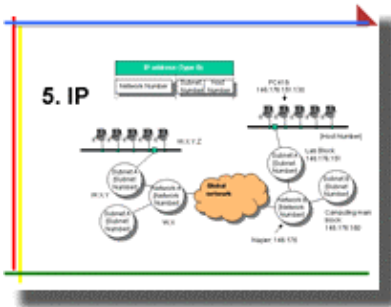


Slide 1 Introduction



In today's and next week's lecture we will cover two of the most important areas in networking and the Internet: IP and TCP. These cover the network and transport layer of the OSI model, and we will generally be investigating the functions of these layers, and how IP and TCP fit into them.

The Internet was designed by DARPA, and its main purpose was to create a robust network which could withstand a strike on any part of the network. It was created using the IP protocol, which defines IP addresses for each of the nodes on the network. These addresses have to be unique over the whole of the Internet. Its greatest problem since has been its success, as we are now running out of IP addresses to assign to all the devices which want to connect to the network.

You should not really think of the Internet as just being a collection of desktop computers. The future will move away from these large desktop systems, towards small, mobile devices. It is

predicted that within a couple of years that there will be more mobile devices which connect to the Internet, than static desktop systems. In order to cope with this demand we need many more IP addresses. At present most systems use IP Version 4, which has a 32-bit network address, giving around 4,294,967,296 network addresses. The new standard (IP Version 6) will have 128 bits for the network address, which gives over 340 million million million million address (3.4×10^{38}), which should be enough to go around, for a while. Just think of the future? Every car could be truly Internet-enabled, where each embedded computer system (such as the engine management system, or the air conditioning system) could be uniquely addressable over the Internet.

TCP and IP are two of the most important protocols ever developed, and have done more for world peace and any diplomat has ever done. In fact, TCP and IP unite the world and allow everyone in the world to communicate, no matter which computer they use, which operating system they are running, which language they speak, or which network they use.

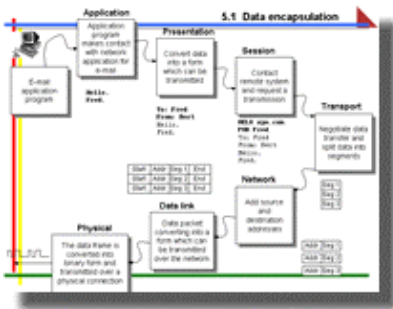
So what really makes the Internet work? Why does the WWW work so well? How can we run so many applications over the Internet at the same

time? How do we know that our data has been received? How does the data actually know how to get to a certain destination? Well, it's to do with TCP, IP and routing protocols. These three parts make the whole of the Internet work, and work reliably. The IP part is responsible for getting the data packets from the source to the destination (using the IP address), the TCP part is then responsible for sending the data to the required application program (using TCP sockets and sequence numbers), and the routing protocols are responsible for passing on information about how to get to destinations (using protocols such as RIP). Isn't it wonderful how a user can run a few WWW browsers, a TELNET session, an FTP session, a video conferencing session, and it all works, seamlessly, even if there are multiple destinations.

As we'll find out, in most cases we need TCP to go along with IP, as IP is only responsible for identify the destination and the sending address, and deciding if the data on the network is for them. It is TCP which either tags all the outgoing data and clearly identifies the virtual connection, or reorders and passes the received data to the required application. It also makes sure that all the data is properly received, and that anything that is sent, that there must be receipt it. If a

receipt is not received, the data is re-sent.

Slide 2 Data Encapsulation



This module is based on the 7-layer OSI model, and each unit focuses on a particular layer of the model. With Ethernet and ATM we are looking at Layer 2 (the data link layer), in this unit we are looking at the network layer, and in the next unit we will look at the transport layer. This unit will provide us with a greater understanding of how the Internet works, as the Internet is basically just an infrastructure of networks which use a standard addressing structure: IP addresses.

To simplify the process of sending and receiving data, the network uses a layered approach, where each layer encapsulates the data with information that is required at the given layer. At the **presentation layer** we would define things like the character set that we were using (such as using an ASCII character set), the type of encryption, and so on. Next at the **session layer** we would

build in data which allowed us to create and maintain a session on the remote system. This might involve us defining our user name and password, or some way to define the destination of our data.

At the **transport layer** a great deal of processing occurs. It is responsible for many things, including:

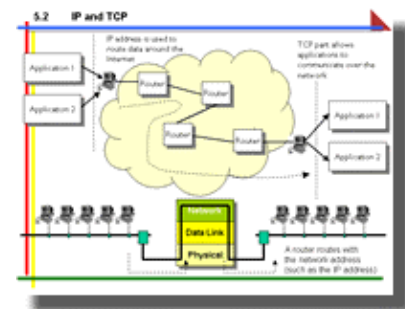
- Splitting the data up into data segments, which are small enough to be able to be sent over the network connection. As we will see Ethernet only allows for a maximum of around 1500 bytes of data to be sent, in an Ethernet frame.
- Identifying each data segment, so that missing segments can be identified, and that out-of-sequence data segments can be rearranged into the correct order.
- Making a connection with a remote system, and acknowledging the data that has been sent.

and several other functions (that will be discussed next week).

The network layer is mainly responsible for network addresses, and routing the data around the Internet, and, at the sending end, basically just adds the network addresses of the source and destination onto the data segments produced from at the transport layer. Eventually the data link layer

takes the data packet produced by the network layer and formats it in a way which can be transmitted over the local network. This normally involves addressing the physical source address of the local system, and the destination address, which is the next device to receive the data frame within the local network.

Slide 3 IP and TCP

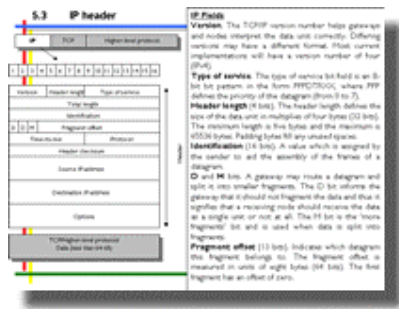


IP and TCP fit into the network and transport layers of the OSI model, respectively. The main purpose of the network layer is to define network addresses, and to determine the *best* path to a destination. The physical address of a node only has local significance, whereas the IP address has significance over the whole of the Internet (or an internet, if it is a locally defined network). Devices called routers, read the destination IP address, and make a decision as to whether they should forward the data packet onto another network. This decision occurs at the network layer, thus routers only require to implement the

first three layers of the OSI model (although, for security reasons, they may also read transport layer information).

Once the data packet is delivered at the destination, the function of the network layer is complete. It does not even ask for a receipt that the data packet has been received correctly. The transport layer then takes over and streams the data segments to the required application stream.

Slide 4 IP Header

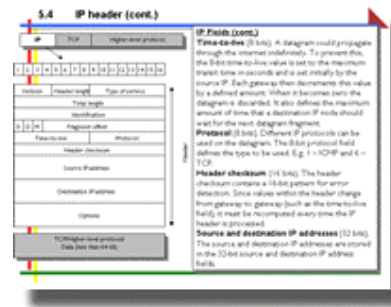


IP (Internet Protocol) is the best known networking layer protocol. Others include **NetBEUI** (for Microsoft networks) and **IPX** (for Novell NetWare networks). It is often a great advantage to use network protocols other than IP, as the data is less prone to access from external users on the Internet (as the Internet uses IP). If a device does not have an IP address, it cannot be accessed over the Internet.

The IP header contains several key areas:

Version. The 4-bit version part contains the IP version (currently this is typically Version 4). It thus allows for different version to be used on the Internet, at a time). The format of the IP header may vary across different version, thus it is important that the version value is defined in the first part of the IP header.

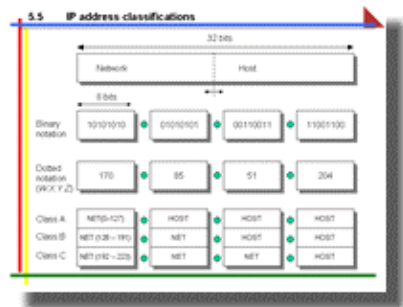
Slide 5 IP Header (cont.)



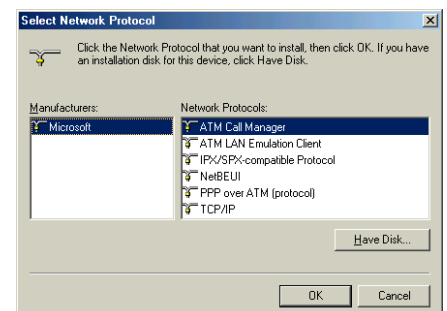
TTL. This stops data packets from transversing the Internet indefinitely. Initially the TTL is set to a given value (it's maximum will be 255, as it is an 8 bit value). Every router which the data packet travels across will then decrement this value by a certain amount. If it travels across too many routers, the TTL value will reduce to zero, after which the data packet will be deleted.

Source and destination IP address. These contain the 32-bit definitions of the source and destination IP addresses.

Slide 6 IP Address Classifications

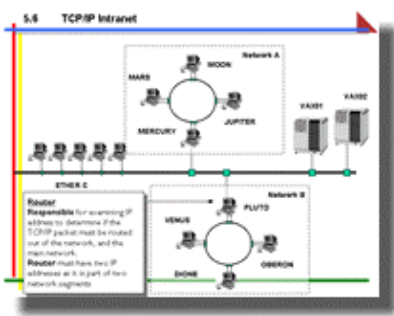


IP addresses have 32-bits, and are granted by an Internet address granting authority. When organisations apply for an address they must define the number of nodes which they will require to be connect to the Internet. It should be noted that a node which does not have an IP address cannot connect directly onto the Internet. A node can still communicate over a network, though, using another protocol, such as NetBEUI and IPX, and that a node can communicate with different systems, using different protocols, at the same time. For example a node might communicate with a Novell NetWare file server using **IPX/SPX**, and with a Microsoft file server with **NetBEUI**, and also with the Internet using TCP/IP.



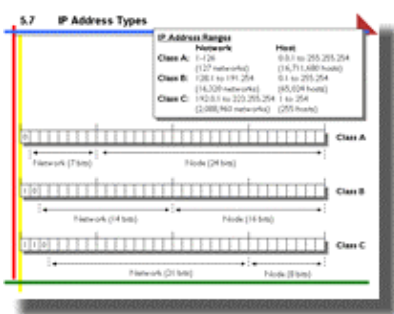
To make the IP address easier to read, the bits are grouped into groups of 8 bits, with a bit separating them. These 8-bit values are typically converted into a decimal format, so that it is easily interpreted by humans. For example 0011 0011.1100 1111.0000 1010. 0000 0001 gives the IP address of: 51.207.10.1.

Slide 7 TCP/IP intranet



In a network the IP addresses are assigned to every node which requires to connect to the Internet. Every network connection requires a unique IP address, thus a router will have an IP address for each of its ports.

Slide 8 IP Address Types



The main IP address classifications are Class A, Class B, and Class C. They typically are granted depending on the size of the organisation. If the first bit (the most significant bit) of the 32-bit IP address is a zero then the address is a Class A address. If it begins with a 10, it is a Class B, and if it begins with a 100, it is a Class C address.

The number of bits that each of the classes reserves for the network and the node within the network also varies. These are defined as:

➤ **Class A.** This address defines 7 bits for the network part, and the rest (24 bits for the node part). There are only thus 2^7 (128 – although some of these cannot be used) network addresses that can be granted, of which there can be 2^{24} (16,777,216 – although some of these cannot be used) nodes on each of these networks. This type of address is granted to organisations which can have a great deal of nodes which connect to the Internet (such as large US universities, or military organisations). A Class A address can be identified with a value which begins with 1.x.x.x to 127.x.x.x.

➤ **Class B.** This address defines 14 bits for the network part, and the rest (16 bits for the node part). There are thus 2^{14} (16,384 – although some

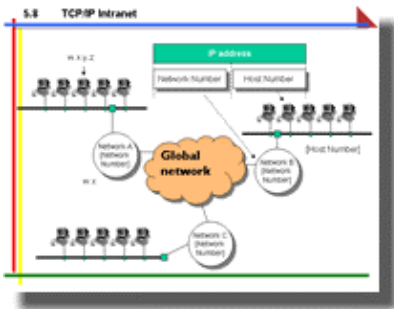
of these cannot be used) network addresses that can be granted, of which there can be 2^{16} (65,536 – although some of these cannot be used) nodes on each of these networks. This type of address is granted to organisations which can have a large number of nodes which connect to the Internet (such as a large commercial company, or university). A Class B address can be identified with a value which begins with 128.x.x.x to 191.x.x.x. Napier has a Class B address, as its address are of the form 146.176.x.x.

➤ **Class C.** This address defines 21 bits for the network part, and the rest (8 bits for the node part). There are thus 2^{21} (2,097,152 – although some of these cannot be used) network addresses that can be granted, of which there can be 2^8 (256 – although some of these cannot be used) nodes on each of these networks. This type of address is granted to organisations which have a small number of nodes connected to each of the assigned network addresses. A Class C address can be identified with a value which begins with 192.x.x.x to 223.x.x.x.

There are other classifications of addresses, such as Class D, which is used for multicasting data packets (sending data packets to a many

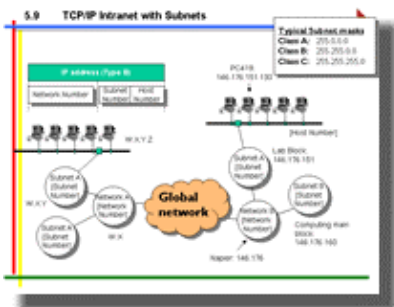
destinations, at the same time.

Slide 9 TCP/IP Intranet



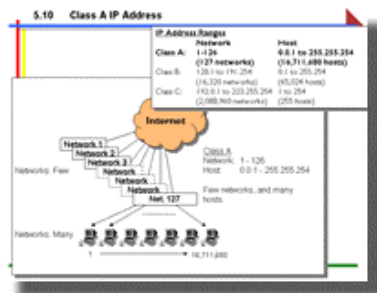
As previously said, the IP address contains a network part, and a node part. By default, the size of these parts are defined by the classification.

Slide 10 TCP/IP Intranet with subnets



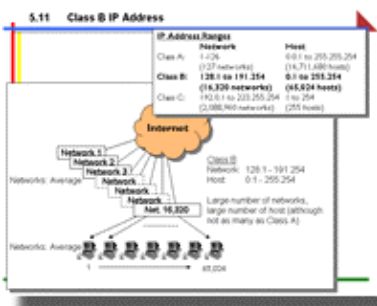
If the IP address were then assigned in a sequential manner, it would be difficult to organise the network in a logical manner. Thus we have the concept of a subnetwork (a subnet), where part of the node address is used to define a subnet within the organisation.

Slide 11 Class A IP Addresses



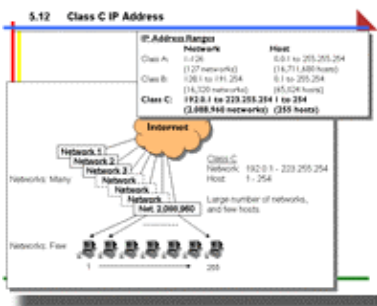
The default addresses of a Class A are thus. To be completed.

Slide 12 Class B IP Addresses



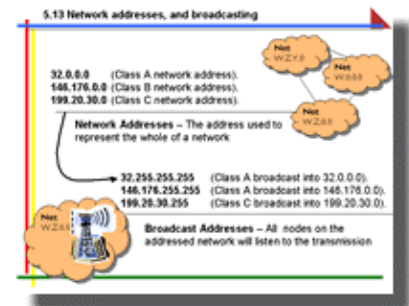
The default addresses of a Class B are thus. To be completed.

Slide 13 Class C IP Addresses



The default addresses of a Class C are thus. To be completed.

Slide 14 Network addresses, and broadcasts



Not all the address can be used, as some of them are reserved for special purposes. The 0 address is used to define the address of a whole network. Thus the Napier address is 146.176.0.0.

The 255 address to the whole of the network is defined as the broadcast address, where all the nodes on the network listen to the data packet. As will find-out this is used by nodes to find the physical address of other nodes on their network (using ARP – which will be covered in the Ethernet unit). A broadcast into the Napier network would thus be 146.176.255.255.

Slide 15

Subnetting for Class B

Subnet address	Default net.	Max sub.	Max nodes (subnet)
11111111 11111111 00000000 00000000	255.255.0.0	2	16382
11111111 11111111 01000000 00000000	255.255.252.0	6	8190
11111111 11111111 01100000 00000000	255.255.248.0	14	4094
11111111 11111111 01110000 00000000	255.255.240.0	30	2046
11111111 11111111 01111000 00000000	255.255.232.0	62	1022
11111111 11111111 01111100 00000000	255.255.224.0	126	510
11111111 11111111 01111110 00000000	255.255.216.0	254	254
11111111 11111111 01111111 00000000	255.255.208.0	510	126
11111111 11111111 01111111 01000000	255.255.204.0	1022	62
11111111 11111111 01111111 01100000	255.255.200.0	2046	30
11111111 11111111 01111111 01110000	255.255.192.0	4094	14
11111111 11111111 01111111 01111000	255.255.184.0	8190	6
11111111 11111111 01111111 01111100	255.255.176.0	16382	2

For example, if the network address is 144.32.8.0 and the five bits are used for the subnet mask then:

The network 144.32.0.0/16 (00001000 0000 0000) is the first subnet - 144.32.8.0 (node range 144.32.8.1 to 144.32.55.254)
and 144.32.211.0/16 (000 0000 0000) is the last subnet - 144.32.248.0 (node range 144.32.248.1 to 144.32.255.254)

Subnetting uses bits borrowed from the node part of the address to define the network part. All the devices on the subnet must know which part is the network, and which part is the node. Thus we have the concept of a subnet mask.

A zero bit in a subnet mask position defines the node part, and a one bit defines the network part. Thus by default the subnet mask for the main classifications of IP address are:

- Class A. 255.0.0.0.
- Class B. 255.255.0.0.
- Class C. 255.255.255.0.

Bits can then be borrowed from the node part. Thus if we took 8 bits from a Class B address to define the subnet, then the subnet mask would be:

255.255.255.0

and we could define 254 subnets, with 254 nodes on each subnet (as shown in the figure). The addresses 0 and

255 are used for the network, and the broadcast address, respectively. Thus for a network address of 155.100.0.0, the network address would be:

155.100.1.0
155.100.2.0

and so on, until 155.100.254.0. The first node on the first subnet (155.100.1.0) would be:

155.100.1.1

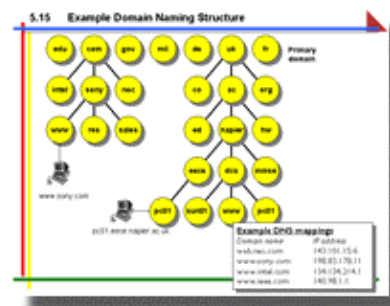
and the last node would be:

155.100.1.254.

Please study the nodes for Class C subnet, and also for creating different sizes of subnet masks.

Slide 16

Example DNS

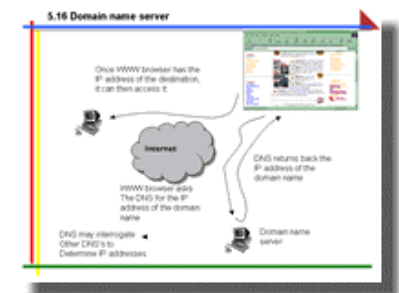


It is extremely difficult to remember IP addresses, thus a domain naming system has been introduced, where organisations register to one of the main domains, which have an associated range of IP addresses. Each organisation must then maintain a domain name system for their own

domain name, in order to define the domain names, and associated IP addresses for their domain. For example Napier would have the napier.ac.uk domain, which is registered to the 146.176.0.0 network address. It can then run its own domain name server to define all the domain names, and associated IP addresses for all the nodes in its domain.

Slide 17

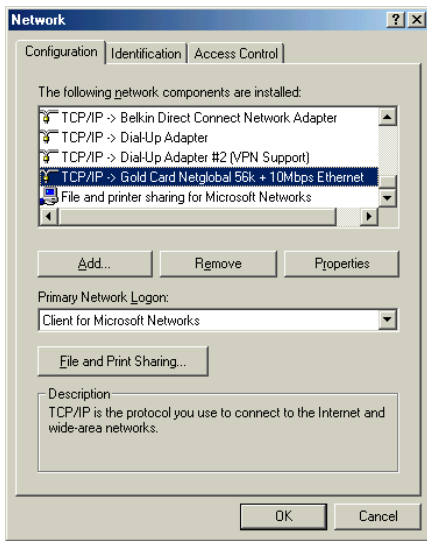
Domain name server



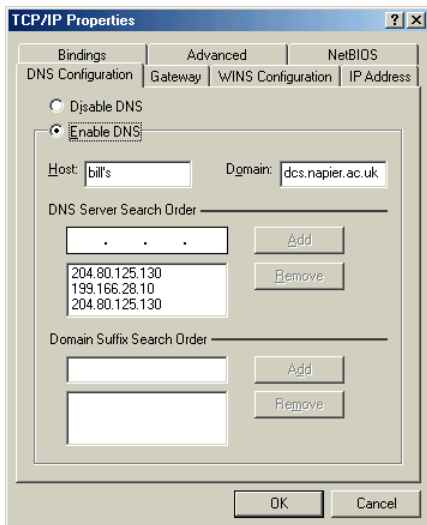
When a program, such as a WWW browser, uses a domain name it must ask a domain name server for the associated IP address. Once it has the address, it can communicate with the destination. This type of system makes the naming structure of the Internet more dynamic, where IP addresses for nodes can be easily changed, and is updated across the whole of the Internet.

The main properties for a node using TCP/IP are the IP address, its subnet mask, the gateway address, and the domain name server. The

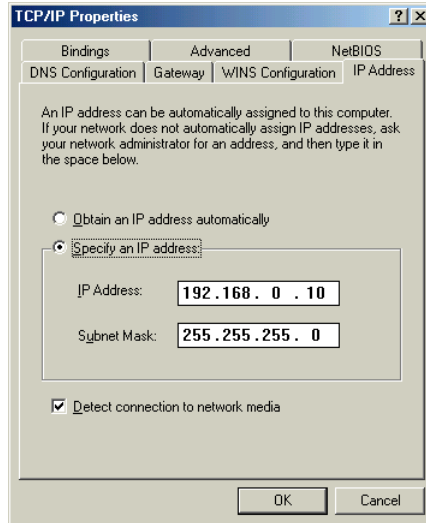
properties are defined from Control Panel -> Network:



The domain name server can be defined by clicking on the Properties tab:



The Gateway address is important, and should be set to the port of the router which the node connects to. The rest of the parameters are accessed from the IP address and Gateway tabs:

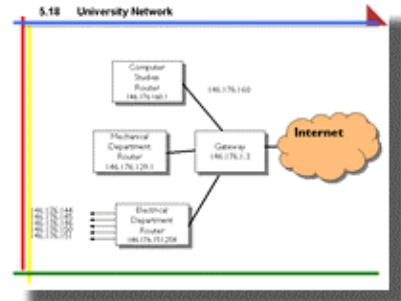


Slide 18 Allocating IP Addresses



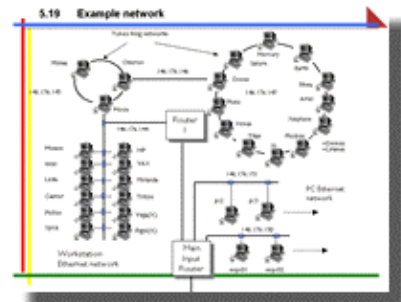
IP addresses are a valuable resource, and can be used to enhance the security of a network. Typically organisation will not allow the user to define their own IP address, but will use a system, such as a **DHCP** (Dynamic Host Control Protocol) server, to grant an IP address to a node, based on its physical (MAC) address. This type of system can also be used to bar certain nodes from accessing the Internet.

Slide 19 University network



The subnetting of a network makes it easier for routers to stream data packets to their required destinations.

Slide 20 Example network



To be completed.