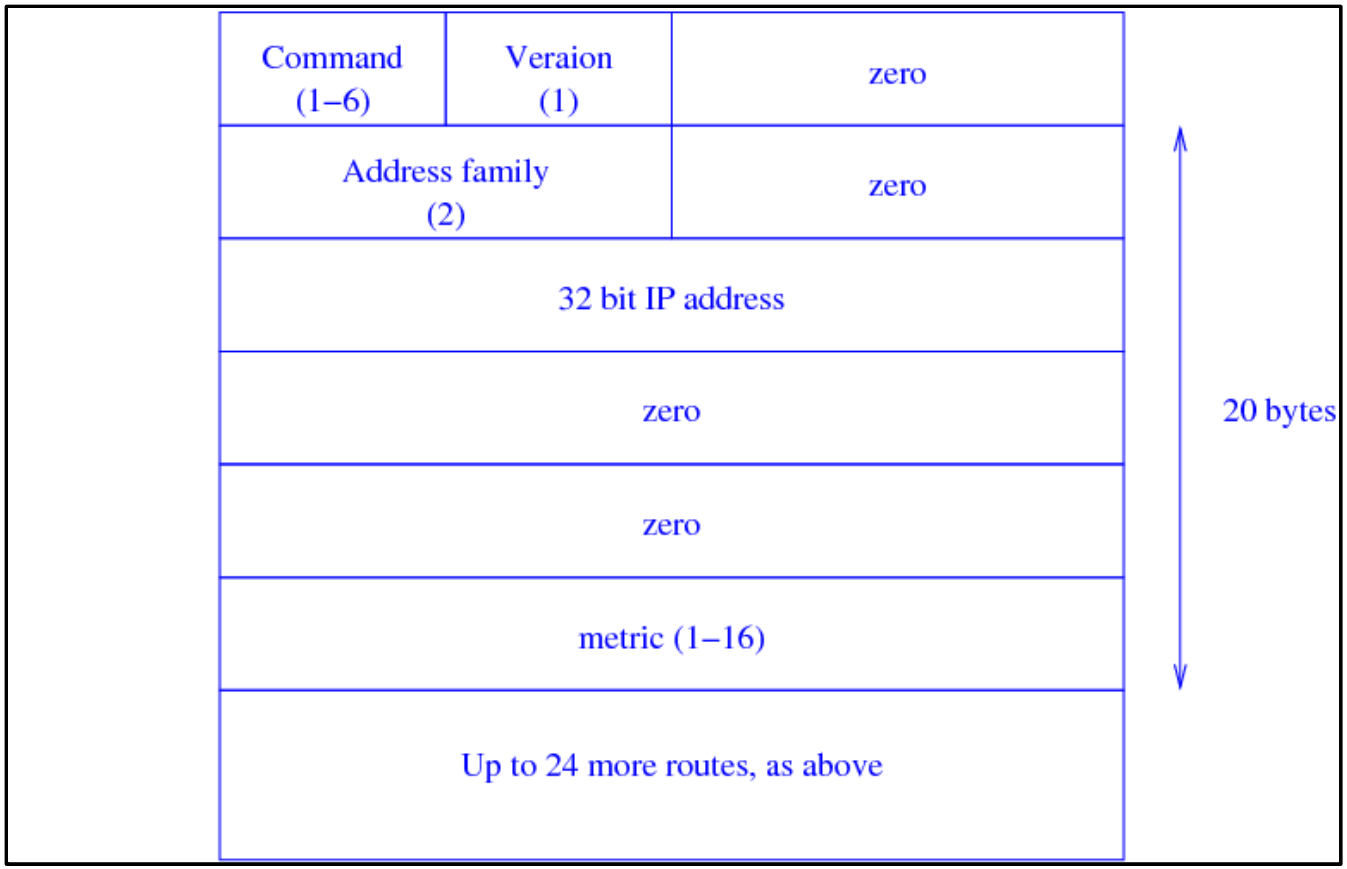
- Command: 1 a request, 2 a reply

Routing IP



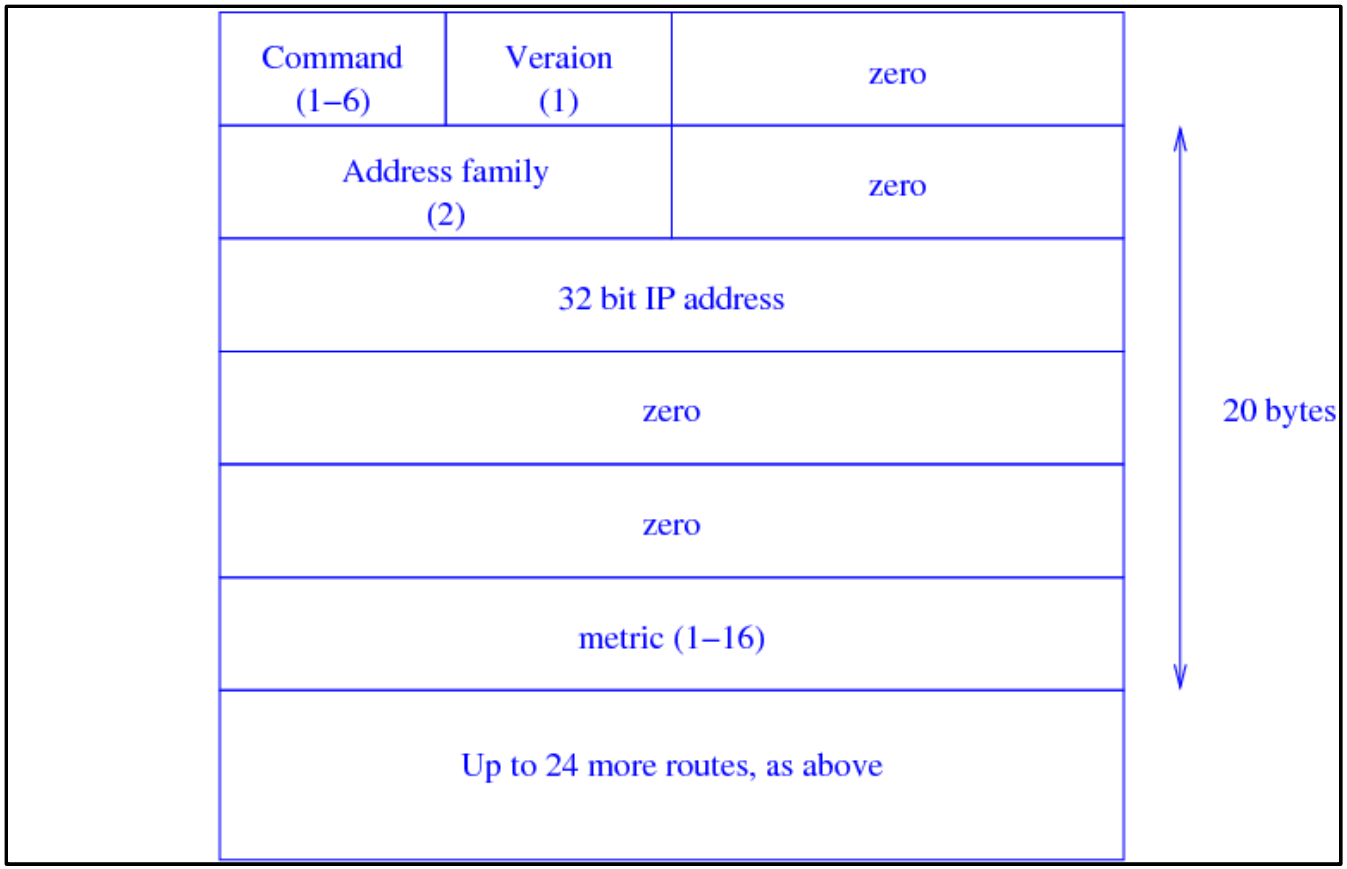
- 20 bytes of route information
- Address family: 2 for an IP address

Routing IP



- 4 bytes of IP address (plus room for expansion)

Routing IP



- Metric is a hop count: “I know a route to that address of this number of hops”
- Fewer hops is a better route

Routing IP

RIP

- When a router starts it broadcasts a RIP request on all its interfaces with address family 0 and metric 16: this is a “send me all your routes” request
- A router receiving such a request replies with its vector of distances

Routing IP

RIP

- Otherwise a request is for a route to a particular address or addresses
- A reply is a metric for a route, or 16 to indicate infinity or “no route”
- Given a reply we can update our routing table: our metric is the received metric plus 1 for the hop to the router that replied

Routing IP

RIP

- If a new route arises with a smaller metric than an existing route we can update our table to use the new route: RIP always deems shorter paths to be better
- RIP also sends a chunk of the routing table every 30 seconds to its neighbours: routes are timed out if they are not confirmed for 3 minutes

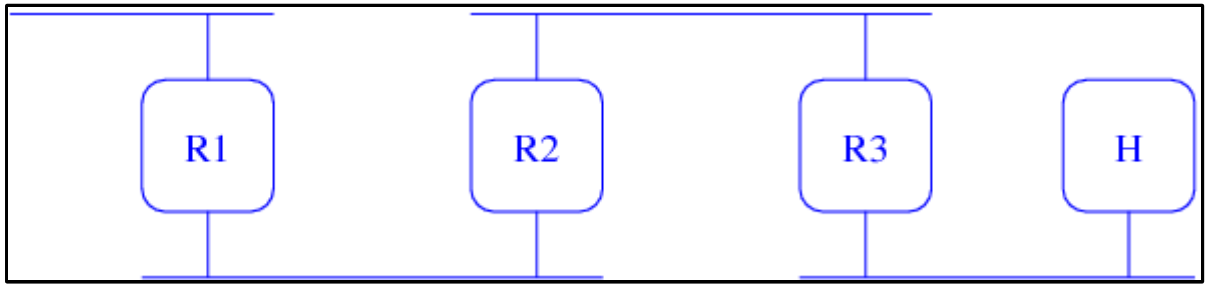
Routing IP

RIP

- A timed-out route is set to metric 16, but not deleted for 60 seconds: this ensures the invalidation is propagated

Routing IP

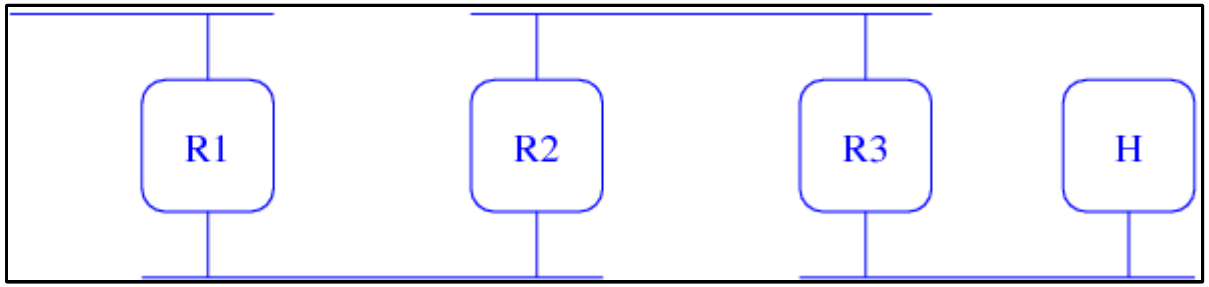
RIP



- R3 knows a route to H with metric 1
- R3 sends a RIP message to R2, R2 learns a route to H via R3 with metric 2
- R2 sends a message to R1, R1 learns a route to H via R2 with metric 3

Routing IP

RIP



- Note that R2 sends a message to R3, too, so R3 learns there is a route to H via R2 with metric 3
- But R3 can ignore this as it already knows a better route

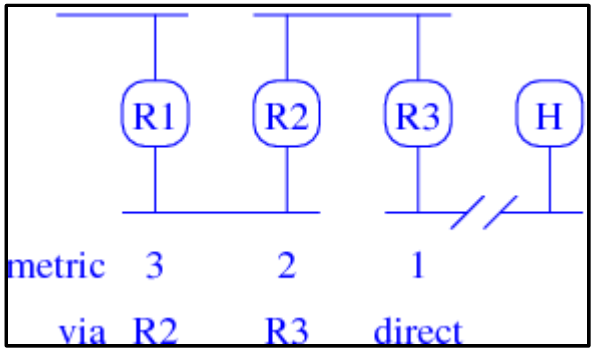
Routing IP

RIP

- RIP has problems, in particular the long time (several minutes) it takes to readjust after a route is added or removed somewhere: the *slow convergence*, or *count to infinity* problem

Routing IP

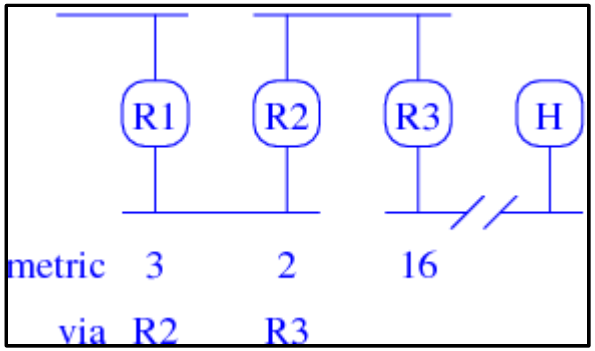
RIP Slow Convergence



- The link from R3 to H breaks

Routing IP

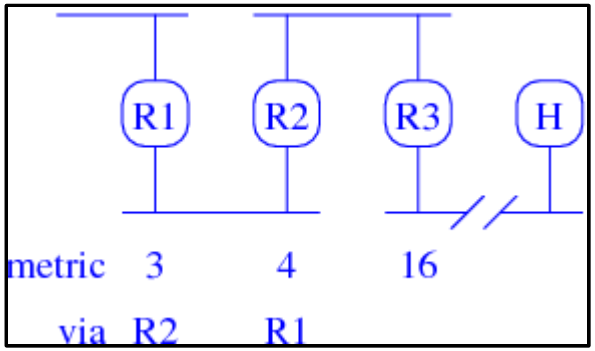
RIP Slow Convergence



- R3 starts sending RIP messages with metric 16 (infinity) for the route to H

Routing IP

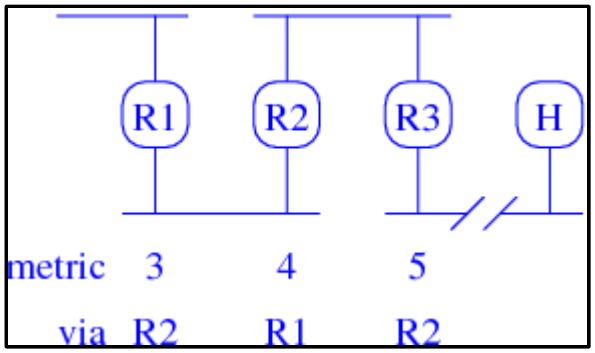
RIP Slow Convergence



- R2 gets this, but it also gets a message from R1 that says it has a route to H of metric 3: so now R2's best route is via R1 with metric 4

Routing IP

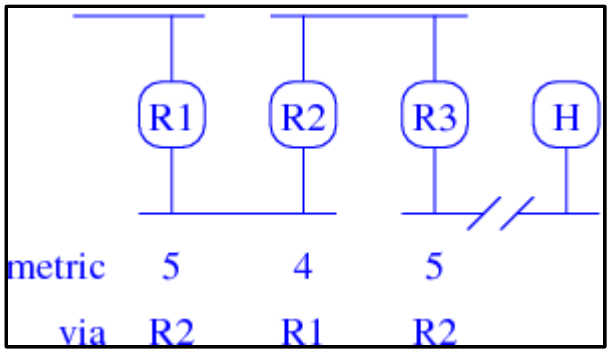
RIP Slow Convergence



- R2 sends messages with metric 4: R3 sees this and “improves” its route to H to be via R2, metric 5

Routing IP

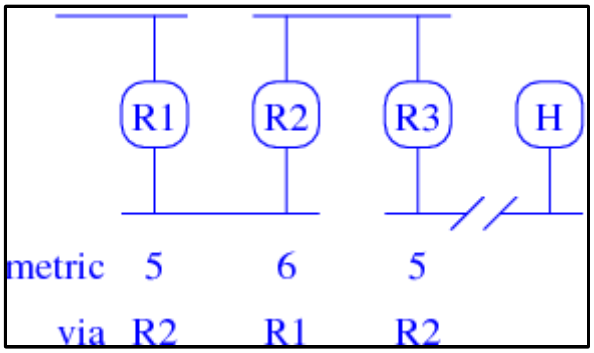
RIP Slow Convergence



- And R1 learns its route via R2 is now of metric 5. R1 sends out messages with its new metric

Routing IP

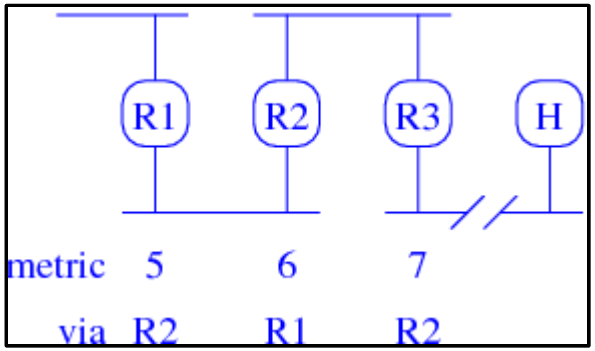
RIP Slow Convergence



- R2 finds its route via R1 has increased its metric, so it updates it tables and sends out messages with metric 6

Routing IP

RIP Slow Convergence



- R3 updates its table, sends out routes with metric 7
- And so on
- The counts eventually reach 16, and after a couple more minutes the routes are finally removed

Routing IP

RIP Slow Convergence

- Meanwhile data packets are being bounced forwards and backwards between the routers until their TTLs reach 0
- The issue is that R2 cannot know that R3's route is via R2: RIP does not provide this information

Routing IP

RIP Problems

- Another problem is that RIP is not suitable for use between ASs as it is ignorant of subnetting and possibly reveals too much about the internal structure of the AS
- As RIP uses broadcast a router only every gets information from its immediate neighbours and this make getting a global view of the network difficult

Routing IP

RIP Problems

- The metric limit of 15 is not big enough for larger networks: making it bigger only makes the count to infinity worse
- The only criterion for choice of route is the metric: this is much too simplistic. Other considerations including network speed, bandwidth and cost should be taken into account

Routing IP

RIP

- However, RIP is popular as it is simple to use
- RIP v2 looks at subnets and EGPs, but OSPF is a better solution

Routing IP

OSPF

- To understand the *Open Shortest Path First* protocol we must first look at *Dijkstra's Algorithm*

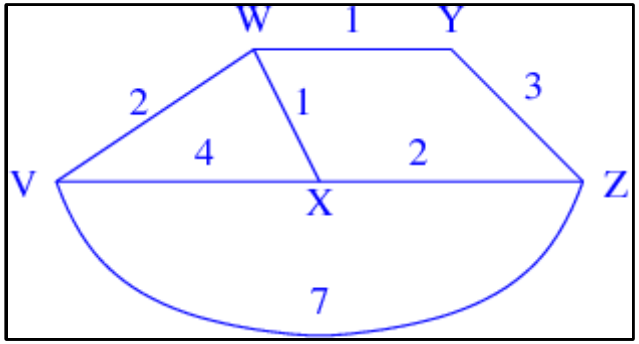
Routing IP

Dijkstra's Algorithm

- This is a method for finding best paths through a network, where “best” means “shortest” or “lowest cost” or whatever you want to measure
- It is used in several routing protocols

Routing IP

Dijkstra's Algorithm



- We want to find the shortest/cheapest path between V and Z, where costs are as given
- We make a table of paths, containing a *predecessor* to a node and a *cost* to get to that node. Also mark each node *determined* or *undetermined*

Routing IP

Dijkstra's Algorithm

0. Initialise all costs to ∞, predecessors to blank and all nodes undetermined
1. Initialise the cost of the starting node V to 0
2. While there are any undetermined nodes:
   a) pick an undetermined node of lowest cost; call it C
   b) mark C determined
   c) for each undetermined neighbour N of C
      if cost to C + cost N to C < cost to N we have found a shorter path to N via C; make the cost of N be cost to C + cost N to C and the predecessor of N to be C

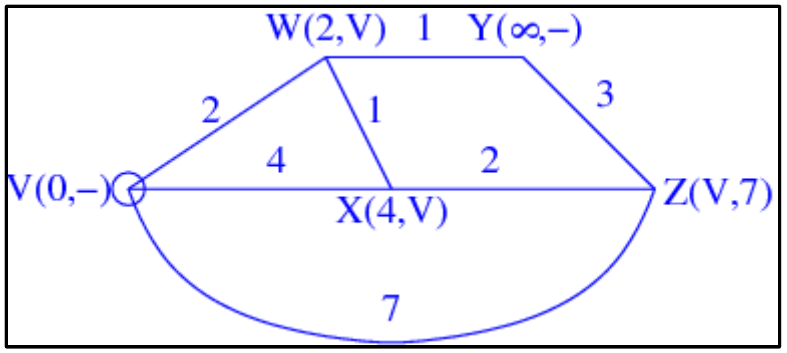
Routing IP

Dijkstra's Algorithm

- Determined nodes are those for which we definitely know the best route; undetermined are those for which we might yet find a better route

Routing IP

Dijkstra's Algorithm

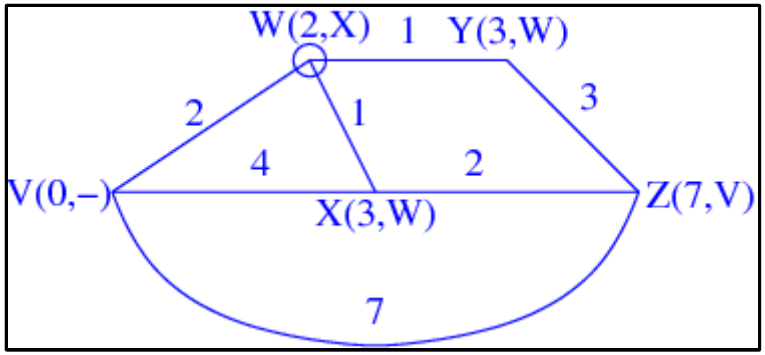


V	W	X	Y	Z
0	2	4	∞	7
-	V	V	-	V
det	undet	undet	undet	undet

- Start. Pick cheapest undetermined node: V
- Make it determined and consider its neighbours W, X and Z
- All have infinite cost, so make their predecessors V and adjust their costs

Routing IP

Dijkstra's Algorithm

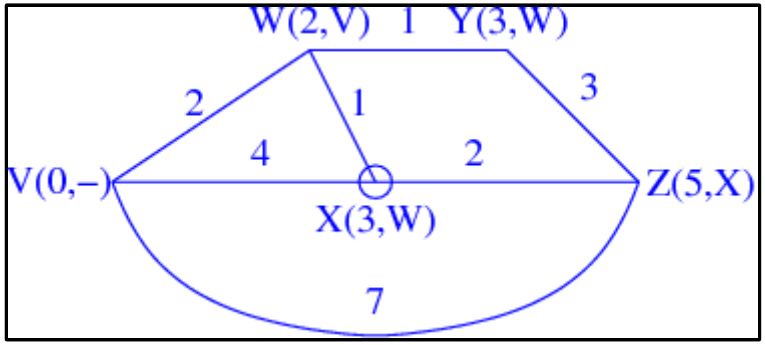


V	W	X	Y	Z
0	2	3	3	7
-	V	W	W	V
det	det	undet	undet	undet

- Pick cheapest undetermined node: W
- Make it determined, and consider its undetermined neighbours X and Y
- Cost to X via W is 2+1<4, so revalue X; cost to Y via W is 2+1<∞, so revalue Y

Routing IP

Dijkstra's Algorithm

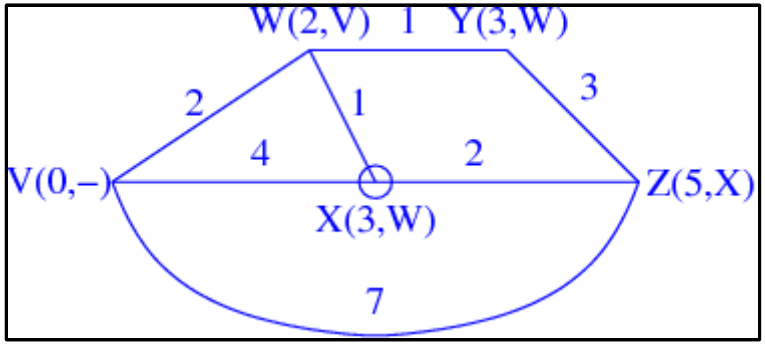


V	W	X	Y	Z
0	2	3	3	5
-	V	W	W	X
det	det	det	undet	undet

- Pick cheapest undetermined node: X (or Y)
- Make it determined and consider its undetermined neighbour Z
- Cost to Z via X is 3+2<7 so revalue Z

Routing IP

Dijkstra's Algorithm

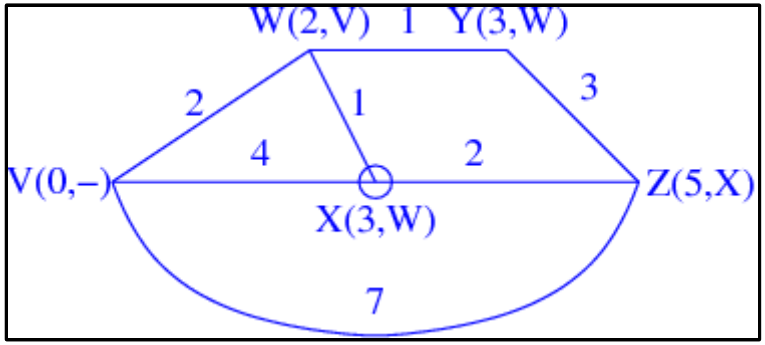


V	W	X	Y	Z
0	2	3	3	5
-	V	W	W	X
det	det	det	det	undet

- Pick cheapest undetermined node: Y
- Make it determined and consider its undetermined neighbour Z
- Cost to Z via Y is 3+3>5 so don't revalue Z

Routing IP

Dijkstra's Algorithm



V	W	X	Y	Z
0	2	3	3	5
-	V	W	W	X
det	det	det	det	det

- Pick cheapest undetermined node: Z
- Make it determined. It has no undetermined neighbours so we are done

Routing IP

Dijkstra's Algorithm

Final costs:

V	W	X	Y	Z
0	2	3	3	5
-	V	W	W	X

- So cheapest path from V to Z is of cost 5
- Also have computed cheapest paths from V to every other node

Routing IP

Dijkstra's Algorithm

Final costs:

V	W	X	Y	Z
0	2	3	3	5
-	V	W	W	X

- And have computed the paths: to get from V to Z read the table in reverse. Start with Z and read successive predecessors: Z X W V
- So path is V W X Z

Routing IP

OSPF

- *Open Shortest Path First* is an interior gateway protocol that addresses the problems of RIP
- RIP counts hops while OSPF considers the states of its neighbours and passes this throughout the AS
- Each router gets this information and uses it to build its own view of network connectivity

Routing IP

OSPF

- OSPF runs directly on top of IP (RIP is layered over UDP over IP) and has many advantages over RIP

Routing IP

OSPF vs RIP

- OSPF can have different routes for different type of IP service
- An interface is assigned a cost; this can be computed from anything relevant, e.g., reliability, throughput, round-trip-time
- If more than one route of equal cost exists OSPF can share traffic equally between them (*load balancing*)

Routing IP

OSPF vs RIP

- OSPF understands subnets
- A simple authentication system can be used to prevent spoofing of routes
- OSPF uses multicast rather than broadcast, so only hosts that are interested have to listen to routing traffic
- Also multicast means that information can be transmitted beyond the local network, anywhere within the AS

Routing IP

OSPF vs RIP

- On the other hand, RIP is simple to set up, while OSPF can be complex
- OSPF uses more compute power as it need to run Dijkstra's algorithm
- OSPF is a complicated system that is hard to implement
- Problems (like routing loops) can still occur if routers' views of the network get out of sync

Routing IP

OSPF

- When a router starts it tests for neighbours by multicasting to 240.0.0.5, the *AllSPFRouters* address, with TTL set to 1
- It periodically multicasts its link state tables to the routers in the AS
- It multicasts updates when changes happen in the network
- It receives updates from other routers in the AS

Routing IP

OSPF

- It keeps its link state tables up-to-date using this information
- It computes its routing table using (say) Dijkstra's algorithm

Routing IP

OSPF

- The cost of a link in OSPF can be set according to any desirable criteria
  - technical (bandwidth, packet size, round trip time, etc.)
  - economic (monetary cost of using a link)
  - political (traffic from Research to Development must not pass through Accounting)
- This is called *policy based routing*

Routing IP

OSPF

- OSPF is good because it reacts quickly to network changes (c.f., RIP counting to infinity)
- It requires minimal network traffic overhead
- It enables multiple routes between hosts
- It enables policy based routing
- It is ideal for large, complex networks

Routing IP

BGP

- The *Border Gateway Protocol* is an *Exterior Gateway Protocol* (EGP), and so is used for routing between ASs
- The problem is now to route between ASs rather than networks, and policy based routing is dominant

Routing IP

BGP

- BGP classifies ASs three ways:

1. A *stub* AS. This has exactly one connection to another AS and only carries local traffic
2. A *multihomed* AS. This has more than one connection, but does not carry anyone else's traffic
3. A *transit* AS. This has more than one connection and will carry traffic from one AS to another, usually with some policy restrictions

Routing IP

BGP

- BGP allows policy based routing (of course!)
- It is a distance-vector protocol
- ASs do not change very much so there is no real problem with slow convergence