IPv4/IPv6 & UDP, DNS

1) An Internet Service Provider (ISP) has bought the right to use the IP addresses in the range from 213.49.0.0 to 214.57.255.255. It uses Classless Inter-Domain Routing (CIDR) to route traffic to these addresses. It receives a number of requests from companies. First company A buys a block of 14,000 addresses, then company B requests 6,000, followed by company C with 850 addresses and company D requests 350 addresses. The ISP processes these requests in the order it receives them. What is the address range allocated for each client? Give the first and last address of the range, the number of significant bits and the subnet mask.

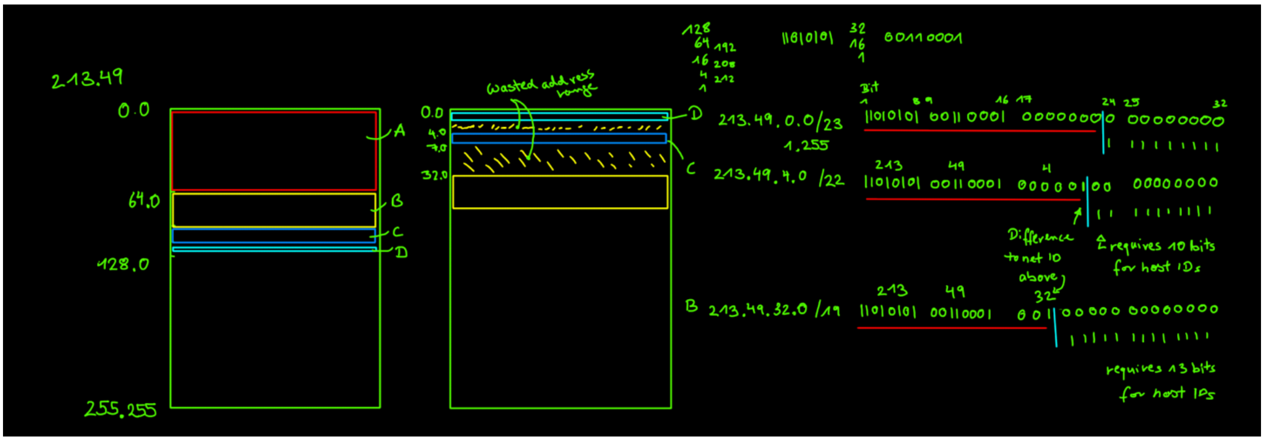
   If CIDR wasn't used, what classes of network addresses would be allocated to each client? How many addresses would be allocated in total? What would be the fraction of addresses actually used by each client? Compare this to the use of CIDR.

First company A buys a block of 14,000 addresses,
then company B requests 6,000,
followed by company C with 850 addresses
and company D requests 350 addresses.

Client A: 213.49.0.0/18	(213.49.0.0-213.49.63.255)
Client B: 213.49.64.0/19	(213.49.64.0-213.49.95.255)
Client C: 213.49.96.0/22	(213.49.96.0-213.49.99.255)
Client D: 213.49.100.0/23	(213.49.100.0-213.49.101.255)

Client A: 14000 – next 2^x: 16384 = 2^{14} 213.49.0.0 213.49.63.255 213.49.0.0/18	Client C: 850 – next 2^x: 1024 = 2^{10} 213.49.96.0 213.49.99.255 213.49.96.0/22
Client B: 6000 – next 2^x: 8192 = 2^{13} 213.49.64.0 213.49.95.255 213.49.64.0/19	Client D: 350 – next 2^x: 512 = 2^9 213.49.100.0 213.49.101.255 213.49.100.0/23

Reverse order of requests results in gaps in allocation:



Without CIDR, each client would require a class B address range, because every request is larger than a class C network with 254 addresses.
without CIDR:
Client A: 14000/65536 = 0.213 ~ 21% usage
Client B: 6000/65536 = 0.077 ~ 9% usage
Client C: 850/65536 = 0.013 ~ 1% usage
Client D: 350/65536 = 0.005 ~ 0.5% usage

with CIDR:
Client A: 14000/16384 = 0.854 ~ 85% usage
Client B: 6000/8192 = 0.61~ 73% usage
Client C: 850/1024 = 0.83 ~ 83% usage
Client D: 350/512 = 0.68 ~ 68% usage

2) Assume you have a dial-up connection and want to send a UDP datagram of 5000 bytes to a server on the Internet. The connection between the two nodes that includes a PPP link with an MTU of 512 bytes, two Ethernet links with an MTU of 1500 bytes and an FDDI ring with an MTU of 4096 bytes. Draw a diagram of the connections, describe the fragmentation of the datagram as it is transferred to its destination and show the effect of the loss of a fragment. Contrast the behaviour of the 512-MTU-bytes dial-up link with 1500-MTU-bytes ADSL connection.

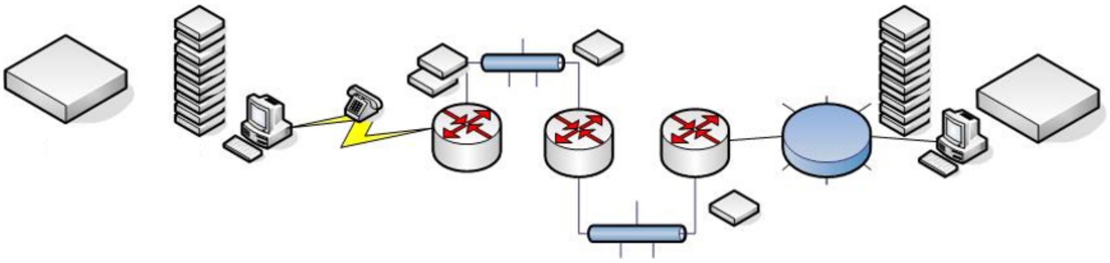


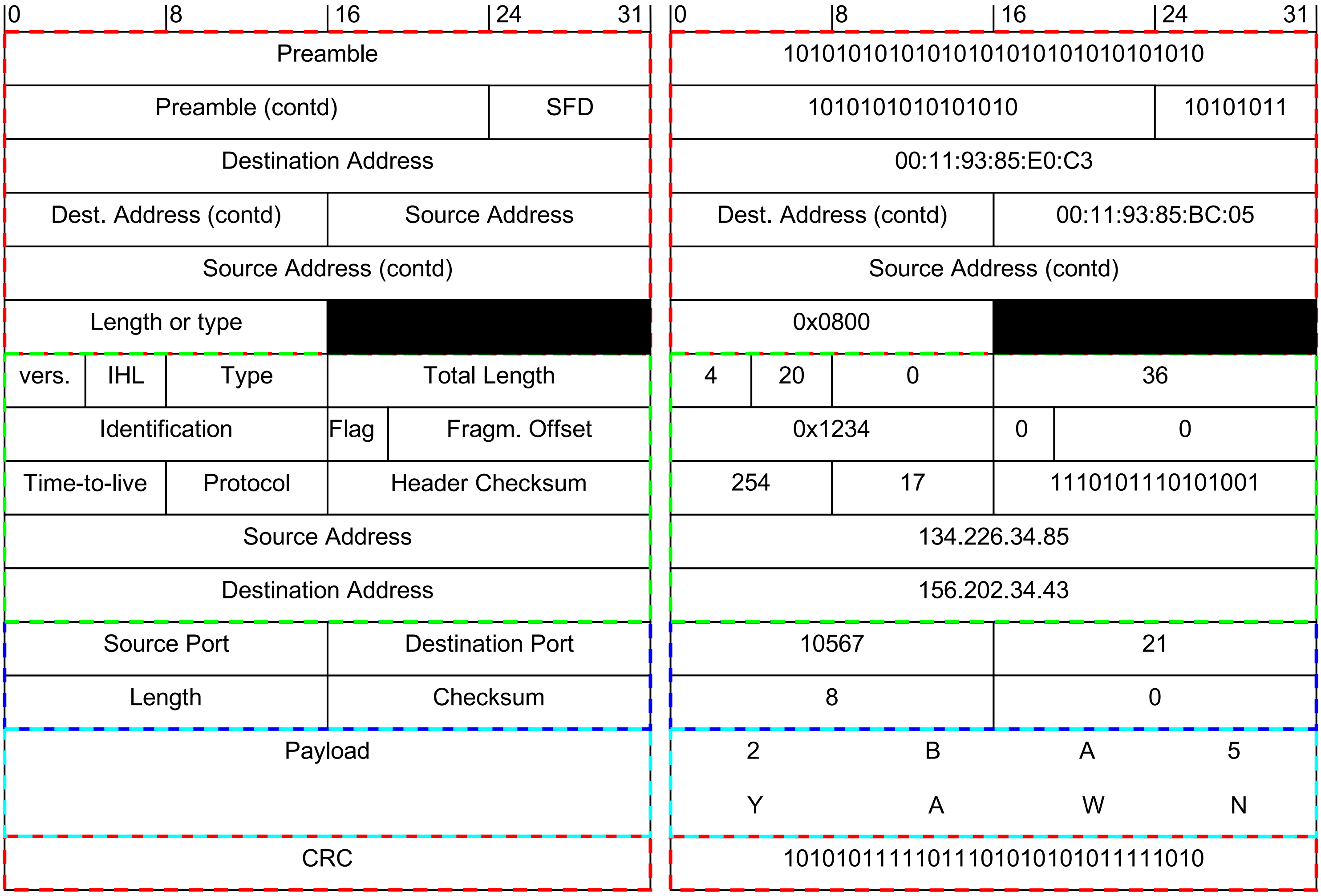
Figure 1: Fragmentation of a packet of 5000 bytes into 10 492-byte fragments and one 80-byte fragment

The original IPv4 packet with 5000 bytes payload and 20 bytes IPv4 header would be split into 10 packets each with its own header: 512 bytes in total or 20 bytes IPv4 header + 492 bytes payload; and one packet with the remaining 80 bytes payload plus 20 bytes header. These packets will be forwarded at each hop; because the MTUs of the subsequent networks are all larger than the first connection, the packets will not be split up any further. The destination node will attempt to assemble the fragments, once all fragments have been received. If one of the fragments is dropped by a router along the path, the destination will discard the received fragments once a timer has expired.
In this example, the small MTU of the first connection leads to a large number of small fragments and these fragments are then individually forwarded towards the destination, requiring individual routing decisions etc at every hop. If the MTU of the initial connection would be larger, the number of fragments would be smaller; so, fewer routing decisions etc would be required along the path. In the past, this scenario used to be common with dial-up connections where the MTUs were small; at the

moment, some protocols for IoT devices exhibit similarly small MTUs and may lead to a high number of fragments at a traffic source.

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 Y	B A	A W	5 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

4) Assume that a node A intents to communicate with a node E over a number of intermediate nodes, B to D, as shown in figure 2. The IPv4 addresses and hardware addresses of the interfaces of the individual nodes are shown in figure 3. Node B acts as a NAT gateway.

   a) Describe the information that node B will keep in order to act as NAT gateway and how this information is used by B to process incoming and outgoing IPv4 packets.
   b) Describe the IPv4 packet that A would issue and the routing process of the IPv4 packet from A to E, at the intermediate hops.
   c) Describe the Link layer frames encapsulating the IPv4 packet assuming that all links use Ethernet and the resolution of the IPv4 addresses to Ethernet addresses at every hop

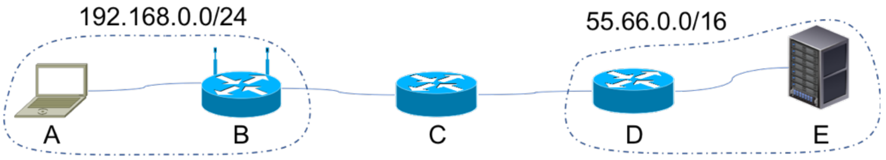


Figure 2: Topology with 5 nodes, A to E, that could represent a connection of a home network to a server over a 1-hop interconnecting network.

.

A

192.168.0.160
0F:0E:0D:AA:CC:BB

Destination	Port Out	Next Hop Address
192.168.0.0	1	-
0.0.0.0	1	192.168.0.1

B

Inside: 192.168.0.1
0D:0E:AA:90:00:AB
Outside: 75.50.25.1
0D:0E:AA:90:00:0D

Destination	Port Out	Next Hop Address
192.168.0.0	1	-
0.0.0.0	2	1.2.3.4

C

1.2.3.4
0D:0E:AA:90:00:0B

Destination	Port Out	Next Hop Address
75.50.25.1	1	75.50.25.1
55.66.0.0	2	55.66.0.1

D

55.66.0.1
0D:0E:AA:90:00:0B

Destination	Port Out	Next Hop Address
55.66.0..0	1	-
0.0.0.0	2	1.2.3.4

E

55.66.0.100
0D:0E:AA:90:00:0B

Destination	Port Out	Next Hop Address
55.66.0.0	1	-
0.0.0.0	1	55.66.0.1

Figure 3:IP addresses, hardware addresses and routing information of the nodes shown in figure 2.

a) Node B needs to keep a table that associates outgoing traffic with the source of traffic e.g. it needs to record the source IP address and port number, the destination IP address and port number and the port number the gateway uses to forward the packets. This information is recorded for outgoing traffic and used for incoming traffic to direct it towards the original source – generally, for connection-oriented protocols that expect replies to outgoing traffic. When a packet arrives from the outside e.g. from node C, the gateway will look for a match in its table. If a match is found i.e. if a packet arrives at a port that was used to forward traffic and that matches the destination IP address and port number that traffic was forwarded to, the gateway will place the IP address and port number of the original source in the packet as destination IP address and port number and forward the packet in its local network. If a match is not found in the table, the packet is dropped at the gateway.

b) Node A would issue a packet with its IP address and port number as source e.g. 192.168.0.160:50000, and node E's IP address and port number as destination e.g. 55.66.0.100:7. Node B would record the outgoing packet in its table, replace the source IP address and port number with its own IP address and the port number used to forward the traffic e.g. 75.50.25.1:55123. It would then consult the routing table and forward the packet to 1.2.3.4. The router 1.2.3.4 would consult its routing table and match the destination address 55.66.0.100 with the second entry and forward the packet to 55.66.0.1. The router 55.66.0.1 will consult its routing table and deliver the packet to 55.66.0.100 on its local network.

c) Node A will determine that the destination address 55.66.0.100 is outside its local network and use the default route to forward the original packet i.e. it will attempt to send the packet to its default router 192.168.0.1. In order to create a Ethernet frame to this default router, Node A will issue an ARP request for 192.168.0.1. Node B will respond with the hardware address for this interface, 0D:0E:AA:90:0D:AB, Node A will create an Ethernet frame addressed to 0D:0E:AA:90:0D:AB, carrying the IP packet addressed to 55.66.0.100:7. Node B will receive the Ethernet frame, extract the IP packet, record the NAT information, replace the source address and port number, and decide to forward it to 1.2.3.4. In order to deliver it to 1.2.3.4, it will send an ARP request for the hardware address of 1.2.3.4 – unless this has not been cached from previous communications. It will then create an Ethernet frame addressed to

0D:0E:AA:90:00:08, carrying the IP packet addressed to 55.66.0.100:7. etc

5) The depletion of IPv4 addresses was a topic for discussions in the early 1990s. Discuss the causes for the depletion of these addresses and the effect that the use of CIDR and NAT had on address depletion.

The topic is discussed at length in various textbooks, so for a detailed discussion consult a textbook. A short description:
The introduction and use of classful addresses resulted in few of the available IP address ranges being used efficiently e.g. very few networks that used a class B network, actually had 65536 nodes, etc. The use of NAT, especially for residential networks, resulted in one globally unique IP address being shared by a larger number of end devices i.e. not all devices communicating over the Internet have a globally unique IP address. CIDR allows IP address ranges to be allocated to the power of 2 i.e. in contrast to classful addresses that specify the network ID in either 1, 2 or 3 bytes, classless addresses specify the network ID by indicating the bits in the address that identify the network ID. This allows aggregation of address ranges in routers and the creation of address ranges that are smaller or larger than address ranges in classful addressing. This resulted in fewer IP addresses being wasted by customers using address ranges that have significantly more addresses than the customer has nodes in a network.