Notes for Firewall Simulator Developers

Based on course notes by Ben Bower for the subject INFT154 System Security in the Diploma of IT (Network Engineering) at CIT Reid.

IP Addresses

What is an IP address?

Every TCP/IP host is identified by a logical address called an IP address. The IP address identifies a system's location on the network in the same way a street address identifies a house on a city block. Just as a street address must identify a unique residence, an IP address must be unique.

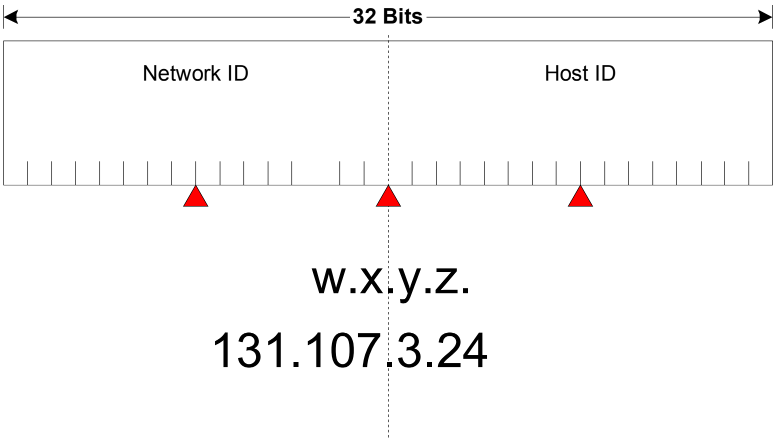
Each IP address defines the network ID and host ID. The network ID identifies the systems that are located on the same physical segment. All systems on the same physical segment must have the same network ID. The network ID must be unique to the internetwork.

The host ID identifies a workstation, server, router, or other TCP/IP host within a segment. The address for each host must be unique to the network ID.

Network ID and Host ID

Each IP address is 32 bits long and is composed of four 8-bit fields, called octets. Octets are separated by periods. The octet represents a decimal number in the range 0-255. This format is called dotted decimal notation. Following is an example of an IP address in binary and dotted decimal formats.

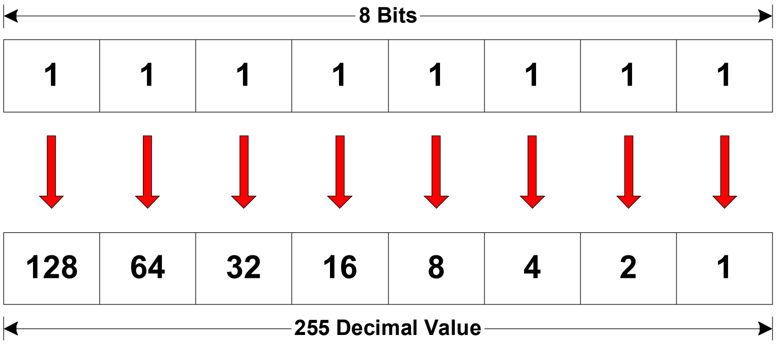
Class B Address



Binary Format	**Dotted Decimal Notation**
10000011 01101011 00000011 00011000	131.107.3.24

Converting IP Addresses from Binary to Decimal

Each bit position in an octet has an assigned decimal value. A bit that is set to 0 always has a zero value. A bit that is set to 1 can be converted to a decimal value. The low-order bit represents a decimal value of one. The high-order bit represents a decimal value of 128. The highest decimal value of an octet is 255 – that is when all bits are set to 1.



The following table shows how the bits in one octet are converted from binary code to a decimal value.

Binary Code	**Bit Values**	**Decimal Value**
00000000	0	0
00000001	1	1
00000011	1+2	3
00000111	1+2+4	7
00001111	1+2+4+8	15
00011111	1+2+4+8+16	31
00111111	1+2+4+8+16+32	63
01111111	1+2+4+8+16+32+64	127
11111111	1+2+4+8+16+32+64+128	255

Address Classes

The Internet community has defined five IP address classes to accommodate networks of varying sizes. The class of address defines which bits are used for the network ID and which bits are used for the host ID. It also defines the possible number of networks and the number of hosts per network.

The following table shows the network and host ID fields for class A, B and C IP addressing.

Class	**IP Addresses**	**Network ID**	**Host ID**
A	w.x.y.z	W	x.y.z
B	w.x.y.z	w.x	y.z
C	w.x.y.z	w.x.y	Z

Class A

Class A addresses are assigned to networks with a very large number of hosts. The high-order bit in a class A is always set to zero. The next seven bits (completing the first octet) complete the network ID. The remaining 24 bits (the last three octets) represent the host ID. This allows for 126 networks and approximately 17 million hosts per network.

Class B

Class B addresses are assigned to medium-sized to large-sized networks. The two high-order bits in a class B address are always set to binary 10. The next 14 bits (completing the first two octets) complete the network ID. The remaining 16 bits (last two octets) represent the host ID. This allows for 16,384 networks and approximately 65,000 hosts per network.

Class C

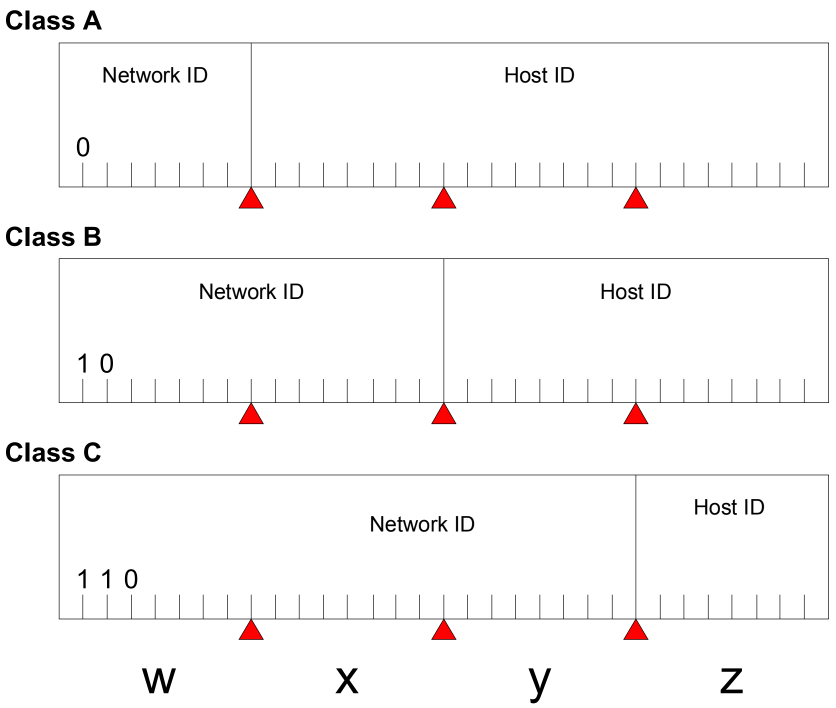
Class C addresses are used for small local area networks (LANs). The three high-order bits are always set to binary 110. The next 21 bits (completing the first three octets) complete the network ID. The remaining 8 bits (last octet) represent the host ID. This allows for approximately 2 million networks and 254 hosts per network.

Class D

Class D addresses are used for multicast group usage. A multicast group may contain one or more hosts, or none at all. The four high-order bits in a class D address are always set to binary 1110. The remaining bits designate the specific group in which the client participates. There are no network or host bits in the multicast operations. Packets are passed to a selected subset of hosts on a network. Only those hosts registered for the multicast address accept the packet.

Class E

Class E is an experimental address not available for general use: it is reserved for future use. The high-order bits in Class E addresses are set to 1111.



Address Class Summary

The following table summarises the number of networks and number of hosts per network, and the range of network IDs in class A, B, and C IP addresses.

Class	**Number of Networks**	**Number of Host per Network**	**Range of Network Ids (First Octet)**
Class A	126	16,777,214	1 – 126
Class B	16,384	65,534	128 – 191
Class C	2,097,152	254	192 – 223

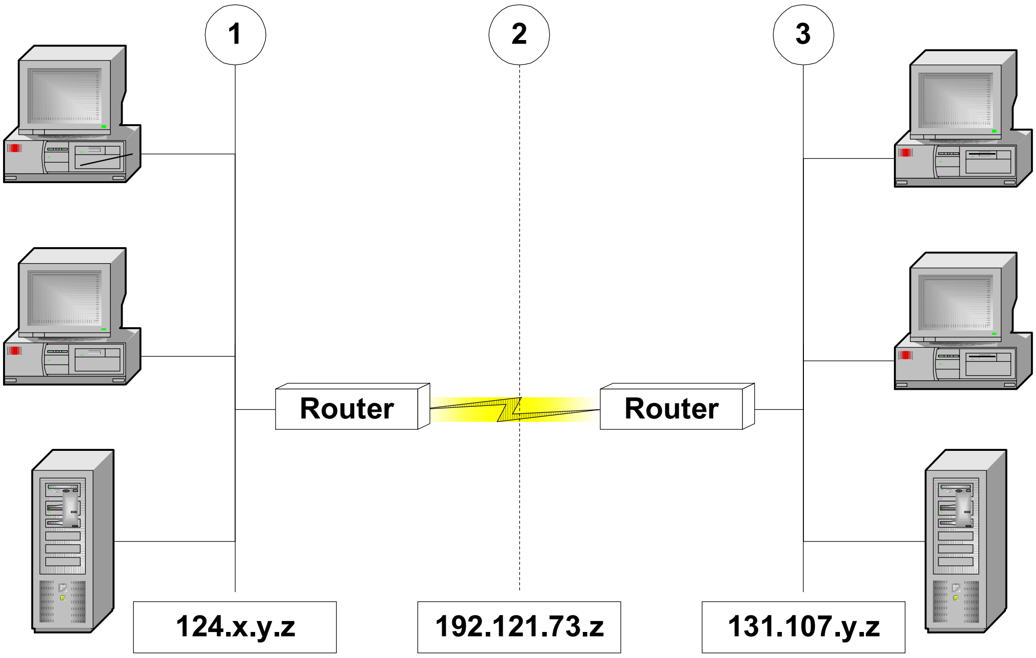
Addressing Guidelines

Follow these guidelines when assigning network IDs and Host IDs.

- The network ID cannot be 127. This ID is reserved for loopback functions.
- The network ID and host ID bits cannot all be 1's. If all bits are set to 1, the address is interpreted as a broadcast rather than a host ID.
- The network ID and host ID bits cannot all be 0's. If all bits are set to 0, the address is interpreted to mean "this network only."
- The host ID must be unique to the local network ID.

Assigning Network IDs

The network ID identifies the TCP/IP hosts that are located on the same physical network. All hosts on the same physical network must be assigned the same network ID to communicate with each other.



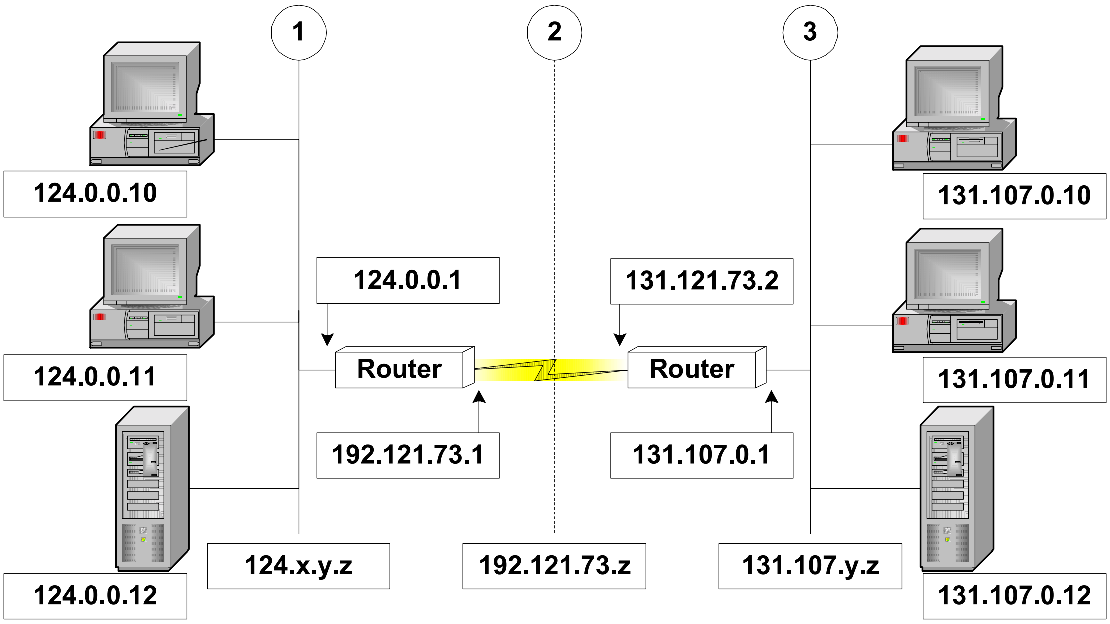
If routers connect your networks, a unique network ID is required for each wide area connection. For example in the graphic:

- Networks 1 and 3 represent two routed networks
- Network 2 represents the wide area network connection between the router. Network 2 requires a network ID so that the interfaces between the two routers can be assigned unique host IDs.

Note: If you plan to connect your network to the worldwide Internet, you must obtain the network ID portion of the IP address to guarantee IP network ID uniqueness. For domain name registration and IP network number assignment, visit AUNIC's online registration page at http://aunic.net.au.

Assigning Host IDs

The host ID identifies a TCP/IP host within a network and must be unique to the network ID. All TCP/IP hosts, including interfaces to routers, require unique host IDs.



The host ID of the router interface is the IP address configured as a workstation's default gateway when TCP/IP is installed. For example, for the host on subnet 1 with an IP address of 124.0.0.11, the IP address of the default gateway is 124.0.0.1.

Valid Host IDs

The following table lists the valid ranges of host IDs for a private internetwork.

Address Class	**Beginning Range**	**Ending Range**
Class A	w.0.0.1	w.255.255.254
Class B	w.x.0.1	w.x.255.254
Class C	w.x.y.1	w.x.y.254

Suggestions for Assigning Host IDs

There are no rules for how to assign valid IP addresses. You can number all TCP/IP hosts consecutively, or you can number them so they can easily be identified – for example:

- Assign host IDs in groups based on host or server type
- Designate routers by their IP address

An example of how to assign a Class C range could be:

Range	**Possible Use**
w.x.y.1 – w.x.y.30	Servers and print devices
w.x.y.101 – w.x.y.200	DHCP scope for workstations
w.x.y.201 – w.x.y.254	Comms equipment. Default gateway for a network typically uses the .254 address

Subnetting

What is a Subnet?

A subnet is a physical segment in a TCP/IP environment that uses IP addresses derived from a single network ID. Typically, an organisation acquires one network ID from the InterNIC.

Dividing the network into subnets requires that each segment uses a different network ID, or Subnet ID. A unique subnet ID is created for each segment by partitioning the bits in the host ID into two parts. One part is used to identify the segment as a unique network, and the other part is used to identify the hosts. This is referred to as subnetting or subnetworking.

Subnetting Benefits

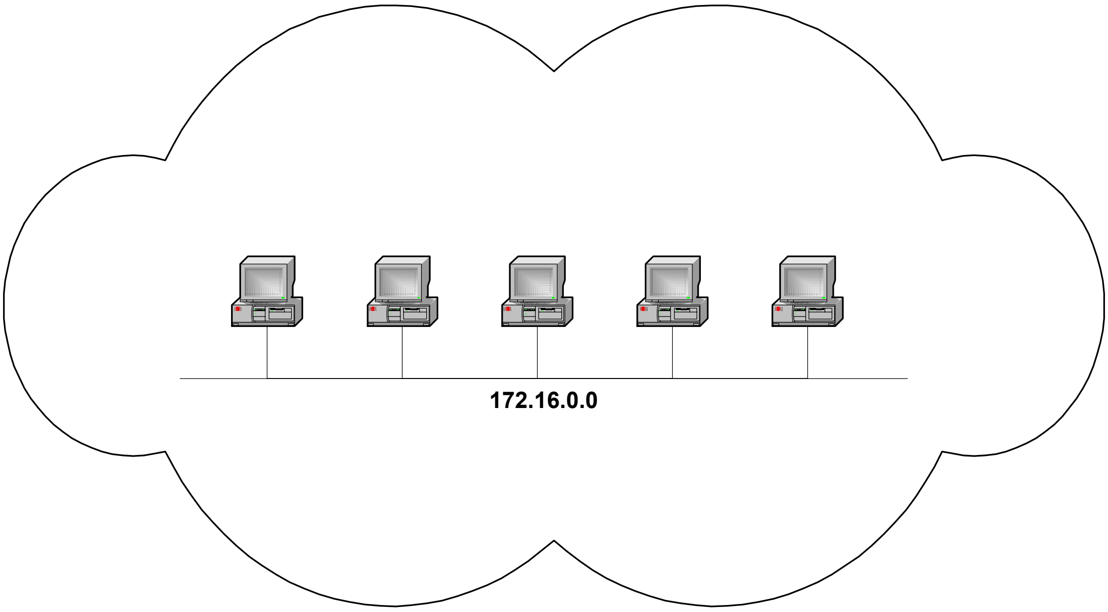
Organisations use subnetting to apply one network across multiple physical segments. Thus you can:

- Mix different technologies, Ethernet and token ring
- Overcome limitations of current technologies, such as exceeding the maximum number of hosts per segment.
- Reduce network congestion by redirecting traffic and reducing broadcasts.

Note: Subnetting is defined in RFC 950.

Addressing Without Subnets

For a network without subnets, the outside world sees the organisation as a single network, and no detailed knowledge of the internal structure is required. All packets addressed to 172.16 are treated the same way, regardless of the third or fourth octet of the address.

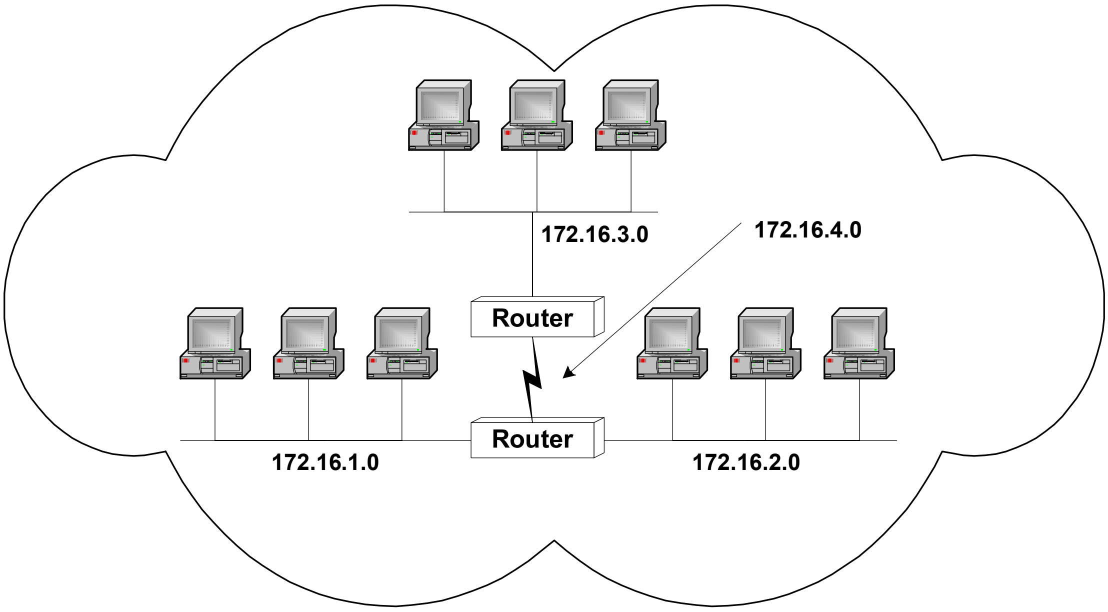


Network addressing with the scheme we have set up so far has no way of distinguishing individual segments (wires) within the network. Inside the cloud having no subnets we see a single large broadcast domain – all systems on the network encounter all the broadcasts on the network. This type of configuration can result in relatively poor network performance.

By default this Class B address space defines on wire with 65,000 workstations on it. What is needed is a way to divide this wire into segments.

Addressing With Subnets

With subnets, the network address use is more efficient. There is no change to how the outside world sees the network, but within the organisation, there is additional structure.



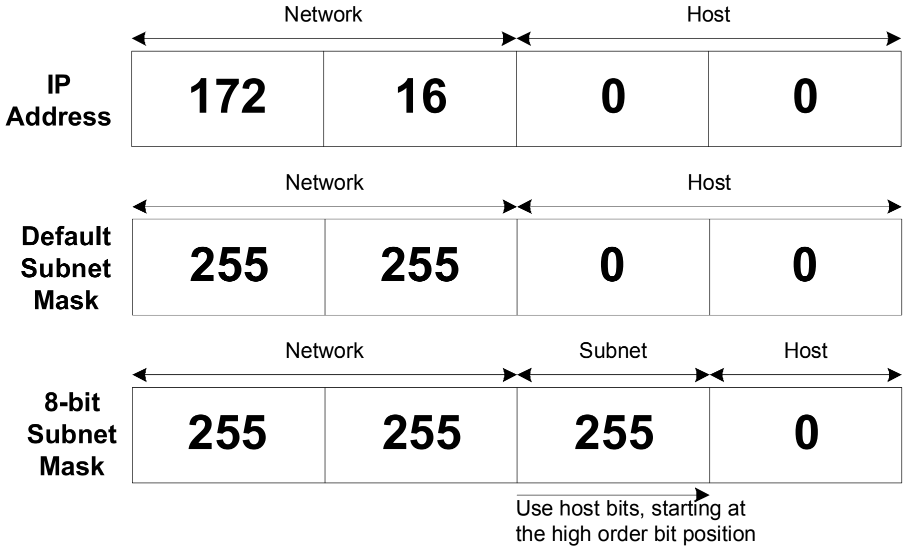
In this example, the network 172.16.0.0 is subdivided or broken up into four subnets, 172.16.1.0, 172.16.2.0, 172.16.3.0, and 172.16.4.0. Routers determine the destination network using the subnet mask address, limiting the amount of traffic on the other network segments.

What is a Subnet Mask?

A subnet mask is a 32-bit address used to:

- Block out a portion of the IP address to distinguish the network ID from the host ID.
- Specify whether the destination host's IP address is located on a local network or a remote network.

Each host on a TCP/IP network requires a subnet mask – either a default subnet mask, which is used when a network is not divided into subnets, or a custom subnet mask, which is used when a network is divided into subnets.



	Network		Host	
IP Address	172	16	0	0
	Network		Host	
Default Subnet Mask	255	255	0	0
	Network		Subnet	Host
8-bit Subnet Mask	255	255	255	0

Use host bits, starting at the high order bit position

The layout of the subnet mask field is as follows:

- Binary 1 for the network bits
- Binary 1 for the subnet bits
- Binary 0 for the host bits

Subnet masks indicate which of the bits in the host field are used to specify different parts (subnets) of a particular network.

Default Subnet Masks

A default subnet mask is used on TCP/IP networks that are not divided into subnets. All TCP/IP hosts require a subnet mask, even on a single-segment network. The default subnet mask you will use depends on the address class.

Address Class	**Bits Used for Subnet Mask**	**Dotted Decimal Notation**
Class A	11111111 00000000 00000000 00000000	255.0.0.0
Class B	11111111 11111111 00000000 00000000	255.255.0.0
Class C	11111111 11111111 11111111 00000000	255.255.255.0

All bits that correspond to the network ID are set to 1. The decimal value in each octet is 255. All bits that correspond to the host ID are set to 0.

Determining the Destination of a Packet

ANDing is the internal process that TCP/IP uses to determine whether a packet is destined for a host on a local network or a remote network.

When TCP/IP is initialised, the host's IP address is ANDed with its subnet mask. Before a packet is sent, the destination IP address is ANDed with the same subnet mask. If both results match, IP knows that the packet belongs to a host on the local network. If the results don't match, the packet is sent to the IP address of an IP router.

To AND the IP address to a subnet mask, TCP/IP compares each bit in the IP address to the corresponding bit in the subnet mask. If both bits are 1's, the resulting bit is 1. If there is any other combination, the resulting bit is 0 - for example:

Bit Combination	**Result**
1 AND 1	1
1 AND 0	0
0 AND 0	0
0 AND 1	0

ANDing Default Subnet Mask

	Network		**Host**	
172.16.2.160	**10101100**	**00010000**	**00000010**	**10100000**
255.255.0.0	**11111111**	**11111111**	**00000000**	**00000000**
	10101100	**00010000**	**00000000**	**00000000**
Network	**172**	**16**	**0**	**0**

The router extracts the IP destination address from the packet and retrieves the internal subnet mask. The router performs a logical AND operation to obtain the network number. During the logical AND operation, the host portion of the destination address is removed. Routing decisions are then based on the network number only. In this example, using the default subnet mask, the network number "extracted" is 172.16.0.0.

ANDing Custom Subnet Mask

	Network		Subnet	Host
172.16.2.160	10101100	00010000	00000010	10100000
255.255.255.0	11111111	11111111	11111111	00000000
	10101100	00010000	00000010	00000000
Network	172	16	2	0

With eight bits of subnetting, the extracted network (subnet) number is 172.16.2.0. This example shows eight bits "borrowed" to extend the network portion of the address.

Planning Subnetting

The IP addressing scheme used for subnets is referred to as subnetting. Before you implement subnetting, you need to determine your current requirements and plan for future requirements. Follow these guidelines:

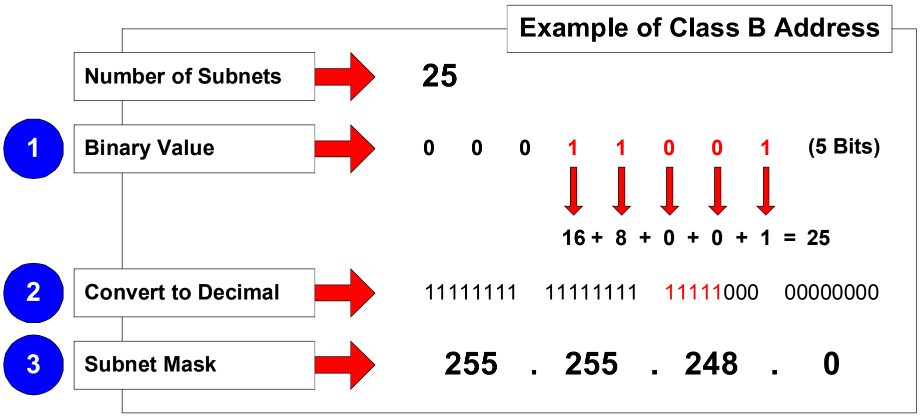
- Determine the number of required network IDs
  a. One for each subnet
  b. One for each wide area network connections
- Determine the number of required host IDs per subnet
  a. One for each TCP/IP host
  b. One for each router interface
- Define one subnet mask based on requirements
- Define a unique subnet ID for each physical segment based on the subnet mask
  a. Each subnet will have a wire or network address. This is all host bits set to 0
  b. Each subnet will have a broadcast address. This is all host bits set to 1
- Define valid host IDs for each subnet based on the subnet ID

Subnet Calculation Example

The organisation we are working for has a class B network address. They have 25 branch offices around the globe with 20 - 30 nodes in each site. We need to develop a subnetting scheme that meets the client's needs.

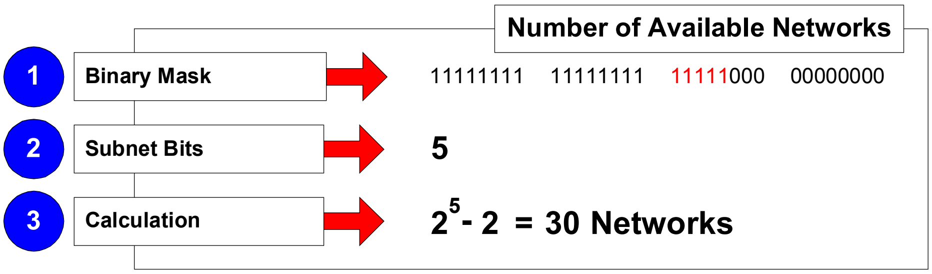
First we work out how many bits we need to borrow to support 25 networks.

1. Convert the number of segments to binary
2. Count the number of required bits
3. Convert the required number of bits to decimal (high order)



Calculating Number of Available Networks

Once you have the number of bits required it is easy to then calculate the number of available networks.



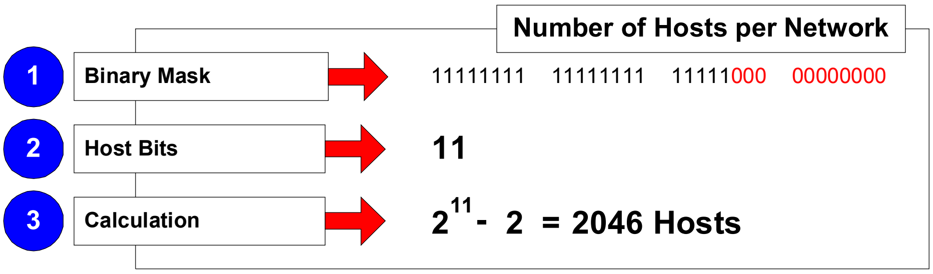
1. Convert the subnet mask to binary.
2. Count the number of subnet bits.
3. Calculate 2 to the power of *Subnet Bits* – 2.

According to RFC 950 you cannot use the first and last subnet (this is why the –2 is in the calculation). You can use these subnets on devices that support them.

Note: Subnet IDs comprised of all 0's or all 1's are called *special-case subnet addresses*. A Subnet ID of all 1's indicates a subnet broadcast, and a subnet ID of all 0's indicates this subnet. When subnetting, it is recommended not to use these subnet ID's. However, it is possible to use these special-case subnet addresses if they are supported by all routers and hardware on your network. RFC950 discusses the limitations imposed when using special-case addresses.

Calculating Number of Hosts per Network

When you have the number of bits in the subnet mask it is also easy to calculate the number of hosts allowed on the network.

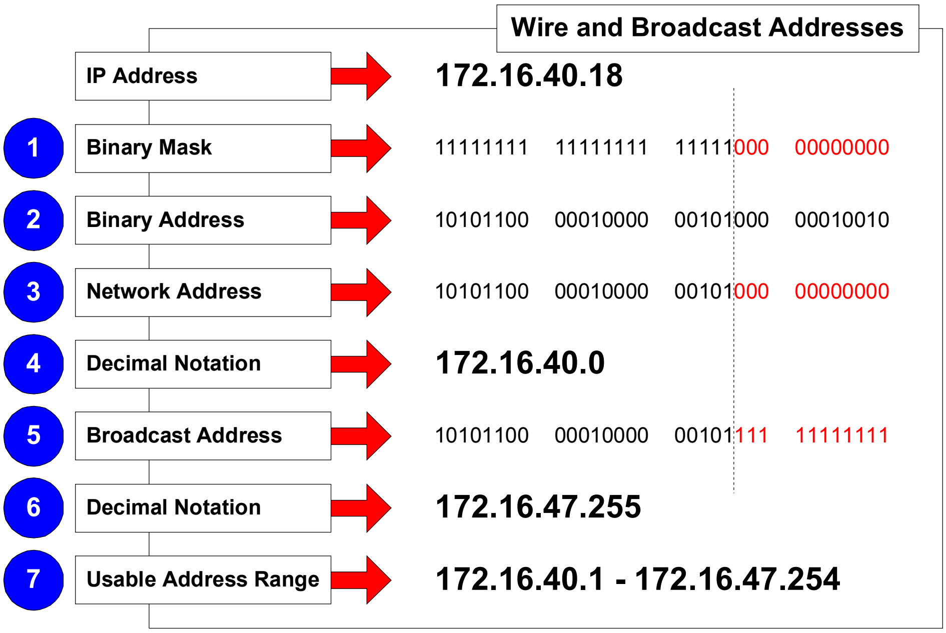


1. Convert the subnet mask to binary.
2. Count the number of subnet bits.
3. Calculate 2 to the power of *Host Bits* – 2.

The –2 in this calculation is because you cannot use the first or last address in a subnet. The first address is the wire or network address and the last is the broadcast address for the subnet.

Calculating Wire and Broadcast Addresses

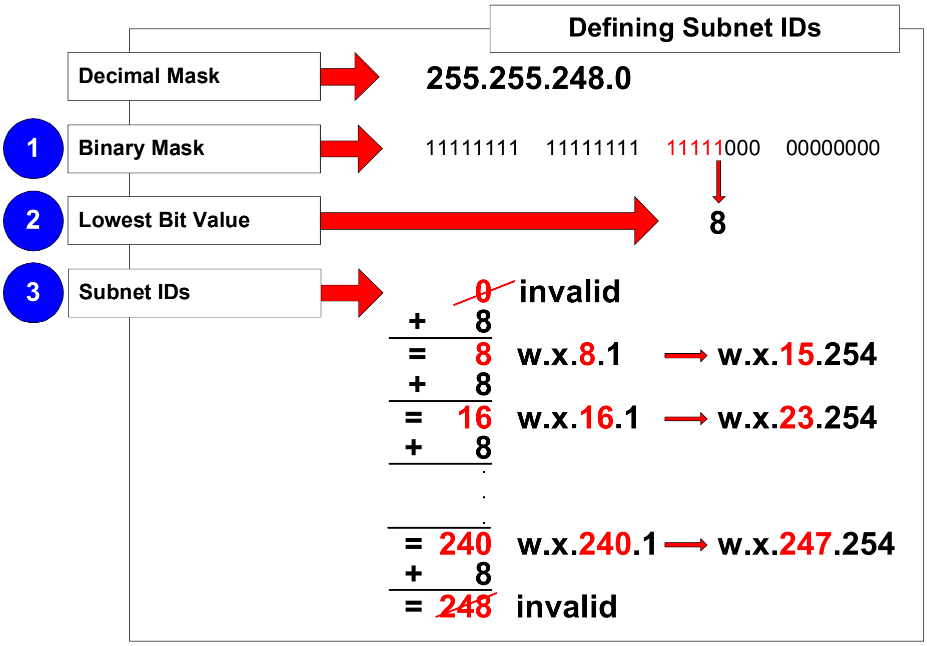
If you know an IP address and subnet mask you can then calculate the Wire and Broadcast addresses for that subnet.



1. Convert the subnet mask to binary
2. Convert the IP address to binary and draw a line through the two addresses after the last subnet bit
3. On the IP address change everything to the right of the line to a zero
4. Convert this to decimal notation and you now have the Wire address for the subnet
5. On the IP address change everything to the right of the line to a one
6. Convert this to decimal notation and you now have the broadcast address for the subnet
7. The usable address range starts at the first address after the wire address and finishes at the last address before the broadcast address.

Calculating Subnet IDs

You can also use the subnet mask to calculate the Subnet IDs. Once you have the subnet IDs it is also easy to use this method to discover the Usable address range, Wire address and Broadcast address for each subnet.



1. List the number of bits in high order used for the subnet ID. For example, if 5 bits are used for the subnet mask, the binary octet is 11111000.
2. Convert the bit with the lowest value to decimal format. This is the increment value to determine each subnet. For example, if you use five bits, the lowest value is 8.
3. Starting with zero, increment the value for each bit combination until the next increment is 256.

OSI and Networking Devices

During this module we will look at the way data travels through the layers of the OSI model. We will then investigate the different devices available at the first three layers.

What is the OSI model?

The OSI model is a set of guidelines for vendors to follow to allow simple interconnection between different systems.

Layers of OSI and TCP/IP Models

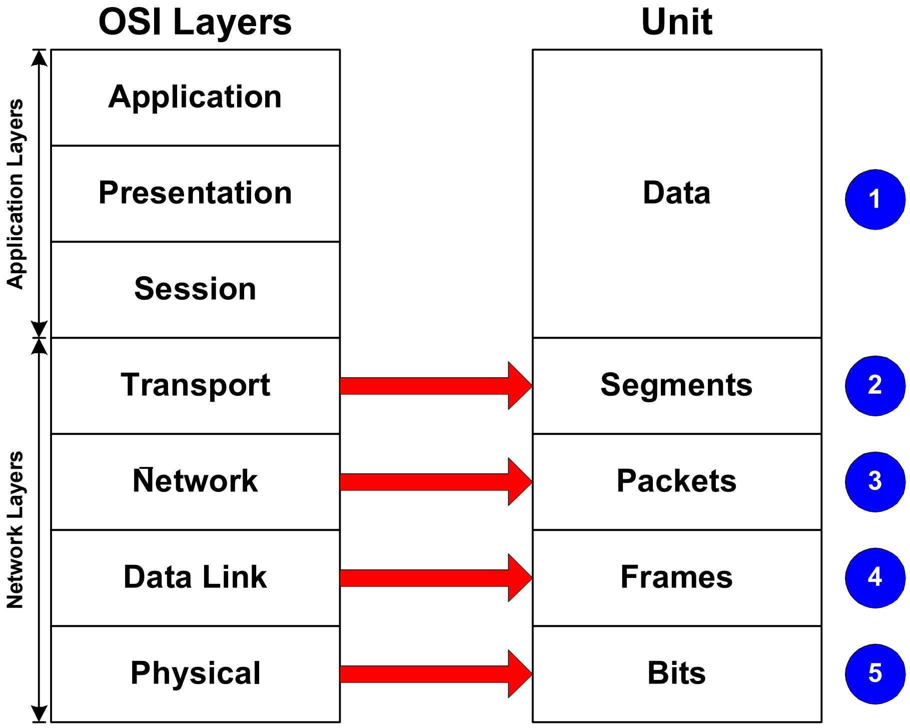
OSI	TCP/IP
Application	Application
Presentation	
Session	
Transport	Tranport
Network	Network
Data Link	Physical
Physical	

Benefits of Layered Model

- A change made to one layer does not affect other layers.
- Provides a guideline for all vendors to follow
- Provides guidelines about how to communicate with other machines. It does not specify types of technology.

Data Encapsulation

As Internetworks perform services for users, the flow and packaging of the information changes. In this example of internetworking, five encapsulation steps occur:



1. As a user sends an e-mail message, its alphanumeric characters are converted to use the Internetwork. This is the data.

2. One change packages the message “data” for the Internetwork transport subsystem. By using segments, the transport function ensures that the message hosts at both ends of the e-mail system can reliably communicate.

3. The next change prepares the data so the transport function can use the Internetwork by putting the data into a packet or datagram that contains a network header with source and destination logical addresses. These addresses help network devices send the packets across the network along a chosen path.

4. Each network device must put the packet into a frame so it can communicate over its interface to the network. The frame allows connection to the next directly connected network interface on the link. Each device in the chosen network path requires framing to connect to the next device. Frame types are media dependent.

5. The frame must be converted into a pattern of ones and zeroes for transmission on the medium (usually a wire). Some clocking function enables the devices to distinguish these bits as they traverse the medium.

The medium on the physical Internetwork can vary along the path used. For example, the e-mail message can originate on a LAN, cross a campus backbone, go out a low-speed WAN link, and use a higher-speed WAN link until it reaches its destination on another remote LAN.

Physical Layer

The physical layer describes how information is placed on the wire. The unit of information at the physical layer is bits.

Physical - Bits
Repeater
Hub

Hubs and Repeaters

In the Ethernet world there are two devices that work at the physical layer. These are Repeaters and Hubs. A Repeater takes information and from one piece of wire and regenerates the signal and the n propagates it down another piece of wire. A Hub is a multi-port repeater. A signal that is received on one port is regenerated and flooded out all other ports. This makes capturing traffic on a Hub very simple. Each port sees all the traffic within the collision domain.

When using a sniffer like Net X-ray or Network Monitor on a Hub Network it is possible to capture traffic that is not destined for your PC. Your network card needs to have a driver that support promiscuous mode.

Hubs are very cheap today. Most of the Hubs available come in 8/16/24/48 ports depending on the supplier. Hubs are also capable of 100Mbps Half-Duplex transmission.

Data Link Layer

The Data-Link layer describes connectivity between layer two devices. The unit of information at the Data-Link layer is frames. Data-Link layer devices provide a boundary for Collision Domains. A network sniffer is only able to see unicast traffic on the same collision domain as the sniffer. Broadcasts are still sent down all Ethernet ports.

Data Link - Frames
Bridge
Switch

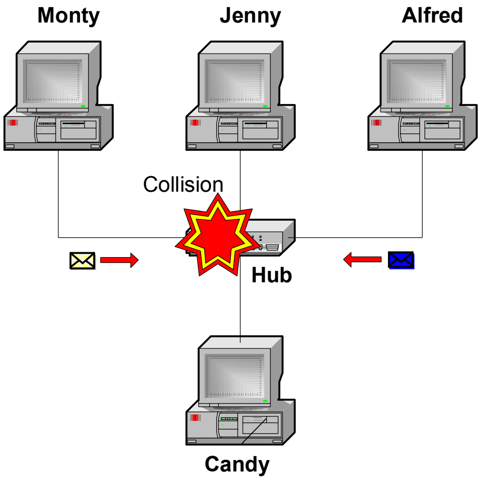
Data-Link Technologies

The two devices that operate at the Data-Link layer are Bridges and Switches. The following are technologies that work at the Data-Link Layer:

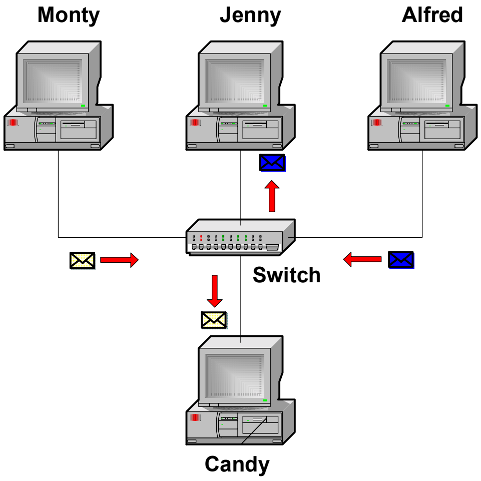
- Ethernet (Point to Multi-Point)
- HDLC
- Frame Relay
- ATM
- PPP
- SLIP
- Token Ring
- FDDI

Collision Domains

Bridges are typically used to segment collision domains. A collision domain describes the area where it is possible to have an Ethernet collision. When collisions become excessive it can dramatically reduce network performance.



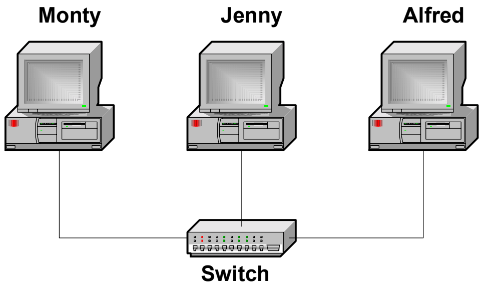
In the diagram Monty is sending a frame to Candy and Alfred is sending a frame to Jenny at the same time. The result is a collision. The network cards will detect the collision using CSMA/CD. The collision registers as a higher voltage on the wire. Once the collision is detected the network cards will each wait for a random amount of time before re-attempting transmission of the frame.



In this diagram we have replaced the Hub with a switch. A switch provides Micro-segmentation of collision domains. Each port on the switch is its own Collision domain (Whereas every port on a hub is a member of the same collision domain). With the switch in place (and assuming that the switch knows the MAC addresses of all connected stations) Monty is able to send the frame to Candy and Alfred is able to send the frame to Jenny at the same time. If the switch is running in Half-Duplex mode it is still possible to have a collision between the switch and a workstation. For this to occur the workstation would have to send a frame at the same time as the switch over that link. However if we use Full-Duplex capable switches and network cards we can also eliminate collisions on individual links. In Full-Duplex Ethernet the Transmit and Receive wires are crossed so that the transmit sends directly to the receive at the other end.

MAC Address Table

In the Ethernet world Bridges and switches make a decision about how to direct a frame based on the MAC address of the target machine. When the device is first switched on the MAC address table on the device is empty.



The process works as follows:

1. Monty sends a frame to Jenny.
2. The switch does not know the MAC address of Monty's PC so it sends the frame to both Jenny and Alfred. The frame sent from Monty has the MAC address of Monty's network card. This is added to the MAC table on the switch.
3. Jenny replies to Monty. The frame this time goes only to Monty's PC as the switch knows the MAC address of Monty's PC. The switch is now also able to add the MAC address of Jenny's Workstation to the MAC table.
4. Monty now wants to send a message to Alfred. The switch still does not know the MAC address of Alfred's PC so the frame is sent to all connected workstations.
5. Alfred replies to Monty. The switch now has all the MAC addresses of workstations on the network. All traffic is now sent directly to the target station.

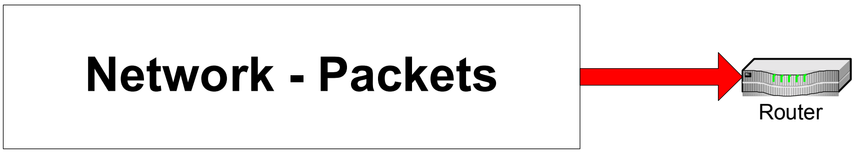
Port Level Security

Many switches have a feature called port level security. Port level security ensures that only the correct PC can be connected to a switch port. The switch learns each MAC address that is connected to each port. The switch administrator can freeze the switch learning at a given point. Each port at this time will only allow a certain MAC address to connect to it. If another MAC address is detected on the port the port will be disabled.

This is an excellent feature in a high security environment. Once learning is frozen no new machines can be connected to the network. There are two great results coming from this. One: A person trying to get an unauthorised machine on the network will not be able to. Two: The administration group are the only ones who can move equipment around the network.

Network Layer

The network layer is responsible for logical addressing and routing through an internetwork.



Routers

Routers are the devices that reside at layer 3 within the OSI model. Routers are the real workhorses of large networks. A router has two reasons for being, One: Routers learn about paths to other networks. Two: Routers take traffic from one network and place it on another.

Segments Broadcast Domains

Routers perform segmentation of Broadcast domains. No router will pass a broadcast packet by default.

Backbone of the Internet

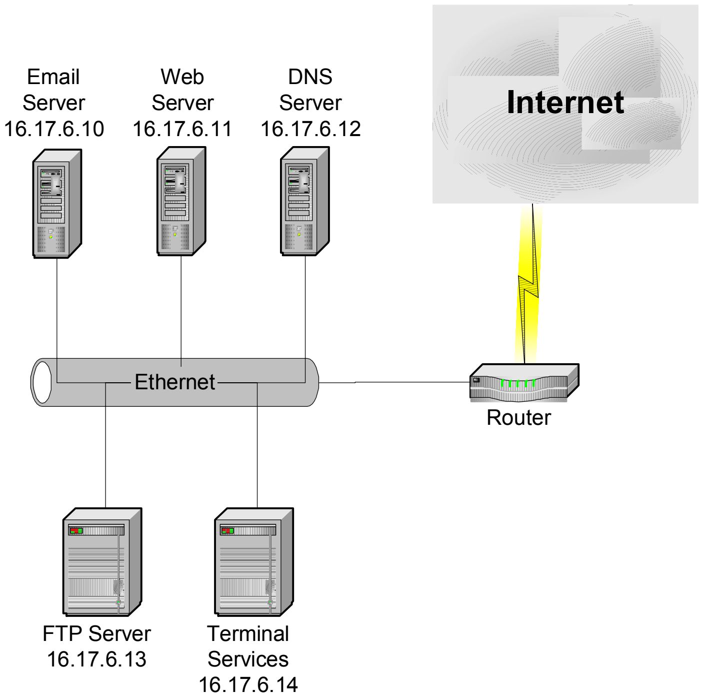
Routers are the backbone of the Internet. The Internet is an incredibly complex Internetwork of changing networks. In order for traffic to be able to get to hosts in a changing environment the routers need to be able to learn when certain networks are up or down.

Logical Addressing

The network layer is responsible for logical host addressing.

Packet Filtering

The ability to filter packets is vital to network security. Packet filtering works by denying certain types of traffic through the router.



The organisation in the diagram is offering services to the Internet. They have Email, Web, DNS, FTP and Terminal Services. We can use packet filtering on the router to limit the access that users connecting through the Internet are allowed.

By default all traffic entering the router from the Internet will be denied. This is great for security but it doesn't allow Internet users to access the resources we are making available. To enable users to access these services we need to open TCP ports 23 (Telnet), 25 (SMTP), 80 (HTTP), 21 (FTP) and UDP port 53 (DNS). We can tie this down by only allowing certain types of traffic to hit specific IP addresses. For example :

- Traffic on TCP port 23 will only be able to hit server 16.17.6.14.
- Traffic on TCP port 25 will only be able to hit server 16.17.6.10
- Traffic on TCP port 80 will only be able to hit server 16.17.6.11
- Traffic on TCP port 53 will only be able to hit server 16.17.6.12
- Traffic on TCP port 21 will only be able to hit server 16.17.6.13

These access filters will allow users from the outside to use the services offered on this network. However we also need to enable Internet traffic for internal users to enter the router. At the moment internal users will not be able to access any web resources. To overcome this we need to add a rule that will allow any packets that belong to established TCP sessions to enter the network. This will only permit TCP sessions that have been initiated from the Internal network.

IP Tunnelling / Virtual Private Networks

VPNs allow secure communication through insecure media such as the Internet. Microsoft has two implementations of VPNs, Point-to-Point Tunnelling Protocol (PPTP) used in Windows NT 4.0 implementations and Layer 2 Forwarding (L2F) used in Windows 2000. Both of these methods work by taking the packet, encrypting it and encapsulating it inside a PPP frame. This frame is then encapsulated inside another TCP/IP packet and treated normally by the intermediate network.

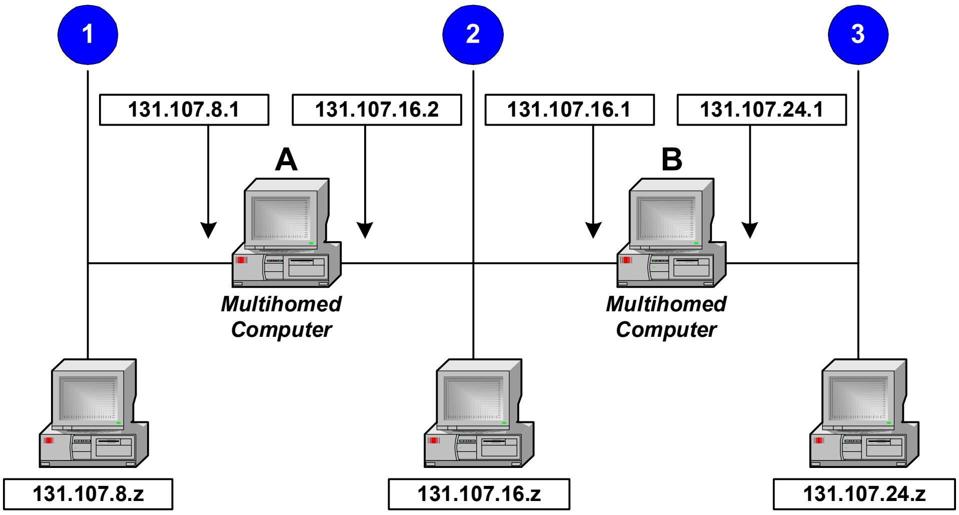
IP Routing

What is IP Routing?

Routing is the process of choosing a path over which to send packets. Routing occurs at a TCP/IP host when it sends IP packets and routing occurs at an IP router. A router is a device that forwards a packet from one physical network to another. By default a router can only send packets to network that is has a configured interface for. Routers are also commonly referred to as gateways.

Static IP Routing

Static routing is a function of IP. Static routers require that routing tables are built and updated manually. If a route changes, static routers do not inform each other of the change.



By default a static router can communicate only with networks to which it has a configured interface. As illustrated in the graphic:

- Computer A has only local connections to networks 1 and 2. As a result, hosts on network 1 can communicate with hosts on network 2, but cannot communicate with hosts on network 3.
- Computer B has only local connections to networks 2 and 3. Hosts on network 3 can communicate with hosts on network 2, but cannot communicate with hosts on network 1.

To route IP packets to other networks, each static router must be configured with one of the following:

- An entry in each router's routing table for each network in the Internetwork
- A default gateway address of another router's local interface.

Dynamic IP Routing

- With dynamic routing, routers automatically exchange routes to known networks with each other. If a route changes, routing protocols automatically update a routers routing table and inform other routers on the Internetwork of the change. Dynamic routing is typically implemented on large Internetworks because minimal configuration is required by a network administrator. Dynamic routing requires a routing protocol.

Routing Protocols and Routed Protocols

Routing protocols are the protocols that handle how routers exchange information with each other. Routed protocols are the protocols that carry user data.

Routing Protocols:

- RIP (Routing Information Protocol)
- OSPF (Open Shortest Path First)
- BGP (Border Gateway Protocol)

- IGRP (Interior Gateway Routing Protocol)
- EIGRP (Enhanced Interior Gateway Routing Protocol)

Routed Protocols:

- TCP/IP
- IPX/SPX
- Appletalk

Distance Vector and Link State Routing

There are two main classifications of routing protocols (Distance Vector and Link State).

Distance Vector

The Distance Vector routing algorithm is sometimes referred to as Bellman- Ford. In Distance Vector routing, each entity keeps a routing database with one entry for every possible destination in the system.

The Distance Vector routing protocol specifies that each router advertises to its adjacent neighbors its routing table. For each network destination, the receiving routers pick the neighbor advertising the lowest cost, and then add this entry to its routing table.

The problem with Distance Vector routing is slow convergence. In Distance Vector routing, when a change is made, the changes must be propagated to each router. This propagation causes all routing tables affected by this change to be recalculated. Distance Vector routing can be very slow converging after a topical change.

Link State

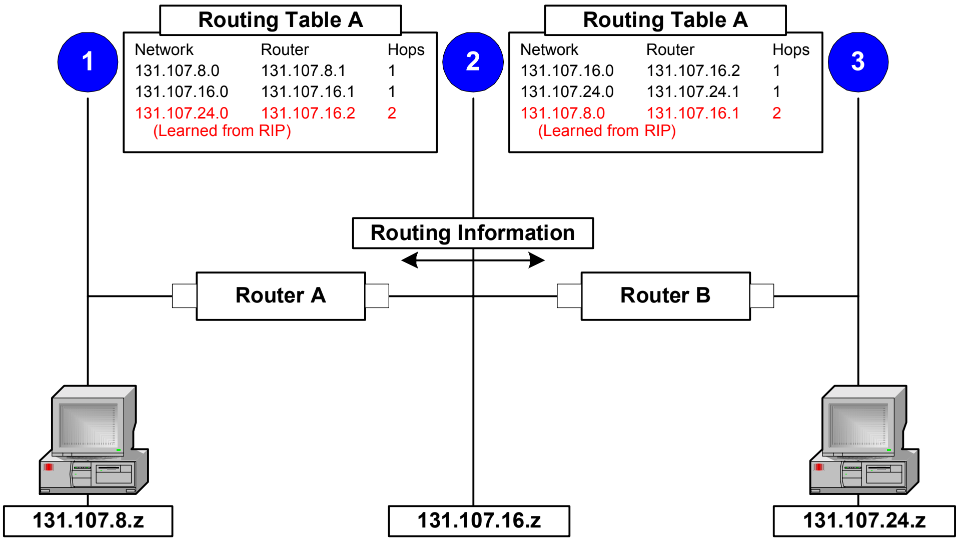
Routers that use a link state routing protocol maintain a database of their individual autonomous system (AS) topology. An autonomous system is a group or collection of networks under common administration using the same routing protocol, and is sometimes called a routing domain. Autonomous systems are commonly divided into areas tied together by a backbone area. Each router in an autonomous system has an identical link state database (LSB). The link state database is composed of each router's local state.

Routers distribute their local state by flooding the autonomous system with link state advertisements (sometimes referred to as link state packets). Link state advertisements (LSAs) are special packets which contain information about neighbors and route cost. These LSAs can also contain routing information gathered by other routing protocols and static routes.

All routers in an autonomous system run the same routing algorithm. From the link state database, each router builds a tree of shortest paths with itself as the root. The tree contains the route to each destination in the autonomous system. Common link state routing protocols are open shortest path first (OSPF) and intermediate system-to-intermediate system interdomain routing protocol (IS-IS).

RIP (Routing Information Protocol)

Routing Information Protocol (RIP) is the protocol we are going to configure on our NT routers. RIP-enabled routers exchange the Network IDs of the networks the router can reach and the distance to these networks. RIP uses a hop count field, or metric, in its routing table to indicate the distance to a network ID. The hop count is the number of routers that must be traversed to reach the desired network. The maximum hop count for RIP is 15. Anything of 16 hops or greater is marked as unreachable.



Routing Table A

Network	Router	Hops
131.107.8.0	131.107.8.1	1
131.107.16.0	131.107.16.1	1
131.107.24.0	131.107.16.2	2

(Learned from RIP)

Routing Table A

Network	Router	Hops
131.107.16.0	131.107.16.2	1
131.107.24.0	131.107.24.1	1
131.107.8.0	131.107.16.1	2

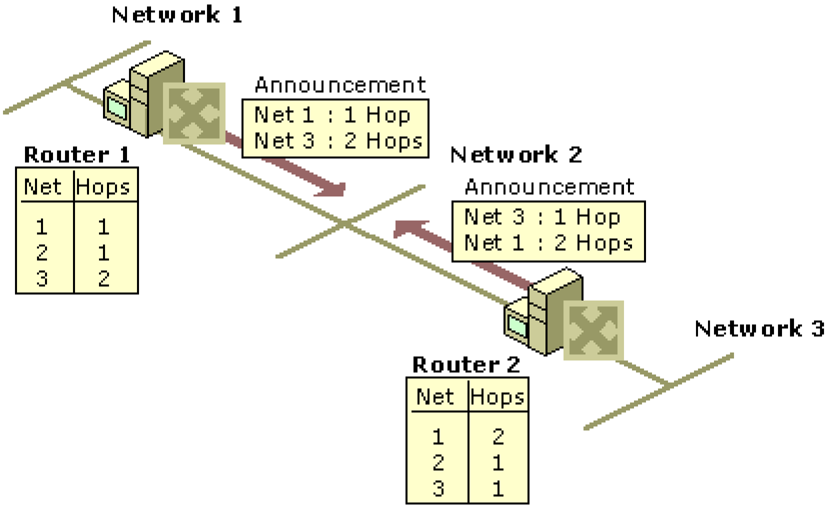
(Learned from RIP)

In the graphic three subnets are connected by two computers running Windows NT server software with RIP routing enabled. Each router is configured with the default update interval; therefore, every 30 seconds each router broadcasts its routing table. The receiving router will then add the new routes to its routing table. All RIP traffic is broadcast to UDP port 520.

Count to Infinity Problem

The classic distance vector convergence problem is known as the count-to-infinity problem and is a direct result of the asynchronous announcement scheme. When RIP for IP routers add routes to their routing table, based on routes advertised by other routers, they keep only the best route in the routing table and they update a lower cost route with a higher cost route only if is being announced by the same source as the current lower cost route. In certain situations, as illustrated in Figures 1 through 5, this causes the count-to-infinity problem.

Assume that the internetwork in Figure 1 has converged. For simplicity, the announcements sent by Router 1 on Network 1 and Router 2 on Network 3 are not included.



Router 1

Net	Hops
1	1
2	1
3	2

Router 2

Net	Hops
1	2
2	1
3	1

Figure 1 Converged Internetwork

Now assume that the link from Router 2 to Network 3 fails and is sensed by Router 2. As shown in Figure 2, Router 2 changes the hop count for the route to Network 3 to indicate that it is unreachable, an infinite distance away. For RIP for IP, infinity is 16.

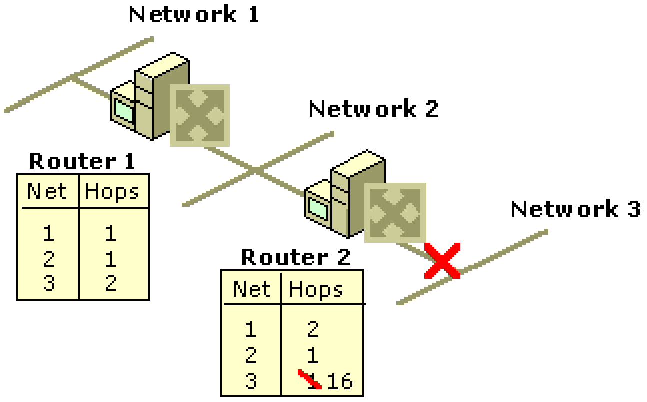


Figure 2 Link to Network 3 Fails

However, before Router 2 can advertise the new hop count to Network 3 in a scheduled announcement, it receives an announcement from Router 1. The Router 1 announcement contains a route to Network 3 which is 2 hops away. Because 2 hops away is a better route than 16 hops, Router 2 updates its routing table entry for Network 3, changing it from 16 hops to 3 hops, as shown in Figure 3.

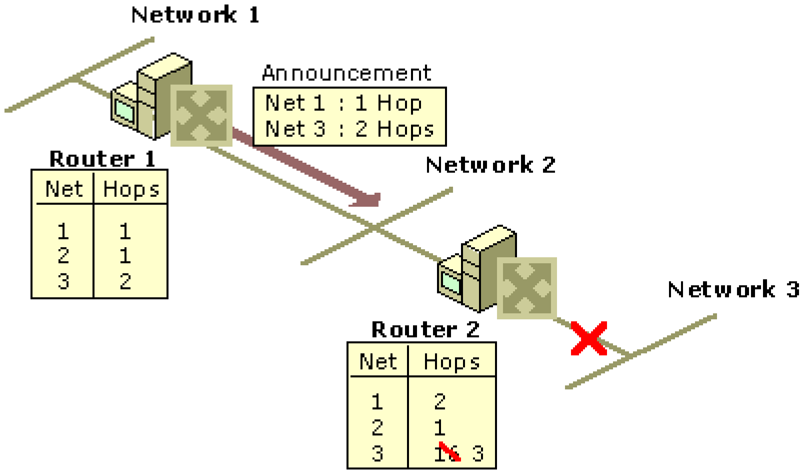


Figure 3 Router 2 After Receiving Announcement From Router 1

When Router 2 announces its new routes, Router 1 notes that Network 3 is available 3 hops away through Router 2. Because the route to Network 3 on Router 1 was originally learned from Router 2, Router 1 updates its route to Network 3 to 4 hops. (See Figure 4.)

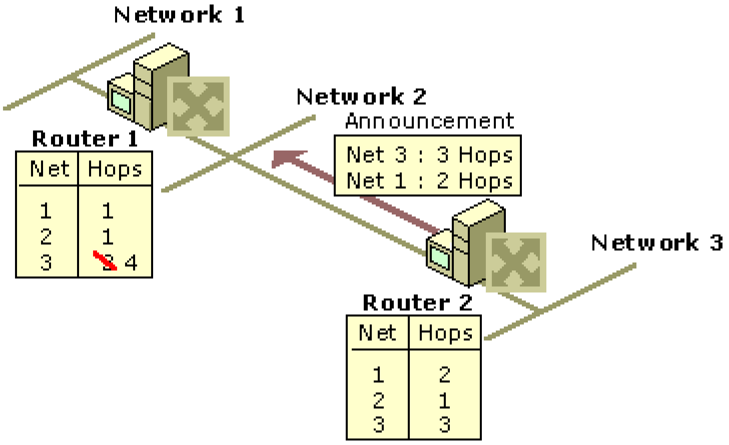
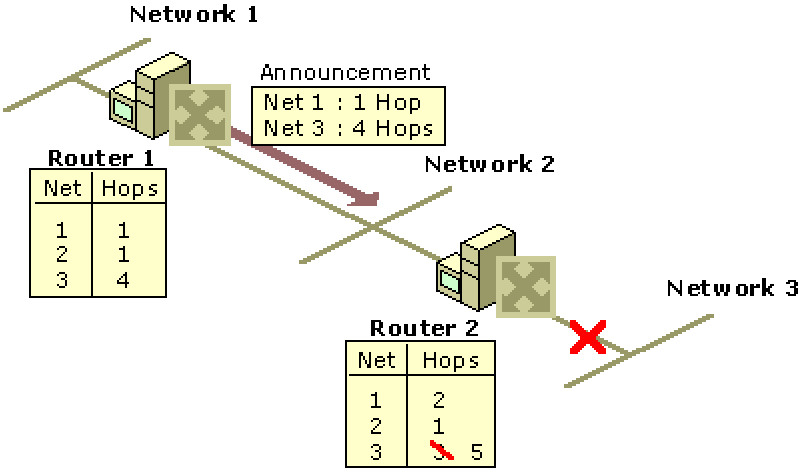


Figure 4 Router 1 After Receiving Announcement From Router 2

When Router 1 announces its new routes, Router 2 notes that Network 3 is available 4 hops away through Router 1. Because the route to Network 3 on Router 2 was originally learned from Router 1, Router 2 updates its route to Network 3 to 5 hops. (See Figure 5.)



Router 1

Net	Hops
1	1
2	1
3	4

Router 2

Net	Hops
1	2
2	1
3	~~3~~ 5

Figure 5 Router 2 After Receiving Another Announcement from Router 1

The two routers continue to announce routes to Network 3 with higher and higher hop counts until infinity (16) is reached. Then, Network 3 is considered unreachable and the route to Network 3 is eventually timed out of the routing table. This is known as the count-to-infinity problem.

The count-to-infinity problem is one of the reasons why the maximum hop count of RIP for IP internetworks is set to 15 (16 for unreachable). Higher maximum hop count values would make the convergence time longer when count-to-infinity occurs. Also note that during the count-to-infinity in the previous example, the route from Router 1 to Network 3 is through Router 2. The route from Router 2 to Network 3 is through Router 1. A routing loop exists between Router 1 and Router 2 for Network 3 for the duration of the count-to-infinity problem.

Split Horizon

Split horizon helps reduce convergence time by not allowing routers to advertise networks in the direction from which those networks were learned. The only information sent in RIP announcements are for those networks that are beyond the neighboring router in the opposite direction. Networks learned from the neighboring router are not included.

Split horizon eliminates count-to-infinity and routing loops during convergence in single-path internetworks and reduces the chances of count-to-infinity in multi-path internetworks. Figure 6 illustrates how split horizon keeps the RIP router from advertising routes in the direction from which they were learned.

Poison Reverse

Split horizon with poison reverse differs from simple split horizon because it announces all networks. However, those networks learned in a given direction are announced with a hop count of 16, indicating that the network is unreachable. In a single-path internetwork, split horizon with poison reverse has no benefit beyond split horizon. However, in a multi-path internetwork, split horizon with poison reverse greatly reduces count-to-infinity and routing loops. Count-to-infinity can still occur in a multi-path internetwork because routes to networks can be learned from multiple sources.

In Figure 7, split horizon with poison reverse advertises learned routes as unreachable in the direction from which they are learned. Split horizon with poison reverse does have the disadvantage of additional RIP message overhead because all networks are advertised.

Encapsulation and OSI

In the following process we will follow the path of our web page request as it travels from our client to the target web server and back again. For the purpose of this exercise we will assume that the TCP three way hand shake has happened and all name/address resolution has occurred. To simplify the description of the encapsulation process we have a simple network as shown in Figure 1

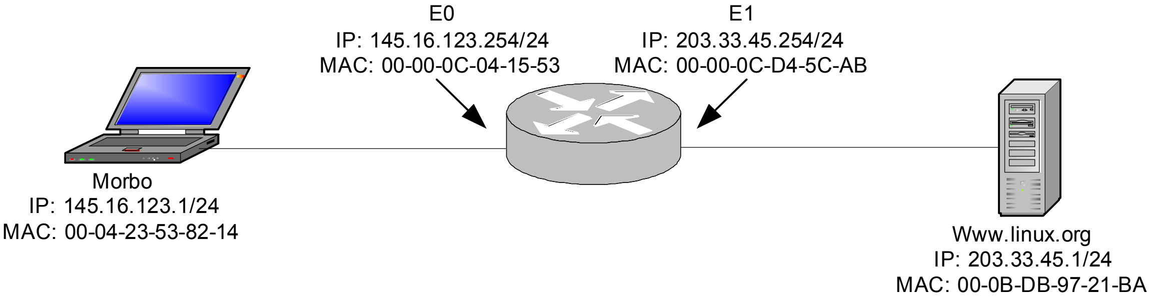


Figure 1 - Simple Network

The network has two Ethernet segments and three hosts. The laptop "Morbo" is running a web browser and is used to access web pages on the server www.linux.org which is located on the second Ethernet segment. The two segments are different IP subnets separated by a router.

Methodology

The communication between the client and the server will be divided into six stages. The six stages are detailed within this document.

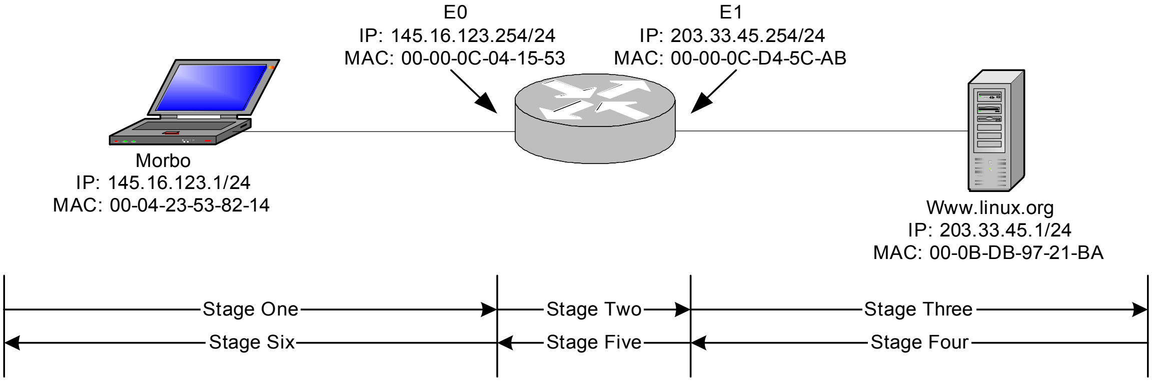


Figure 2 - Methodology

Stage One – From Morbo to the Router

1. User types the URL into the web browser.

Note: At this point a DNS lookup to find the target IP address would occur. We are not covering this process in this exercise. Just assume that the lookup worked properly☺.

2. HTTP Get request is passed from the upper layers to the Transport layer. The Data is encapsulated into a TCP Segment. The TCP header is added.

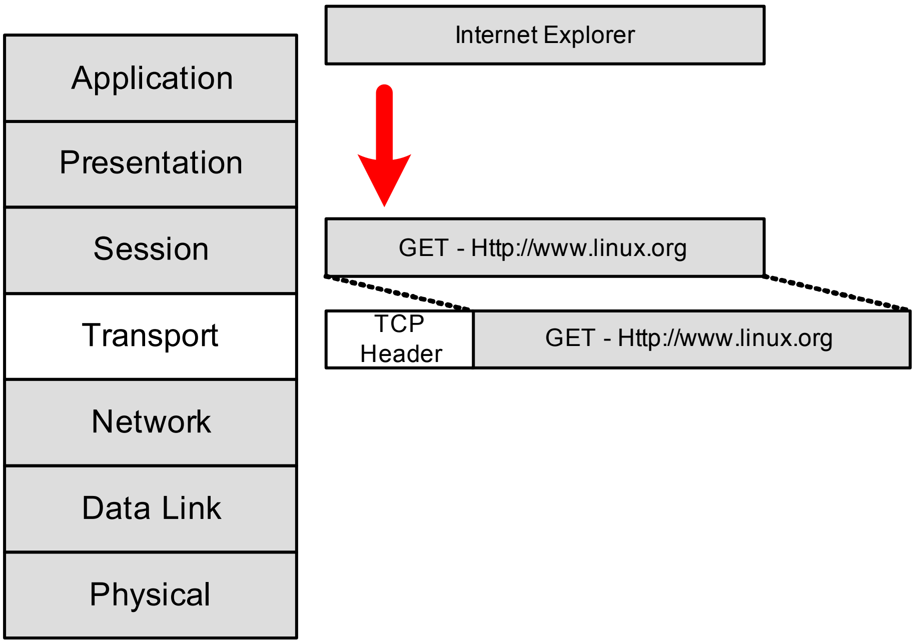


Figure 3 - Add TCP Header

Figure 3 shows the TCP header being added to the data as it moves from the data layers to the transport layer. For detailed information about the makeup of the TCP header refer to RFC793.

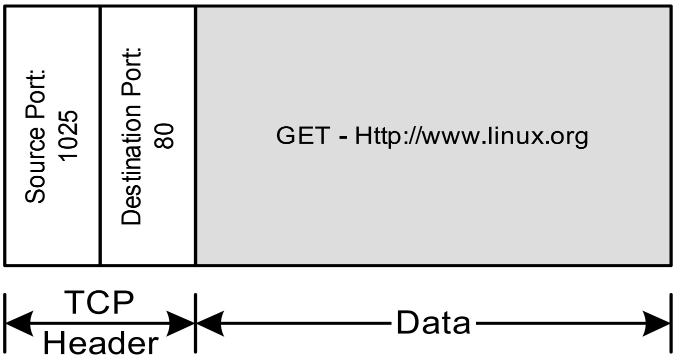


Figure 4 - TCP Segment

Figure 4 shows the two addresses added at the TCP layer. These addresses are the source and destination ports. The destination port is used to tell the receiving host which process to pass the data to. As this is a HTTP session, port 80 is the destination. The source port is used by the client to keep track of different TCP sessions. These source ports are also known as “Ephemeral” ports. On a windows host the ephemeral or source port will always be between 1024 and 5000.

3. The TCP Segment is passed down to the Network layer. The TCP Segment is now encapsulated into an IP packet. The IP header is now added as shown in Figure 5.

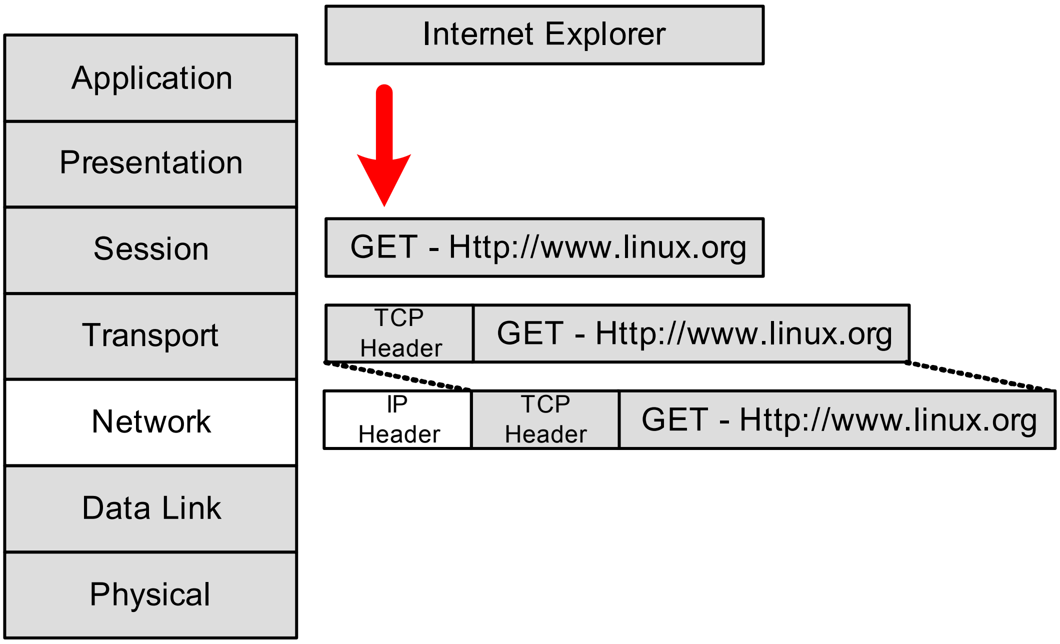


Figure 5 - Add IP Header

The IP Header contains the source and destination IP addresses for the packet. In this instance the source address is the client machine. The destination IP address is the address of the web server.

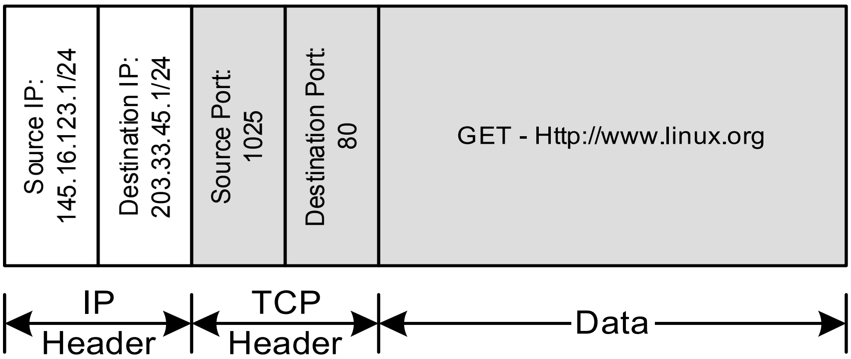


Figure 6 - IP Packet

Figure 6 shows the IP addresses that were added at this time. Take note here of the IP addresses in the packet. The source is the client machine and the destination is the target web server. Remember the role of the network layer in OSI model. It is responsible for finding the path through the network. To do this there needs to be a method of maintaining the address of the final destination of the data. This address is the IP address.

4. The IP Packet is encapsulated inside an Ethernet Frame as the IP Packet is passed to the Datalink layer. Figure 7 shows the Ethernet headers and footers being added to our IP packet. Two parts are added when the packet is encapsulated inside an Ethernet frame. The Ethernet header contains the source and destination MAC addresses. The FCS (Frame Check Sequence) contains error checking information so the integrity of the frame can be verified by the destination. Once again there is additional information added to the frame that is not shown here. For detail on this visit www.ietf.org.

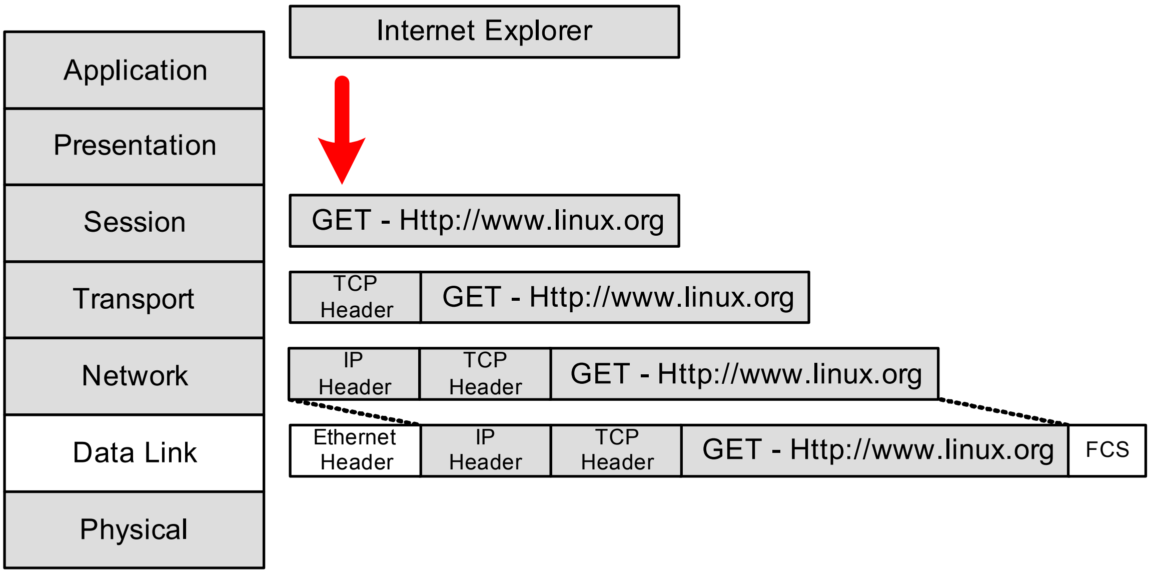


Figure 7 – Add Frame Header and FCS

MAC addresses are added here. The source address is the client and the destination address is the local interface on the router.

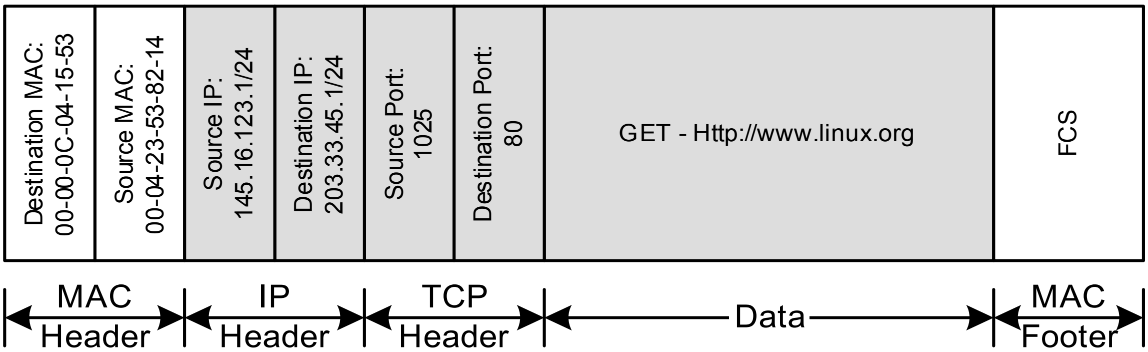


Figure 8 - Ethernet Frame

Figure 8 shows the MAC addresses added at this time. The source MAC address is the web client machine. The destination MAC address is not the address of the web server. It is the MAC address of the router. The data link layer is responsible for the next hop in the chain. This layer knows nothing of the larger network. It only deals with this point to the next.

Note: A correctly configured client will contain the IP address for the client's default gateway. If a target network is not found in the clients routing table the traffic will be sent to the default gateway. It is the gateways responsibility to then get the traffic to the appropriate network. In order to get the MAC address of the default gateway the ARP (Address Resolution Protocol) is used. For the purpose of this exercise we will assume that the ARP lookup was successful.

5. The Ethernet frame is now converted to bits and transmitted across the wire.

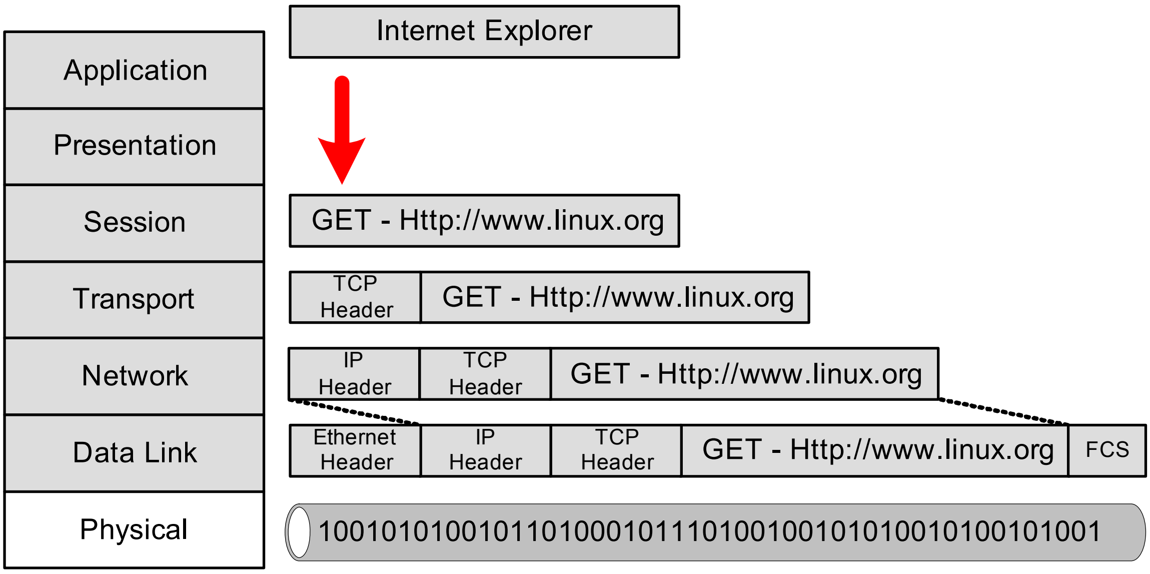


Figure 9 – Bits

Stage 2 – In the Router

1. Having enjoyed their brief journey across the Ethernet cable the bits are now grabbed and reassembled by the router on interface E0.

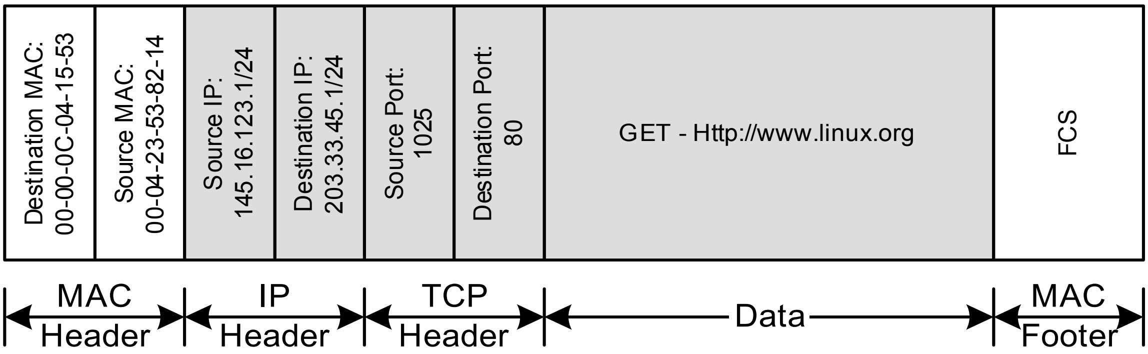


Figure 10 - Ethernet Frame

The first thing the router does with the frame is to check the destination address. If the destination address is not the routers address it will disregard the frame. In this instance the destination MAC address is the address of the router. The frame is now passed higher up to the network layer.

2. The router now removed the Ethernet header and footer. This leaves the IP packet shown in Figure 11.

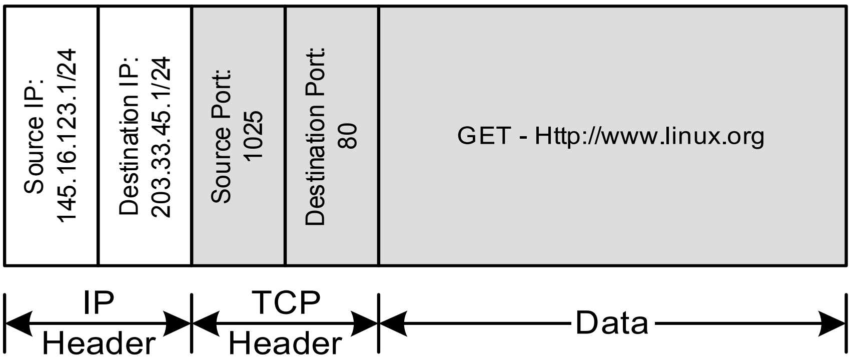


Figure 11 - IP Packet

The router now analyses the destination IP address. If the destination IP address is the address of the router the packet will be passed to the transport layer. In this instance the destination IP address is the address of the web server not the router. The router now needs to do some extra work.

Note: Routers have only two main reasons for existing. The first function routers serve is to discover other networks and how to get to them. The second function is to direct packets to other networks as required. Detailed information on routers is outside the scope of this document however.

3. The router now compares the destination address to its routing table. The router asks “Do I know how to get to that network?”. In this case the router is directly connected to that network so the answer is yes. The router now passes the packet to the interface that is connected to that network. No inspection of

the packet other than the IP information that the router uses to determine the destination of the packet.

4. The IP packet is now passed back down to the datalink layer.

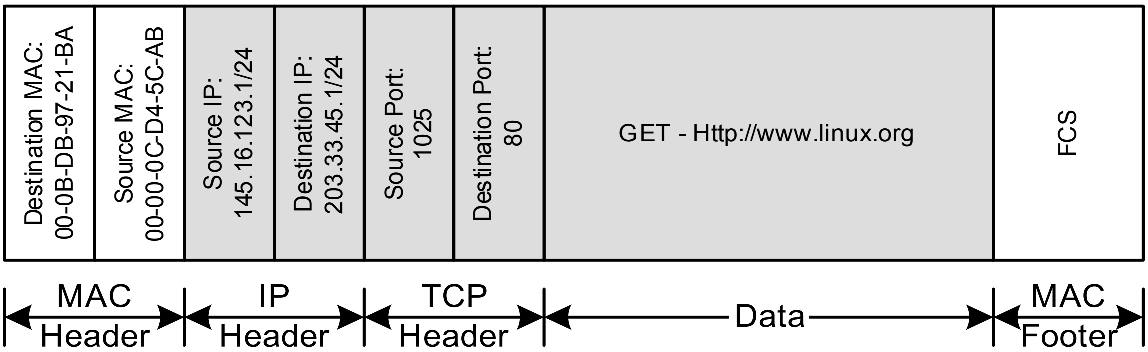

Destination MAC: 00-0B-DB-97-21-BA
Source MAC: 00-00-0C-D4-5C-AB
Source IP: 145.16.123.1/24
Destination IP: 203.33.45.1/24
Source Port: 1025
Destination Port: 80
GET - Http://www.linux.org
FCS
MAC Header
IP Header
TCP Header
Data
MAC Footer

Figure 12 - Ethernet Frame

New Ethernet headers and footers are now added. This time the MAC addresses added are the addresses of the router interface on the 203.33.45.0/24 network and the MAC address of the web server.

5. The frame is now converted to bits again and sent out interface E1 on the router.

Stage 3 – From the Router to the Web Server

1. The bits are collected from the wire and passed up to the data link layer.

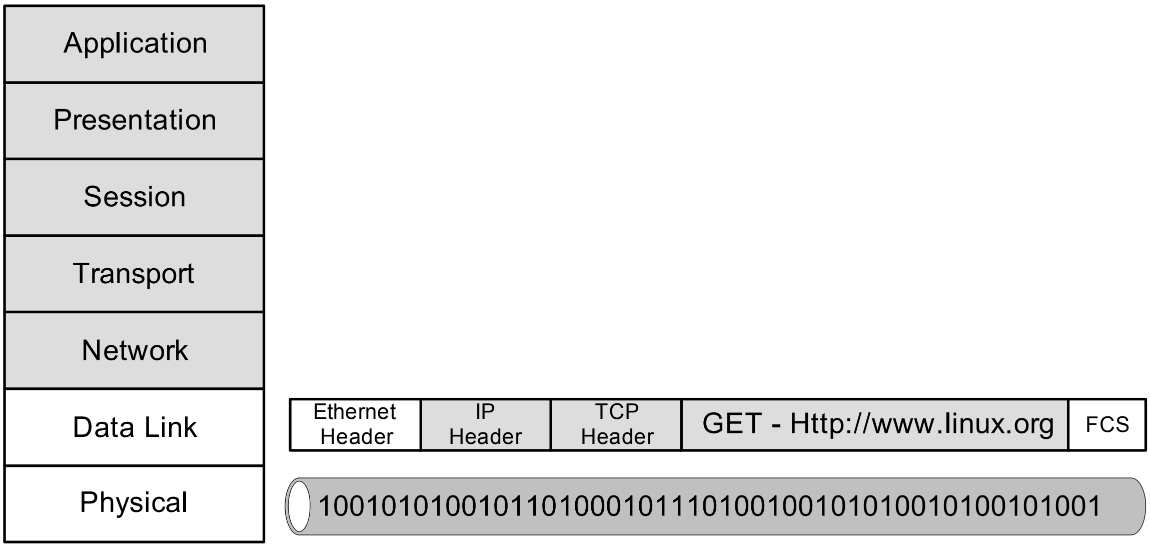


Figure 13 – Re-assemble the frame

The Ethernet addresses are checked. If the destination address is the address of the web server the frame will be passed up to the network layer.

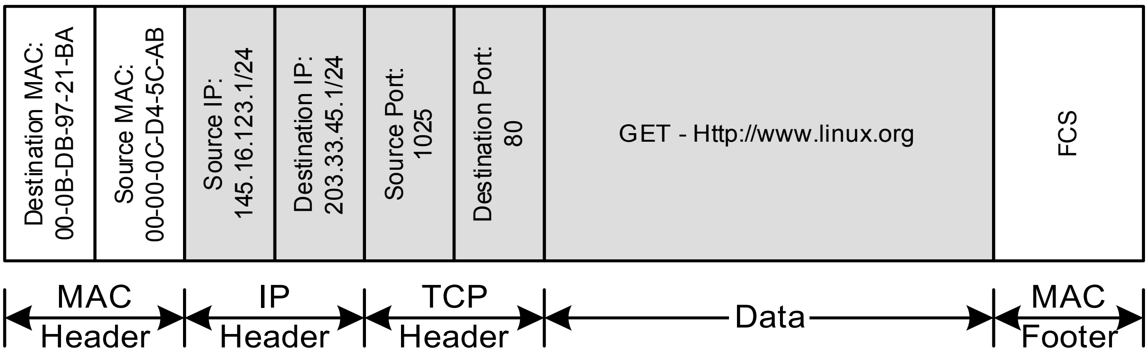

Destination MAC: 00-0B-DB-97-21-BA
Source MAC: 00-00-0C-D4-5C-AB
Source IP: 145.16.123.1/24
Destination IP: 203.33.45.1/24
Source Port: 1025
Destination Port: 80
GET - Http://www.linux.org
FCS
MAC Header
IP Header
TCP Header
Data
MAC Footer

Figure 14 - Ethernet Frame

As seen in Figure 14 the destination address is the MAC address of the web server.

2. The Ethernet frame is now passed to the network layer. The Ethernet headers and footers are now removed and the original IP packet is exposed as shown in Figure 15.

Note: As the size of an IP packet is a maximum of 64k and an Ethernet frame payload is a maximum of 1500 bytes there may need to be some re-assembly of the packet happen at this layer. This is outside of the scope of this exercise.

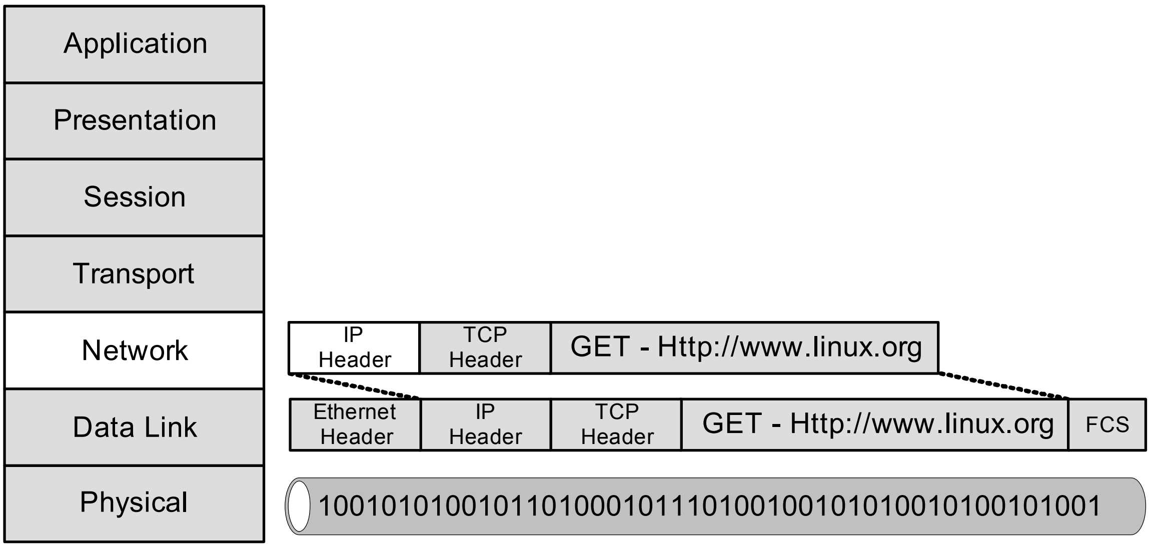


Figure 15 - Remove Ethernet Frame

The IP packet is now examined. The web server looks at the destination address and asks "Is this packet addressed to me? As seen in Figure 16 the destination address is the address of the web server.

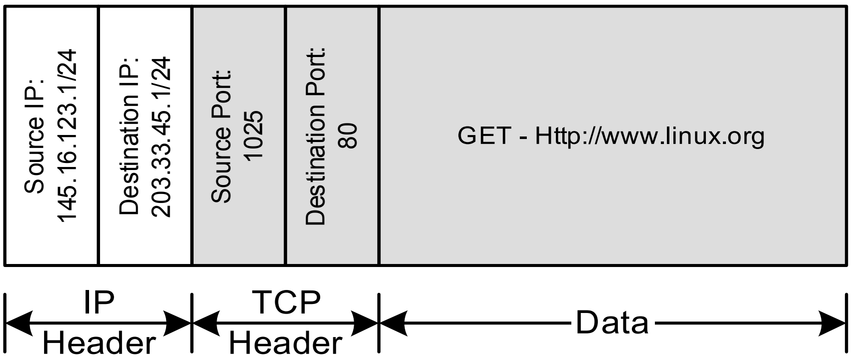

Source IP: 145.16.123.1/24
Destination IP: 203.33.45.1/24
Source Port: 1025
Destination Port: 80
GET - Http://www.linux.org
IP Header
TCP Header
Data

Figure 16 - IP Packet

3. The IP packet is now passed up to the transport layer. The IP header is removed as shown in Figure 18.

Note: As the size of an IP packet is a maximum of 64k and a TCP segment can be much larger, there may need to be some re-assembly of the segment at this layer. This is outside of the scope of this exercise. It is worth remembering though as some attacks take advantage of this need to reassemble the segment.

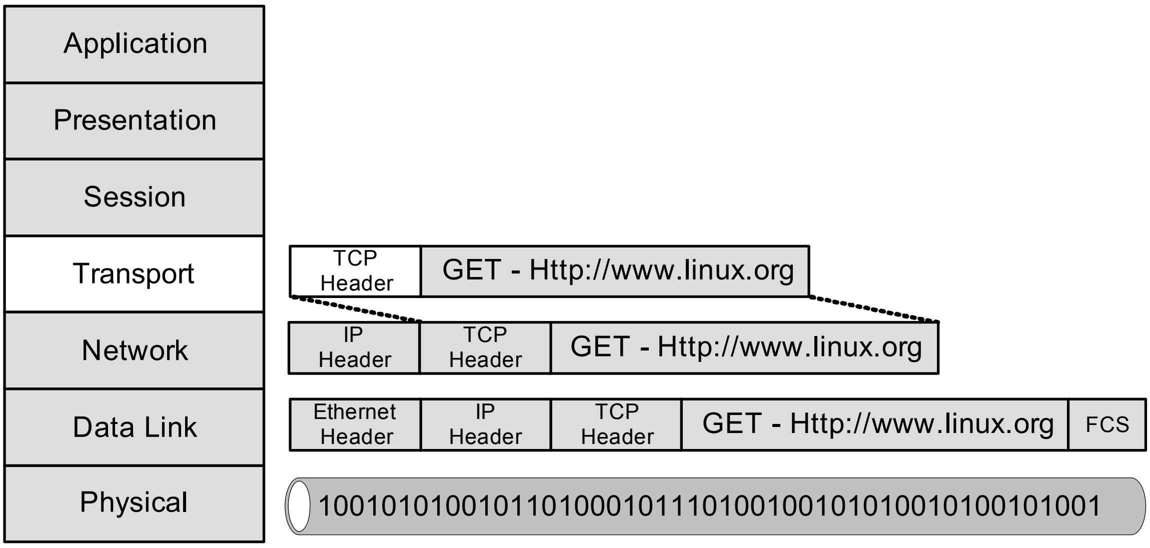


Figure 17 – Remove IP Header

This leaves the original TCP segment that was generated on the client machine. The web server now needs to determine which application or process to pass the data to.

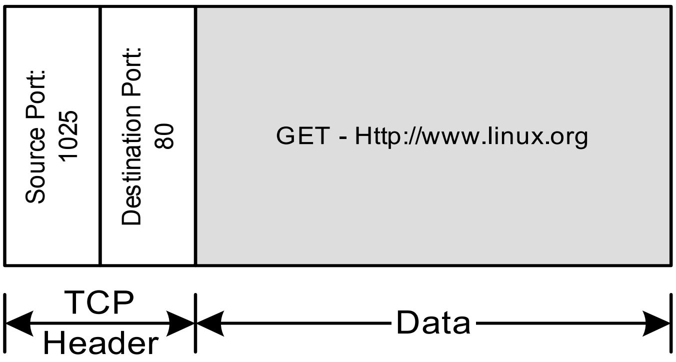


Figure 18 - TCP Segment

To determine which process is the intended recipient of the segment, the web server checks the destination port of the segment. In the segment shown in Figure 18 the destination port is 80. The server now knows to pass the traffic to the process that is listening on port 80.

Note: It is possible to configure services to listen on ports other than the ones reserved for a particular protocol. For the purpose of this exercise we will assume that all services are listening on their default ports. For more information on which protocols are assigned to which ports visit **www.iana.org/numbers.html**.

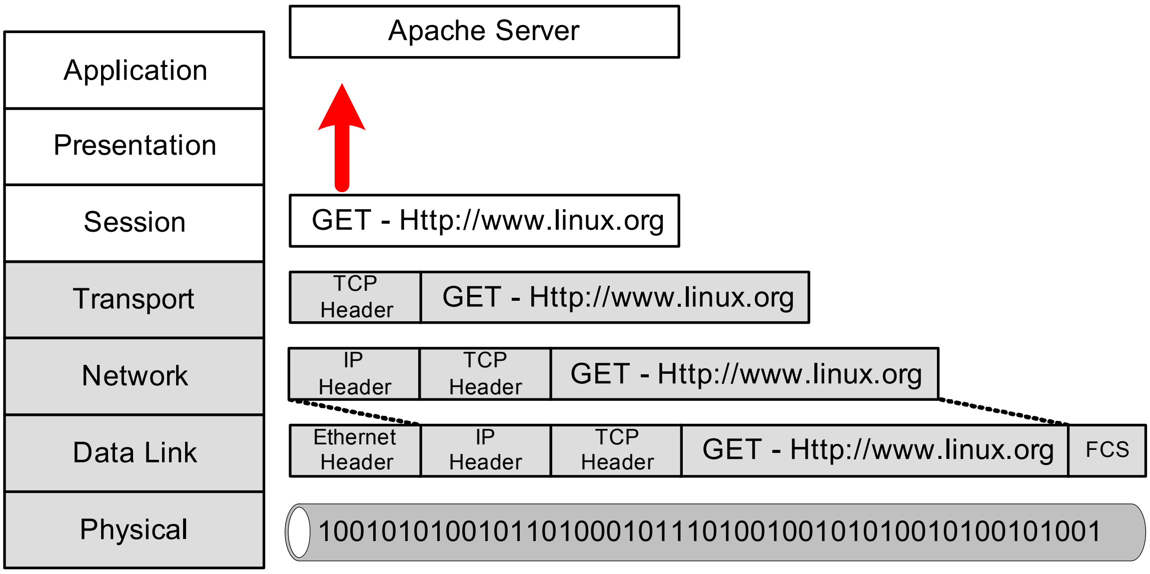


Figure 19 - Send GET request to the web process

4. The original web request is now passed to the process that is listening on Port 80. In this instance it is an apache web server process.

5. At this point the web server processes the request and prepares to respond.

Stage Four – From the web server to the router

1. The web server assembles the web page as per the client request and passes the data to the transport layer. A new TCP segment is generated.

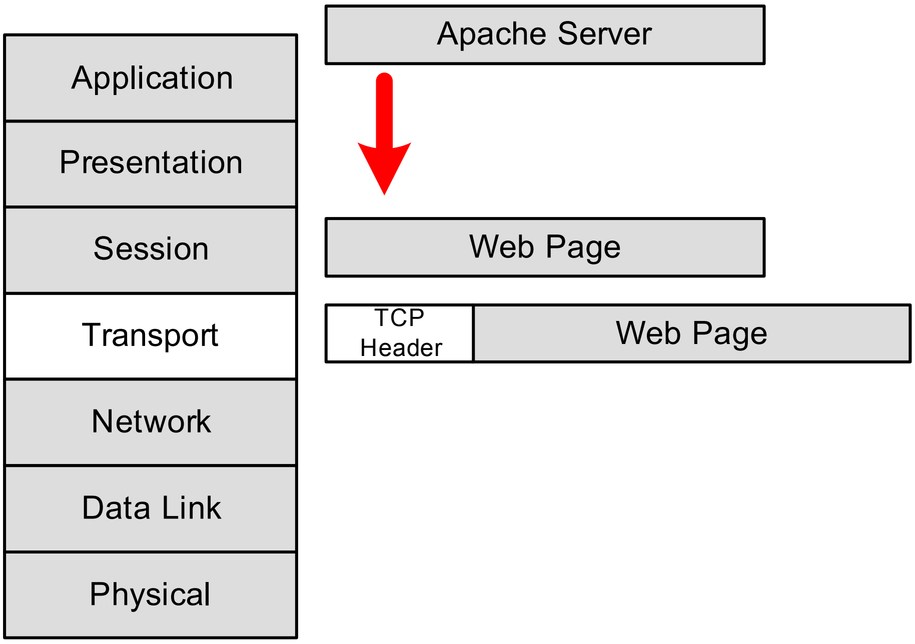


Figure 20 - Encapsulated Web Page

Figure 21 shows the new segment that was generated on the web server. The source port is now 80 and the destination port is 1025 (the ephemeral port used on the web client). This return segment is part of the same TCP session. That is why the port numbers are the same as the initial request.

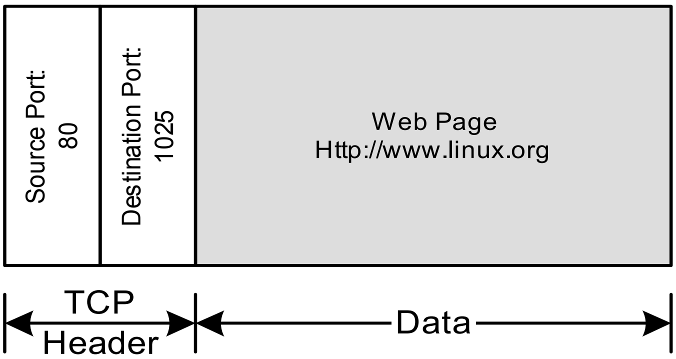


Figure 21 - TCP Segment

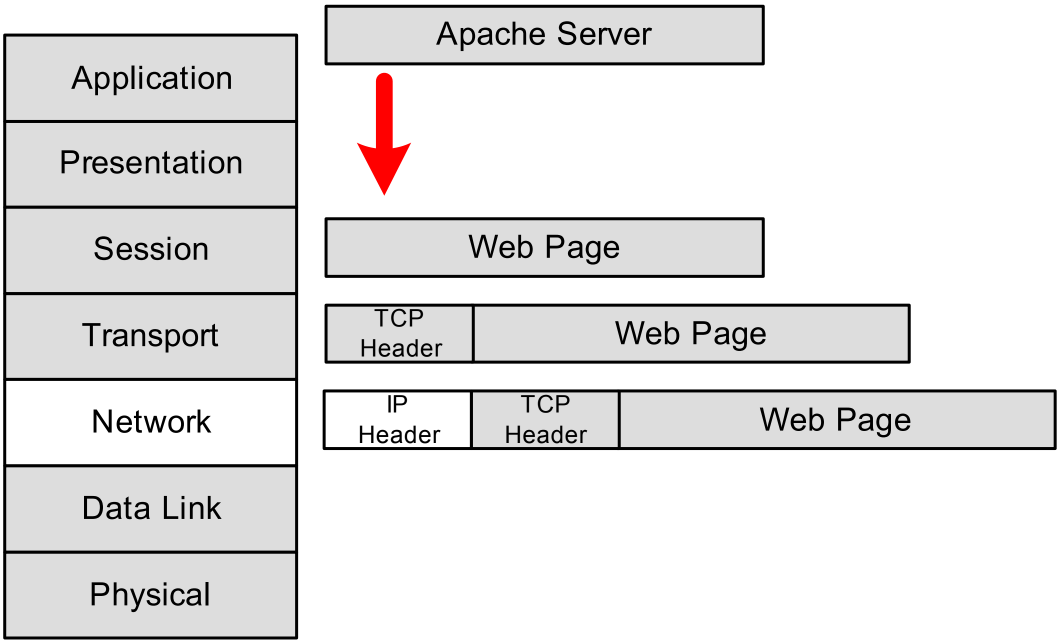


Figure 22 - New IP Packet

2. The TCP segment is now encapsulated inside an IP packet. As shown in Figure 23. The source address for this packet is the IP address of the web server as that is where this packet was generated. The destination is the web client machine.

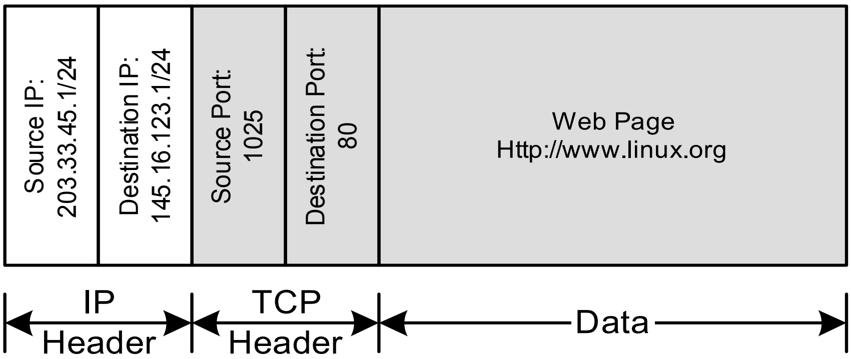


Figure 23 - IP Packet

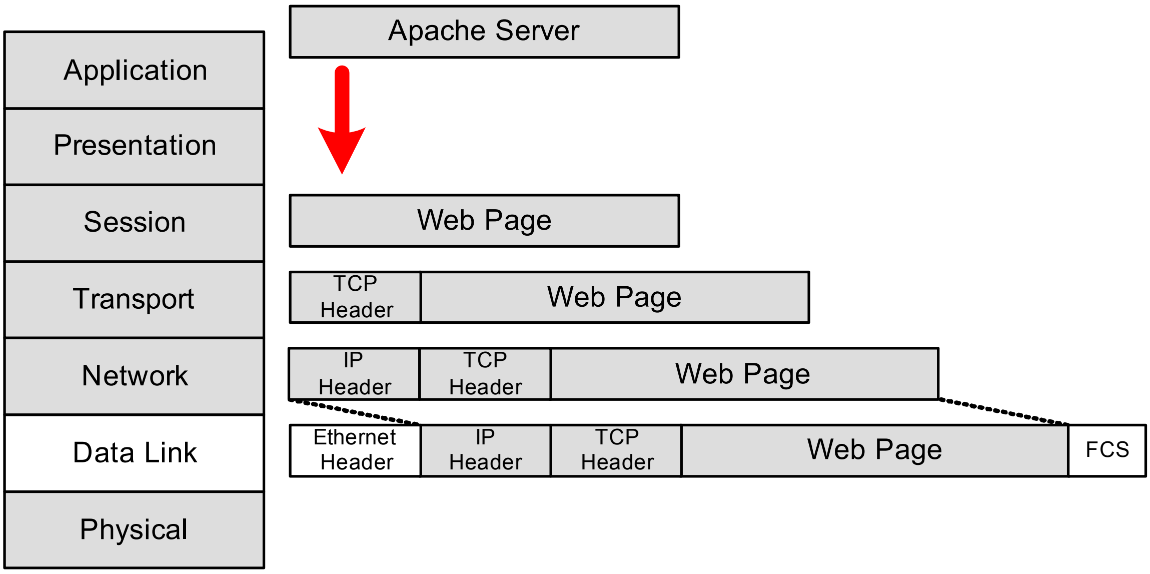


Figure 24 – New Ethernet Frame

3. The IP packet is now encapsulated inside an Ethernet frame. The source MAC address is the address of the web server. The destination MAC address is the address of the router interface E1.

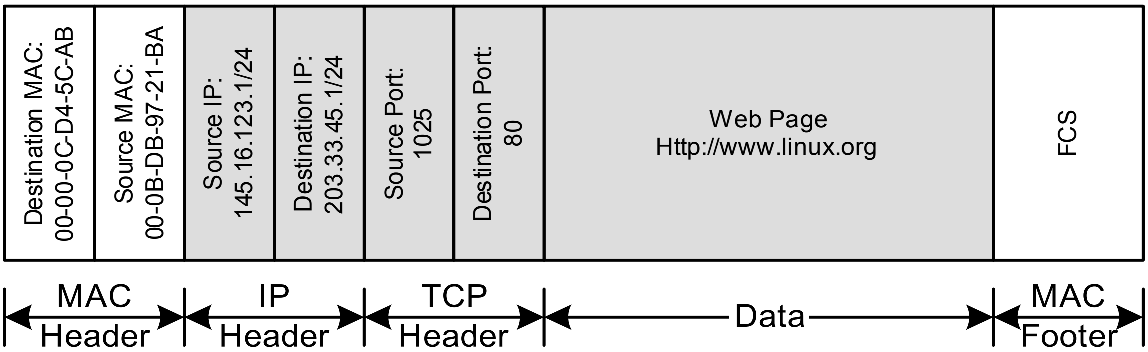

Destination MAC: 00-00-0C-D4-5C-AB
Source MAC: 00-0B-DB-97-21-BA
Source IP: 145.16.123.1/24
Destination IP: 203.33.45.1/24
Source Port: 1025
Destination Port: 80
Web Page Http://www.linux.org
FCS
MAC Header
IP Header
TCP Header
Data
MAC Footer

Figure 25 - Ethernet Frame

4. The Ethernet frame is now converted to bits and transmitted on the wire.

Stage 5 – Inside the router

I wont detail this step as it is identical to Step 2. The direction of traffic flow it the only difference here. For a refresher on how this worked re-read step 2 ☺.

Stage 6 – Web page is displayed on Morbo

1. The bits are received on the wire and assembled into an Ethernet frame.

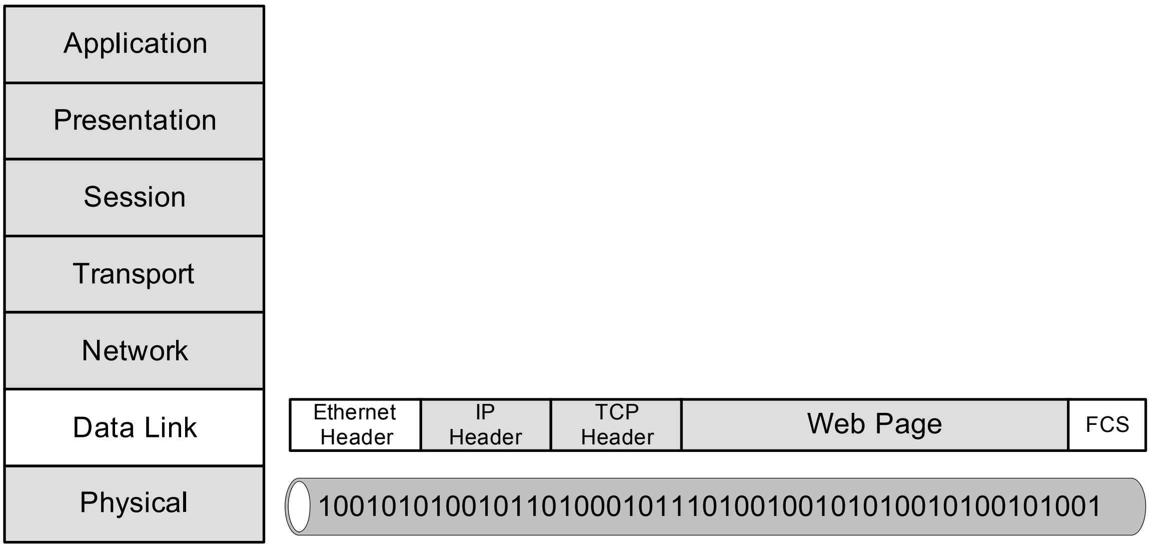


Figure 26 - Frame is assembled

2. The client checks the destination address to ensure that the frame is for it. In this case the address matches so the frame is accepted.

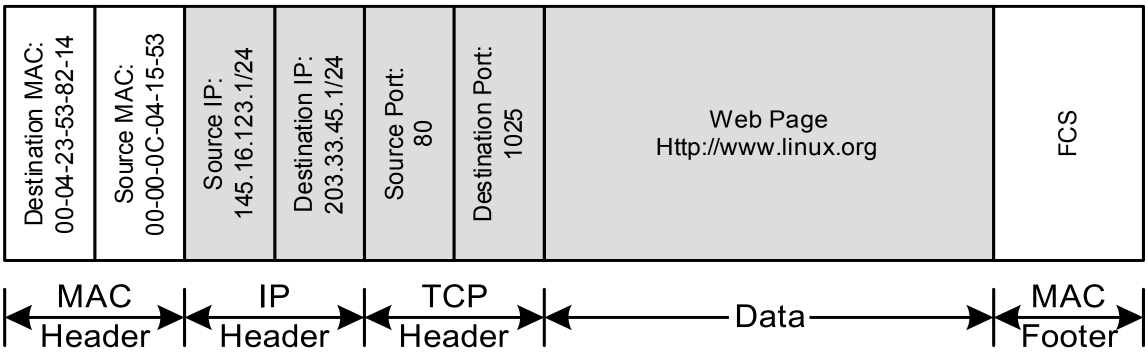

Destination MAC: 00-04-23-53-82-14
Source MAC: 00-00-0C-04-15-53
Source IP: 145.16.123.1/24
Destination IP: 203.33.45.1/24
Source Port: 80
Destination Port: 1025
Web Page
Http://www.linux.org
FCS
MAC Header
IP Header
TCP Header
Data
MAC Footer

Figure 27 - Ethernet Frame

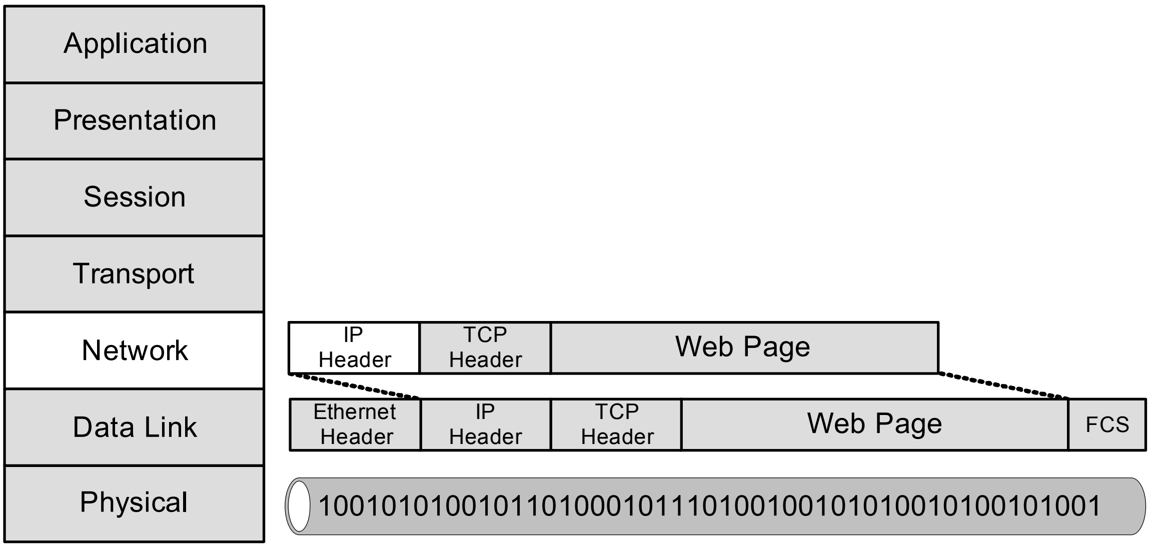


Figure 28 – IP Packet is removed

3. The frame headers and footers are now removed. The packet is passed up to the transport layer.

4. The client now checks the destination IP address to ensure the packet is for it.

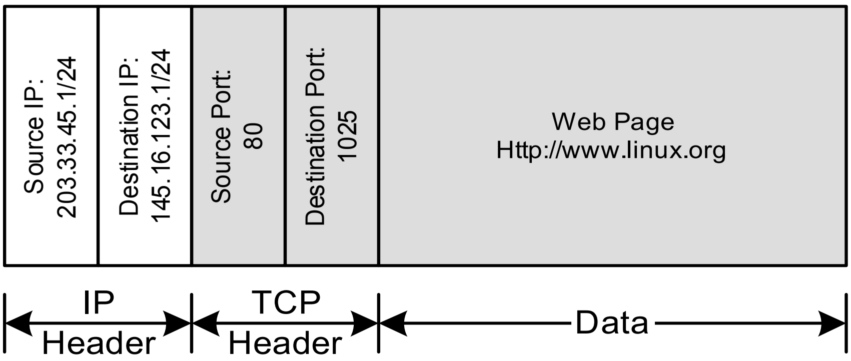


Figure 29 - IP Packet

The destination IP address is of the packet is the address of the client. The packet is accepted.

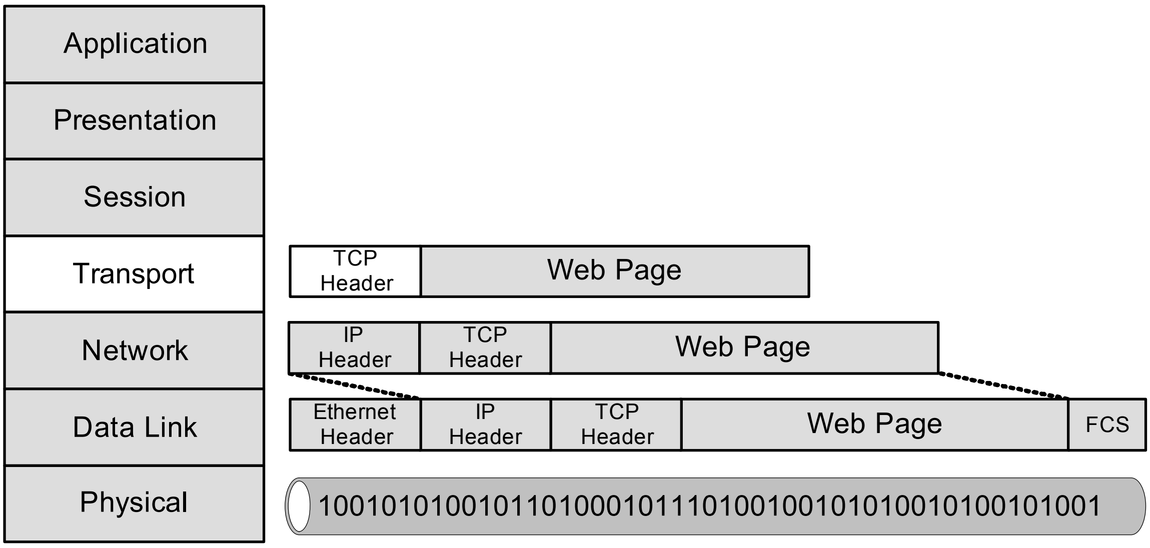


Figure 30 - Remove the TCP Segment

5. The IP header is removed and the TCP segment is now passed to the transport layer.

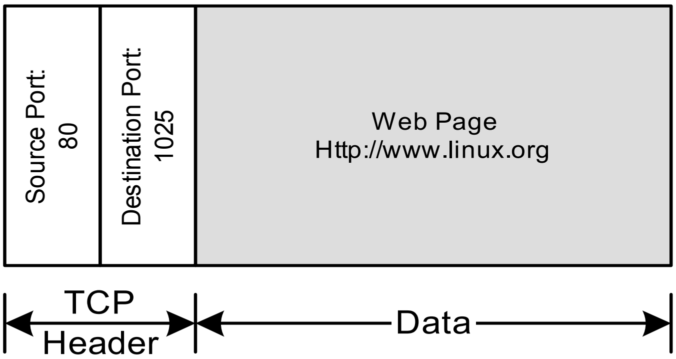


Figure 31 - TCP Segment

6. The destination port is now examined by the client. The client asks "Do I have a session active for this segment?". As this is return traffic from the original request sent the answer is yes. The client now knows to pass this segment to the process that is expecting a reply on port 1025.

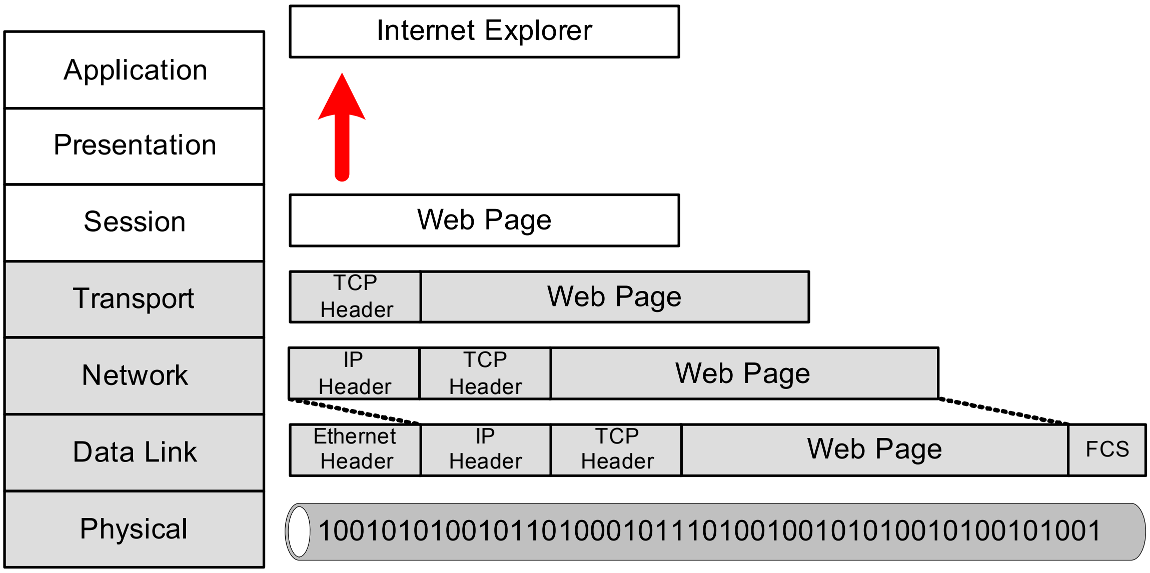


Figure 32 - Display the Web page

7. The data component (Web Page) is now extracted from the Segment and passed up to Internet Explorer on the client.

8. The web page now appears in the user's browser.