Auxiliary protocols

IP is helped by a number of protocols that perform specific tasks that IP does not handle:

- Routing table management (**RIP**, **OSPF**, etc.).
- Congestion and error reporting (**ICMP**).
- Multicasting (**IGMP**).
- Mapping between logical and physical addresses (**ARP**, **RARP** and **DHCP**).

This list is far from complete.

Path determination

An **IP** router routes datagrams based on the contents of its routing table. This table is not managed by IP but by some outside actions, either manual (human) or automated.

There are several ways to describe routing table management. On the one hand, a table can be static (changes done manually very infrequently); if it is not static, it can be modified only at the time of creating a circuit (or path) or it could be updated continuously, possibly even between consecutive transmissions (dynamic)

An IP instance does not know how its table is updated and always routes every datagram independently. Obviously if the table remains fixed, so will be the decision of IP but routing will remain dynamic[a] which allows IP to react to network perturbations.

[a]Static routing is a misnomer as IP must use a routing table even if there is only one entry in it.

Selection criteria

The notion of **cost** of using a path is assumed to have a meaning (it does not). The best path to choose is the cheapest one (in terms of costs); it can be computed if the routing table contains information about all the relevant routers.

To do that, we represent the Internet as a weighted graph (each router is a node and each link is an edge with a cost associated with it).

There are two basic approaches to finding a cheapest path in a graph (true of all optimisation problems):

Global optimisation: the actual minimum–cost path is computed;.

Local optimisation: (“greedy” approach) a not–necessarily–optimal path is computed based on information about the neighbourhood of a router.

Global optimisation

A **shortest path** algorithm should be applied:

- Dijkstra's algorithm for unicast.
- Prim/Kruskal algorithms for minimum–cost spanning tree for broadcast and multicast.

However, shortest path algorithms require that each router knows the current cost of every link in the Internet; a requirement totally unrealistic in the absence of trustworthy information (an Internet is not a static graph).

Given the constraints, global optimisation applied to this problem cannot guarantee optimal solutions.

Local optimisation

If global knowledge is not practical, we have to accept suboptimal solutions. Any algorithm that finds a solution based on partial knowledge (of the Internet graph, in this case) is called **local** optimisation; such algorithms seldom give the true optimum.

The positive side of local optimisation is that it requires only partial knowledge: a router needs to know what its neighbours (however defined) know about the Internet. Such information is much easier to get and is much more reliable than information about distant routers.

Link State Algorithm

The LS algorithm is a global optimisation; it requires each node to **broadcast** the identities and current costs of all the links attached to it (**link state information**). All the routers in the Internet receive these broadcast messages and can use them to compute the single–source shortest paths (Dijkstra) and the minimum–cost spanning tree (Prim/Kruskal) originating from them.

The size of the Internet and congestion in routers[a] make the classic link–state approach useless in practice.

To make is practical, two concepts were introduced:

- The Internet was divided into a (growing) number of subdomains called **autonomous systems**. LS routing, limited to one subdomain, can be practical.
- The impact of congestion is reduced by requiring that all the nodes send LS datagrams at some fixed frequency, at a higher priority than other datagrams.

[a] Information about congested links will arrive with large delays.

Router Hierarchy

As the Internet grew in size, it became necessary to subdivide the routers into smaller entities. These are called **Autonomous Systems**. Routers within an A.S. consider only routing within that system, with the exception of a specially designated set of routers, called **Gateway Routers** which know at least one gateway router belonging to another A.S.

Local routers know that traffic to external sites should be sent to the proper gateway, while gateways know which peer gateway it should be forwarded to.

Consider network 48. One node on this network is a router (let it be 48.1). This node belongs to A.S. 104. The Autonomous System 104 is made of routers only and has a gateway, say 104.12.1, to the A.S. 103. It also has another gateway, 104.15.1, to A.S. 121; this gateway "knows" gateway 121.77.1.

When 48.76 (in 104) sends a packet to 121.45.23, the packet first reaches the router 104.48.1, which routes it to 104.15.1 (possibly through intermediate nodes within 104).

104.15.1 hands the packet to 121.77.1, which routes it to 121.45.x, etc.

OSPF

Designed for passing routing information inside one A.S., OSPF is a *link–state* protocol using broadcast messages (unlike RIP which calls for messages to neighbours).

Advertisements describe the states of links directly attached to the sending router. The costs are set somewhat arbitrarily by the network administrators of the routers.

Some features:

- Authentication: the advertisements sent without a proper authentication are not accepted.
- The lowest–cost path is chosen independently for every datagram (important when there are several same–cost paths).
- Separate link costs for each type of service can be specified.
- A routing hierarchy within a single A.S.

Hierarchy in OSPF

The A.S. is divided into *areas*; each area operates as a separate OSPF entity (advertising is limited to one area).

Each area has its designated routers that exchange information with routers from other areas; these designated routers are called *area border routers*.

The whole A.S. has a number of routers designated for routing inter–A.S. traffic; these routers are called *boundary routers*.

A special area called **backbone** is responsible for routing inter–area traffic. The backbone contains all the area border routers, all the boundary routers. Additionally, the backbone contained some routers not belonging to any area; these are called *backbone routers*.

Hierarchy

The OSPF hierarchy from the point of view of packet travel:

1. Within an area: both the source and the destination are in area $\mathcal{A}$. This traffic is handled by local router and perhaps by area border routers (not acting in this capacity).

2. Inter–area: source in area $\mathcal{A}$, destination in area $\mathcal{B}$. This traffic must pass through at least two area border routers and perhaps a number of backbone routers.

3. Inter–A.S.: source in this A.S.; destination in another. This traffic must pass through a boundary router.

Distance vector algorithm

This is a “local” and suboptimal algorithm.

Each node receives cost information from its direct neighbours, performs its calculations and send the results back to the neighbours. The process continues as long as new information becomes available; every time a node discovers the need to modify its own results, it will keep the process going. The calculation never stops.

The Bellman–Ford algorithm is used in the calculations. The result converges to optimal if the Internet does not change its attributes.

Otherwise it may lead to livelock (datagrams cycling forever inside the network core). Livelock can be prevented by the *poisoned reverse* trick in which if A routes to C through B, it tells B that the cost of sending from A to C is infinite.

Routing Information Protocol

An early protocol for building routing tables confined to one A.S., RIP makes routers exchange messages, called *small RIP advertisements*, every 30 seconds. These advertisements (copies of routing tables) are used to update the routing table of the receiver.

The cost measure is the number of hops with no provision for congestion. Many non–standard implementations exist; some add a congestion flag to the link description.

RIP Advertisement

Each router sends an advertisement to its neighbours. The advertisement message is essentially a copy of the routing table, describing all the paths known to the sender.

Advertisements are sent every 30 seconds. If a router does not get an advertisement from one of its neighbours for a period of 180 seconds, it assumes the neighbour died and starts to advertise this fact.

A unix routing table

UNIX has a daemon called *routed* which uses RIP. Part of the routing table on snowhite:

```
Internet:
Destination      Gateway            Flags     Refs      Use    Mtu
Interface
default          131.104.48.18      UGS         39  1256145    de0
127.0.0.1        127.0.0.1          UH           2     3574    lo0
131.104.48/23    link#1             UC          49        0    de0
131.104.48.1     127.0.0.1          UGHS         1    11441    lo0
131.104.48.18    0:90:2b:7f:14:0    UHL         40        0    de0
131.104.48.24    0:80:c8:e2:75:a5   UHL          0     1629    de0
131.104.48.46    00:1f:f3:cc:5a:82  UHLW         0 16211676    de0
131.104.49.60    0:0:e8:dd:a0:5b    UHL          6     9638    de0
```

The default route is galileo (.18) and it seems to be popular.

Border Gateway Protocol

OSPF and **RIP** function within one domain (autonomous system). They do not attempt to manage traffic between domains.

BGP is an interdomain protocol for exchanging routing information between gateway routers, either inside an AS (**IBGP**) or outside (**EBGP**).

If you want to know more about BGP try the tutorial.

ICMP

The Internet Control Message Protocol (RFC792) is used in IP error messages and in control messages reporting congestion of traffic obstacles.

ICMP sends a message in a single IP datagram with the ICMP message as its payload and setting the protocol field of the IP header to 1.

Ethernet header	IP header	ICMP header	ICMP data	padding

Most systems add padding to the data field so that an ICMP datagram has a fixed size (for a given OS: 64B in Linux, 40B in Windows).

ICMP header

The header is 4 bytes long.

Type	Code	ICMP Checksum

The type and data fields indicate the condition reported by ICMP.

The checksum is computed over the first two bytes of the header and the data part.

Some conditions reported by ICMP

There are ca. 40 types of conditions reported; most of them do not require and additional description (in the code field).

Type		Code	
0	Echo reply	0	
9	Router advertisement	0	
11	Time exceeded	0	in transit
		1	during reassembly
12	Bad IP header	0	Pointer to error
		2	Bad length

3	Destination unreachable	0	Network unreachable
		1	Host unreachable
		2	Protocol unreachable
		3	Port unreachable
		4	Fragmentation required and DF flag set
		6	Network unknown
		7	Host unknown
		9	Network prohibited
		10	Host prohibited
		13	Access prohibited

Multicasting

IGMP (RFC1112 and many subsequent RFCs) adds **multicasting**, i.e. sending packets to a specific group of hosts that declared their membership in an Internet multicast group. Originally used for discussion groups (“usenet”) the multicast groups evolved to facilitate distribution of online video and online gaming.

IGMP does not manage the movement of multicast datagrams but manages the membership of groups (through “create”, “join” and “leave” requests) and the presence of multicast routers in subnets.

More details on IGMP can be found in a short tutorial or in a long tutorial (CISCO). The latter describes IGMP snooping in detail.

Address resolution techniques

There are three basic methods of finding the physical address corresponding to an IP address:

- Table lookup: the router has a table of all nodes known to it.
- Computational mapping: addresses are chosen in such way that there is a mathematical function that, when applied to an IP address, gives the physical address.
- Message exchange: the router sends a broadcast (or multicast) message asking for the physical equivalent of the IP address. A node that knows the answer sends a reply.

Address Resolution Protocol

The **ARP** standard (RFC826) defines the format of address resolution messages and the way these messages are handled.

ARP is used in various situations. Denoting by $\mathcal{H}$ a host and by $\mathcal{R}$ a router, the following address queries may be needed:

1. $\mathcal{H} \xrightarrow{?} \mathcal{H}$.
2. $\mathcal{H} \xrightarrow{?} \mathcal{R} \leadsto \mathcal{H}$.
3. $\mathcal{R} \xrightarrow{?} \mathcal{R} \leadsto \mathcal{H}$.
4. $\mathcal{R} \xrightarrow{?} \mathcal{H}$.

Here, the $^?$ superscript indicates the hardware address needed when an IP is known.

Case 3 will normally be resolved when the routing table is inspected (in the “Interface” entry). Case 4 occurs when the local interface is a broadcast link.

The ARP message format is general to allow the translation of arbitrary NL (“protocol”) addresses to any type of DLL (“hardware) addresses,

In order to accommodate any pair of protocol address types, the ARP header has two fields (HLEN and PLEN) which give the sizes of hardware (DLL) and protocol (NL) addresses. In reality ARP is almost always used to translate a 32–bit IP address to a 48–bit **MAC** (*Ethernet*) address.

ARP for IP$\longrightarrow$MAC matching

Hardware addr. type		Protocol addr. type	
Hardware len	Protocol len	Operation	
Sender Hardware address			
Sender Hardware addr.		Sender prot. addr.	
Sender prot. addr.		Target Hardware addr.	
Target Hardware address			
Target Protocol address			

Hardware address type = 1 for *Ethernet* and Protocol address type = 2048 for IP.

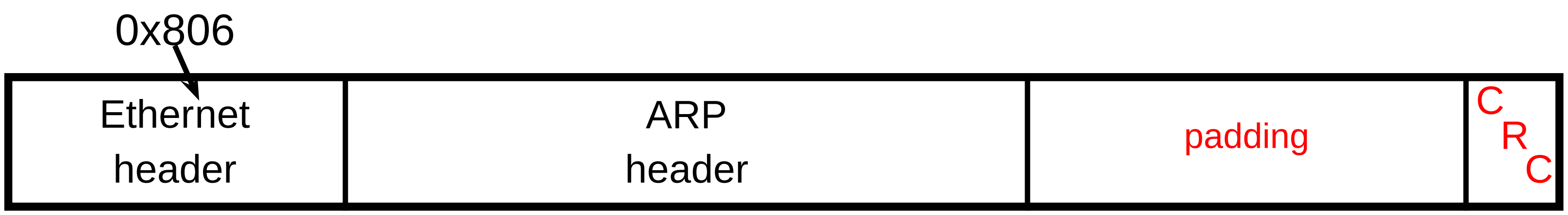
Operation: 1=request, 2=response.

ARP request for a MAC address

1		2048
6	4	1
Sender’s MAC address		
Sender’s MAC addr.		Sender IP addr.
Sender IP addr.		0xFFFF.
0xFFFFFFFF		
Target IP address		

A Request frame is broadcast locally (inside the LAN). An entity with a matching IP address responds with a unicast Reply frame which looks the same but frame has 2 as “operation” and the addresses reversed (the MAC address requested showing in the “Sender’s MAC address” field).

An ARP message is put directly into a DLL frame.



Every frame format has a *frame type* field which is set to ARP (*0x806* in *Ethernet* frames). Padding of 18B is needed to bring the frame size to 64B

ARP responses are *cached* in a first–in–first–out local cache.