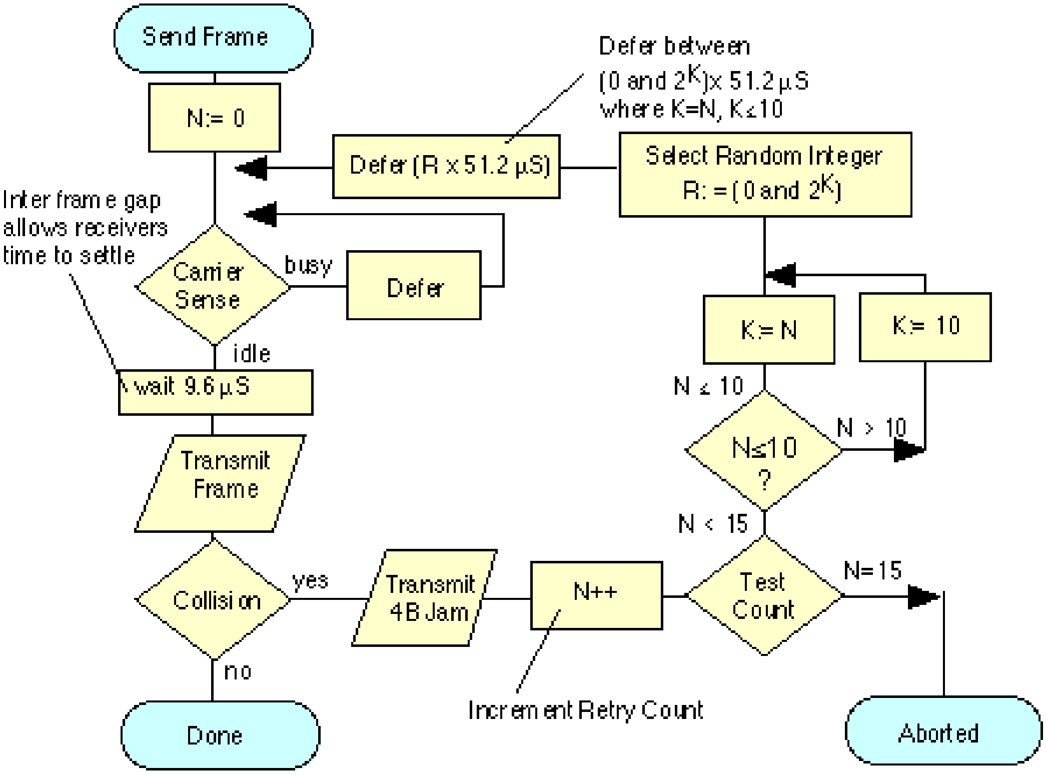
***Worked solutions for EG4546***

***Question 1***

(a) ***10*** ***Explain the method by which a number of systems may share a common and provide random access to send frames of data to each other.***



To control which computers are allowed to transmit at any given time, a protocol is required. The simplest protocol is known as ALOHA. ALOHA allows any computer to transmit at any time, but states that each computer must add a checksum at the end of its transmission to allow the receiver(s) to identify whether the frame was correctly received. Ethernet uses a refinement of ALOHA, known as Carrier Sense Multiple Access (CSMA), which improves performance when there is a higher medium utilisation. When a node has data to transmit, the node first listens to the cable (using a transceiver) to see if a carrier (signal) is being transmitted by another node. This may be achieved by monitoring whether a current is flowing in the cable (each bit corresponds to 18-20 milliAmps (mA)). The individual bits are sent by encoding them with a 10 (or 100 MHz for Fast Ethernet) clock using Manchester encoding. Data is only sent when no carrier is observed (i.e. no current present) and the physical medium is therefore idle. Any computer which does not need to transmit, listens to see if other computers have started to transmit information to it.

However, this alone is unable to prevent two nodes transmitting at the same time. If two noes simultaneously try transmit, then both could see an idle physical medium (i.e. neither will see the other's carrier signal), and both will conclude that no other node is currently using the network. In this case, both will then decide to transmit and a collision will occur. A second element to the Ethernet access protocol is used to detect when a collision occurs. When there is data waiting to be sent, each transmitting node monitors its own transmission. If it observes a collision (excess current above what it is generating, i.e. > 24 mA for coaxial Ethernet), it stops transmission immediately and instead transmits a 32-bit jam sequence. The purpose of this sequence is to ensure that any other node which may currently be receiving this frame will receive the jam signal in place of the correct 32-bit MAC CRC, this causes the other receivers to discard the frame due to a CRC error. To ensure that no node may completely receive a frame before the transmitting node has finished sending it, Ethernet defines a minimum frame size (i.e. no frame may have less than 46 bytes of payload). The minimum frame size is related to the distance which the network spans, the type of media being used and the number of repeaters which the signal may have to pass through to reach the furthest part of the LAN. Together these define a value known as the Ethernet Slot Time, corresponding to 512 bit times at 10 Mbps.

The transmitter initialises the number of transmissions of the current frame (n) to zero, and starts listening to the cable (using the carrier sense logic (CS) - e.g., by observing the Rx signal at transceiver to see if any bits are being sent). If the cable is not idle, it waits (defers) until the cable is idle. It then waits for a small Inter-Frame Gap (IFG) (e.g., 9.6 microseconds) to allow to time for all receiving nodes to return to prepare themselves for the next transmission. Transmission then starts with the preamble, followed by the frame data and finally the CRC-32. After this time, the transceiver Tx logic is turned off and the transceiver returns to passively monitoring the cable for other transmissions. During this process, a transmitter must also continuously monitor the collision detection logic (CD) in the transceiver to detect if a collision occurs. If it does, the transmitter aborts the transmission (stops sending bits) within a few bit periods, and starts the collision procedure, by sending a Jam Signal to the transceiver Tx logic. It then calculates a retransmission time. If all nodes attempted to retransmit immediately following a collision, then this would certainly result in another collision. Therefore a procedure is required to ensure that there is only a low probability of simultaneous retransmission. The scheme adopted by Ethernet uses a random back-off period, where each node selects a random number, multiplies this by the slot time (minimum frame period, 51.2 $\mu$S) and waits for this random period before attempting retransmission. The small Inter-Frame Gap (IFG) (e.g., 9.6 microseconds) is also added.

On a busy network, a retransmission may still collide with another retransmission (or possibly new data being sent for the first time by another node). The protocol therefore counts the number of retransmission attempts (using a variable N in the above figure) and attempts to retransmit the same frame up to 15 times. For each retransmission, the transmitter constructs a set of numbers:

{0, 1, 2, 3, 4, 5, ... L} where L is ([2 to the power (K)]-1) and where K=N; K<= 10;

A random value R is picked from this set, and the transmitter waits (defers) for a period R x (slot time) i.e. R x 51.2 Micro Seconds

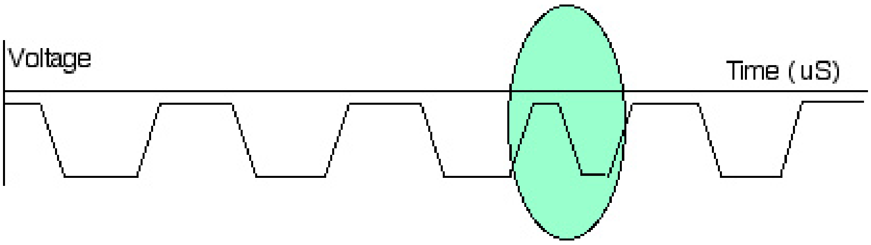
***(b) 4 Explain what is mean by the terms "Broadcast domain" and "Collision Domain" when applied to an Ethernet Local Area Network.***

Collision Domain = Set of systems that share access to a cable. This comprises one or more cable segments connected via repeaters and/or hubs.

Broadcast Domain = All broadcast packets must be propagated to all parts of the broadcast domain. Systems can share a common collision domain, or employ a network that combines several collision domains using bridges/switchees as well as repeaters and/or hubs. et of systems that are part of a L3 IP network, although a single broadcast domain may support more than one IP network. [Actually a single IP network can work over multiple broadcast domains using NBMA mode - but this is well beyond the scope of the course, and is really a kludge!]

(*c*) *6* ***Sketch an Ethernet frame and explain the function of the bytes that precede the first byte of the Layer 2 address.***

In Ethernet, the preamble consist of 8 bytes, the last of which has a special sequence,



known as the Start of Frame Delimiter (SFD) which indicates that actual information follows this.

The purpose of the preamble is to allow time for the receiver in each node to achieve lock of the receiver Digital Phase Lock Loop which is used to synchronise the receive data clock to the transmit data clock. At the point when the first bit of the preamble is received, each receiver may be in an arbitrary state (i.e. have an arbitrary phase for its local clock). During the course of the preamble it learns the correct phase, but in so doing it may miss (or gain) a number of bits. A special pattern (11), is therefore used to mark the last two bits of the preamble. When this is received, the Ethernet receive interface starts collecting the bits into bytes for processing by the MAC layer.

A Digital PLL (DPLL) circuit may consist of a serial shift register which receives digital input samples (extracted from the received signal), a stable local clock signal which supplies clock pulses to the shift register to drive it and a phase corrector circuit which takes the local clock and regenerates a stable clock in phase with the received signal by slowly adjusting the phase of the phase of the regenerated clock to match the received signal.

This circuit is useful when the data and clock are sent together over a common cable (as in Manchester encoding), since it allows the receiver to separate (regenerate) the clock signal from the received data. The regenerated clock signal is then used to sample the received data and determine the value of each received bit.

The start of frame delimiter contains the encoded sequence '11' which results in a rapid transition, and therefore a unique marker than can be used by the receiver to determine the start of the MAC header in the frame.

.

*Question 2*

*(a) 8 In the context of Fast Ethernet explain how the following sequence of bits* **{1 0 0 1 1 0 } are encoded using Multi-Level Threshold, MLT-3 line encoding.**

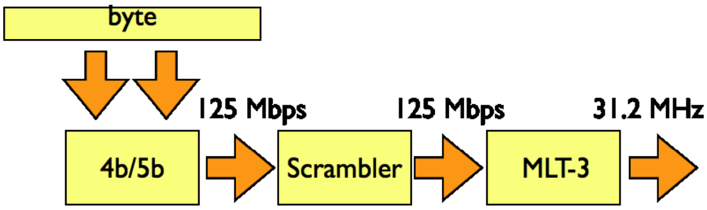
The bi-phase Manchester encoding can consume up to approximately twice the bandwidth of the original signal (20 MHz). While this was of little concern in coaxial cable transmission, the limited bandwidth of CAT5e cable necessitated a more efficient encoding method for 100 Mbps transmission using a 4b/5b MLT code. This uses three signal levels (instead of the two levels used in Manchester encoding) and therefore allows a 100 Mbps signal to occupy only 31 MHz of bandwidth.

4B/5B encoding is a type of 'Block coding'. This processes groups of bits rather than outputting a signal for each individual bit (as in Manchester encoding). A group of 4 bits is encoded so that an extra 5th bit is added. Since the input data is taken 4-bits at a time, there are 2^4, or 16 different bit patterns. The encoded bits use 5-bit, and hence have 2^5 or 32 different bit patterns. As a result, the 5-bit patterns can always have two '1's in them even if the data is all '0's a translation occurs to another of the bit patterns. This enables clock synchronisation, required for reliable data transfer.

Since there are (2^5) 32 possible combinations of 5 bits, and there are only (2^4) 16 combinations of 4 bits one half the patterns are unused. The chosen set of 16 5-bit patterns have the most transitions, this ensures clocking information is present in the signal (for locking the receiver DPLL). This results in a bandwidth increase of 25%.

Cross-Talk requirements / RF Emission led to the need for a scrambler. The data is finally sent as a 3-level physical waveform known as MLT-3. MLT-3 cycles through a set of voltage levels {-1, 0, +1}, to indicate a 1-bit. The signal stays the same when transmitting a 0 bit. It takes four 1 bits to generate a complete cycle, this the maximum fundamental frequency is reduced to one fourth of the baud rate.

This combined scheme of 4b/5b with MLT-3 encoding leads to a waveform of 31.25 MHz, well within the specification for Unshielded Twisted Pair Cabling.



Fast Ethernet Line Interface for 100 BT

{1 0 0 1 1 0 } are encoded as: +++0-- (assuming a zero start and positive waveform)

```
+ve   +-+-+-+
      |     |
0    -+     +-+
              |
-ve           +-+-+
```

**(b) 8 Provide a description of how a Network Interface Card (NIC) processes the addresses in a frame received.**

Frames that have a broadcast address or with a destination address that matches the node's source address are forwarded to the host.

(The destination address is a unicast frame to the node's own MAC address.
This may be configured in software, but is usually the manufacturer-assigned ID.)
Frames that match one of the configured multicast addresses are also forwarded to the host.

Before forwarding the frame is checked.
The length must be greater than the minimum and less than the maximum
There must be no residue bits (i.e. there must be an integer number of bytes).

The frame must have a valid type in the MAC packet type field
The frame is then demultiplexed based on the specified MAC packet type

It is passed to the appropriate protocol layer (e.g. LLC, ARP, IP)
(Packets destined for IP have a type field of 0x0800.)

**(c) 4 What is promiscuous mode?**
**Provide two examples of when this is useful.**

Promiscuous mode is entered by setting a flash in the NIC. This disables the receive address filters in the network interface. The interface therefore sees all frames, irrespective of their destination address and these are passed to the network layer to be received by the host system.

This mode is required for bridging in software and is useful for network operations, administration and management applications.

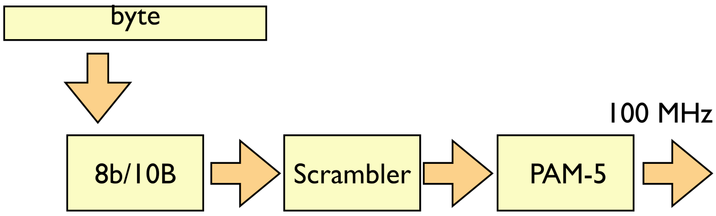
Two examples of promiscuous mode usage are:

An Ethernet Bridge, that receives and selectively forwards each frame.
A traffic analysis /sniffer tool that captures and decodes packets.

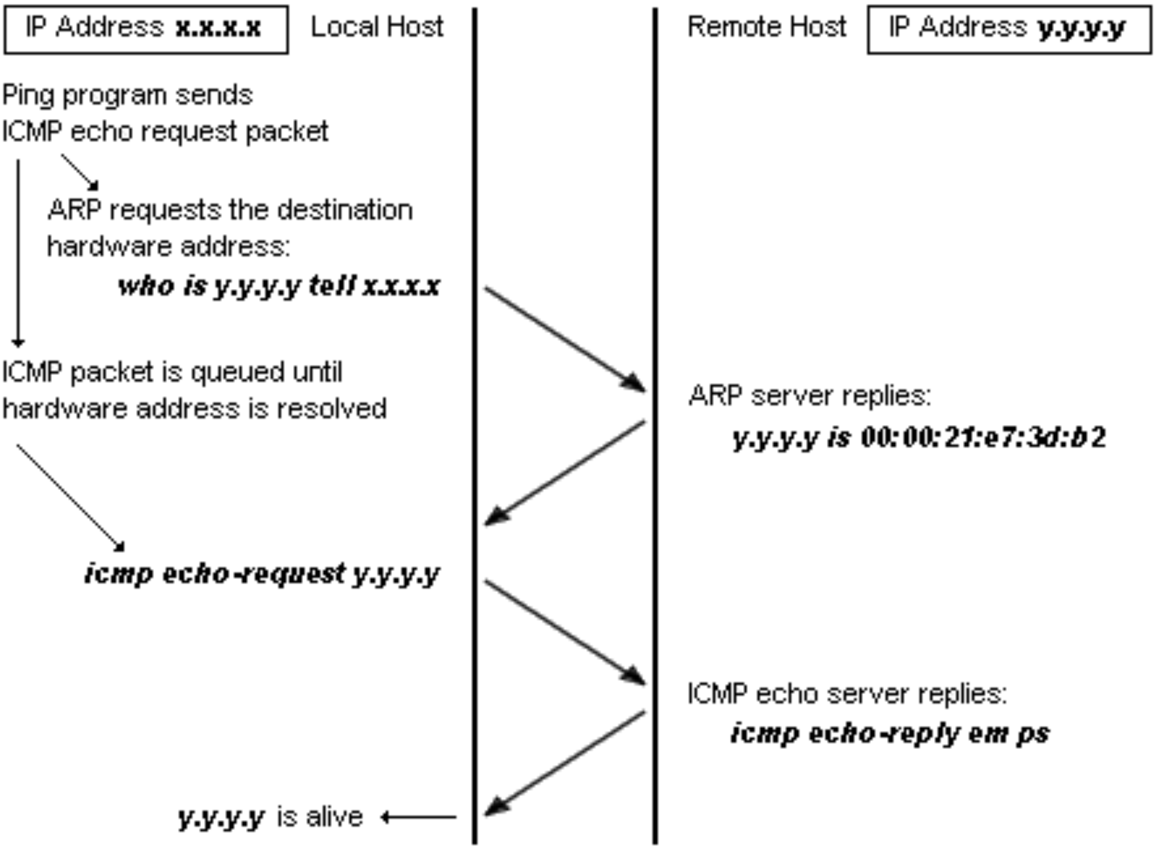
## Question 3

*(a) 8 Explain the operation of the IP ARP Protocol*

The term address resolution refers to the process of finding an address of a computer in a network. The address is “resolved” using a protocol in which a piece of information is sent by a client process executing on the local computer to a server process executing on a remote computer. The information received by the server allows the server to uniquely identify the network system for which the address was required and therefore to provide the required address. The address resolution procedure is completed when the client receives a response from the server containing the required address.



This diagram illustrates this using a ping request to generate a packet that triggers the use of arp at the network driver:


IP Address x.x.x.x
Local Host
Remote Host
IP Address y.y.y.y
Ping program sends
ICMP echo request packet
ARP requests the destination
hardware address:
who is y.y.y.y tell x.x.x.x
ICMP packet is queued until
hardware address is resolved
ARP server replies:
y.y.y.y is 00:00:21:e7:3d:b2
icmp echo-request y.y.y.y
ICMP echo server replies:
icmp echo-reply em ps
y.y.y.y is alive

The Ethernet network uses two hardware addresses which identify the source and destination of each frame sent by the Ethernet. The destination address (all 1's) may also identify a broadcast packet (to be sent to all connected computers) or a multicast packet (msb=1) (to be sent only to a selected group of computers). The hardware address is also known as the Medium Access Control (MAC) address, in reference to the IEEE 802.x series of standards which define Ethernet. Each computer network interface card is allocated a globally unique 6 byte address when the factory manufactures the card (stored in a PROM). This is the normal source address used by an interface. A computer sends all packets which it creates with its own hardware source address, and receives all packets which match its hardware address or the broadcast address. When configured to use multicast, a selection of multicast hardware addresses may also be received.

A protocol known as address resolution protocol (arp) is therefore used to translate between the two types of address. The arp client and server processes operate on all computers using IP over Ethernet. The processes are normally implemented as part of the software driver which drives the network interface card.

***(b) 2 Why is a cache used at the ARP requester?***

To reduce the number of address resolution requests, a client normally caches resolved addresses for a (short) period of time. The arp cache is of a finite size, and would become full of incomplete and obsolete entries for computers that are not in use if it was allowed to grow without check. The arp cache is therefore periodically flushed of all entries. This deletes unused entries and frees space in the cache. It also removes any unsuccessful attempts to contact computers which are not currently running.

**(c) 10 Provide diagrams and a detailed explanation on either of the two following topics:**

**Either**

**Explain the operation of a Domain Name System (DNS) resolver**

Indicative marks:
2 marks for understanding of DNS service
2 marks for understanding of query function
2 marks for understanding of response function
2 marks for understanding of cache
2 marks for explaining usage in wider Internet context or recursion

A name is a human-readable label assigned to a system.
An address is the basic routing identifier used to locate a system in the network.
- 2 marks each for clarity of the above definitions

DNS Service:
Mapping between the two is performed using the domain name service. This is an example of a client/server system which is used by the Internet Protocol (IP) Suite to resolve the logical names of nodes in an IP network to an IP address (see also arp - which is used to resolve Ethernet addresses to IP addresses).

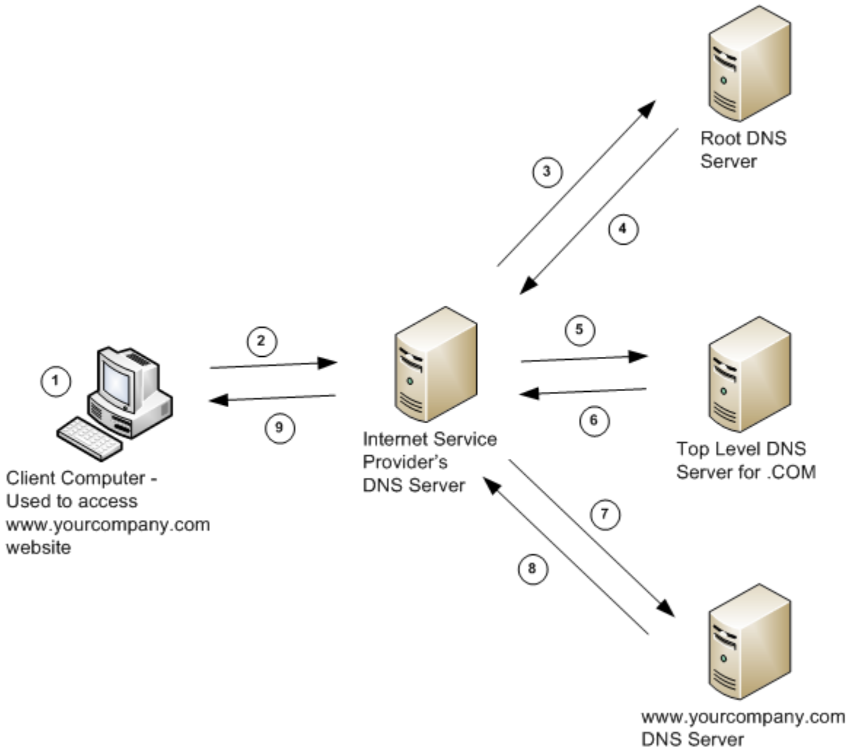
Resolution query:
The term address resolution refers to the process of finding an address of a computer in a network. The address is "resolved" using a protocol in which a piece of information is sent by a client process executing on the local computer to a server process executing on a remote computer. The client resolver must be pre-configured with the IP address of the DNS server.

Resolution response:
The information received by the server allows the server to uniquely identify the network system for which the address was required and therefore to provide the required address. Resolution may require recursive lookup on one or more DNS servers to finally receive an authoritative answer. Recursion involves searching multiple databases until the result is retrieved or it is concluded the name is unknown.

Cache:
A key point to be noted is that the system is requested by an application and the results are cached - so that the lookup does not need to be performed for every single use. In DNS the information provider determines the cache time - vastly different values are used for different applications (small where there is a churn of addresses, large for main infrastructure stability where change is not envisaged).

Use of information:
The address resolution procedure is completed when the client receives a response from the server containing the required address. this is then used as the IP destination address. Next hop resolution provides a MAC address based on this IP address.

The following diagram shows these exchanges, numbering each exchange in turn:


Root DNS Server
3
4
1
2
9
Internet Service Provider's DNS Server
5
6
Top Level DNS Server for .COM
Client Computer - Used to access www.yourcompany.com website
7
8
www.yourcompany.com DNS Server

1. Client enters 'www.yourcompany.com' internet address. Client computer needs the IP address translation of 'yourcompany.com' and first checks its own DNS cache for this information. If this is the first time using this website or the cache has been cleared it cannot find the IP address here.
2. The client computer (or "query"?) is then redirected to the Internet Service Provider's (ISP's) DNS Server. The ISP's DNS server checks its own cache but it will not be there if the site has not been accessed before.
3. The ISP's DNS server redirects the query to the Root DNS Server. Every DNS server has a file that contains a list of all of the root DNS servers.
4. The root DNS server maintains information about where a top-level (.com) DNS server is located and returns this information to the ISP's DNS Server.
5. The ISP's DNS server redirects the query to a top-level (.com) DNS server.
6. The top-level (.com) DNS server knows the IP address of the DNS server for the yourcompany.com domain and returns that information to the ISP's DNS server.
7. The ISP's DNS server redirects the query to the actual DNS server for the yourcompany.com domain.
8. The DNS server for www.yourcompany.com returns the IP address of the host of www.yourcompany.com to the ISP's DNS server.
9. Last, the ISP's DNS server sends the IP address to the client computer so the client can access www.yourcompany.com.

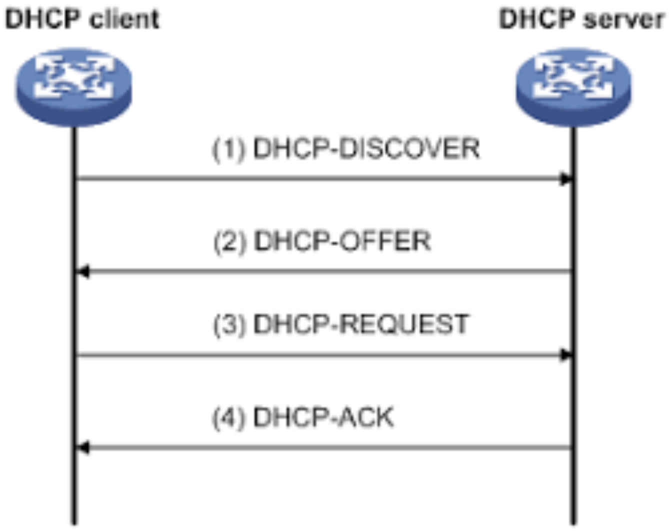
**or**

**How can an IP network node automatically determine IP network address that it should use on a specific LAN segment?**

Indicative marks:

- 2 marks for understanding of address usage
- 2 marks for understanding a DNS query (broadcast or multicast in v6)
- 2 marks for understanding a DNS response (unicast)
- 2 marks for understanding of lease and cache
- 2 marks for explaining usage in the wider Internet context or recursion
-

A unicast/broadcast IPv4 address is a 32 bit value (i.e. four bytes) which is allocated to each system in the Internet. The 32-bit value uniquely identifies this system, and therefore no two systems may have the same IP address. Some systems have more than one IP address, in which case they may be reached by any of their IP addresses. The Dynamic Host Configuration Protocol is a protocol for assigning dynamic IP addresses to devices on a network. With dynamic addressing, a device can have a different IP address every time it connects to the network. In some systems, the device's IP address can even change while it is still connected. DHCP also supports a mix of static and dynamic IP addresses. When a computer or other networked device connects to a network, its DHCP client software in the operating system sends a broadcast query requesting necessary information. The client broadcasts messages on the network subnet using the destination address 255.255.255.255 or the specific subnet broadcast address.

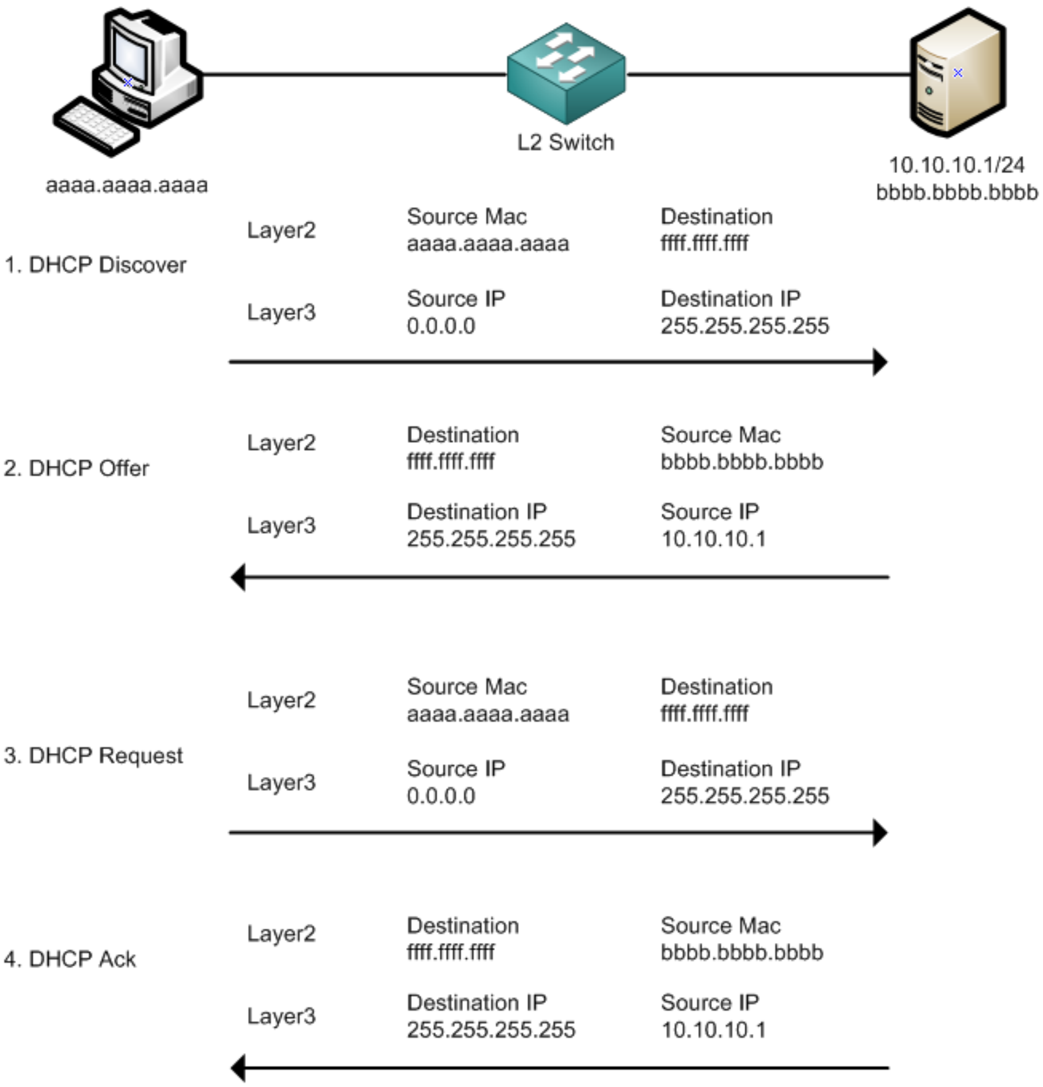


Any DHCP server on the network may service the request. The DHCP server manages a pool of IP addresses and information about client configuration parameters such as default gateway, domain name, the name servers, and time servers. When a DHCP server receives a DHCP-DISCOVER message from a client, which is an IP address lease request, the server reserves an IP address for the client and makes a lease offer by sending a DHCP- OFFER message to the client. This message contains the client's MAC address, the IP address that the server is offering, the subnet mask, the lease duration, and the IP address of the server making the offer.

On receiving a request (broadcast), a server may respond with specific information for each client, as previously configured by an administrator, or with a specific address and any other information valid for the entire network, and the time period for which the allocation (lease) is valid. When other DHCP servers receive a request message, they withdraw any offers that they might have made to a client. The client ACKs the completed request. A host typically queries for this information immediately after booting, and periodically thereafter before the expiration of the information. When an assignment is refreshed by the client computer, it initially requests the same parameter values, but may be assigned a new address from the server, based on the assignment policies set by administrators.

Cache: A DHCP cache is used to record the binding until the lease expires.

**DHCP Message Exchange**

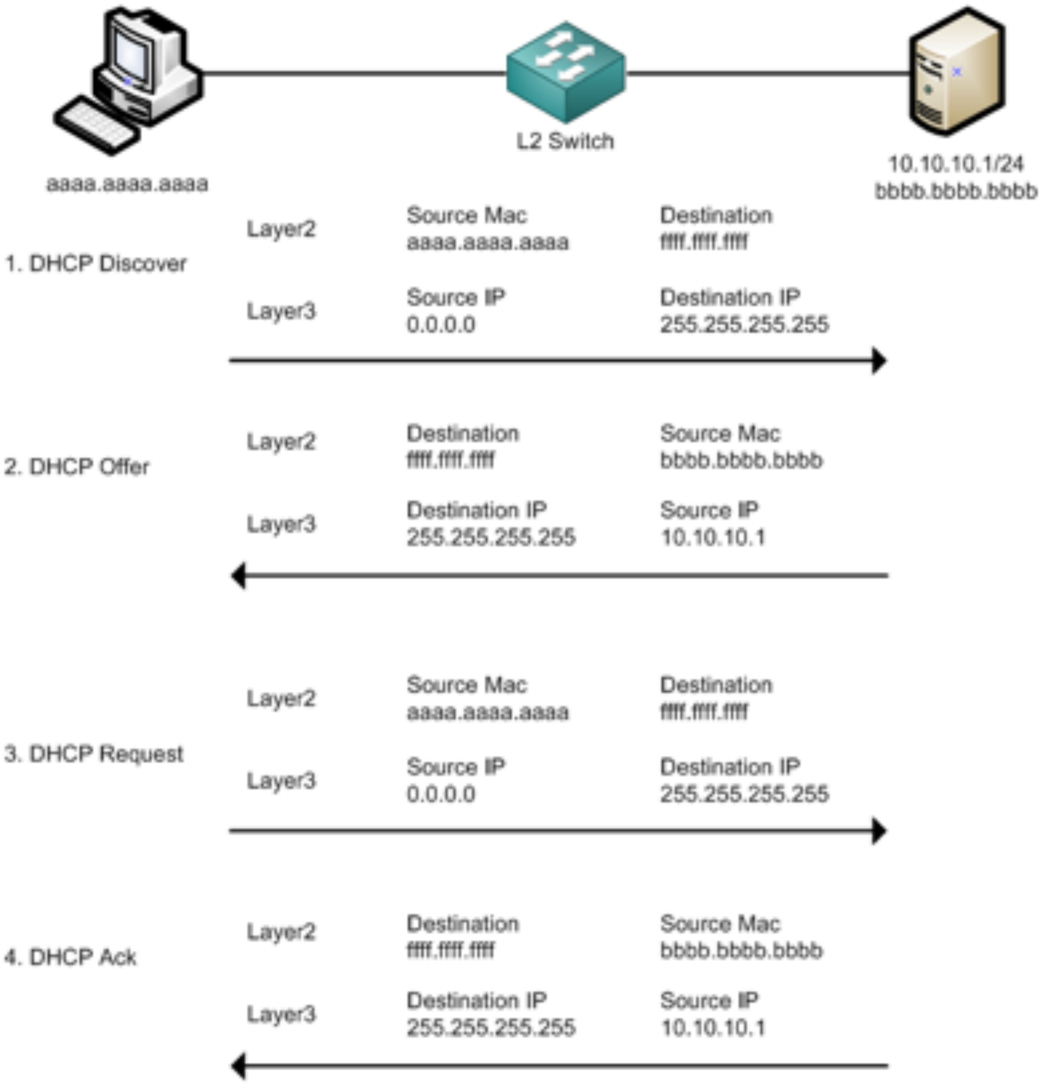


| | | | |
|---|---|---|---|
| 1. DHCP Discover | Layer2 | Source Mac<br>aaaa.aaaa.aaaa | Destination<br>ffff.ffff.ffff |
| | Layer3 | Source IP<br>0.0.0.0 | Destination IP<br>255.255.255.255 |
| 2. DHCP Offer | Layer2 | Destination<br>ffff.ffff.ffff | Source Mac<br>bbbb.bbbb.bbbb |
| | Layer3 | Destination IP<br>255.255.255.255 | Source IP<br>10.10.10.1 |
| 3. DHCP Request | Layer2 | Source Mac<br>aaaa.aaaa.aaaa | Destination<br>ffff.ffff.ffff |
| | Layer3 | Source IP<br>0.0.0.0 | Destination IP<br>255.255.255.255 |
| 4. DHCP Ack | Layer2 | Destination<br>ffff.ffff.ffff | Source Mac<br>bbbb.bbbb.bbbb |
| | Layer3 | Destination IP<br>255.255.255.255 | Source IP<br>10.10.10.1 |

Use of information: The address is used by the client as the source IP address of all packets sent to the network, it also uses the addressing information to set the local broadcast address, netmask and to configure the local router. It may use this information to set the DNS server.

This is a more detailed diagram:

**DHCP Message Exchange**



| | | | |
|---|---|---|---|
| 1. DHCP Discover | Layer2 | Source Mac<br>aaaa.aaaa.aaaa | Destination<br>ffff.ffff.ffff |
| | Layer3 | Source IP<br>0.0.0.0 | Destination IP<br>255.255.255.255 |
| 2. DHCP Offer | Layer2 | Destination<br>ffff.ffff.ffff | Source Mac<br>bbbb.bbbb.bbbb |
| | Layer3 | Destination IP<br>255.255.255.255 | Source IP<br>10.10.10.1 |
| 3. DHCP Request | Layer2 | Source Mac<br>aaaa.aaaa.aaaa | Destination<br>ffff.ffff.ffff |
| | Layer3 | Source IP<br>0.0.0.0 | Destination IP<br>255.255.255.255 |
| 4. DHCP Ack | Layer2 | Destination<br>ffff.ffff.ffff | Source Mac<br>bbbb.bbbb.bbbb |
| | Layer3 | Destination IP<br>255.255.255.255 | Source IP<br>10.10.10.1 |