Queen Mary
University of London

IP Addressing and Internet Protocol

EBU5211: Ad Hoc Networks

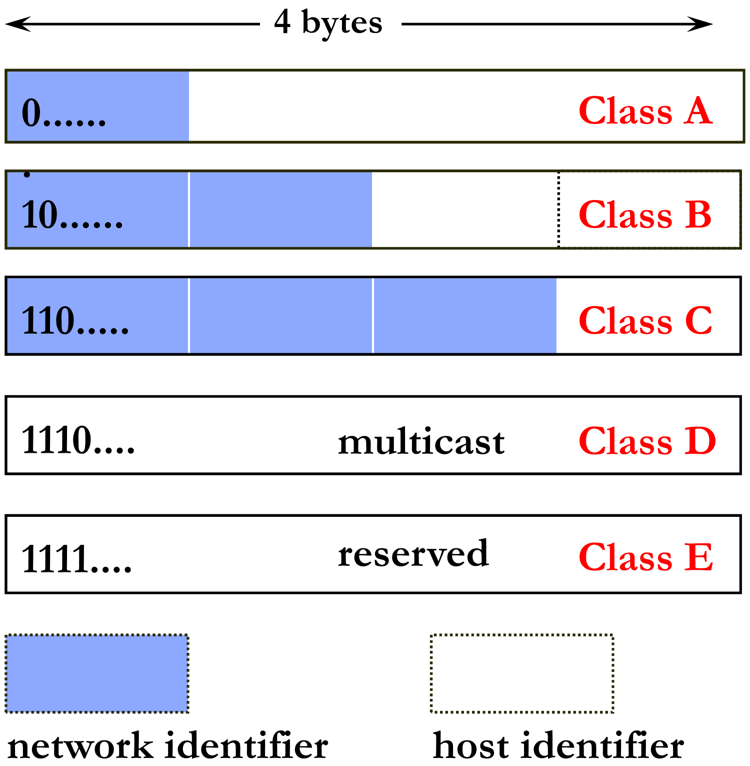
Dr. Yan SUN (Cindy)

IP address

- a unique address of each network interface connected to a network which supports Internet Protocol.
- a host on the internet can have more than one interfaces connected to different networks. i.e a router must have two or more physical interfaces for interconnecting LANs and/or WAN transmission facilities
- the original version of IP, IP Version 4 (IPv4), uses a 32-bit binary (base 2) address.
- each IP address is composed of a network identifier and a host identifier.

Example: 138.37.32.112

Classful IP Address Structure



- Dotted decimal notation xxx.xxx.xxx.xxx
- Starting number, n, shows whether Class A, B or C
  - Class A: $n < 128$ (supports up to 16 million hosts)
  - Class B: $128 \leq n < 192$ (support up to 65K hosts)
  - Class C: $192 \leq n$ (support up to 254 hosts)

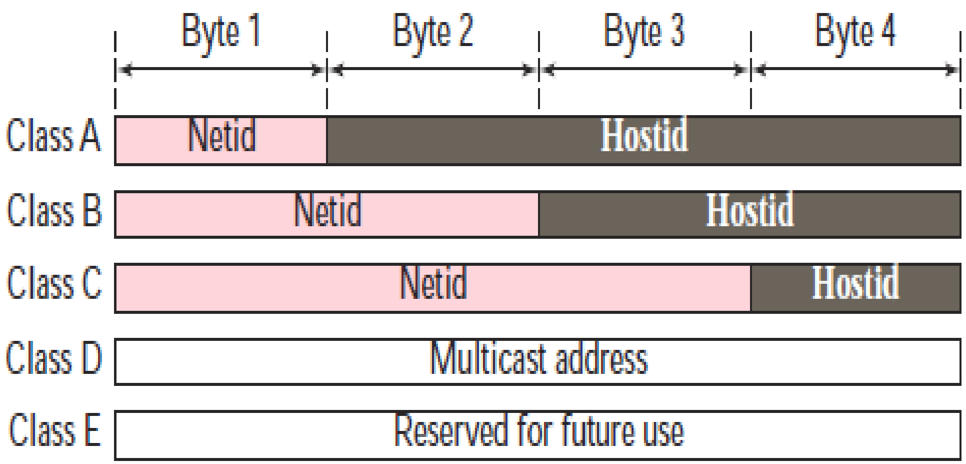
127.x.x.x = local loopback

There is no universal broadcast address. 255.255.255.255 is limited broadcast

Classless Internet Domain Routeing (CIDR) Addressing has the partition anywhere (not just at byte boundaries)

Classful IP Address Structure

Queen Mary University of London



- Dotted decimal notation xxx.xxx.xxx.xxx
- In classful addressing, the IP address space is divided into five classes: A, B, C, D, E.
- Starting number, n (first byte), shows whether Class A, B or C
  - **Class A**: n < 128 (up to 16m hosts)
  - **Class B**: 128 ≤ n < 192 (up to 65K hosts)
  - **Class C**: 192 ≤ n < 224 (up to 254 hosts)

	Octet 1	Octet 2	Octet 3	Octet 4
Class A	0........			
Class B	10......			
Class C	110.....			
Class D	1110....			
Class E	1111....			

Binary notation

	Byte 1	Byte 2	Byte 3	Byte 4
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–299			
Class E	240–255			

Dotted-decimal notation

Example – at QM

138	37	32	112

Network part Host part

- Since I is in the range $128 \le I < 192$ we know this is a Class B address
- We can write the address in a form that is a network address by putting 0s in the host part:

    138.37.0.0

    Usually done with subnets – see later

Private IP addresses

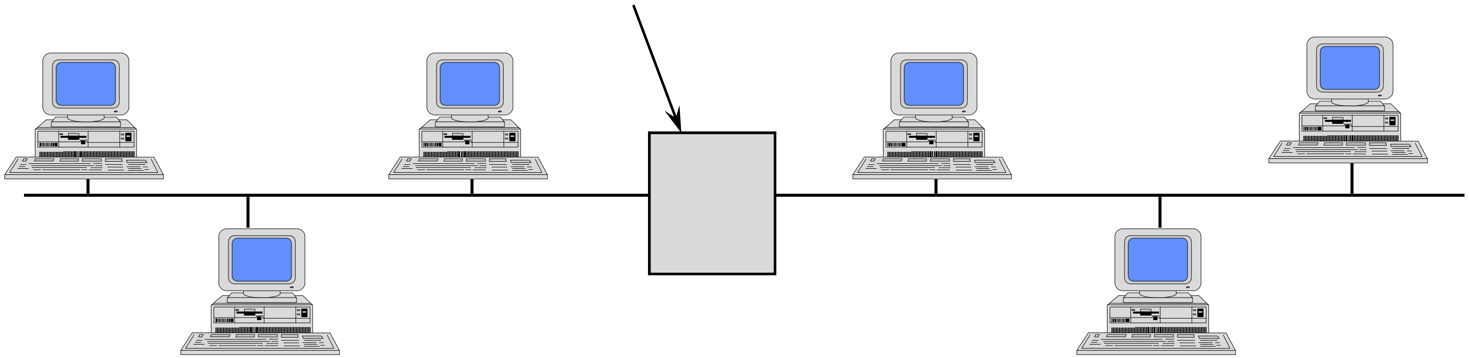
- Reserved for private intranets
  - 10.0.0.0 - 10.255.255.255
  - 172.16.0.0 - 172.31.255.255
  - 192.168.0.0 - 192.168.255.255
- Used within organisations and within the home
- Use NAT (Network Address Translation) at edge of private domain (e.g in ADSL router) if need to connect to Internet.
- IP addresses in the range of 169.254.0.0 -169.254.255.255 are reserved for Automatic Private IP Addressing – a Windows feature.
- Loopback range used for testing: 127.x.x.x

Routers

A Router is a network layer device that controls the routeing of traffic flows between networks

Routing protocols are distributed and dynamic mechanisms for determining the best path a flow should take across an inter-network

This “box” is the router if it separates networks



A router can physically be a workstation or a special-purpose piece of hardware

Routers

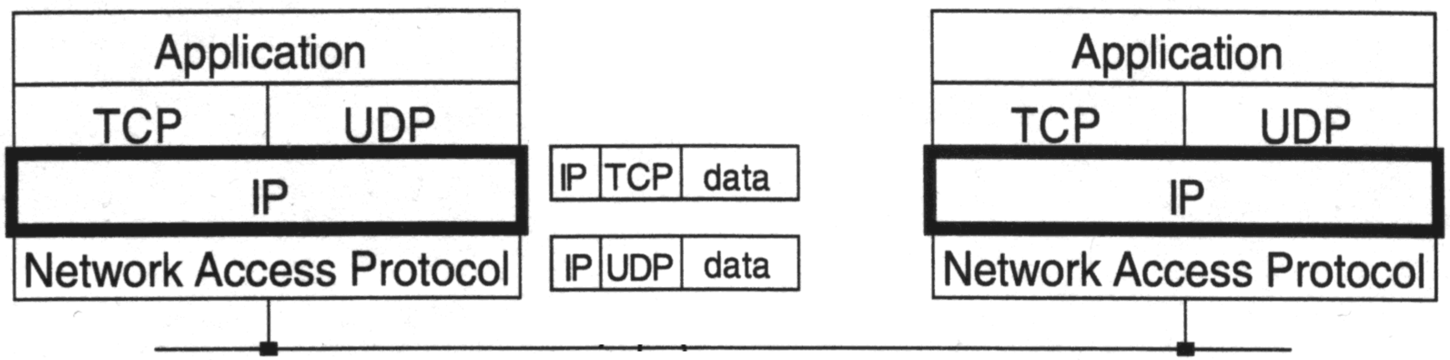
- Routers operate at layer 3.
- a router must have two or more physical interfaces for interconnecting LANs and/or WAN transmission facilities.
- a router uses two types of network protocols
  - **routable protocols**, also known as **routed protocols**, are those that encapsulate user information and data into packets, i.e. IP.
  - **routing protocols** are used between routers to determine available routes, communicate what is known about available routes and forward routed protocol packets along those routes. i.e. RIP, OSPF

Internet Protocol - IP

- Was designed primarily for internetworking
- Essentially, it aims to provide a "**best effort**" service over the network layer in the form of a **datagram** service
- Data from the transport layer (TCP or UDP) is converted into IP datagrams and carried over the network
- The network is considered essentially "dumb" and "**unreliable**" so it is left to the end points or applications to confirm that application data has been delivered correctly and to take action if it has not (i.e. ordered delivery)
- Intermediate nodes within the networks (IP routers) have essentially two simple functions to perform: to **route/forward** IP packets towards their destination and to **fragment** larger PDU data blocks into IP sized blocks for transfer

Internet Protocol Functions

- **Encapsulates TCP and UDP packets**
- **Common logical interface to different networks**
- **Universal 32-bit address mechanism**
  – Uses dotted decimal notation; e.g., 143.119.4.23
- **Connectionless network protocol**



Functions of the Internet Protocol

It provides:

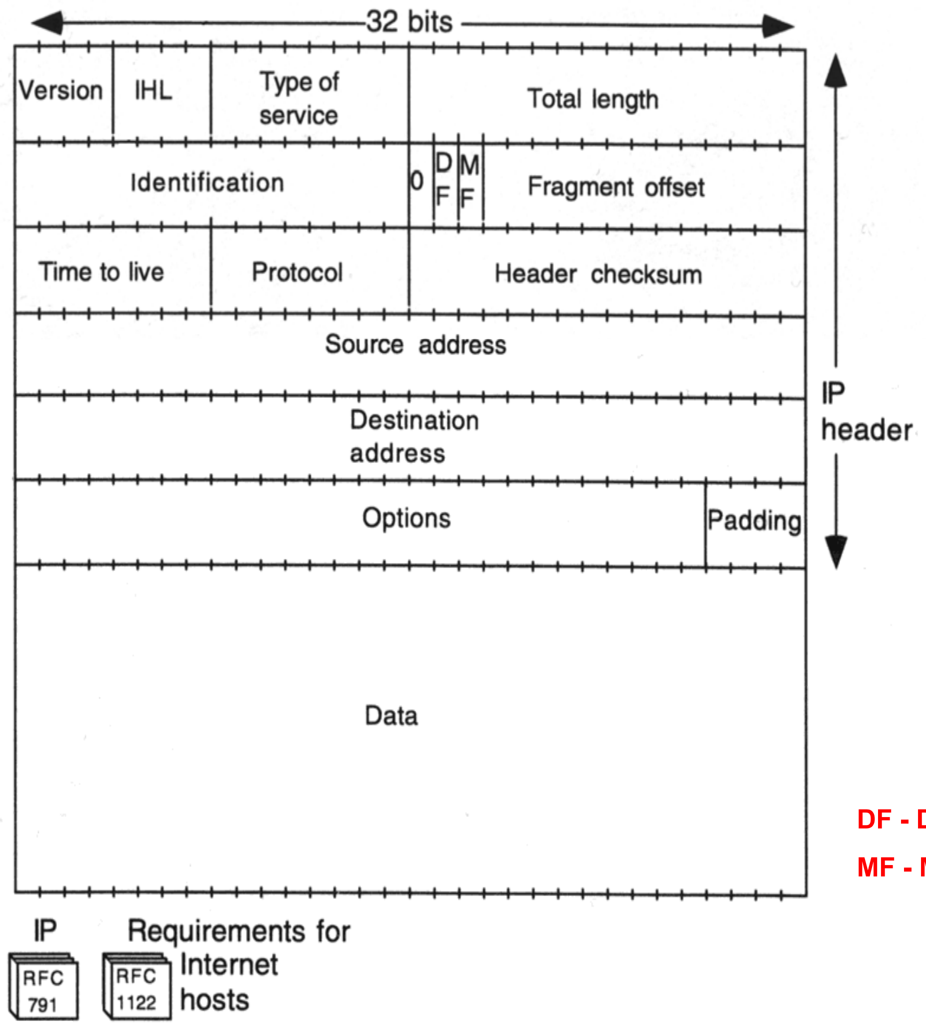
- Physical network independence for higher-layer processing
- Logical address for network stations
- Independence from maximum transmission unit (MTU) size
- Fragmentation and reassembly control
- Datagram service – no QoS guarantees
- Simple core network responsible for routing/forwarding data

Advantages

- Simplicity and less overhead
- Upper layers can build more reliable service
- Adequate for many networks

IP Datagram Structure

The IP datagram consists of a **13-field** variable length header plus the data field itself. The maximum length of any IP datagram is **64kbytes**.



DF - Don't Fragment

MF - More Fragments

IP Datagram Structure

Time To Live (TTL):

Used to count hops and prevent packets from overstaying in the network. Originally, intended to be a time-based count but became a hop-based count A router simply decrements the field by one each time a PDU passes through. When the field reaches Zero, the PDU is discarded

Protocol:

Indicates which transport protocol the datagram is associated with. Fully defined in RFC 1700, a value of 6 indicates that the payload should be passed to TCP. Likewise 17 refers to UDP

Checksum:

Provides a header integrity check (needs to be recalculated after each hop as the TTL field changes). Packets with corrupted headers are discarded

IP Datagram Structure

Source Address:

Unique 32bit address of an originating interface on an internet

Destination address:

Unique 32bit address of an destination interface on an internet

Options:

Allows IP to support various options, such as security and source routing

Padding:

Optional null data to ensure the header length is aligned on a 32bit boundary

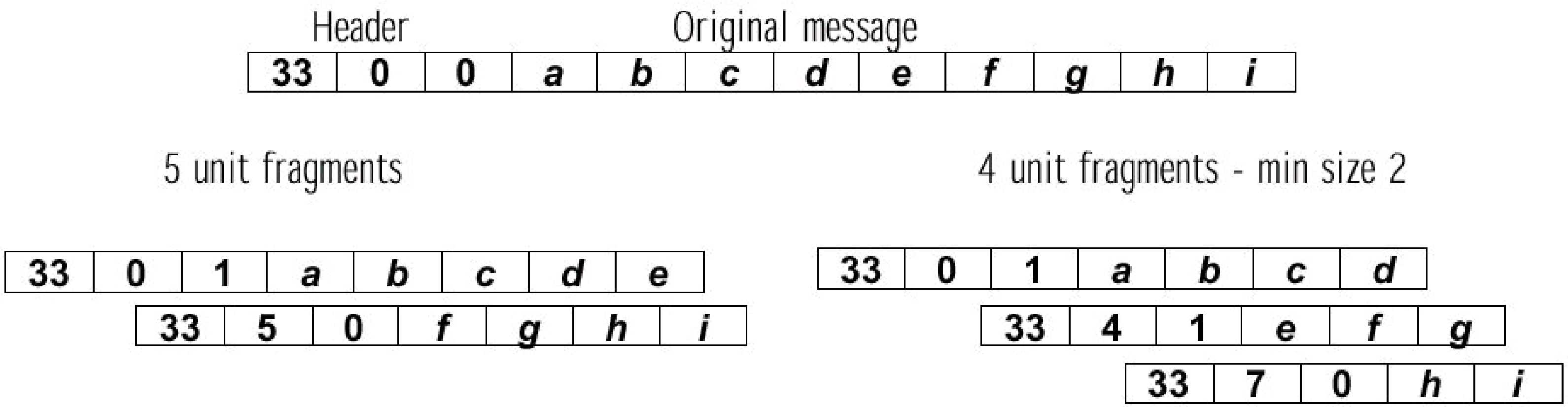
Data:

Contains upper-layer information

IP Datagram Structure

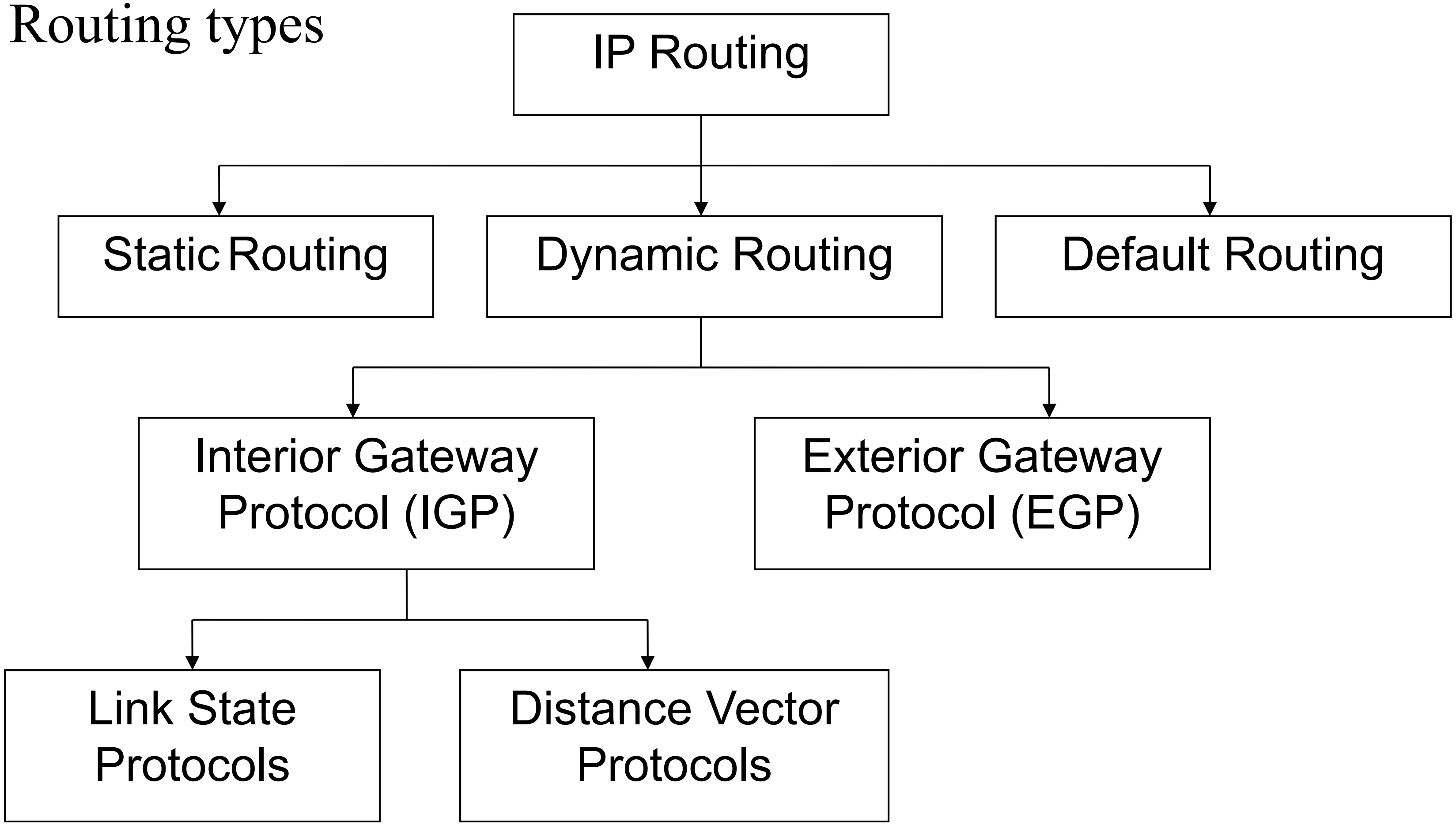
When fragmenting a packet it is necessary to indicate in the fragment header in some way the fragment size, the number of fragments in that packet and the position of these fragments in the final message.

In this example, the header contains three fields: the message number, the number of the first fragment in the packet and an end of packet bit. This is analogous to the fragmentation process that can take place in IP systems.



IP Network Routing

Routing types



IP Routing

Source Routing

Uses a pre-selected set of hops programmed into each datagram

Next-Hop (hop by hop) Routing

Each router along the path chooses the next hop towards each datagram's ultimate destination based on information within a local routing table and contained within the datagram – i.e. the destination IP address

Default Route

- Most end-systems have a single route defined
- Sometimes multiple gateways can be accommodated
- Default routes can also be set up within Routing Tables – as a "catch all" when more specific matches cannot be made

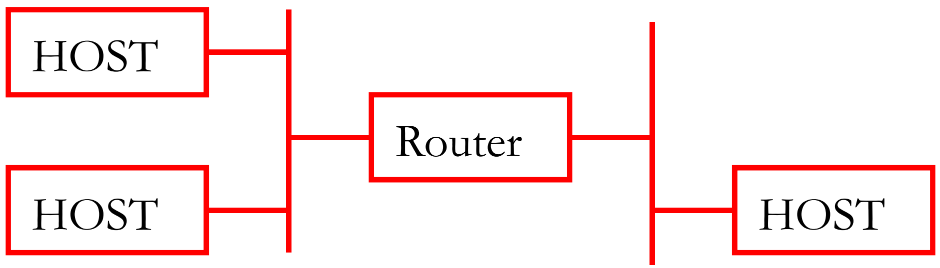
Static IP Routing

If a host cannot send a packet directly to destination, it has to select a router. Most end-systems have a single DEFAULT router that is manually configured. This is frequently known as a DEFAULT GATEWAY.

If gateway fails, end-system may need to be manually reconfigured to utilize an alternative

A Static Routing Table can be placed within each Router. This is simple but not scalable

Host Route Determination



Host checks network part of destination IP Address to determine whether it can reach host directly. If not, the packet is forwarded to the default gateway

Static Routing

- Each router manually configured with a list of destinations and the next hop to reach those destinations
- Static routing ideal for small number of destinations or "stub" networks
- Static routing is simplistic approach
- Good for simple environments and flat topologies
- Shortcomings:
  - Cumbersome to configure
  - Cannot deal with link/node failures, addition of new nodes and change of link metrics
- Solution - Dynamic Routing

Dynamic Routing

- Routers compute routing tables dynamically based on information provided by other routers in the network
- Routers communicate topology to each other via different protocols
- Each router then computes one or more next hops for each destination - trying to calculate the most optimal path
- Better convergence than for manual intervention

General Rule of Thumb

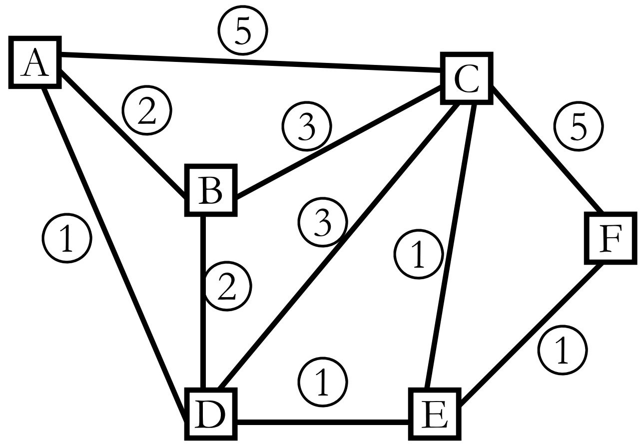
- Use static routes where you *can*.
- Use dynamic routing where you *must*

Convergence

- When something changes (i.e. when a link or router goes down), it takes a while before the change is propagated to all affected routers.
- Convergence is when all the routers have common routing information i.e. they are consistent.
- When a network is not converged, there is network downtime
  - Packets don't get to where there are supposed to be going
  - Occurs when there is a change in the status of a router or the status of a link

Distance Vector Algorithms[1]

- Distance:
  - hop count, queue length, …
- Each node:
  - evaluates distance to all other nodes
  - distributes <destination, cost> information to adjacent nodes
  - finds shortest (lowest distance) route to to all other nodes



DV - Bellman, Ford in 1969

Distance Vector Algorithms[2]

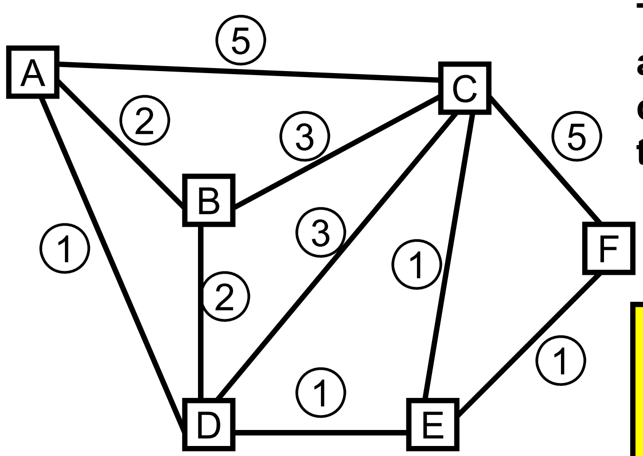


Table entries take into account the additional cost of getting from A to the first hop

node	queue length
B	2
C	5
D	1

Table1:Cost to neighbours from A

Initial local knowledge

to \ from	B	C	D
A	2	3	1
B	0	3	2
C	3	0	2
D	2	2	0
E	3	1	1
F	4	2	2

Table2: Distance vectors received by A

Contents of a single RIP Response message

route via	1st hop distance	remaining distance	total distance
B	2	3	5
C	5	1	6
D	1	1	2

Table 3: Calculation of routing table entry to E from A

Destination	Distance	First hop
A	0	-
B	2	B
C	3	D
D	1	D
E	2	D
F	3	D

Table 4: New routing table for A

Distance Vector Algorithms [3]

- Routing by Rumor
- Distance
  - cost
  - hop count
- Vector
  - “direction” to go
  - next hop
- Listen to neighbouring routers
- Install all routes in table, lowest distance wins
- Broadcast all routes in table - not scalable
- Very simple - slow to converge
- Very stupid - split horizons

Routing Information Protocol (RIP)

RIP is a distance-vector routing protocol that uses a single routing metric (hop count) to measure the distance between the source and a destination network

RIP is an *interior gateway protocol* (IGP), which means that it performs routing within a single autonomous system

Maximum distance is 15 hops – a hop count of 16 is defined as infinity so it is only suitable for small networks

It is formally defined in two documents: RFC1058 and RFC1723

The RIPv1 Packet Format

0 31

<table>
<tr><td>Command(1)</td><td>Version(1)</td><td colspan="2">Unused - All Zeros (2)</td></tr>
<tr><td colspan="2">Address Family Identifier (2)</td><td colspan="2">Unused - All Zeros (2)</td></tr>
<tr><td colspan="4">IP Address (4)</td></tr>
<tr><td colspan="4">Unused - All Zeros (4)*</td></tr>
<tr><td colspan="4">Unused - All Zeros (4)*</td></tr>
<tr><td colspan="4">Metric (4)</td></tr>
</table>

*scope for larger address fields (non-IP)

Command: 1 = Request, 2 = Response

Version: 1 for RIPv1

Address Family: 2 = IP

Routing Information Protocol (RIP)

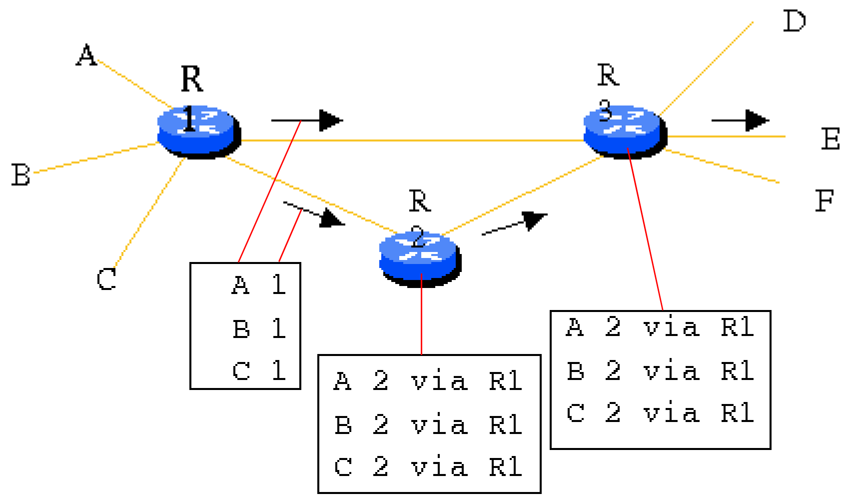
- RIP messages are sent using UDP (port 520) and are normally sent as a broadcast
- Maximum message size of 512 bytes
- 25 entries/message
- Follows a request/response + update mechanism
  - Each router sends out a periodic broadcast response that contains the entire routing table
  - Triggered responses of changed entries are sent when there is a change of link status

RIP Protocol Operation

- At initialisation routers are configured with directly connected network addresses
- RIP Broadcast request to ask for RIP information on each RIP-enabled interface
- Router compares received RIP table with its own. Routers add new entries and update existing destinations, if the (hop count) cost is lower
- RIP Broadcast response packets sent periodically

RIP Protocol Operation

- Nodes set 'distance' to all directly connected neighbours (i.e. 1 hop), assigning infinite distance (i.e. 16 hops) if link is down
- Directly connected neighbours then exchange routing tables (distance vectors) periodically
  - Nodes update distance tables if shorter path is advertised
  - Result: All nodes know lowest cost path to any given node



RIP Protocol Operation

- RIP updates in the form of the complete routing table are sent at regular intervals (i.e. every 30 seconds*)
- If a route is not refreshed within 180 seconds*, the distance is set to 16 and the entry is marked (unreachable) for subsequent removal (typically 60 seconds later)
- RIP requests are of two types
  - Complete routing table requests
  - Specified entries request
- Triggered updates are used for faster convergence when the network topology changes

*** These default times can be changed but MUST be the same throughout the AS routing domain**

Four-Step Routing Table Update

- Check for validity of the update, if invalid ignore the update
- Look for the corresponding destination
- If destination already present update entry if cost is lower. Reset timeout.
- If destination not found add it

If lower cost entry is already found in the table and the recorded advertising (next hop) router is the same as the one issuing the response then the route is marked as unreachable for a holddown period. If the neighbour is still advertising the higher hop-count at the end of this time, the new metric is accepted.

RIP Routing Table Structure

- Each RIP routing table entry includes:
  - Destination IP Address
  - Metric (hop count 1-15, 16 unreachable)
  - Next Hop, Advertising Router – split horizons
  - Timeout (seconds)
- Directly connected networks typically have a metric of 1 – indicating a one hop cost.
- If a route times-out the metric is set to 16 ("infinity") and deleted after 1 minute by default.

```
PC3> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       B - BGP, > - selected route, * - FIB route

R>* 47.0.0.0/8 [120/2] via 49.0.0.11, ep0, 00:25:58
R>* 48.0.0.0/8 [120/2] via 49.0.0.11, ep0, 00:25:58
C>* 49.0.0.0/24 is directly connected, ep0
C>* 127.0.0.0/8 is directly connected, lo0
S>* 224.0.0.5/32 [1/0] via 127.0.0.1, lo0
S>* 224.0.0.6/32 [1/0] via 127.0.0.1, lo0
PC3>
```

RIP Router Types

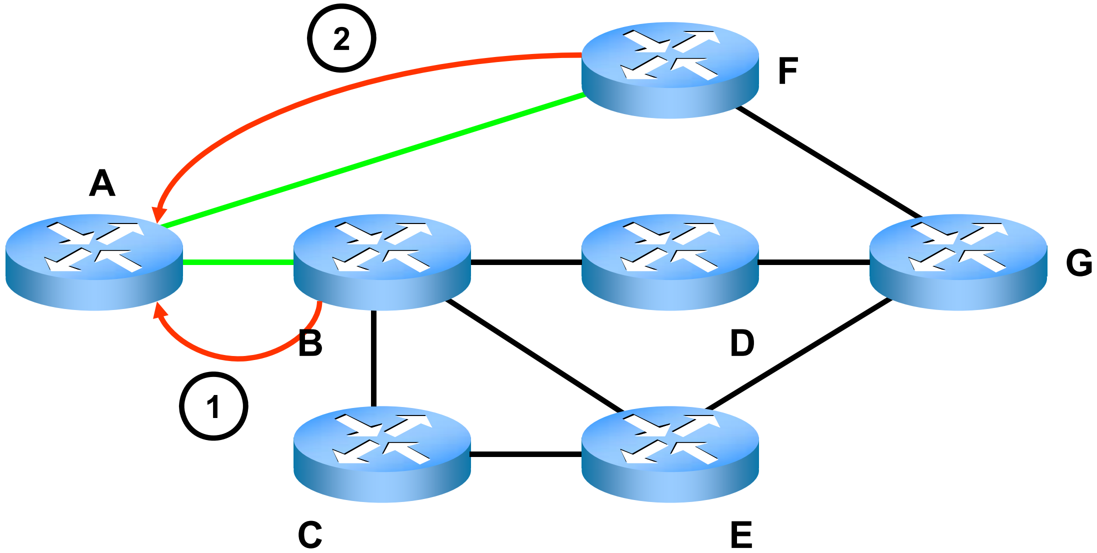
Active Gateways

Advertise routes by broadcasting a message every 30 seconds. The message contains Distance/Vector pairs of an IP network address along with a hop count (metric)

Passive Hosts

Listen and update their routing tables based upon the router advertisements. Passive hosts don't advertise

Converging: An Example



Initialize directly connected destinations

RT @ A

B – 1 via B (dc)
F – 1 via F (dc)
dc - directly connected

1

C – 1
D – 1
E – 1
G – 2
F – 3

RT @ A

B – 1 via B
C – 2 via B
D – 2 via B
E – 2 via B
G – 3 via B
F – 1 via F

2

B – 3
C – 3
D – 2
E – 2
G – 1

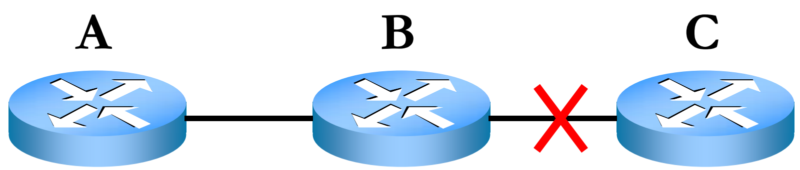
RT @ A

B – 1 via B
C – 2 via B
D – 2 via B
E – 2 via B
G – 2 via F

Better Offer

Routing Information Protocol (RIP)

- RIP specifies a number of stability features:
  - RIP implements the **split-horizon** and **hold-down** mechanisms to prevent incorrect routing information from being propagated
  - The RIP hop-count limit prevents routing loops from continuing indefinitely
- RIP uses numerous timers to regulate its performance including:
  - routing-update timer, route timeout, and route-flush timer



Upon link failure B can't see C

A tells B – It can see C 2 hops away

B wrongly concludes that A has another route to C

Split Horizons

Routing Table Extract at A

Dest	Next	Cost	Time
C	B	2	xx

1 Failure arises on link between B and C

Routing Table Extract at B

Dest	Next	Cost	Time
C	C	16	xx

2 B announces unreachable link to A but A has a lower cost link (via B!!)

Dest	Next	Cost	Time
C	B	2	xx

3 A tells B it can see C 2 hops away and B thinks A has an alternative path

Dest	Next	Cost	Time
C	A	3	xx

4 A's low cost path times out and A now accepts the path announced via B that's 3 hops away (via A!!)

Dest	Next	Cost	Time
C	B	4	xx

Etc...

Dest	Next	Cost	Time
C	A	5	xx

The process repeats until the cost progressively escalates to 16 and the route is finally considered truly unreachable

Motto: Store who you learn things from. Don't tell them what they've told you

Convergence Time Improvements

Split Horizon Update

- Router does not broadcast routing information over the same interface over which it was received

Poison Reverse

- Router retains an unreachable route in its table for two broadcast periods and broadcasts the destination as unreachable

Triggered Updates

- Route changes require a router to issue a broadcast message rather than waiting for the normal broadcast interval

Hold-Down Timer

- Router does not change information about a route following a message indicating that the destination is unreachable (for 60 seconds)

What's wrong with RIP

- Broadcasts (not scalable)
- Infinity of 16 (not large enough)
- Routing loops
- Poor robustness – only one path to a destination is stored
- Does not use variable length subnet masks - classful
- Insecure
- Reliance on fixed metrics to calculate routes
- Slow convergence – instability when routes change rapidly

Link State Routing

- Every node knows cost to direct neighbours (link state)
- Link state of each node flooded to all routers in the network
- Nodes have enough information to build complete network topology with link metrics
- Link state requires:
  - Reliable broadcasting
  - Calculation of lowest cost path from LS information eg: OSPF

Link State Vs Distance Vector

- Distance Vector
  - Routers compute the best path from information passed to them from neighbours
  - Adds distance vectors from router to router
  - Frequent, periodic updates; slow convergence
  - Passes copies of routing table to neighbour routers
- Link-State
  - Gets common view of entire network topology
  - Calculates the shortest path to other routers
  - Event-triggered updates: faster convergence
  - Passes link-state routing updates to other routers
  - Link State routers each have a local copy of the entire network map from which the best routes are computed

Dijkstra's Algorithm

Uses three datastructures:

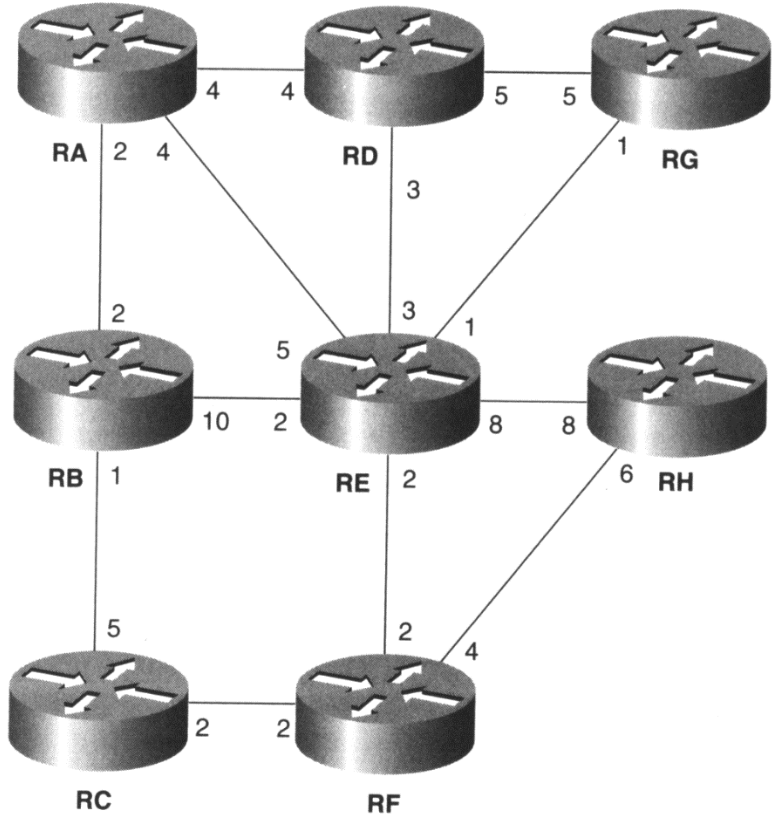
Link State Database **Input Reference**	Repository of all known links
Candidate Database **Temporary Structure**	Working SP tree to which links are added from the Link State Database
Tree Database **Finished Output**	Contains complete shortest path tree once the algorithm finishes

Dijkstra's Algorithm

1. A router initializes the Tree database by adding itself as the root. This entry shows the router as its own neighbour, with a cost of 0.
2. All triples in the link state database describing links to the root router's neighbours are added to the Candidate database.
3. The cost from the root to each link in the Candidate database is calculated. The link in the Candidate database with the lowest cost is moved to the Tree database. If two or more links are an equally low cost from the root, choose one.
4. The Neighbour ID of the link just added to the Tree database is examined. With the exception of any triples whose Neighbour ID is already in the Tree database, triples in the link state database describing that router's neighbours are added to the Candidate database.
5. If entries remain in the Candidate database, return to step 3. If the Candidate database is empty, then terminate the algorithm. At termination, a single Neighbour ID entry in the Tree database should represent every router, and the shortest path tree is complete.

Dijkstra's Algorithm Example

Example
Network



Dijkstra's Algorithm Example

Candidate	Cost to Root	Tree	Description
		RA,RA,0	Router A adds itself to the tree as root.
RA,RB,2 RA,RD,4 RA,RE,4	2 4 4	RA,RA,0	The links to all of RA's neighbors are added to the candidate list
RA,RD,4 RA,RE,4 RB,RC,1 ~~RB,RE,10~~	4 4 3	RA,RA,0 RA,RB,2	(RA,RB,2) is the lowest-cost link on the candidate list, so it is added to the tree. All of RB's neighbors except those already in the tree are added to the candidate list. (RA,RE,4) is a lower-cost link to RE than (RB,RE,10), so the latter is dropped from the candidate list.
RA,RD,4 RA,RE,4 RC,RF,2	4 4 5	RA,RA,0 RA,RB,2 RB,RC,1	(RB,RC,1) is the lowest-cost link on the candidate list, so it is added to the tree. All of RC's neighbors except those already on the tree become candidates.

New Candidate(s)

Pick the best, based on root cost and add its neighbours as candidates

Dijkstra's Algorithm Example

Candidate	Cost to Root	Tree	Description
RA,RE,4 RC,RF,2 RD,RE,3 RD,RG,5	4 5 7 9	RA,RA,0 RA,RB,2 RB,RC,1 RA,RD,4	(RA,RD,4) and (RA,RE,4) are both a cost of 4 from RA; (RC,RF,2) is a cost of 5. (RA,RD,4) is added to the tree and its neighbors become candidates. Two paths to RE are on the candidate list; (RD,RE,3)is a higher cost from RA and is dropped.
RC,RF,2 ~~RD,RG,5~~ RE,RF,2 RE,RG,1 RE,RH,8	5 9 6 5 12	RA,RA,0 RA,RB,2 RB,RC,1 RA,RD,4 RA,RE,4	**(RA,RE,4)** is added to the tree. All of RE's neighbors not already on the tree are added to the candidate list. The higher-cost link to RG is dropped.
RE,RF,2 RE,RG,1 ~~RE,RH,8~~ RF,RH,4	6 5 12 9	RA,RA,0 RA,RB,2 RB,RC,1 RA,RD,4 RA,RE,4 RC,RF,2	(RC,RF,2) is added to the tree, and its neighbors are added to the candidate list. (RE,RG,1) could have been selected instead because it has the same cost (5) from RA. The higher-cost path to RH is dropped.

Pick the best, based on root cost and add its neighbours as candidates

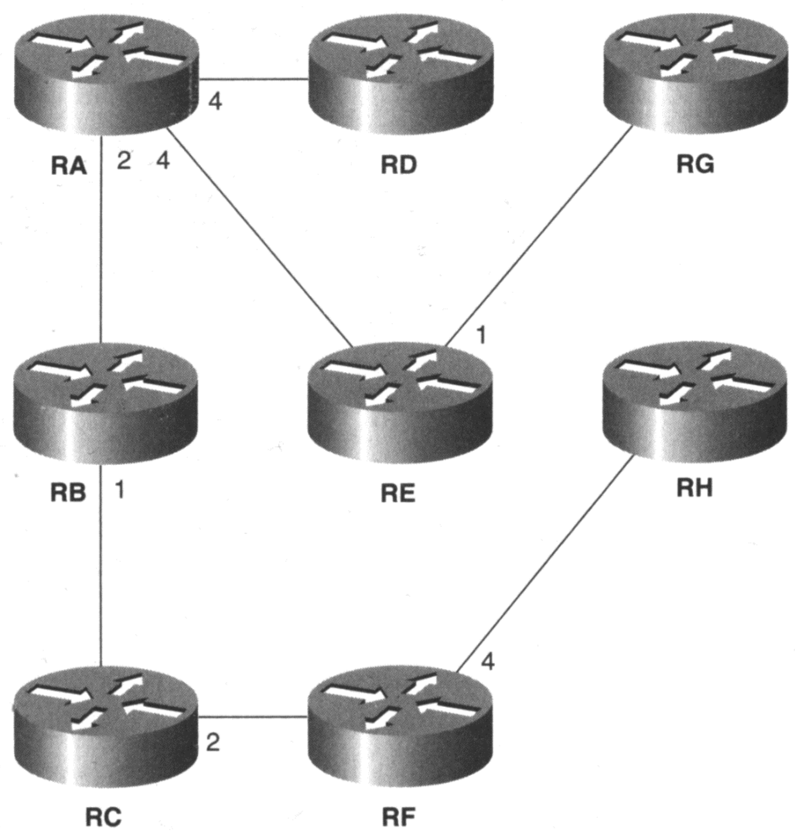
Dijkstra's Algorithm Example

Candidate	Cost to Root	Tree	Description
RF,RH,4		RA,RA,0 RA,RB,2 RB,RC,1 RA,RD,4 RA,RE,4 RC,RF,2 RE,RG,1	(RE,RG,1) is added to the tree. RG has no neighbors that are not already on the tree, so nothing is added to the candidate list.
		RA,RA,0 RA,RB,2 RB,RC,1 RA,RD,4 RA,RE,4 RC,RF,2 RE,RG,1 RF,RH,4	(RF,RH,4) is the lowest-cost link on the candidate list, so it is added to the tree. No candidates remain on the list, so the algorithm is terminated. The shortest path tree is complete.

Pick the best, based on root cost
and add its neighbours as candidates

Dijkstra's Algorithm Example

Shortest Path Tree derived from the algorithm



OSPF – How it Works

1. OSPF-speaking routers send Hello packets out all OSPF-enabled interfaces. If two routers sharing a common data link agree on certain parameters specified in their respective Hello packets, they will become neighbours – discovery mechanism

2. Adjacencies, which may be thought of as virtual point-to-point links, are formed between some neighbours. OSPF defines several network types and several router types. The establishment of an adjacency is determined by the types of routers exchanging Hellos and the type of network over which the Hellos are exchanged

OSPF – How it Works

3. Each router sends link state advertisements (LSAs) over all adjacencies. The LSAs describe all of the router's links, or interfaces, and the state of the links. These links may be to stub networks (networks with no other router attached), to other OSPF routers, to networks in other areas, or to external networks (networks learned from another routing process). Because of the varying types of link state information, OSPF defines multiple LSA types

4. Each router receiving an LSA from a neighbour records the LSA in its link state database and sends a copy of the LSA to all of its other neighbours

OSPF – How it Works

5. By flooding LSAs throughout an area, all routers will build identical link state databases

6. When the databases are complete, each router uses the SPF algorithm to calculate a loop-free graph describing the shortest (lowest cost) path to every known destination, with itself as the root. This graph is the SPF tree

7. Each router builds its routing table from its SPF tree

When all link state information has been flooded to all routers in an area--that is, the link state databases have been synchronized and the routing tables have been built, OSPF is a quiet protocol

OSPF Update – How it Works

- Once an adjacency is established, trade information with your adjacent neighbours
- Topology information is packaged in a "link state announcement" – not complete routing table
- Announcements are sent ONCE, and only updated if there is a change (Complete RT every 30 minutes)

Summary **Change occurs**

"Broadcast" change

Run SPF algorithm

Install output into routing table

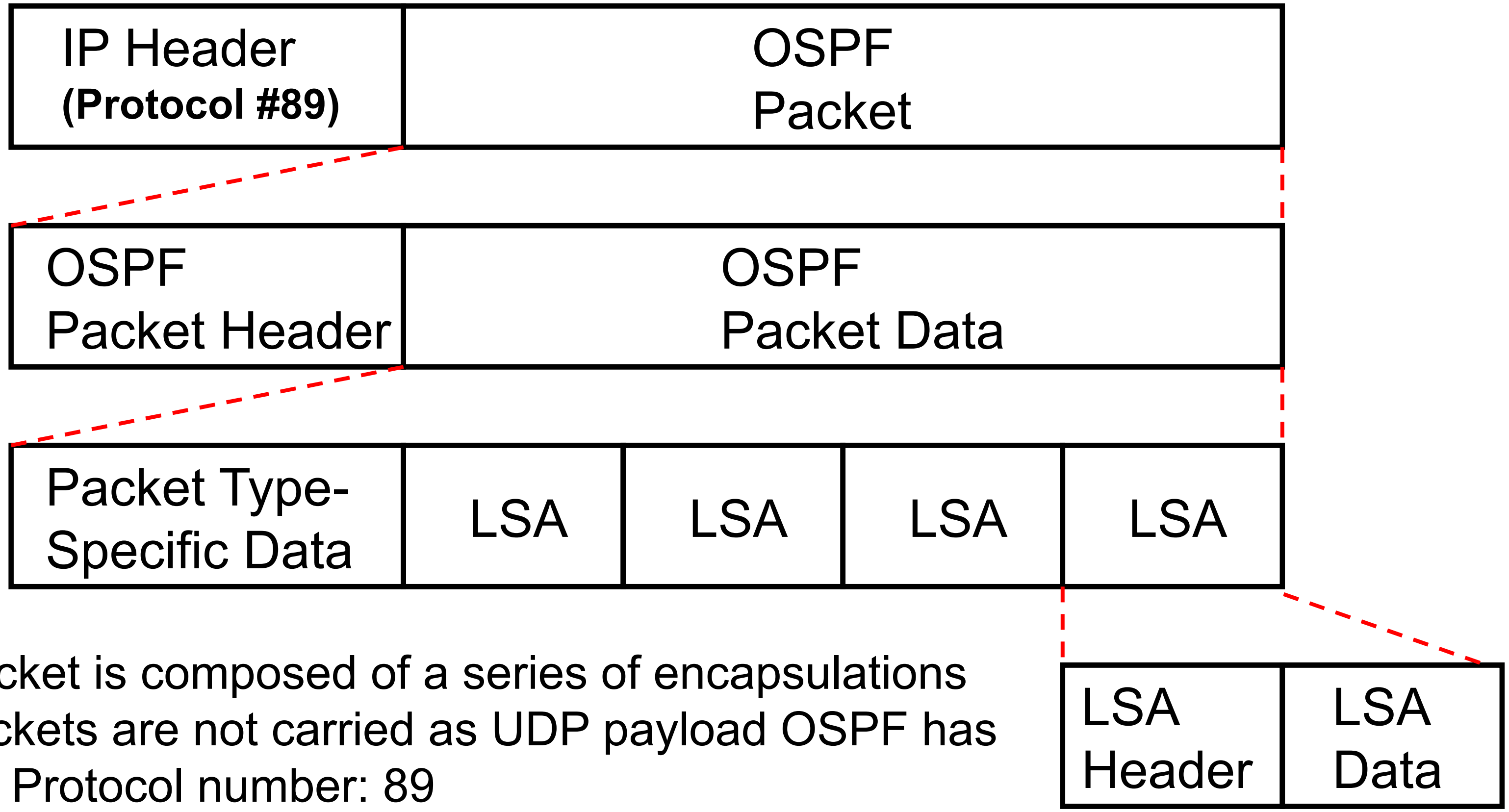
Hello Messages

- Broadcast to neighbours – discovery mechanism
- Receive ACK
- Can exchange routing protocol negotiation information with these neighbours - establishes a 2-way communication
- Repeated periodically – "keep-alive" integrity check
- Enables the establishment of adjacencies

Link State Advertisement

- Flooding (along adjacencies) is implemented using Link State Advertisements (LSAs)
- Nodes add information to LS database and then forward link state information to direct neighbours, who forward to their direct neighbours, and so on.
- Process continues until all nodes have all information
- LSA Contains:
  - Node ID
  - Sequence number
  - Calculation of path done using Dijkstra's shortest path algorithm

OSPF Packet Format



- OSPF Packet is composed of a series of encapsulations
- OSPF packets are not carried as UDP payload OSPF has its own IP Protocol number: 89
- TTL set to 1 in most cases
- Destination IP Address is 224.0.0.5 (AllSPFRouters), 224.0.0.6 (AllDRRouters) or neighbour’s IP address

OSPF Message Types

Type

1 Hello – used for discovery and parameter negotiation

2 Database Description – Used to exchange topology information in summary form

3 Link State Request – Asks for an update – such as when database information is considered too old

4 Link State Update – Sends link state advertisements

5 Link State Acknowledgement – Used to confirm the reception of OSPF messages

Scaling

Each link transition causes a broadcast and SPF run

- OSPF can group routers to appear as one single router using OSPF areas
- Can build OSPF hierarchy to segregate broadcasts

Some Points:

- Rule of thumb: no more than 150 routers/area
- Reality: no more than 500 routers/area
- Backbone "area" is an area
- Always 'area 0'
- Proper use of areas reduce bandwidth & CPU utilization

Route Summarisation limits instability within each Area

Why Subnet?

- Consider the case when an organisation requires address a number of separate networks. One approach is to assign each segment its own Class B or C addresses depending upon the number of hosts. The latter could be used if the number of hosts was sure to be less than 254 otherwise Class B addressing could be used. Both require external routers to have knowledge about the address assignment and thus restricts any changes that could be made by the organisation's network managers.

- If Class B addressing was used then 2^{16} -2 hosts can be addressed. This would be inefficient use of IP addresses if the number of hosts was quite low.

- For example, consider the case when a segment requires 300 addresses; clearly Class B addressing is required but such an allocation leaves 65234 addresses unused. One solution to the above problem is the use of Subnets

Subnetting

- In the IP domain, the term subnet has a particular meaning: it refers to the when a network is split up internally (in addressing terms) into a number of sub networks but remains as a single entity when viewed by the outside world, i.e. nodes not within that network.

- Effectively part of the host address space over which the network manager has control becomes reserved to describe not a host machine but a subnet within the wider organisation.

- This means that only routing tables of nodes within the network need be changed to reflect the new three (rather than two) level hierarchy that exists. Through the use of a subnet mask, local routers can determine which other local router a particular packet is destined for - it is not required to know exactly which host unless the host is associated with its subnet.
- Subnetting is defined in RFC 950

IP Subnets and Network Masks

- Subnets provide extra flexibility to network administrators by subdividing IP networks into smaller subnetworks – reducing congestion
- IP subnets define two or more physical networks that share a common netid field (portion of 32-bit address that is assigned by the NIC)
- Subnetting allows routers to hide complexity of multiple LANs from the rest of the Internet and Enterprise WANs
- As subnetting is an internal operation and hidden from the rest of the world, the internal configuration of the subnets is left up to the network managers - only the internal routers need to be programmed with the appropriate masks.

IP Subnets and Network Masks

- Subnet masks are used to specify the number of bits used to define a subnet
- Subnet masks use the same format and representation techniques as IP addresses (e.g. 255.255.255.0)
- Subnet masks have 1's in the netid and subnet fields, and 0's in the hostid field
- Class B Subnetting example:
  - Before Subnetting: [10:netid:hostid];
  - After Subnetting: [10:netid:subnet:hostid];

Converting binary to/from decimal (8 bits)

Convert 10100010 to decimal

Factor	128	64	32	16	8	4	2	1
Binary	1	0	1	0	0	0	1	0
Decimal	128	0	32	0	0	0	2	0

Add together 128 + 32 + 2 = 162

Convert decimal to binary – if number is bigger than or equal to factor subtract factor and put 1 in binary column. Example 155

Factor	128	64	32	16	8	4	2	1
Decimal	155-128	27	27	27-16	11-8	3	3-2	1
Binary	1	0	0	1	1	0	1	1

155 in decimal is 10011011

Subnet Mask Construction

- Assign a value of 1 to all the bits in the netid field (i.e. first 8/16/24 bits of Class A/B/C networks)
- Assign a value of 1 to each bit in the subnet field
- Assign a value of 0 to each bit in the hostid field
- Convert to dotted decimal or hexadecimal notation

Subnet Mask Construction

- Class B Address Info
  - 129.24.0.0 to 129.24.255.255
  - netid = 129.24.
  - hostid = 16 bits (i.e. 65,536 potential IP addresses)
- Subnet Mask Assumptions:
  - netid bits = 16
  - potential hostid bits = 16
  - If we divide the address space into 32 (2^5) subnets we will have 2046 (2^11-2) hostids or IP addresses per subnet
    - Subnet bits = 5
    - Hostid bits = 11

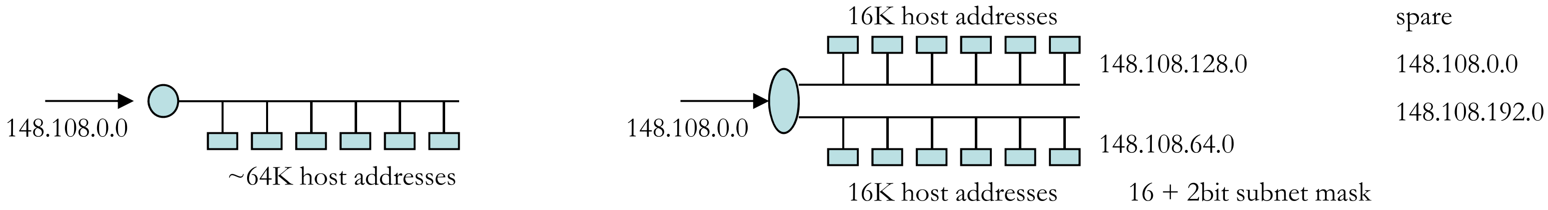
Subnet Mask Construction

Queen Mary University of London

- xxxx xxxx.xxxx xxxx.xxxx xxxx.xxxx xxxx (32-bits)
- 1111 1111.1111 1111.xxxx xxxx.xxxx xxxx (Step 1)
- 1111 1111.1111 1111.1111 1xxx.xxxx xxxx (Step 2)
- 1111 1111.1111 1111.1111 1000.0000 0000 (Step 3)
- 255.255.248.0 (Step 4)

For example, using assigned address: 129.24.0.0
A mask of 255.255.248.0/21 gives 32 subnets (2^5)
A mask of 255.255.224.0/19 gives 8 (2^3) subnets of 8190 hosts (2^13-2)
A mask of 255.255.255.0/24 is the subnet equivalent of Class C addressing

Subnetting Summary



- **Use some of the bits in the *host identifier* field to distinguish between multiple networks called subnets**

- **This leads to more flexibility and the smaller subnets are easier to manage It also reduces the traffic on individual subnets**

- **A *subnet mask* is applied at the Network Layer. It provides local interpretation of the *host identifier* field to determine which bits define the subnet (1s) and which the host (0s)**

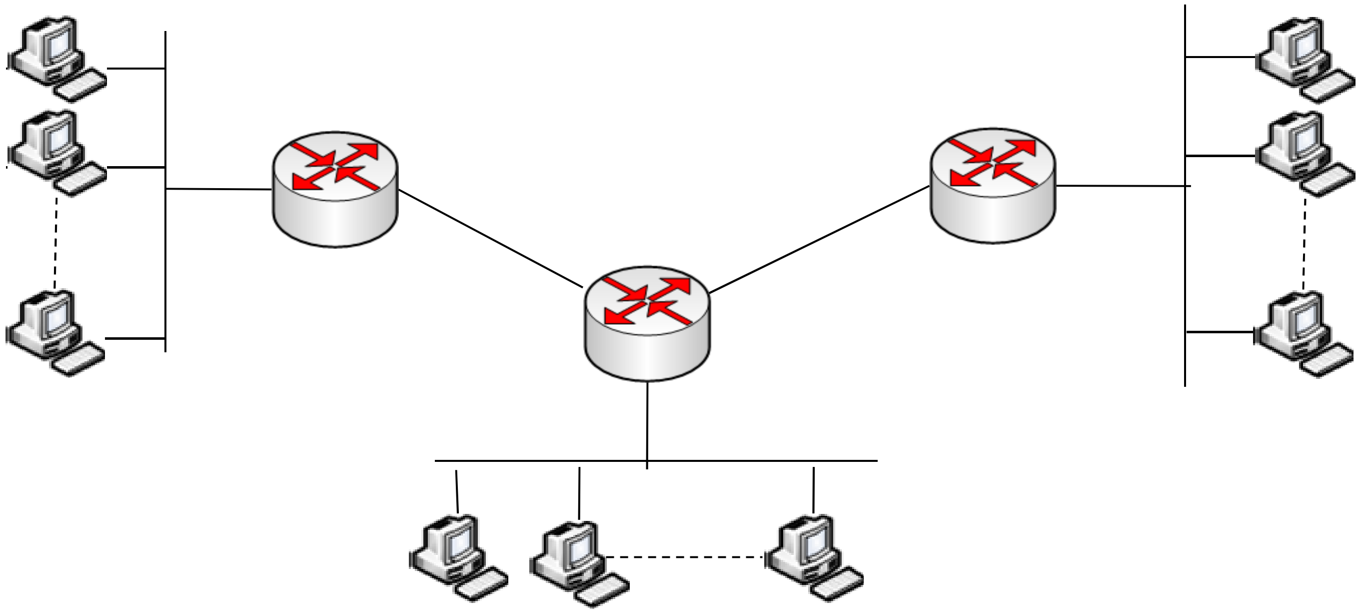
- **Hosts apply the *subnet mask* to the destination address. If it is in the same network (subnet) the Address Resolution Protocol (ARP) is used to find the Medium Access Control (MAC) address. If not, framed packet is forwarded to a router for routeing to another network**

A method of subnetting

three steps:

1. Determine number of networks/hosts and convert to binary
2. Reserve bits in subnet mask and find the increment
3. Use increment to find the network range.

Exercise – subnetting based on networks



An organisation has purchased the Class C address 216.21.5.0 and would like to use it to address this network

A method of subnetting - 3 steps

1. Determine number of networks and convert to binary

5 = 00000<u>101</u> → 3 bits

2. Reserve bits in subnet mask and find the increment

class C default subnet mask :

255.255.255.0 = 11111111.11111111.11111111.00000000

new subnet mask (also called extended-network prefix : → Increment 32

/27 or 255.255.255.224 = 11111111.11111111.11111111.<u>111</u>00000

3. Use increment to find the network range.

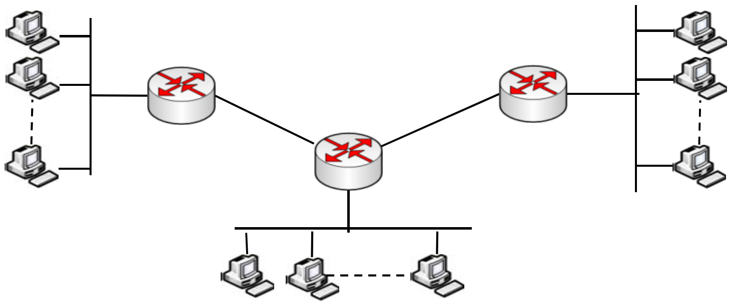
216.21.5.0 – 216.21.5.31

216.21.5.32 – 216.21.5.63

216.21.5.64 – 216.21.5.95

216.21.5.96 – 216.21.5.127

216.21.5.128 – 216.21.5.159



Subnetting Exercise

Assume that you have been assigned the 132.45.0.0/16 network block. You need to establish eight subnets

- How many binary digits are required to define eight subnets.
- Specify the extended-network-prefix that allows the creation of 8 subnets.
- Express the subnets in binary format and dotted decimal notation:

Solution

- 3 binary digits are required to define the eight subnets.
- Specify the extended-network-prefix that allows the creation of 8 subnets /19 or 255.255.224.0
- Express the subnets in binary format and dotted decimal notation:

Subnet #0: 10000100.00101101. 000 00000.00000000 = 132.45.0.0/19

Subnet #1: 10000100.00101101. 001 00000.00000000 = 132.45.32.0/19

Subnet #2: 10000100.00101101. 010 00000.00000000 = 132.45.64.0/19

Subnet #3: 10000100.00101101. 011 00000.00000000 = 132.45.96.0/19

Subnet #4: 10000100.00101101. 100 00000.00000000 = 132.45.128.0/19

Subnet #5: 10000100.00101101. 101 00000.00000000 = 132.45.160.0/19

Subnet #6: 10000100.00101101. 110 00000.00000000 = 132.45.192.0/19

Subnet #7: 10000100.00101101. 111 00000.00000000 = 132.45.224.0/19

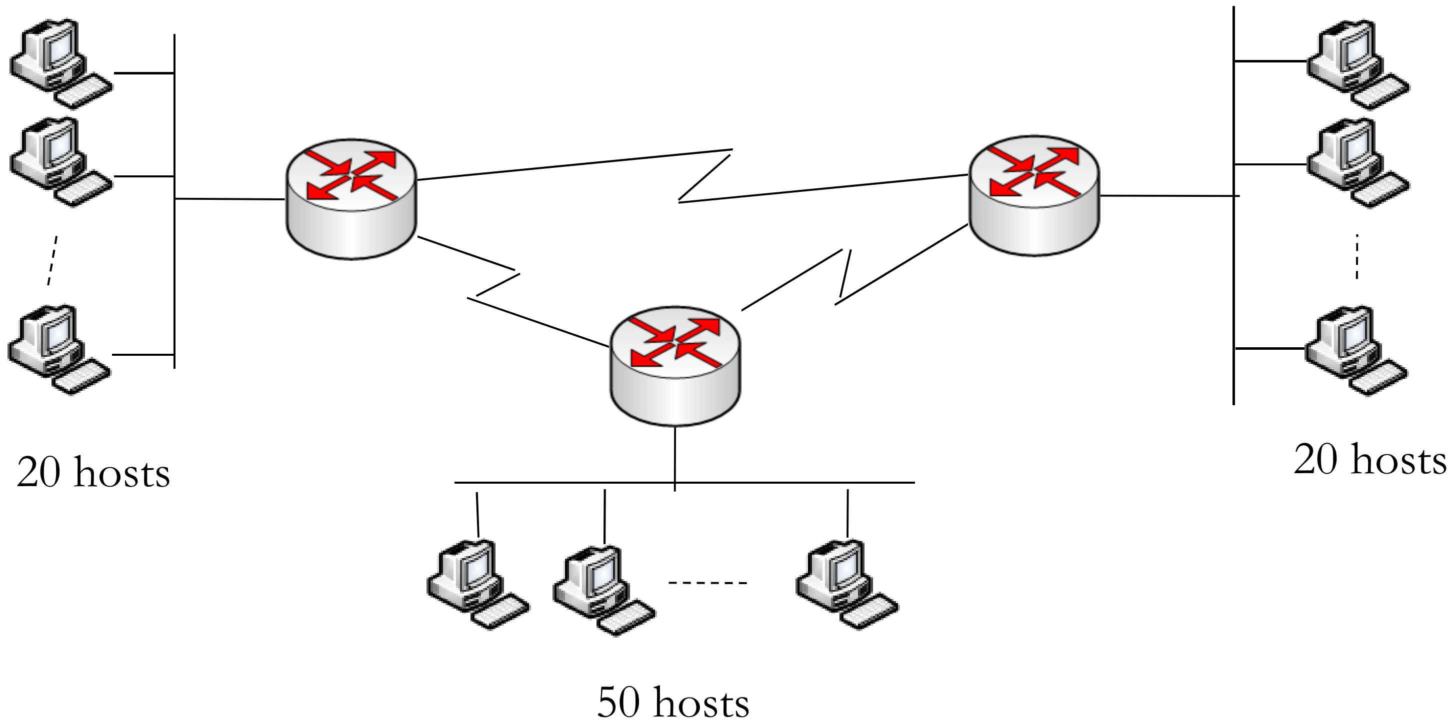
Advanced subnetting - VLSM

Queen Mary
University of London

- In previous subnetting examples, single subnet mask for a entire network is used. It does not support subnets with different sizes.
- Variable Length Subnet Mask (VLSM) was proposed in RFC 1009 as a solution which specified how a subnetted network could use more than one subnet mask.
- VLSM is a means of allocating IP addressing resources to subnets according to their individual need rather than some general network-wide rule.
- VLSM enables a more efficient use of an organisation's IP address space.

VLSM example

Subnet 192.168.1.0/24 to address this network by using the most efficient addressing possible.



Layer 2 / Layer 3 Address Usage

- Hosts on a LAN have a low level "hardware" address (e.g 6 byte Ethernet address) held in the network interface firmware - referred to as network point of attachment (NPA) address, also called MAC address or data link identifier.
- Networks - normally use internet protocol (IP) addresses to forward datagrams between machines
- Internet protocol datagrams have to be carried inside MAC frames

Internet Addressing

IP address limitations

- Address refers to network interface not a physical host

IP address authority

- All IP addresses are assigned by a central authority
- IANA: Internet Assigned Number Authority has ultimate control
- INTERNIC: Internet Network Information Center assigns addresses in USA
- RIPE assigns addresses in Europe

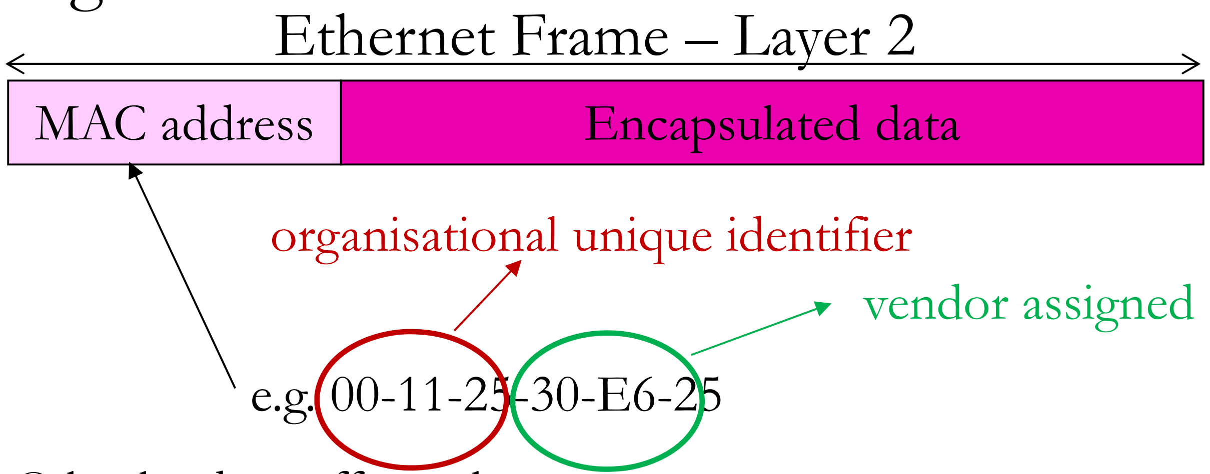
Interworking Logical & Physical Addresses

- IP Addresses define Layer 3 (Network Layer) logical addresses
- Ethernet MAC addresses (etc..) define Layer 2 (Data Link Layer) physical addresses
- Address Resolution (Mapping):

  Translation from logical address (IP address) to an equivalent physical hardware address (Ethernet address) is required for information exchange between host-to-host and host-to-router located on the same physical network

MAC address

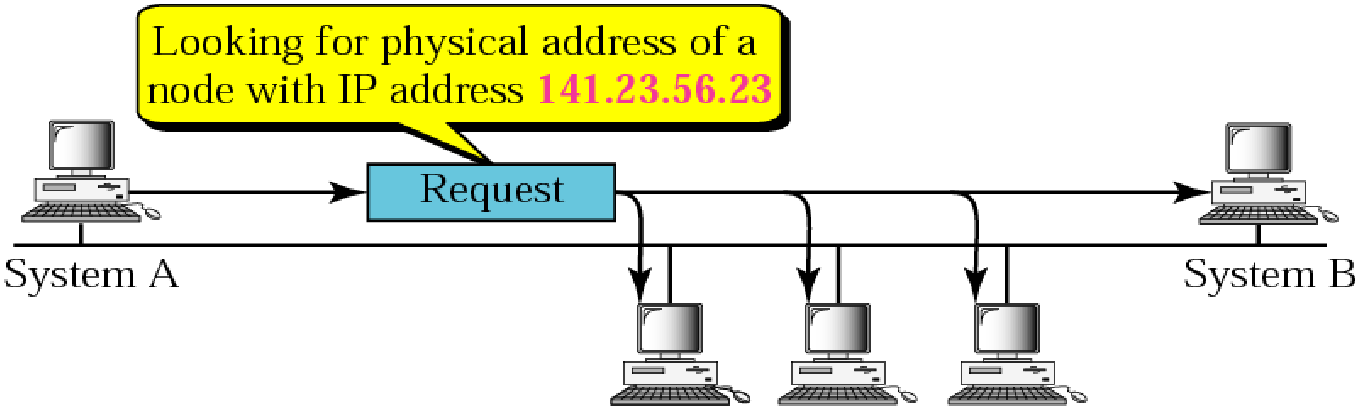
- a unique identifier assigned to network adapters or network interface cards (NICs) by the manufacturer
- 48 bits number, normally expressed as 12 hexadecimal digits



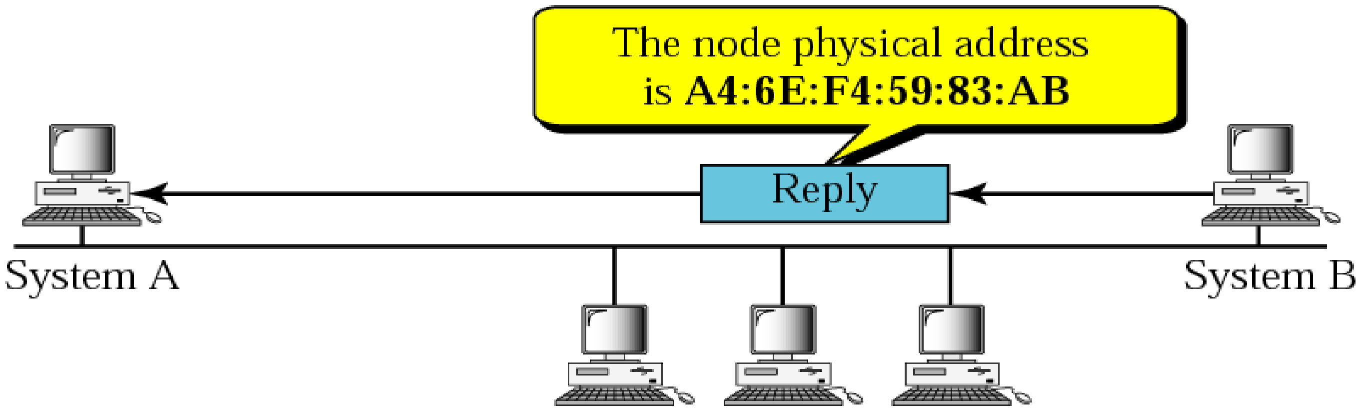
Other header stuff not shown

Each frame has two MAC addresses – source MAC address and destination MAC address

ARP operation



a. ARP request is broadcast



b. ARP reply is unicast

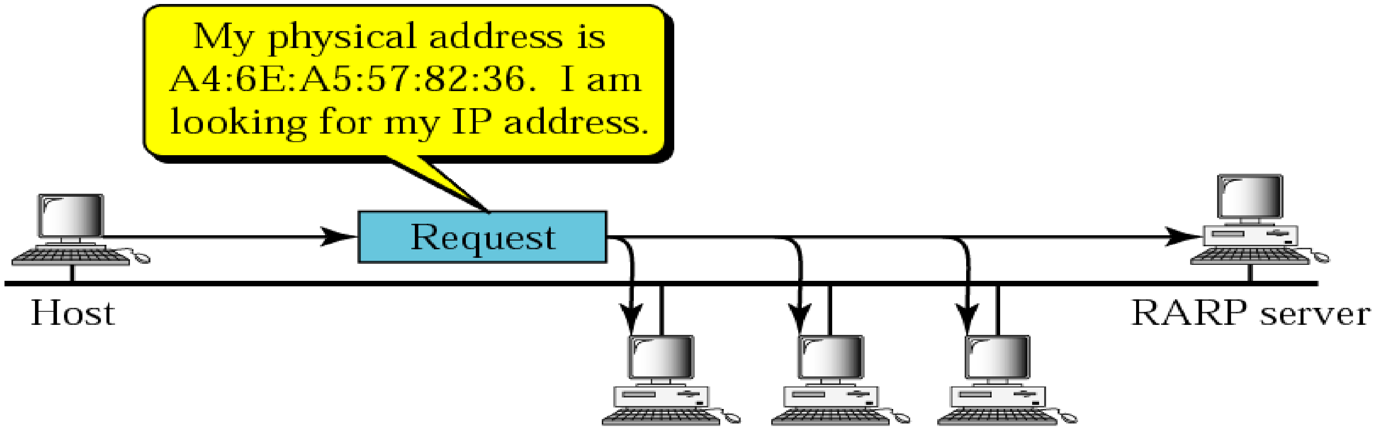
ARP- interaction between IP and MAC

- ARP associates an IP address with its physical address.
- Hosts and gateways store IP/NPA address pairs
- ARP is used to obtain a NPA address given a destination IP address
- The *example* shows the case where the *gateway* receives an IP *datagram* where the NPA address is not known
- ARP request packet: contains the originating gateway IP/NPA pair and the IP address of the target station
- ARP reply message will contain the IP/NPA pair of the target station
- If all is well, the gateway can now encapsulate the IP datagram into a frame with the correct destination MAC address
- An ARP request is broadcast; an ARP reply is unicast

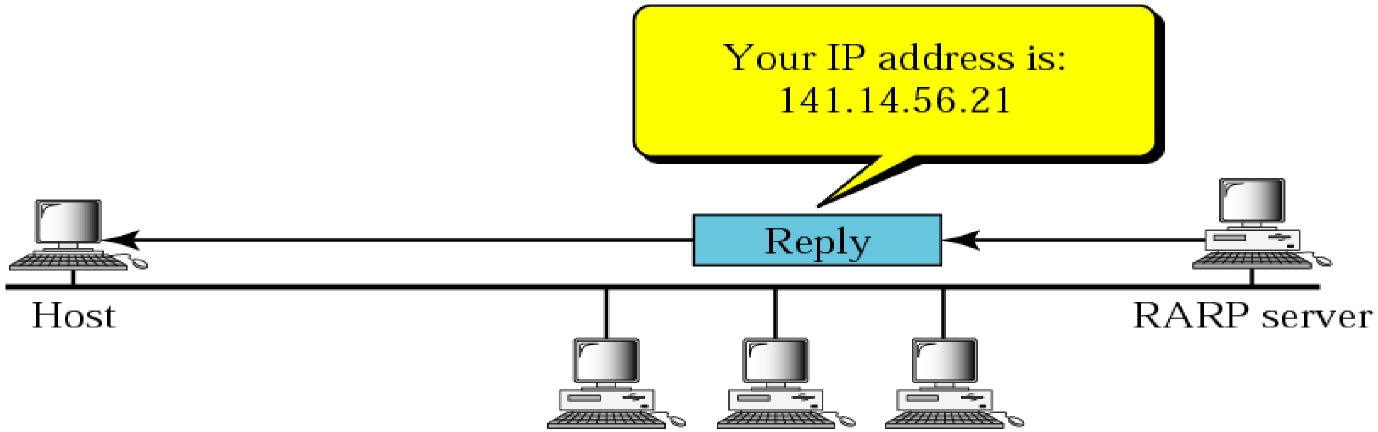
ARP Packet

<table>
<tr><td colspan="2">Hardware Type</td><td colspan="2">Protocol Type</td></tr>
<tr><td>Hardware length</td><td>Protocol length</td><td colspan="2">Operation
Request 1, Reply 2</td></tr>
<tr><td colspan="4">Sender hardware address
(For example, 6 bytes for Ethernet)</td></tr>
<tr><td colspan="4">Sender protocol address
(For example, 4 bytes for IP)</td></tr>
<tr><td colspan="4">Target hardware address
(For example, 6 bytes for Ethernet)
(It is not filled in a request)</td></tr>
<tr><td colspan="4">Target protocol address
(For example, 4 bytes for IP)</td></tr>
</table>

RARP operation



a. RARP request is broadcast



b. RARP reply is unicast

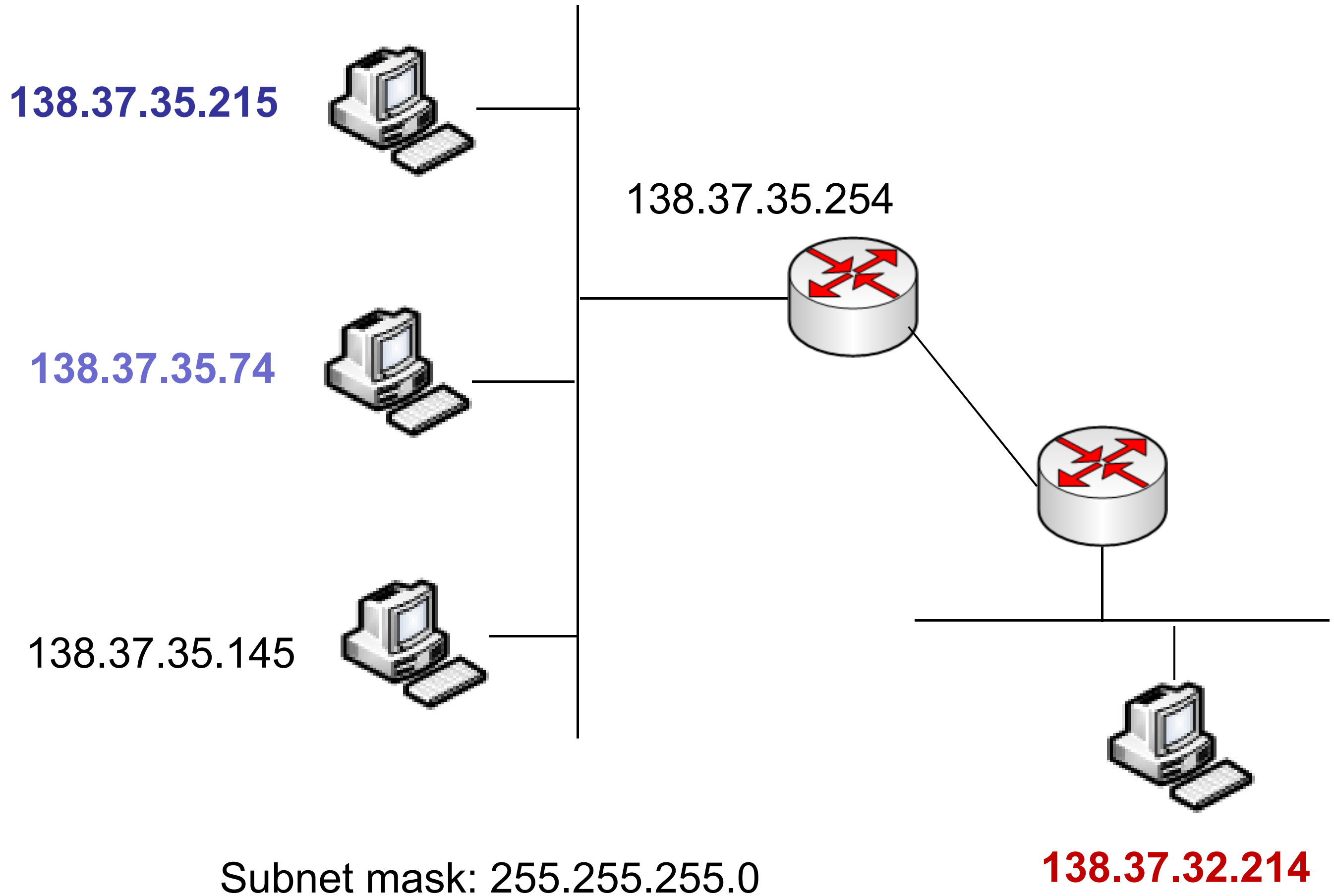
Reverse ARP

- Diskless workstations cannot keep a permanent record of IP/NPA address pairs
- RARP is used by diskless nodes to obtain their IP address from a server
- At boot up, the diskless station broadcasts a message to say does anyone know my IP address - the server receives this *message* and looks up its table for the information and then passes it back to the node. Note other stations on the network can also hear this dialogue and store the information locally - this saves time later
- RARP request: on boot up, station broadcasts a request
- RARP reply: server replies with a message containing station's IP address plus the server's NPA/IP pair
- RARP is now obsolete. It was replaced by Dynamic Host Configuration Protocol (DHCP)
- The RARP request packets are broadcast; the RARP reply packets are unicast.

RARP Packet

Hardware type		Protocol type
Hardware length	Protocol length	Operation Request 3, Reply 4
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP) (It is not filled for request)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled for request)		
Target protocol address (For example, 4 bytes for IP) (It is not filled for request)		

ARP in operation



Subnet mask: 255.255.255.0

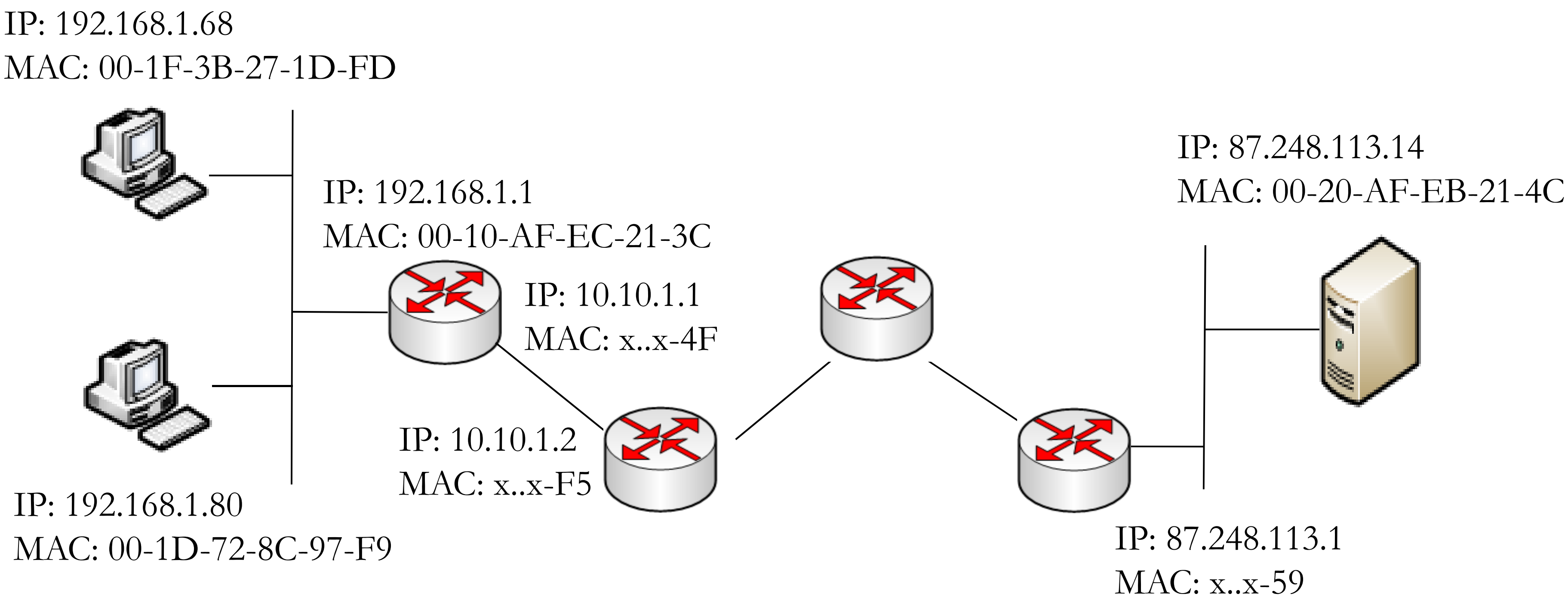
ARP in operation

- Host 138.37.35.215 wants to send a packet to 138.37.35.74
- IP packet needs to be put in an Ethernet frame with MAC address
- Need to find MAC address for 138.37.35.74
- Address Resolution Protocol (ARP) sends broadcast asking for the MAC address
- Usually the destination host will reply with it's own MAC address
- Cached in arp table

ARP – to different subnet

- Host 138.37.35.215 wants to send a packet to 138.37.32.214 (different subnet)
- IP packet needs to be put in an Ethernet frame with MAC address as before
- Different subnet so will need to go through a router
- Routing table (see later) provides address of router
  - (138.37.35.254 here)
- ARP will find the MAC address of the router if not found in the cached arp table.

The two address concept



Subnet mask: 255.255.255.0