IPv6 & User Datagram Protocol (UDP)

1) The communication will consist of two steps: a) Name resolution of www.google.com, and b) Transmission of the UDP packet to the destination.

   a) As a first step, *fastnet* has to resolve the name “www.google.com”. In order to resolve the name, it would contact its name server – in this case 134.226.32.58. It would usually receive the IP address of a name server with its configuration or when it is assigned an IP address by a DHCP server. In order to communicate with 134.226.32.58, it would consult its routing table and determine that it needs to route the IP packet to its default gateway because it can only contact nodes the subnetwork 134.226.62.0 directly. To forward the IP packet to its default gateway 134.226.62.254, it needs to create an 802.11 frame with the hardware address of 134.226.62.254; so, it will broadcast an ARP request for 134.226.62.254. Because it is associated with an access point, it will compete with other nodes for access to the medium and an issue a frame addressed to the access point, with a second hardware address with all bits set, indicating a broadcast. The access point acts as a bridge and will broadcast the frame in its connected Ethernet network. The router 134.226.62.254 will receive the ARP request and will respond with an ARP reply directed towards the IP address and hardware address of *fastnet*. Once *fastnet* has received the hardware address of the gateway, it will issue a frame addressed to the gateway, carrying the DNS request for www.google.com addressed to 134.226.32.58. The gateway will receive the frame, remove the IP packet and determine that the destination of the IP packet is in another network segment to it i.e. 134.226.32.0; so, it will broadcast an ARP request on this segment for 134.226.32.58 and receive the hardware address of the interface in this computer in the ARP reply. It will then forward the DNS query in a frame addressed to the hardware address of 134.226.32.58.

      The DNS server will consult its list of root servers and contact one of the root servers to determine who is responsible for resolving names for .com addresses. Then it will contact one of the name servers for .com addresses to determine the IP address of a name server for .google, and then contact a name server for .google for an IP address for the name www. All DNS queries from the DNS server would be routed from the DNS server 134.226.32.58 over its gateway 134.226.32.254 to router 134.226.1.2 for College, to a router from HEAnet, e.g. 193.1.219.57, and from there towards its destination. Whenever the DNS query would be forward from one router to the next, it would be carried in a Link Layer frame e.g. an Ethernet frame, addressed to the next hop.
      Once the DNS server receives the DNS reply from .google, it will deliver the IP address in a DNS reply to *fastnet* returning over the gateway.

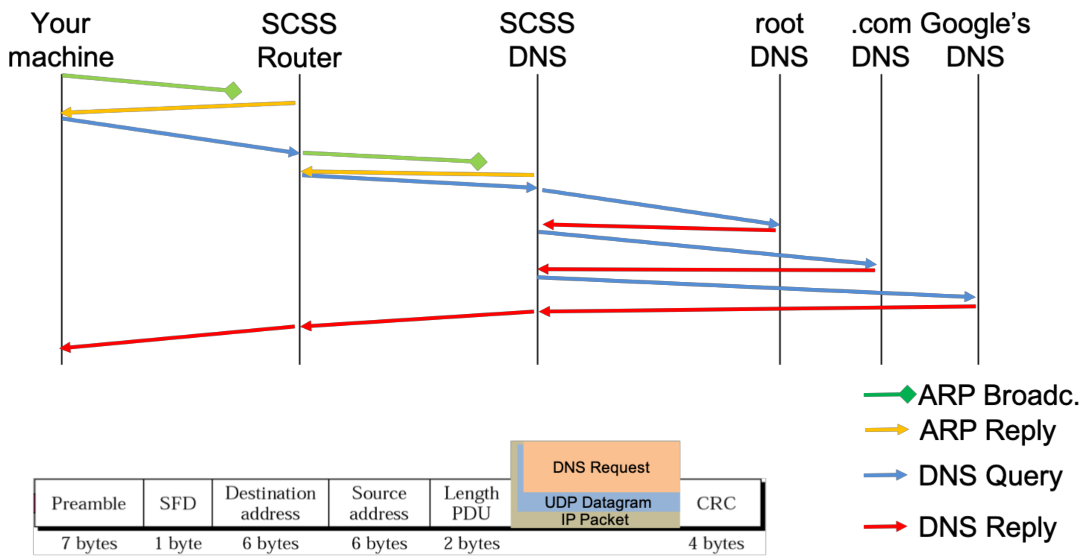


Figure 1: The figure demonstrates the forwarding of the DNS queries and the initial ARP requests and replies associated with the forwarding of the IP packets within Link Layer frames e.g. Ethernet frames.

b) Once *fastnet* receives the IP address of www.google.com e.g. 74.125.24.104, it can issue an IP packet to this address e.g. carrying an ICMP ping. This packet again would be carried in a Link Layer frame addressed to the default gateway, which would forward it to the College router 134.226.1.2. In the case of the setup of Colllege's network, this router may apply an Access Control List (ACL) and only forward the packet if it has been issued by a computer with an IP address that has been registered. If so, this router would then forward the packet to a router at HEAnet, which may determine from its routing table that the most suitable next hop for this packet is a router at the Irish Internet Exchange (INEX). A router at INEX may determine that Google peers at INEX and that it has a connection to a router at Google and would forward the IP packet to this router as next hop.

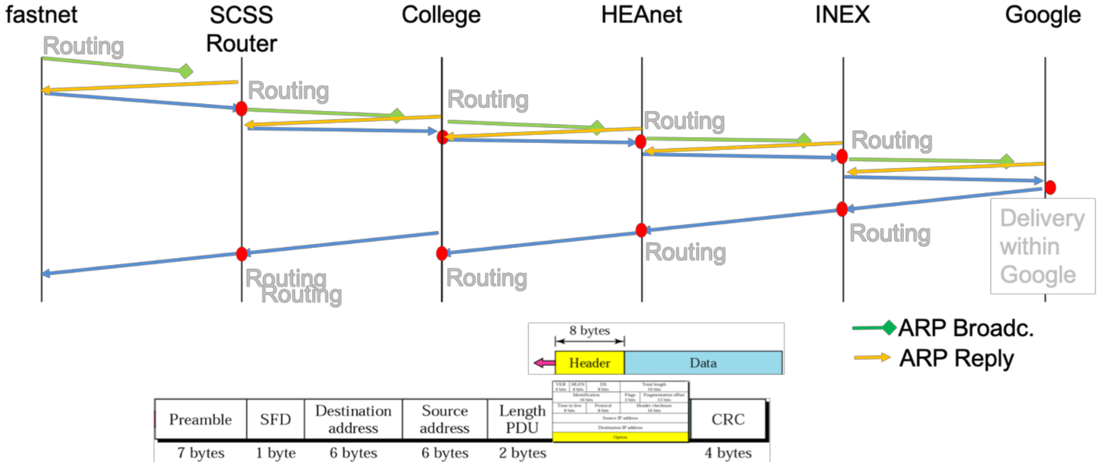
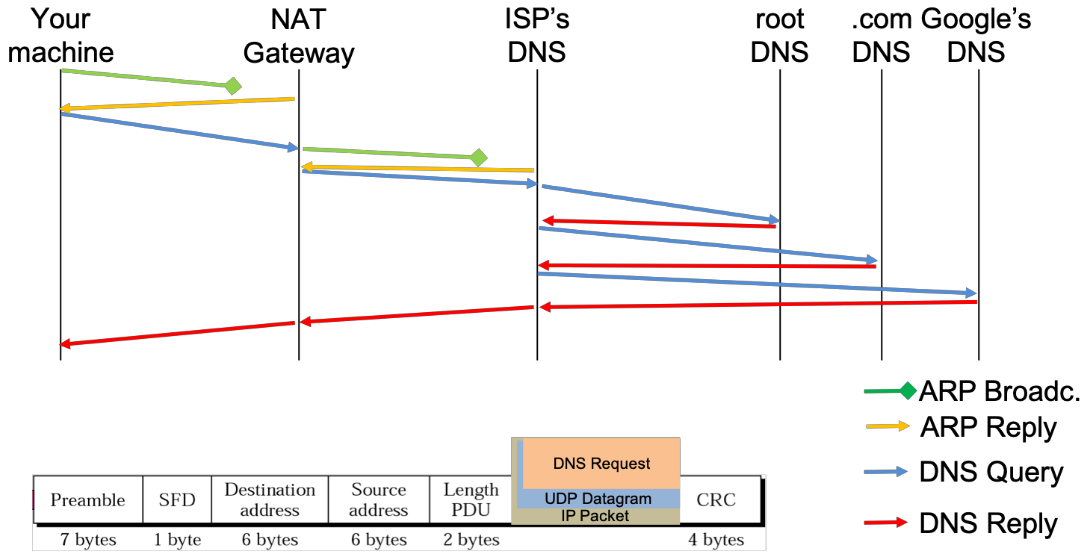
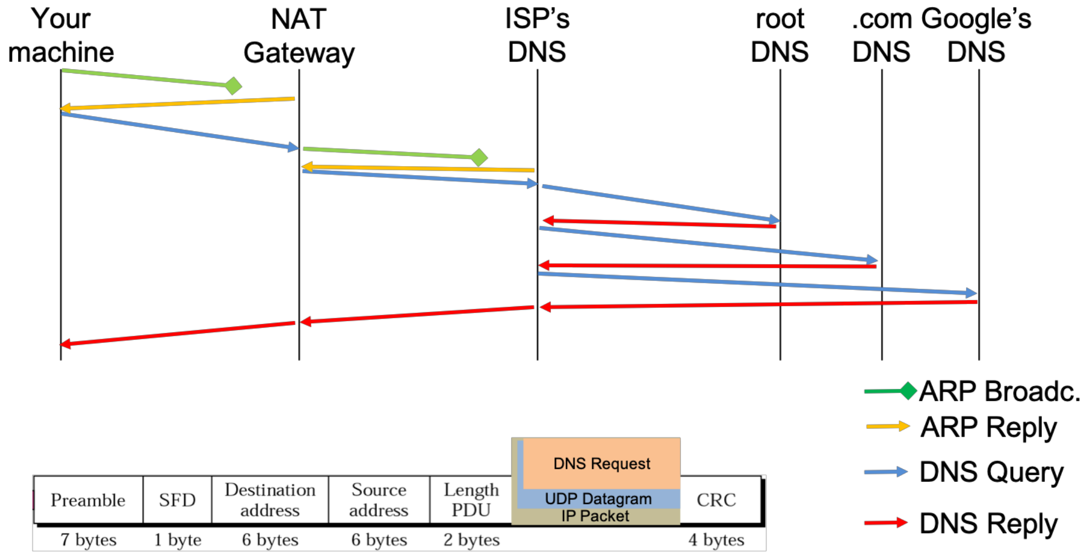


Figure 2: The figure demonstrates the forwarding of the IP packet issued by fastnet with an IP address at Google, over a number of hops. At each hop, the forwarder would require the hardware address of the next hop in order to create the Link Layer frame that carries the IP packet forward.

The question addresses a number of important points for CS2031: 1) It combines all layers covered in the module and is one of the most complete examples of the connection between the material covered in the module, 2) it is an example for communications that underlie transmissions such as requests for web pages etc, 3) it is a good demonstration for the encapsulation of UDP packets inside IP packet which are then carried forward by Link Layer frames such as Ethernet frames and that these frames are addressed to the next hop which may require additional communication to enable this.

2) <more information to come>





3) Protocol Encapsulation

Draw a diagram of the individual headers i.e. UDP, IP, Ethernet header, of an Ethernet packet that includes an UDP packet addressed to an application on host 156.202.34.43 port 21 from the local application on address 134.226.34.85 port 10567. Assume values for fields of the individual headers if these values are not given above. For each value give a short explanation why you chose this particularly value.

0	8	16	24 31
Preamble			
Preamble (contd)			SFD
Destination Address			
Dest. Address (contd)		Source Address	
Source Address (contd)			
Length or type			
vers.	IHL	Type	Total Length
Identification	Flag	Fragm. Offset	
Time-to-live	Protocol	Header Checksum	
Source Address			
Destination Address			
Source Port		Destination Port	
Length		Checksum	
Payload			
CRC			

0	8	16	24 31
10101010101010101010101010101010			
1010101010101010			10101011
00:11:93:85:E0:C3			
Dest. Address (contd)		00:11:93:85:BC:05	
Source Address (contd)			
0x0800			
4	20	0	36
0x1234	0	0	
254	17	1110101110101001	
134.226.34.85			
156.202.34.43			
10567		21	
8		0	
2	B	A	5
Y	A	W	N
10101011111011101010101011111010			

🟥 Ethernet header

🟩 IP header

🟦 UDP header

🟦 Payload

Notes to the values:

- Preamble + Start Frame Delimiter (SFD) are predefined in the Ethernet standard
- Ethernet Source and Destination address are random 48bit addresses
- Type field contains the value for an IP packet in the Ethernet payload
- The CRC is a random value in this example – generally, it would be calculated over the whole Ethernet frame

- The version field in IPv4 is set to 4

- The header length is 20 bytes
- The type of the packet is not set
- The total length of the IP packet is the sum of the IP header length (20 bytes), the UDP header (8 bytes) and the payload (8 bytes)
- The identification of the packet is a random value in this example
- None of the flags are set
- The fragmentation offset is 0 because this packet is not a fragment of some larger packet
- The time-to-live is 254
- The protocol field contains the value for a UDP packet as payload
- The header checksum is a random value in this example – generally, it would calculated over the IP header
- The source and destination address are taken from the question above

- The source and destination port are taken from the question above
- The length of the payload is 8 byte in this example
- The checksum is not used in this example