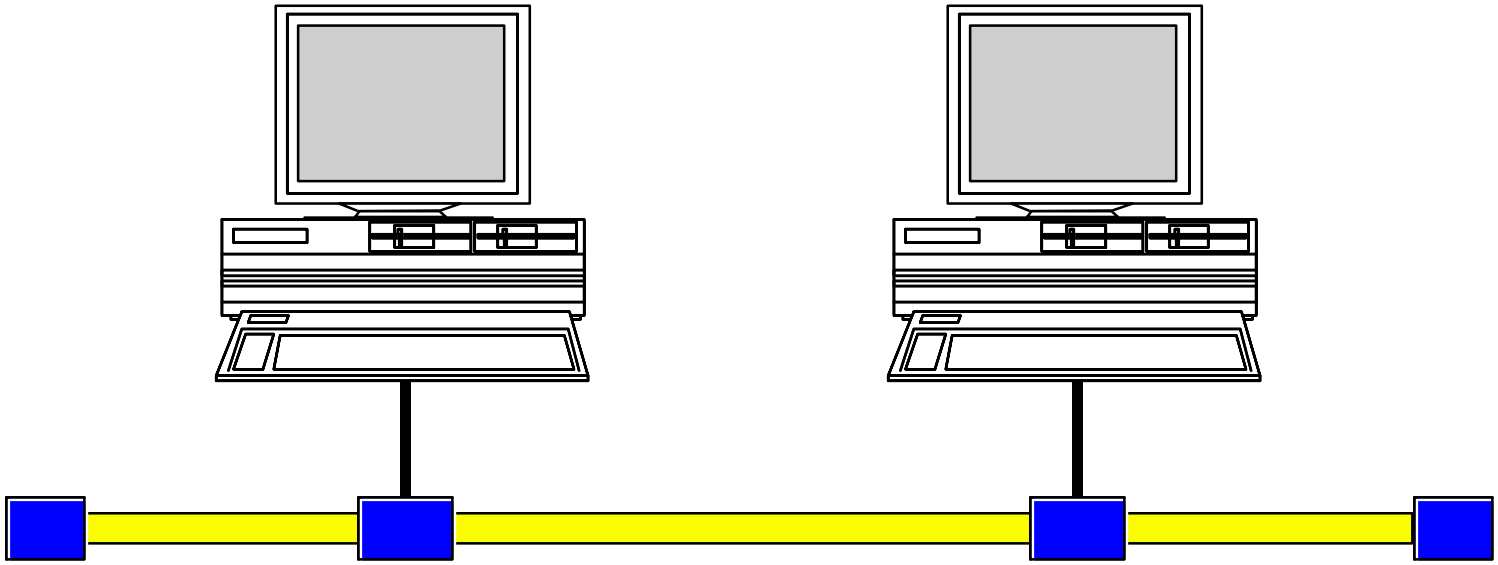

FORE
SYSTEMS

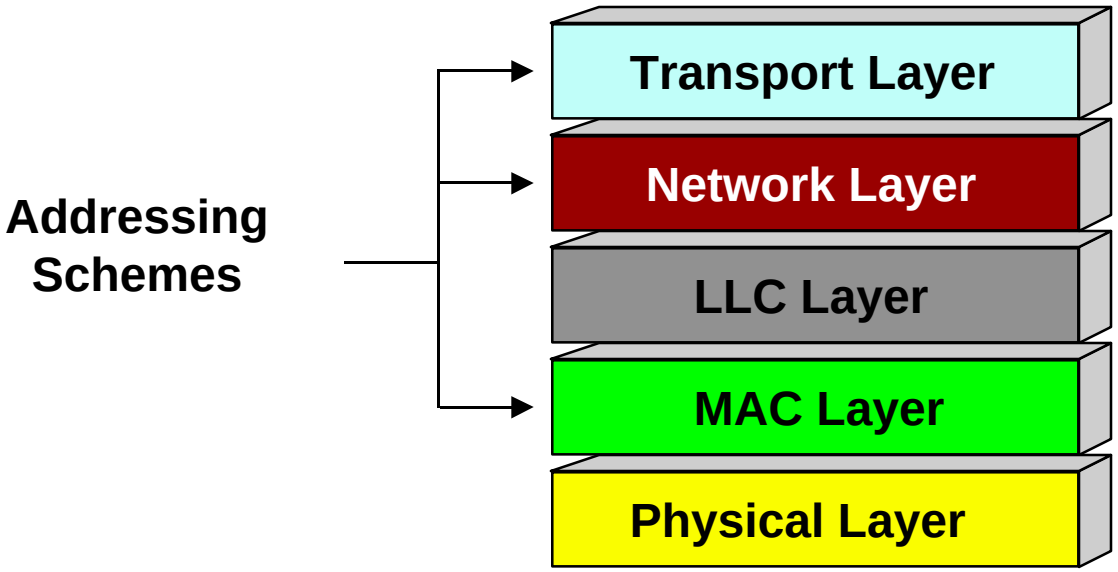
Addressing In Local Area Networks

V1.1: Geoff Bennett

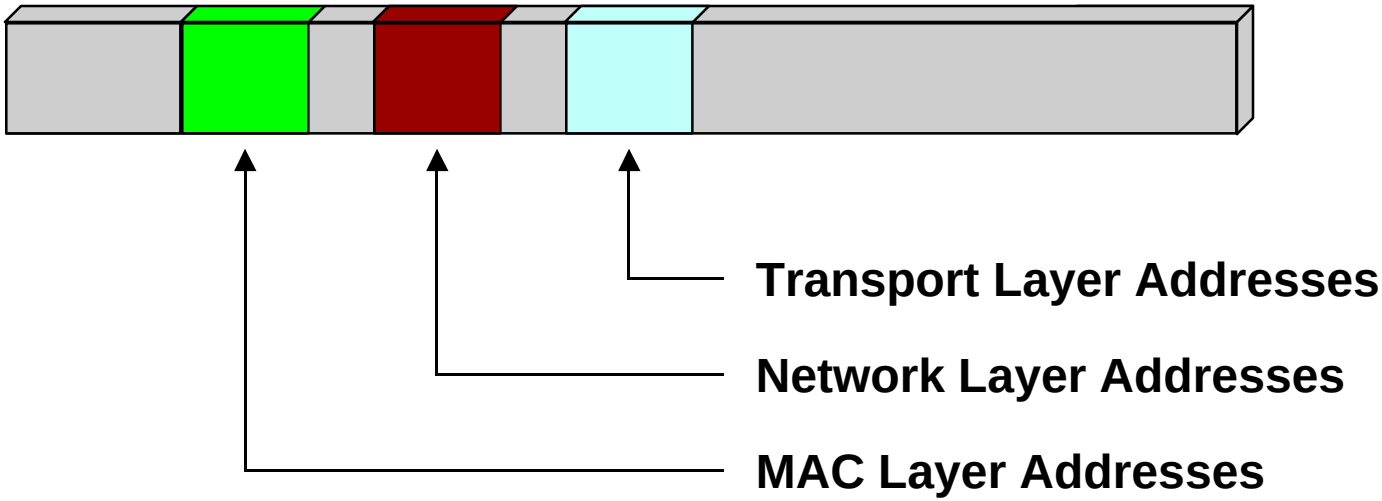


Networks allow us to transmit information between one computer and another.

Part of this procedure is the use of *addressing* to make sure messages get to the right place.



As we'll see in this tutorial, addressing schemes exist at multiple layers of the OSI Model. A typical TCP/IP packet will contain addresses that are designed to be used at the MAC Layer, the Network Layer and the Transport Layer.



The addresses are stored in specifically defined parts of the IP packet and the LAN frame.

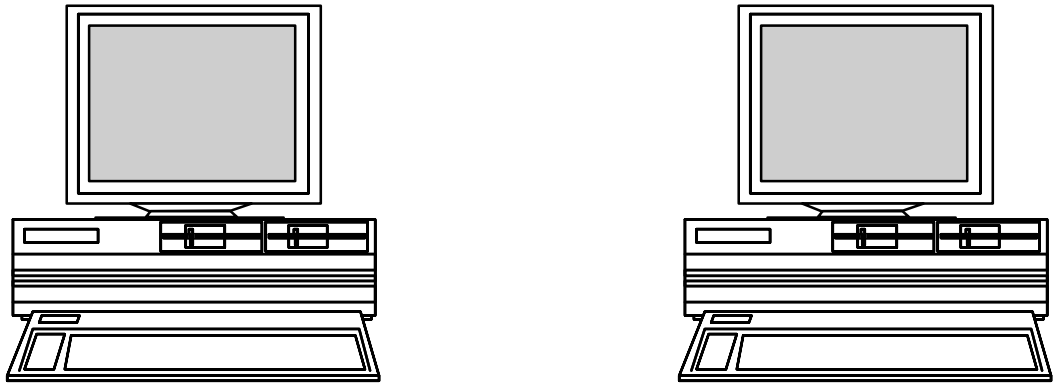
The consistent position of addresses is a key factor that allows software to interpret addressing information correctly. In other words, if we put addresses in the wrong format, or in the wrong place, our communication software will not work correctly.

FORE SYSTEMS

Why Do We Need Addressing?

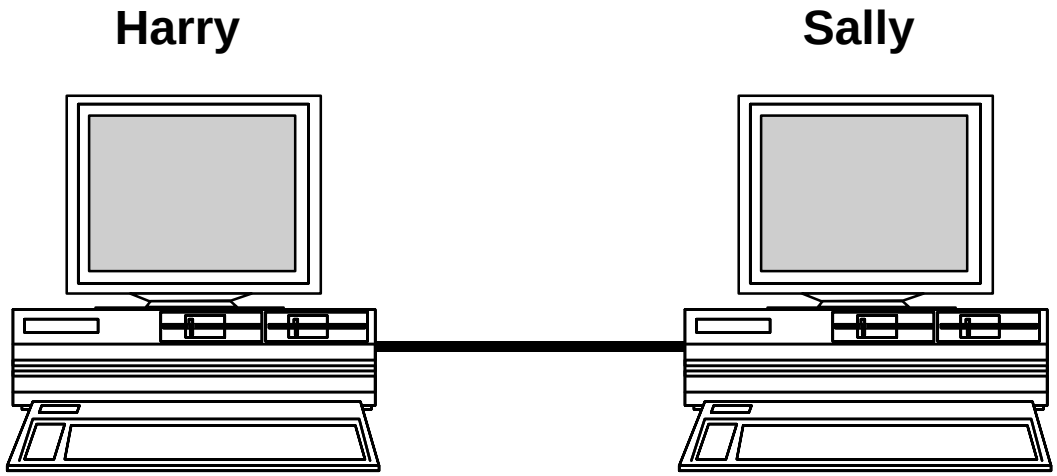
FORE SYSTEMS

Harry **Sally**



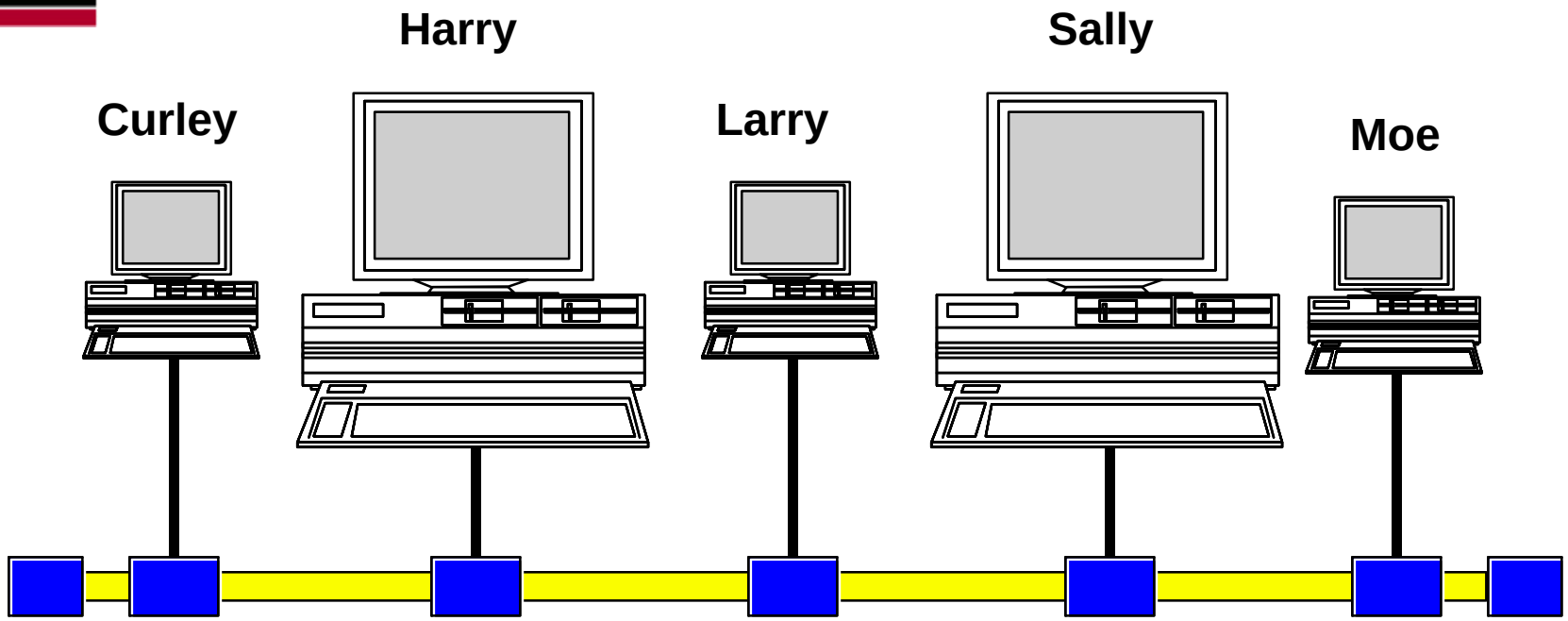
In this diagram, let’s assume that Harry wants to send information to Sally.

Lets further assume that both computers are equipped with suitable interface circuits that allow them to insert messages into the network.



If we just connect a cable between the two computers, then Harry's software can simply push the information into the cable, and it will inevitably end up at the right place.

You may have even used such a configuration if you've ever downloaded information using modems, and software such as Kermit.



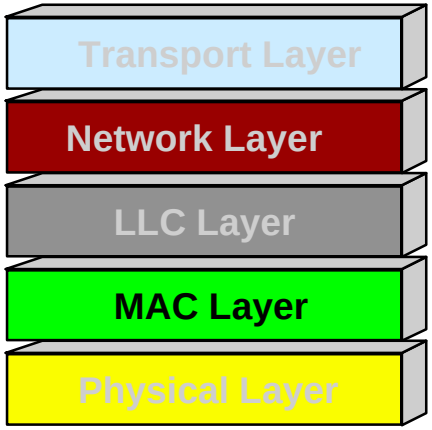
On a *shared* network, such as Ethernet, Harry and Sally are not the only users of the communication channel.

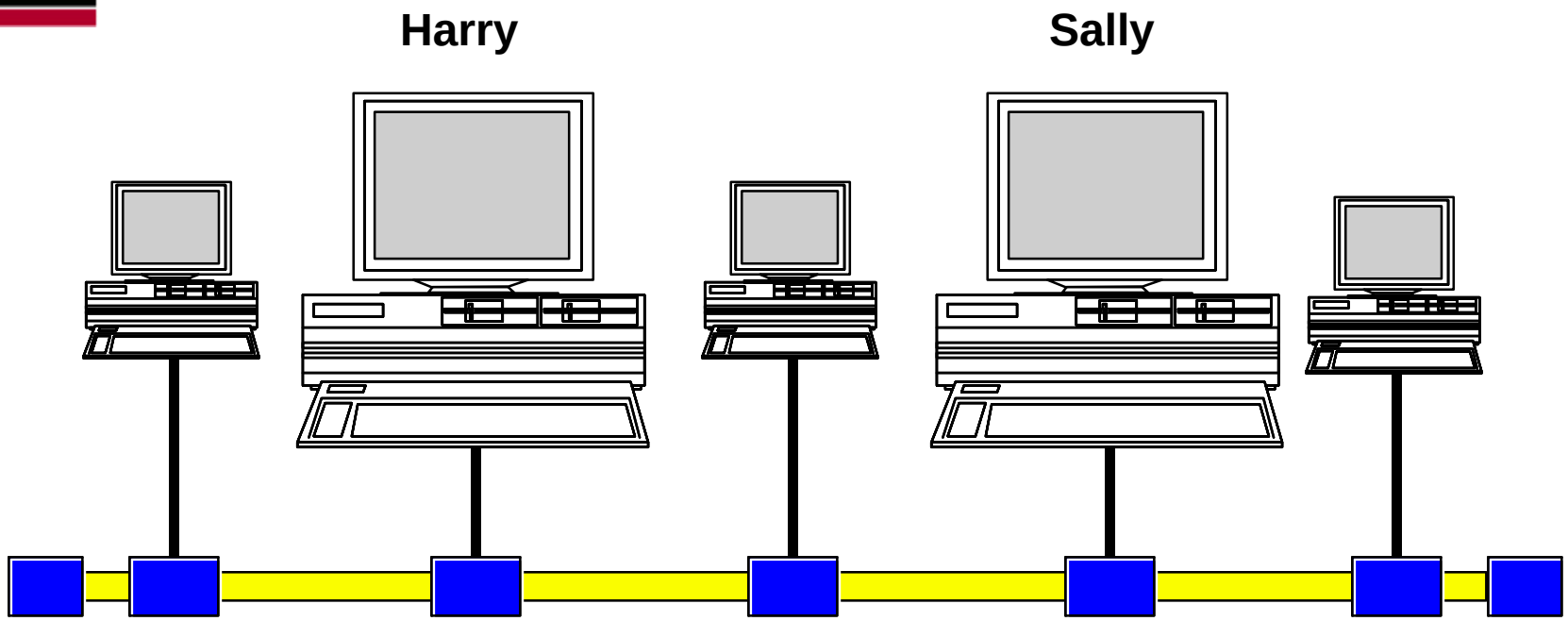
All of the computers attached to this network share the same communication channel.

Addressing is used to ensure that messages between any two of these machines are not received by other users.

FORE SYSTEMS

MAC Layer Addressing





MAC Layer Addressing is often explained in terms of security. In other words, all the stations are on the same network and MAC Addressing ensures that one station cannot receive messages intended for another station. However, this is a naive way to think of MAC Addressing, since it is so easy to bypass this security.

Instead, we should regard MAC Addressing as a way to ensure that other LAN users are not forced to process messages that are actually being sent to someone else.

- **Local Wire Address**

There are several phrases used to describe MAC Layer Addressing. They are all identical in meaning.

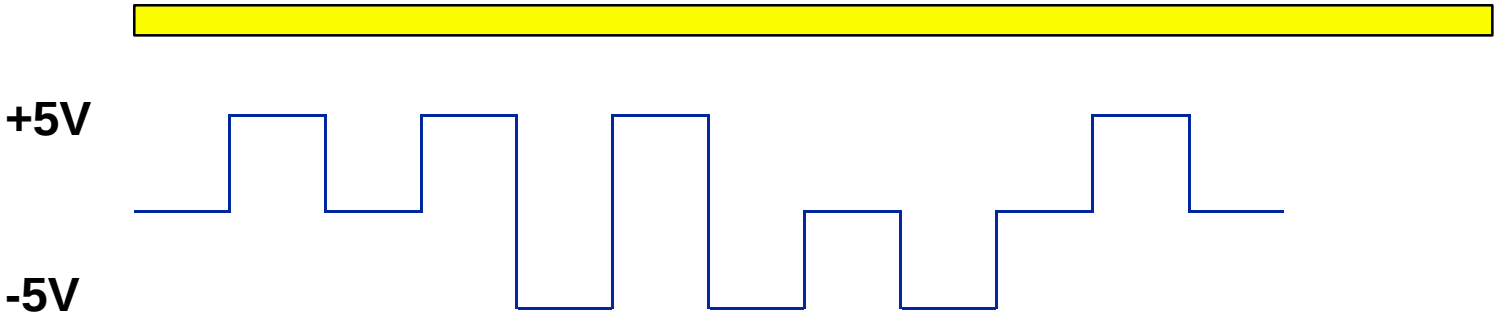
Local Wire Address is a slang phrase, and refers to the fact that the reason for the addressing scheme is to differentiate LAN stations that are attached to the same cable. I tend not to use this term because it is not such a good description in these days of LAN switches and multiport bridges.

- Local Wire Address
- **Physical Address**

Physical Address is the term used in RFC documents to describe the MAC Address. I believe it came into use because the MAC Address is tied to the *physical* host from which the frame originates, or to which it is directed.

- Local Wire Address
- Physical Address
- **MAC Address**

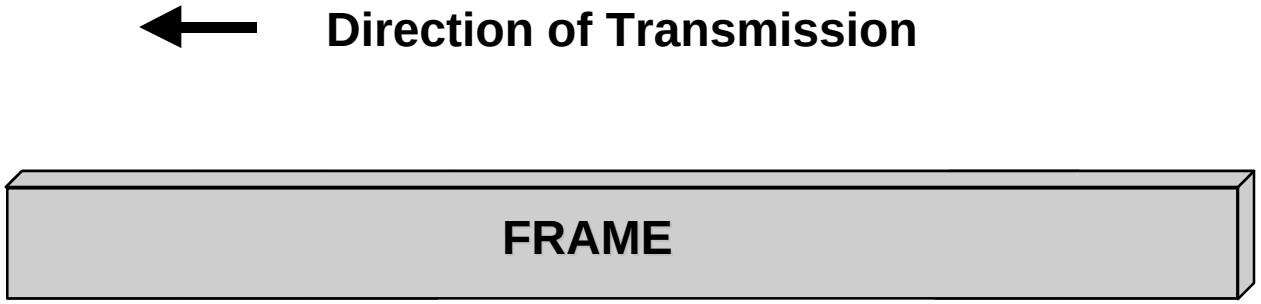
MAC (Media Access Control) address is a term that's used throughout the industry, and it's the one I've grown used to. It is at the MAC Layer of the OSI Model that these addresses have significance.



...1101011010...

At the Physical Layer of the OSI Model, electrical signals are interpreted as a series of binary 1's and 0's.

The Physical Layer functions don't make any attempt to interpret these bits in any way.



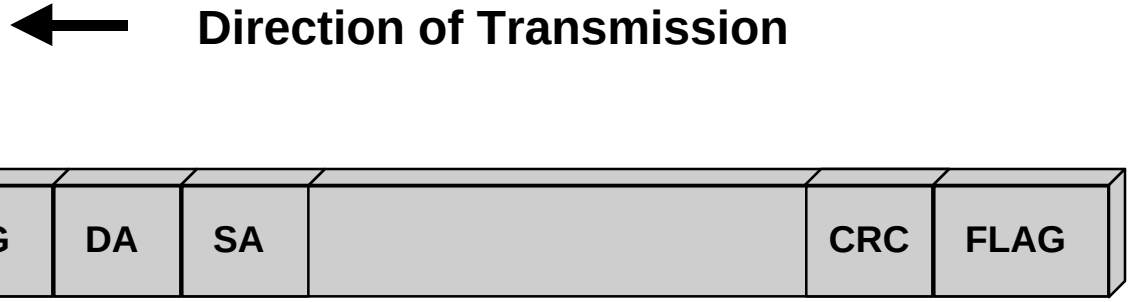
At the MAC Layer, the 1's and 0's are interpreted into a structure called a *Frame*.

Frames are the lowest level collection of information on a LAN.

Frames can be quite long. On Ethernet, they are up to 1.5kB (about 12 000 bits), on Token Ring up to 18kB and on FDDI up to 4kB.

The smallest frame size is also specific to a given LAN technology. Ethernet has a minimum of 64 bytes.

FORE SYSTEMS



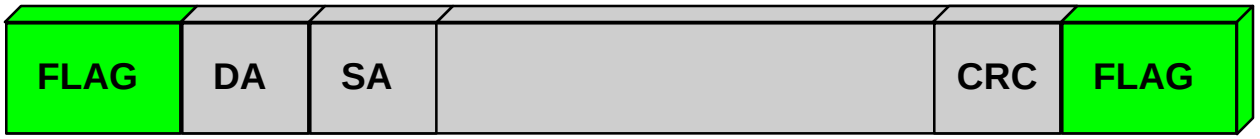
Frames have a structure that is specific to the LAN technology. Ethernet, Token Ring and FDDI frames are all slightly different in structure.

This diagram is a generic view of a frame.

The bits in the frame are transmitted in order from left to right. This is the typical convention used in most textbooks.

FORE SYSTEMS

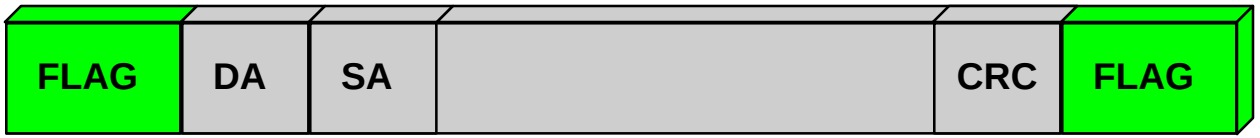
Direction of Transmission



The first feature of a frame is some form of *delimiter*, or *flag*. Flags are some special bit pattern, or line encoding, that allows the LAN circuits to identify the beginning and ending of the frame.

FORE SYSTEMS

Direction of Transmission



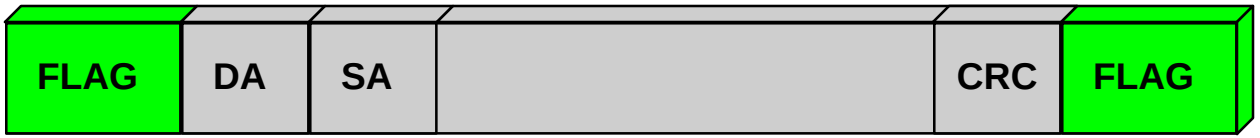
On Ethernet, the flag at the start of the frame is a series of 62 bits alternating 1 and 0, and then two bits set to 1. Ethernet and IEEE standards refer to this field as the *preamble*.

Another major use for the preamble is to allow LAN adapters to “lock on”, or synchronise with the clock signal that is contained within the bitstream encoding.

The ending delimiter is actually a “gap” in transmission - this must last at least 9.6 microseconds, but will be longer if no other station is ready to transmit. Ethernet and IEEE standards refer to this as the *interframe gap*.

FORE SYSTEMS

Direction of Transmission



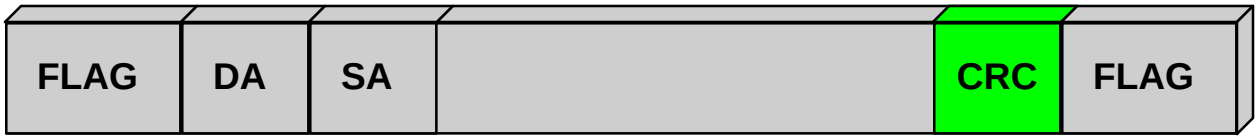
In Token Ring and FDDI technologies, the flags are represented by special line coding.

For Token Ring, the coding is actually a controlled violation of the Manchester Encoding scheme.

For FDDI, special 5-bit symbol patterns are reserved for flags.

FORE SYSTEMS

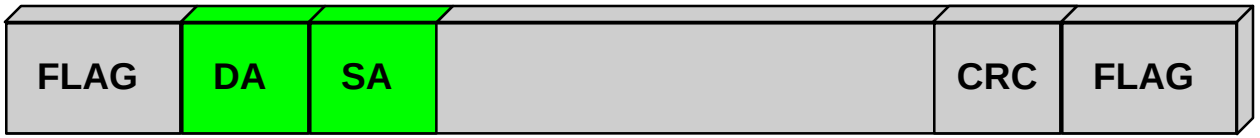
Direction of Transmission



Towards the end of the frame is a field called the *Cyclic RedundancyCheck* (CRC). This is used to check for frame corruption.

FORE SYSTEMS

Direction of Transmission

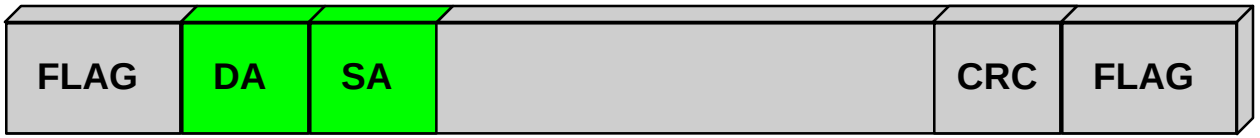


At the beginning of the frame are the two *MAC Address* fields. The first of these fields is the *Destination Address* (DA), and the second is the *Source Address* (SA).

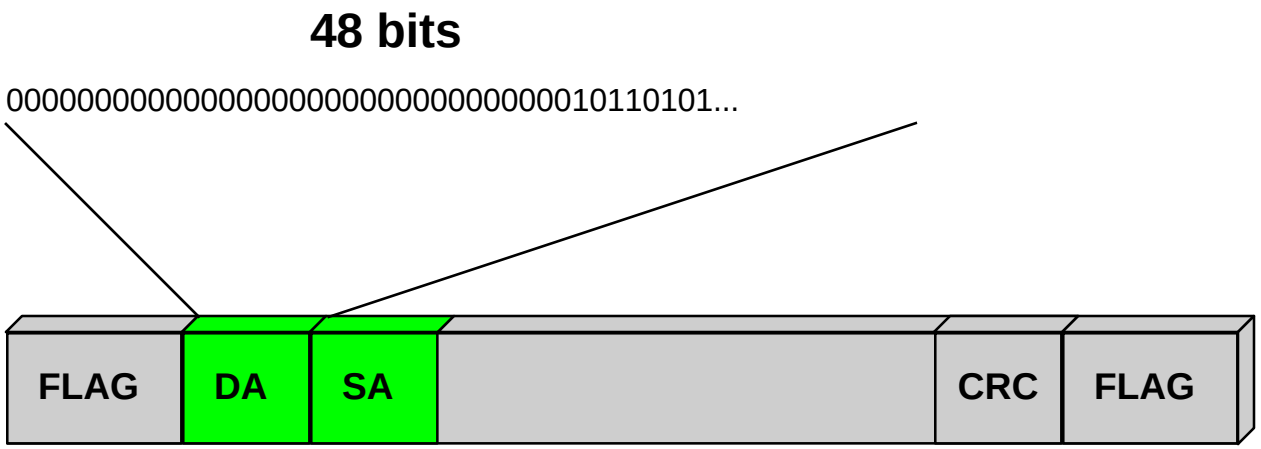
For Harry's message to Sally, Harry would insert Sally's MAC Address in the DA field, and his own MAC Address in the SA field.

FORE SYSTEMS

MAC Address Structure



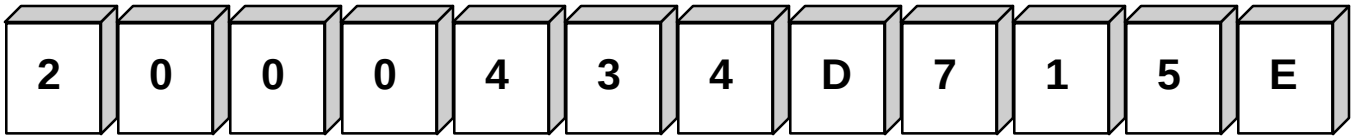
Although frames are specific to a given LAN technology, the most popular modern technologies (Ethernet, Token Ring and FDDI) all use the same address structure.



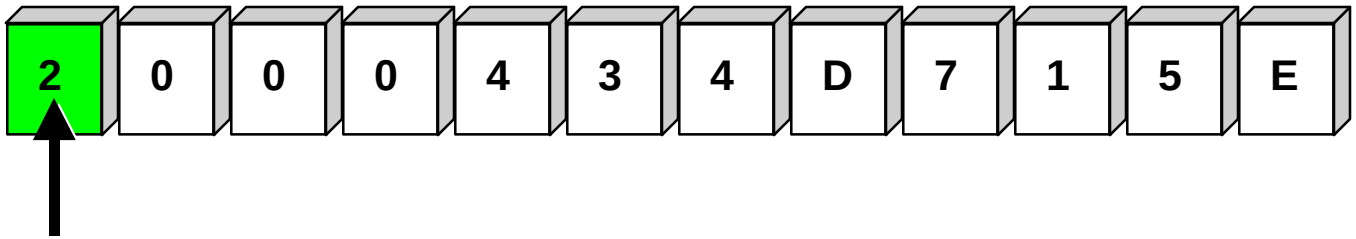
This MAC address structure is defined by the IEEE 802 committee, and is *48 bits* long.

As you can see, representing these addresses in binary is a bit tedious, and so we normally write the address in hexadecimal. A 48 bit address can be written as 12 hex digits.

We use hex rather than decimal because there is a further structure to the 48 bit address.



Here's a typical IEEE MAC address, divided into hex digits. Each *hex digit* is the equivalent of *4 bits*.



Binary Representation= 0010

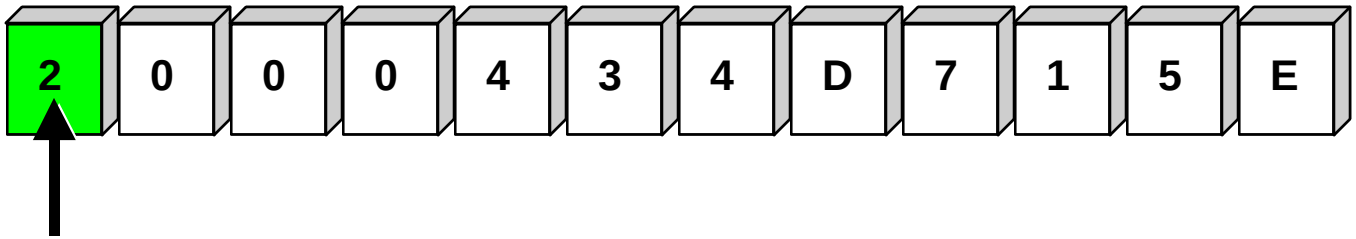
G/I Bit

The first two bits of the address have a special significance.

The first bit is known as the *Group/Individual* (G/I) bit.

If this bit is clear (ie. 0), then the address is a *Unicast* address. This means that the frame is addressed to only one possible LAN interface.

If the G/I bit is set (ie. 1) then the frame is a *Broadcast* or *Multicast*.



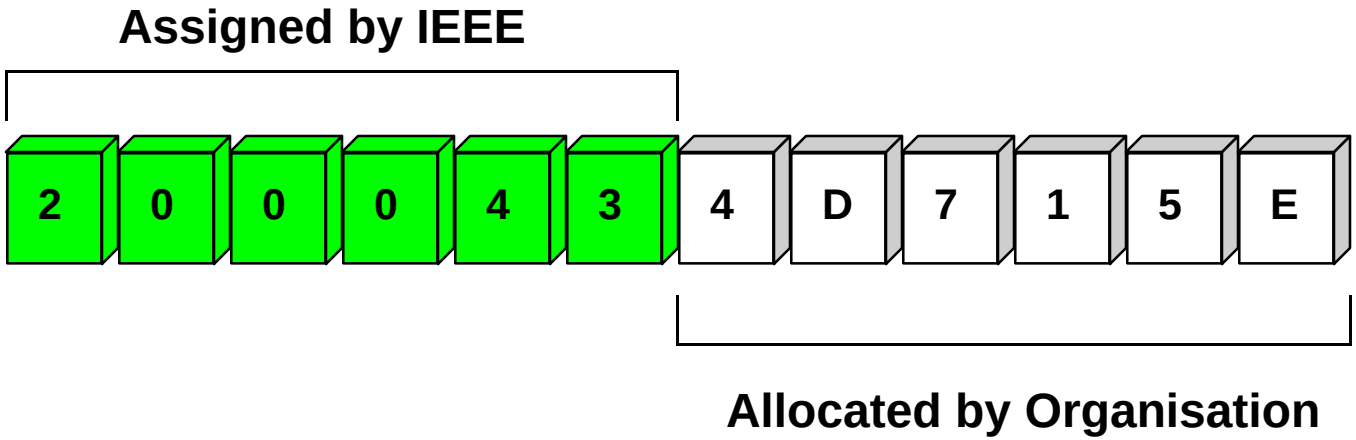
Binary Representation= 0010

G/L Bit

The second bit is known as the *Global/Local* (G/L) bit.

If this bit is clear, then this MAC address has been allocated from a block of addresses which is registered with the IEEE. In this case, no other LAN interface in the universe should have an identical address. In other words, this is a *Globally Administered* address.

If the bit is set, then this address was created by the local LAN administrator, and it may not be globally unique. In other words, it is a *Locally Administered* address.



For globally administered addresses, the IEEE allocates a 24 bit address block to organisations that apply.

Once the block is allocated, the organisation is responsible for uniquely assigning addresses within its own block.

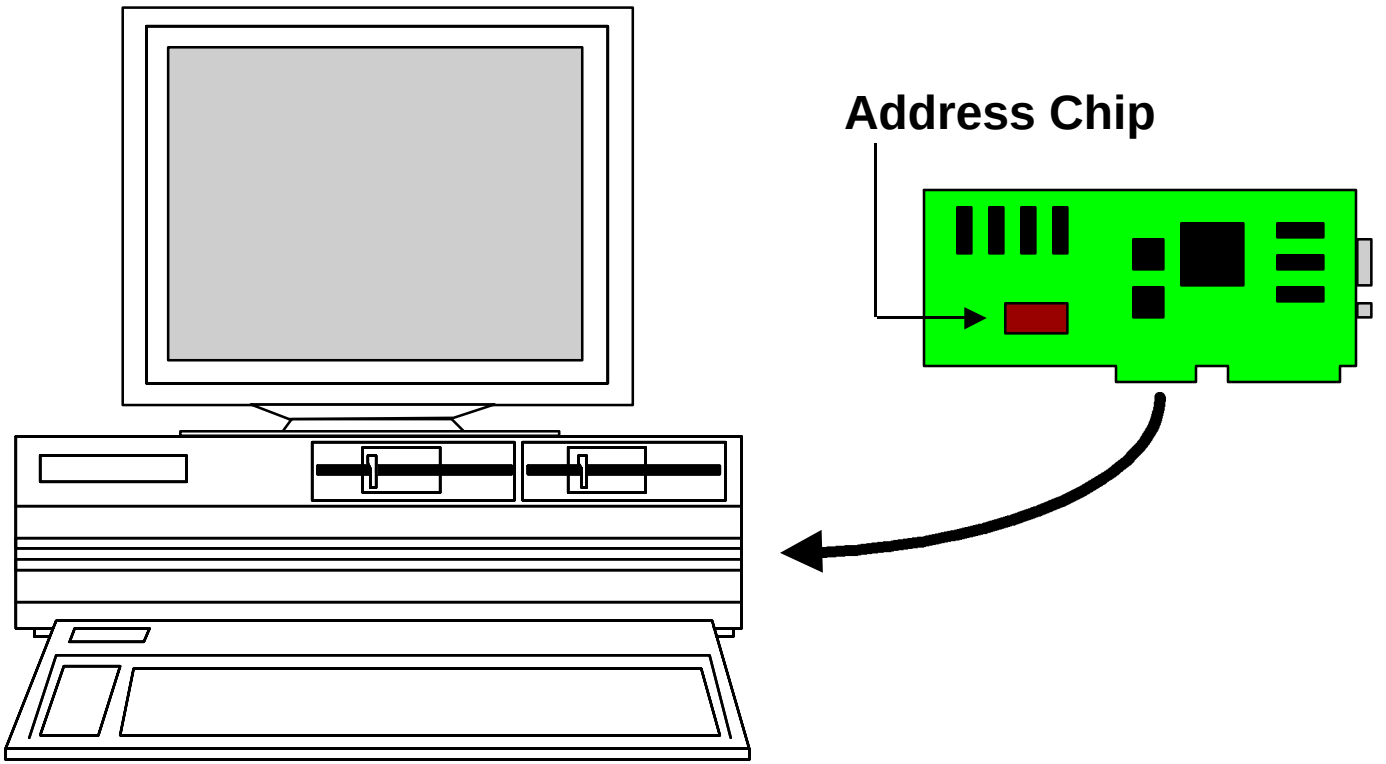
Large organisations (such as DEC and IBM) have multiple 24 bit blocks.

Organisation	Address Block
Cisco	00000Ch
DEC	08002B (et. al.)
IBM	08005A (et. al)
Sun	080020h
Proteon	000093h
Wellfleet	0000A2h

Here are a few examples of IEEE-assigned address blocks. A complete list can be found in the latest version of the “Assigned Numbers” RFC.

FORE SYSTEMS

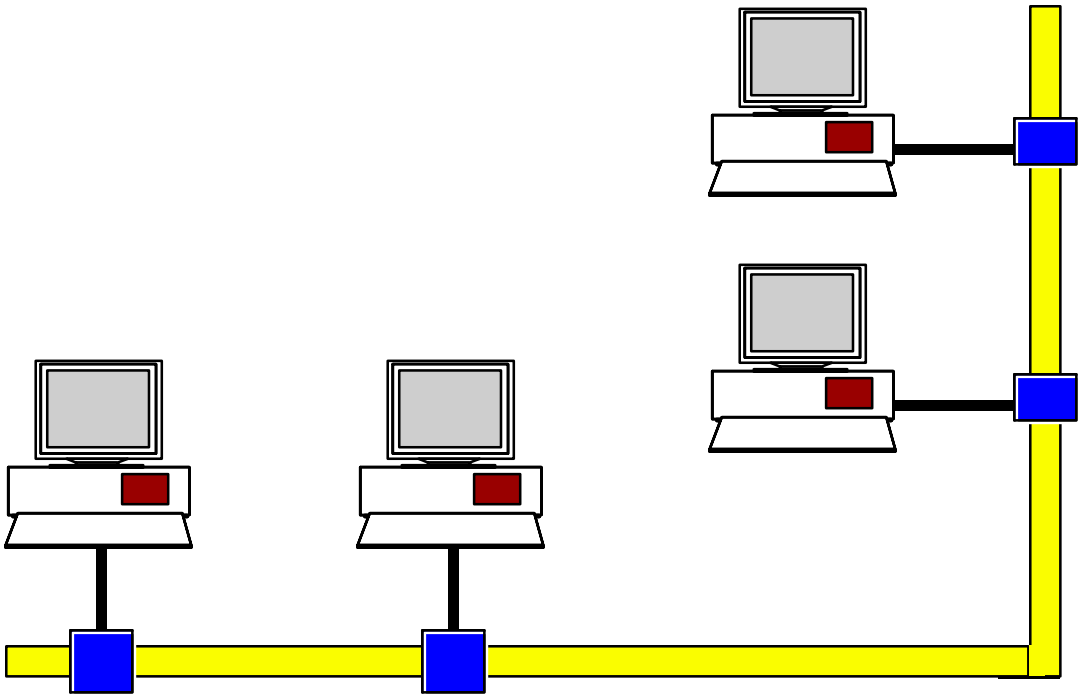
MAC Addresses In Action



Let's say that Joe Bloggs Inc. apply to the IEEE and are given the 24 bit block "200043". No other organisation will *ever* be given the same address block.

Joe Bloggs manufacture an Ethernet interface, and assign the remaining 24 bits. They then "install" this address into a permanent memory device (a PROM or PAL chip) on the interface.

No other LAN interface (even Token Ring or FDDI interfaces) should *ever* be assigned this address by Joe Blogss.

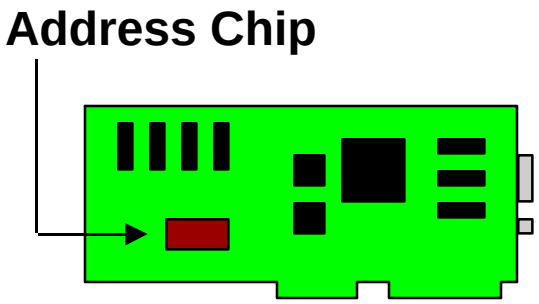


If we were always able to use Globally Administered addresses, we could be sure that no two machines in the world are using the same MAC address.

So you might think that MAC addresses are all we need to send LAN traffic between any two machines in the world.

Unfortunately this is not true, for two reasons...*Local Addressing* and *Scaleability*.

Local Addressing - How?

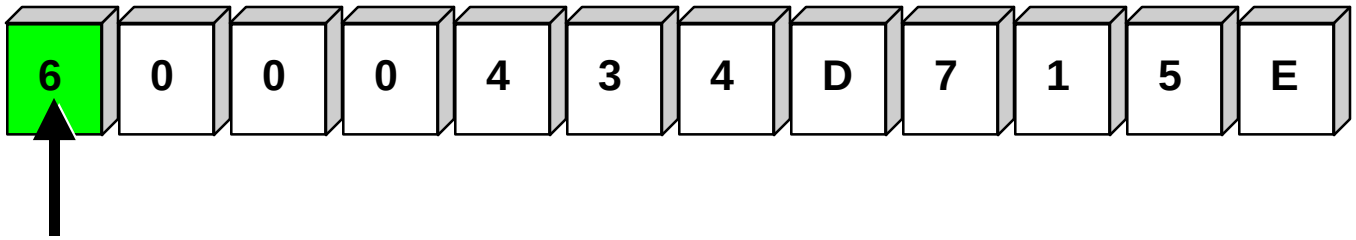


If IEEE-registered addresses are installed in every LAN card, how can we use local addressing?

The answer is simple. When the chipset on a LAN interface is activated, it reads the MAC address from the chip, and makes it available to the communications software. This software can optionally re-write the address with another address that is either provided manually, or (in the case of DECnet, for example), is generated from the Network Layer address (which is, in turn assigned manually).

Most modern LAN chipsets allow multiple MAC addresses to operate simultaneously.

Local Addressing - Why?



Binary Representation= 0<u>1</u>10

G/L Bit

For a locally administered address, we set the G/L bit to “warn” other end stations that this address does not have global significance.

But why should we bother to use local addresses when the IEEE procedure guarantees that addresses will never be duplicated?

There is no single answer to this question, just a set of industry stories...

FORE SYSTEMS

- **IBM’s Functional Addressing**

One instance of re-addressing is within higher levels of SNA functionality. IBM defined functional addresses for specific SNA devices, and these are allocated specific group address bands.

- IBM’s Functional Addressing
- **DECnet Addressing**

In a DECnet node, the Network Layer address that is entered by the network designer is used to form a MAC Layer address, which is then written back to the interface chipset.

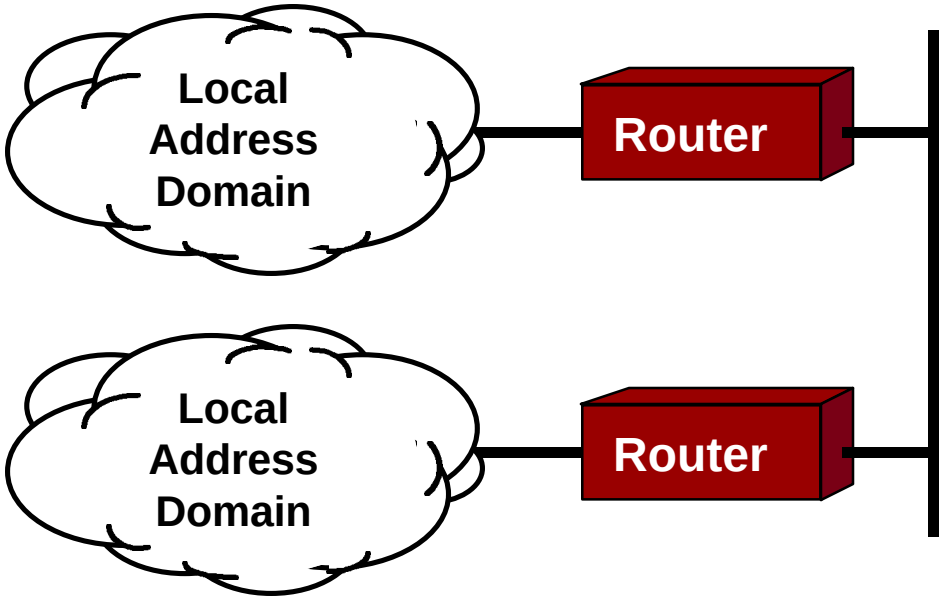
Similar procedures are adopted by other protocol stacks.

FORE SYSTEMS

- IBM’s Functional Addressing
- DECnet Addressing
- **Non-registered manufacturers**

An IEEE address block costs money, and has an administrative overhead associated with it. Some manufacturers don’t bother to register.

Local Addressing: Scaleability



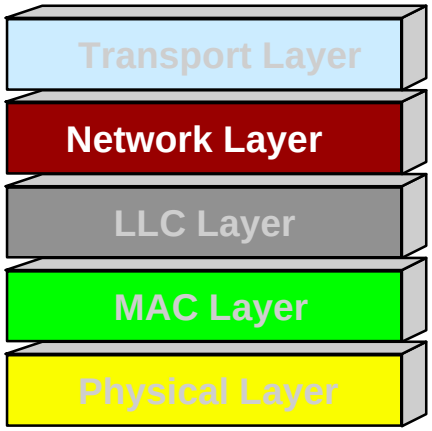
If we adopt local address administration, we may be able to build networks within our own domains of control. Perhaps this domain consists of the building in which we work, or even just the floor where our workgroup is located.

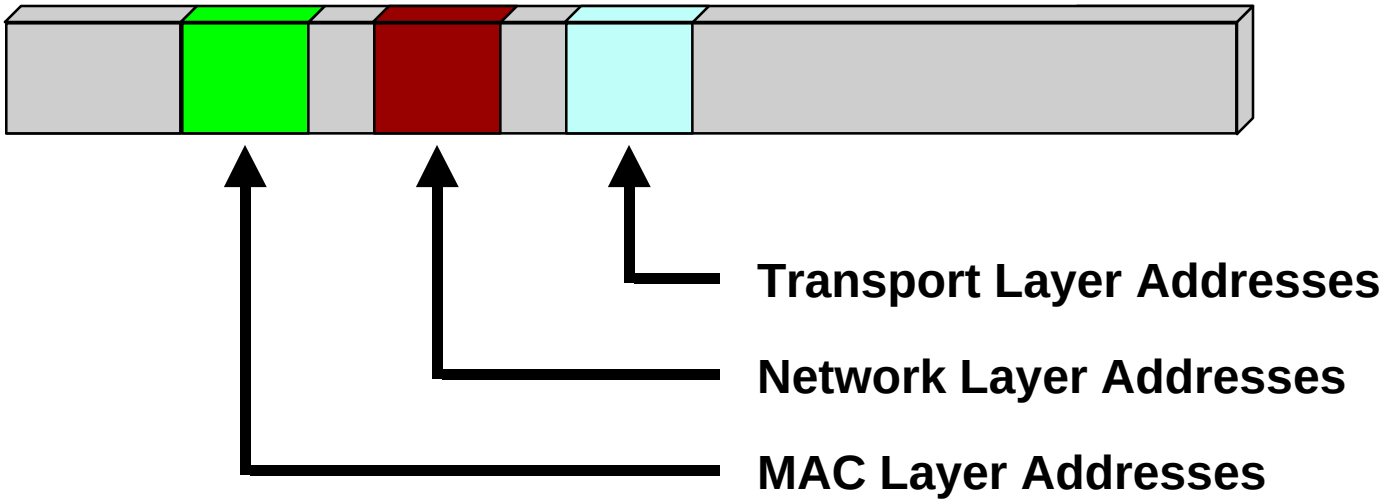
In order to connect local address domains without worrying about MAC Address duplication, we use *Routers*.

Routers make their decisions based on Network Layer Addressing.

FORE SYSTEMS

Network Layer Addressing





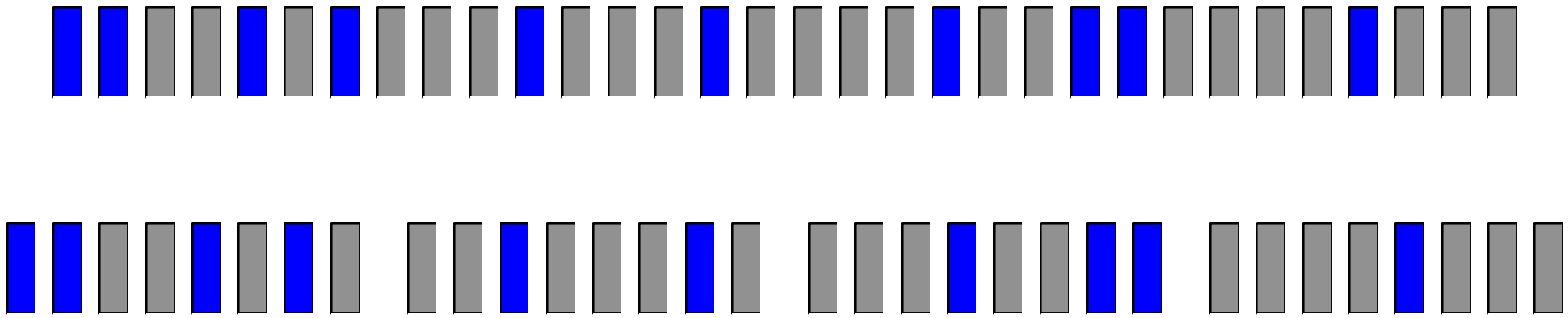
Network Layer addresses are found inside the Data Field portion of the MAC frame.

NOTE: This follows an OSI Model convention; for a Layer N structure, all addressing at higher layers is treated as data by Layer N. So Layer 3 addresses are treated as data by Layer 2, Layer 4 addresses are treated as data by Layers 2 and 3, etc.

FORE SYSTEMS

For IP, the structure of the address is relatively simple.

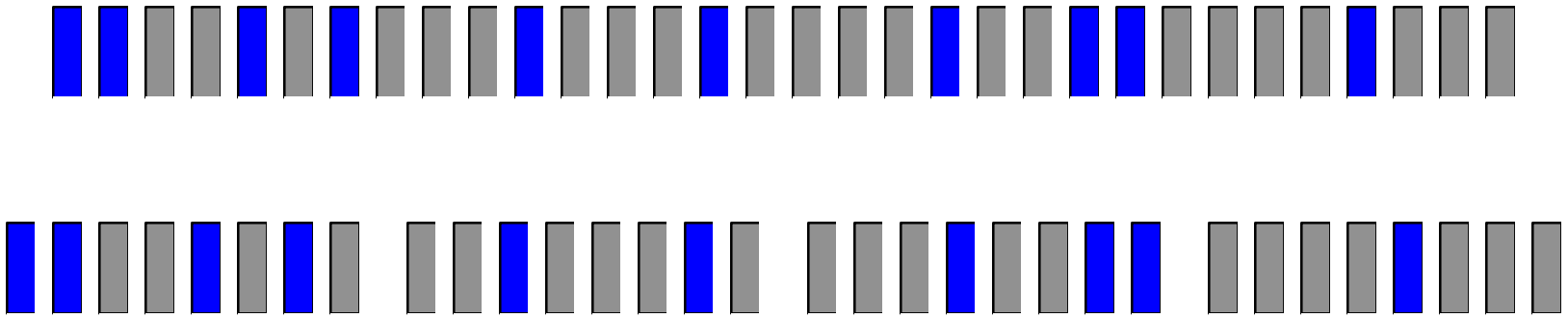
We take a 32 bit address.



For IP, the structure of the address is relatively simple.

We take a 32 bit address

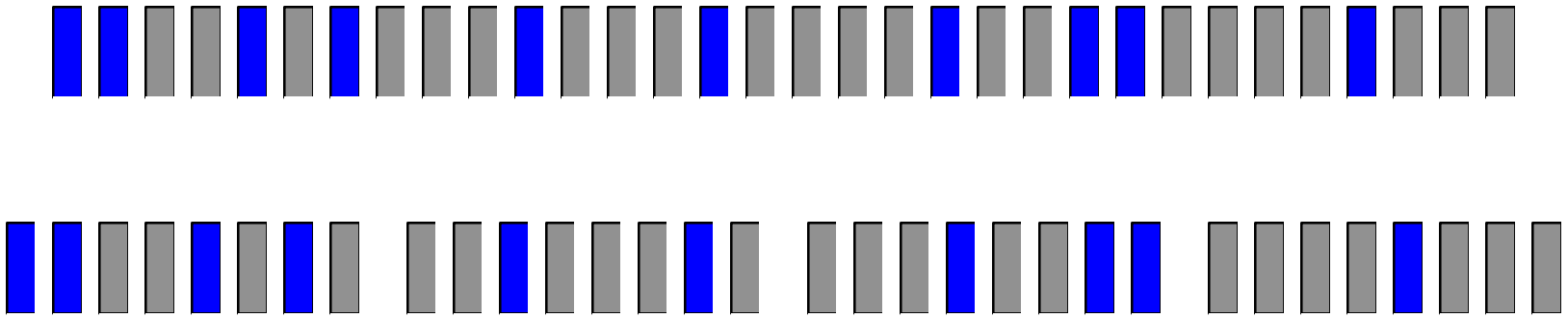
Divide it into 4, 8-bit fields.



202 34 19 8

For IP, the structure of the address is relatively simple.
We take a 32 bit address.
Divide it into 4, 8-bit fields..
Then we evaluate each field separately in decimal.

FORE SYSTEMS



202 . 34 . 19 . 8

For IP, the structure of the address is relatively simple.

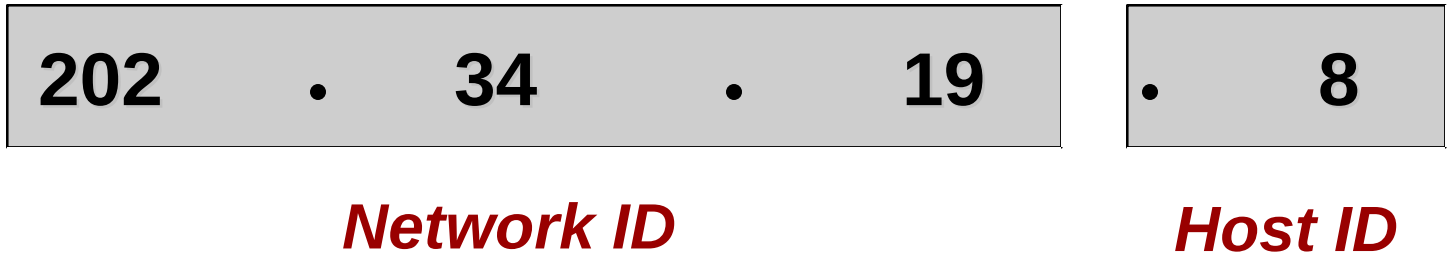
We take a 32 bit address.

Divide it into 4, 8-bit fields.

Then we evaluate each field separately in decimal.

And we write down these values with the individual byte-fields separated by dots. This is called *dotted decimal notation*.

FORE SYSTEMS

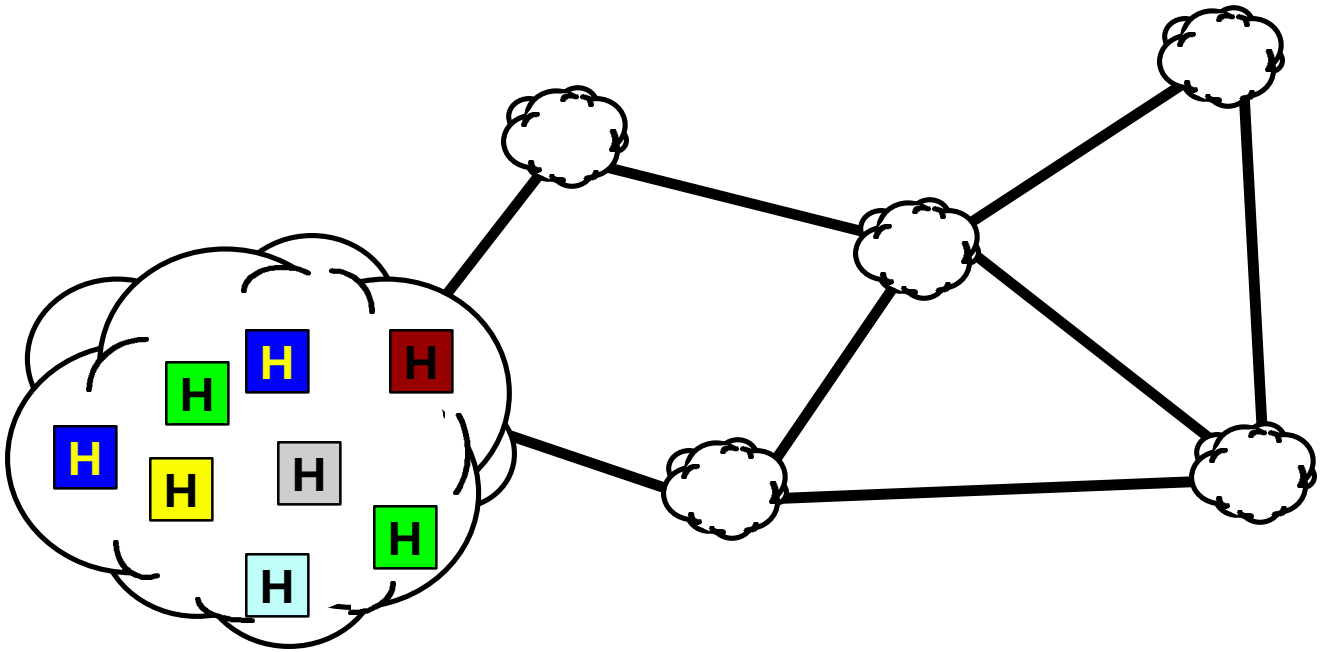


IP addresses have an additional structural element. Part of the address is reserved to indicate the *Network ID*, while the semainder of the address represents the *Host ID*. The relative sizes of the Network and Host ID fields vary with the class of IP address.

Using the Network ID, routers can direct traffic over multiple hops until it reaches the correct network.

The final router in the path will use the Host ID to perform an *Address Resolution*, and find out the correct MAC address of the destination host.

FORE SYSTEMS

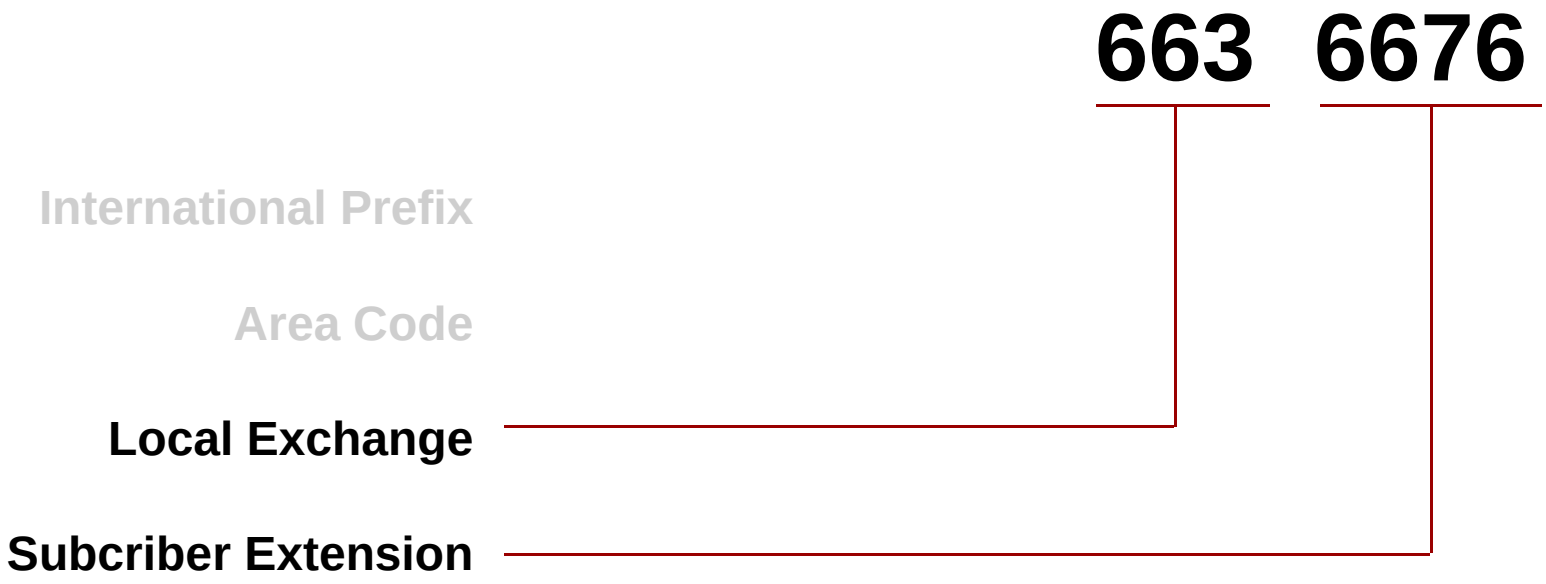


In the case of the worldwide Internet, there are millions of hosts already attached, and the connection rate is still increasing.

Without a hierarchical form of addressing, then internetwork routers would need to remember where every individual host was located.

With hierarchical addressing, each router only needs to track the hosts that are connected to networks on the router.

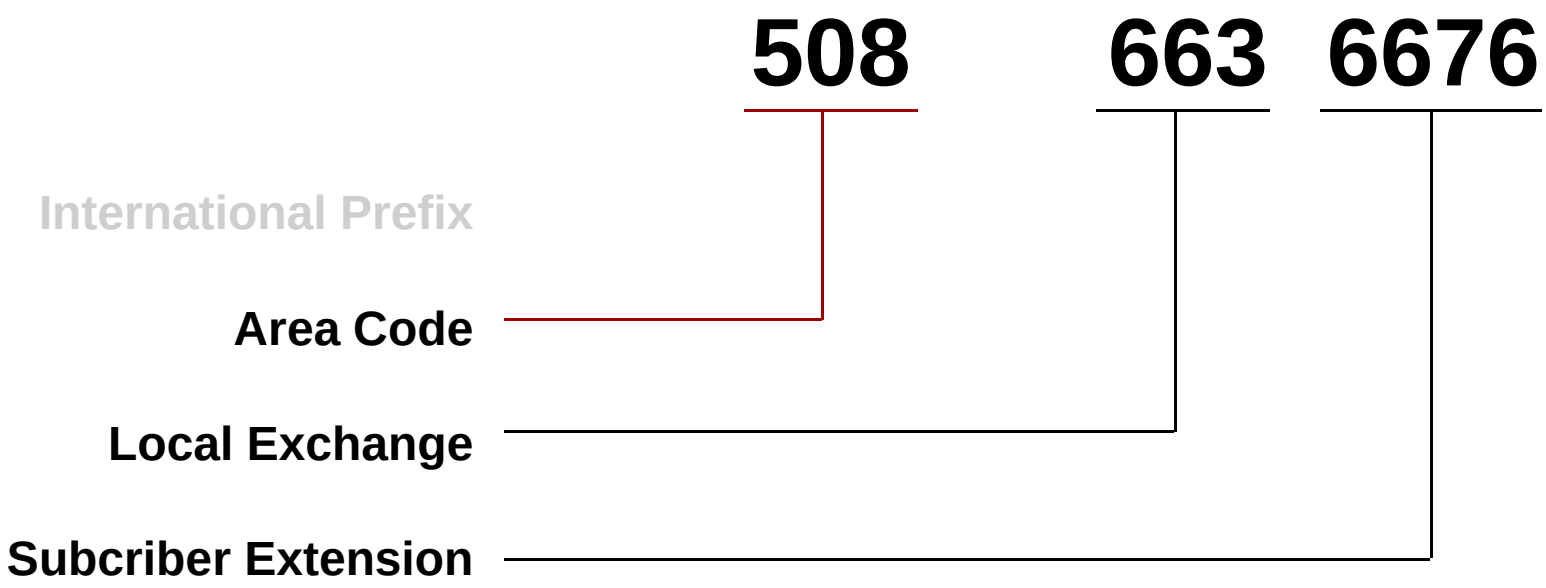
Hierarchical addressing is used in another global network - the Telephone System.



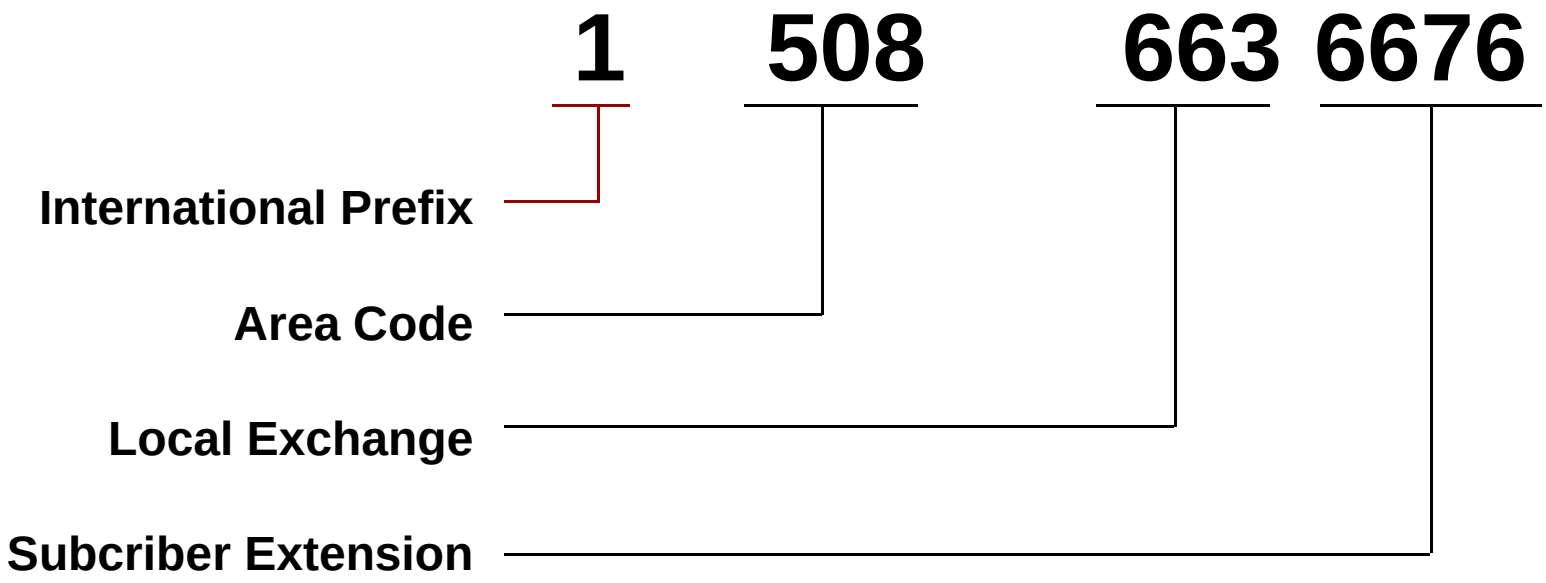
This is a telephone number in the USA. It has 7 digits, which means that up to 10 million subscribers can be addressed individually.

This is a lot, but not enough for a national, or international addressing scheme.

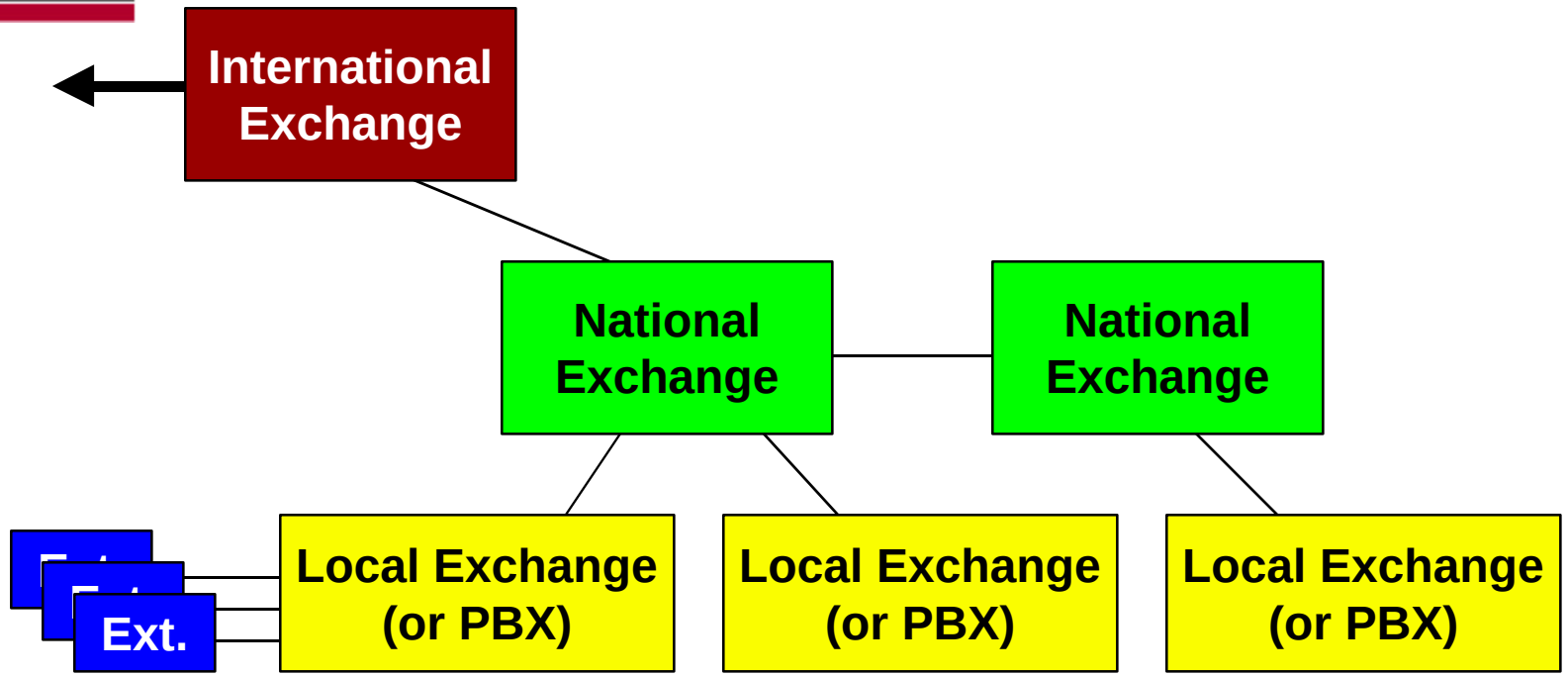
Even the seven digits are actually divided into the *Local Exchange*, and the *Subscriber Extension*.



To extend the numbering scheme, US numbers add an *area code*.
Area codes are three digits long.



If we want to call this number from outside the US, we need to add the *International Prefix*. For the USA, this is 1, for the UK, 44, for Germany 49 etc.

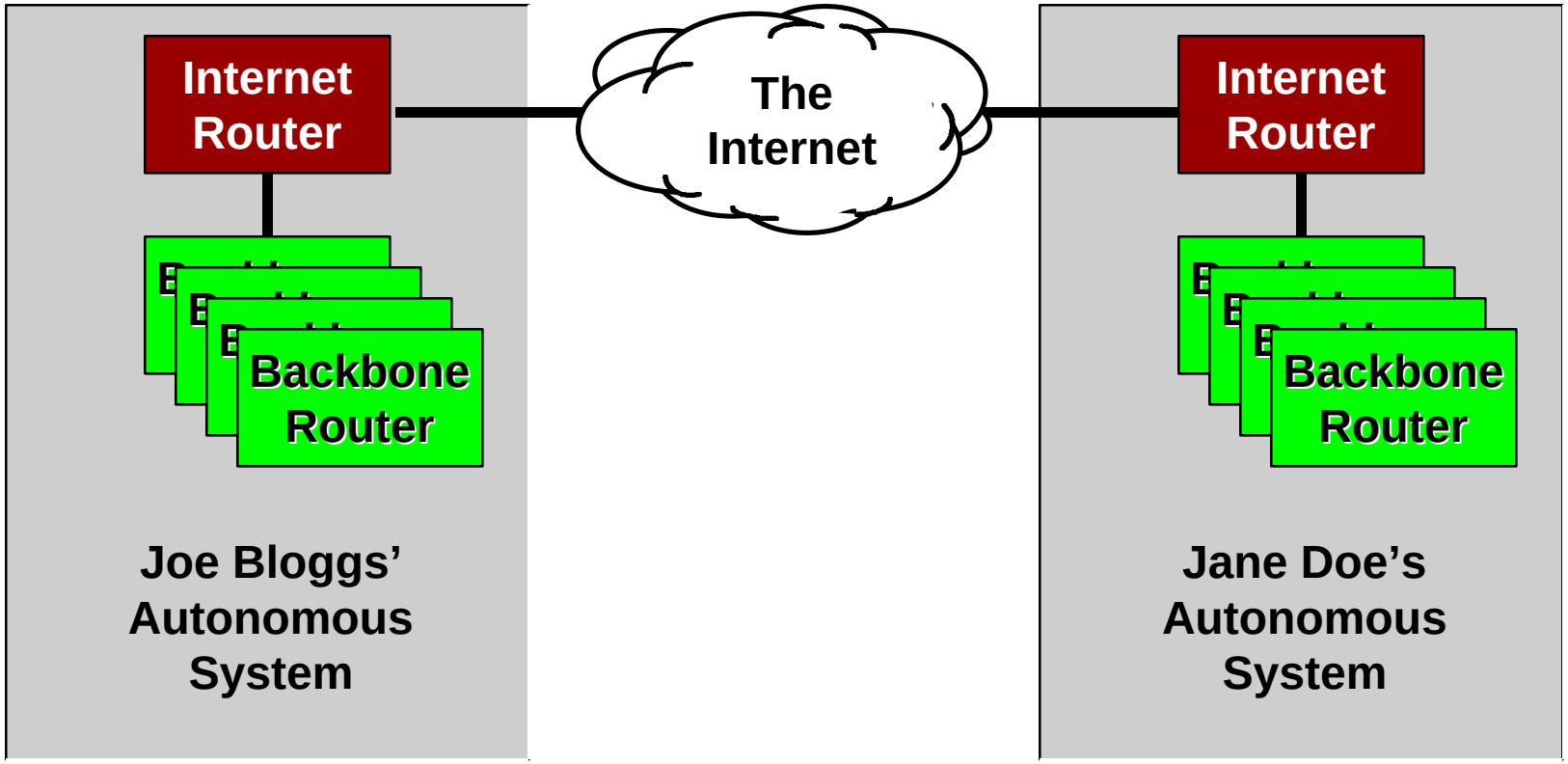


This hierarchical numbering scheme is essential to the telephone system. By isolating the scope of individual telephone numbers, we gain a number of advantages.

First, human users of the system only need remember seven digits for any local number.

Second, a national PTT can adopt any reasonable internal structure for its numbering. It is “protected” from address duplication and confusion by the International Prefix.

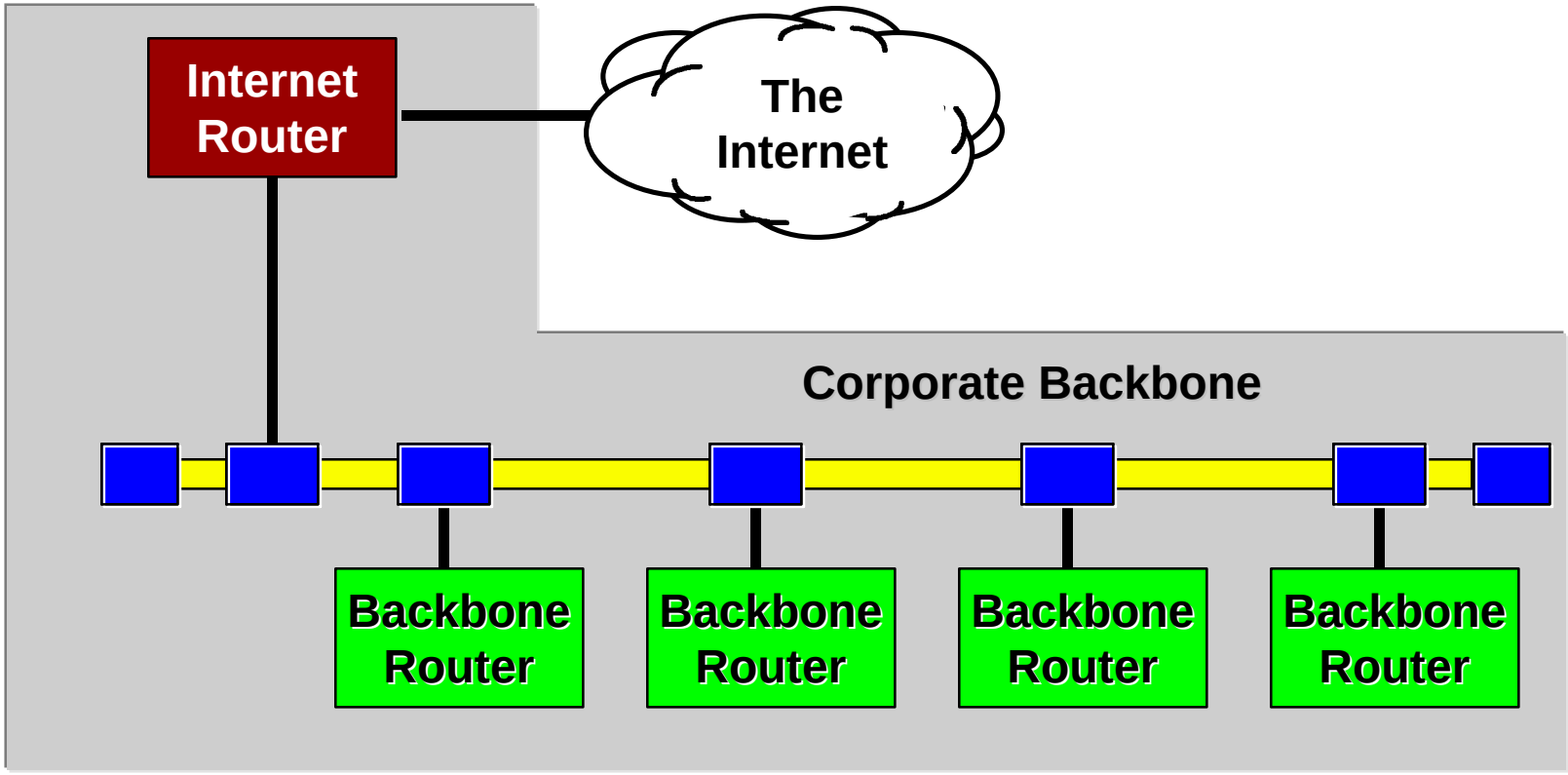
Finally, and most important, any given telephone exchange only needs to know about addresses below it the hierarchy.



I'd like to concentrate on this final advantage, because it's the primary reason we use hierarchical addressing in data networks.

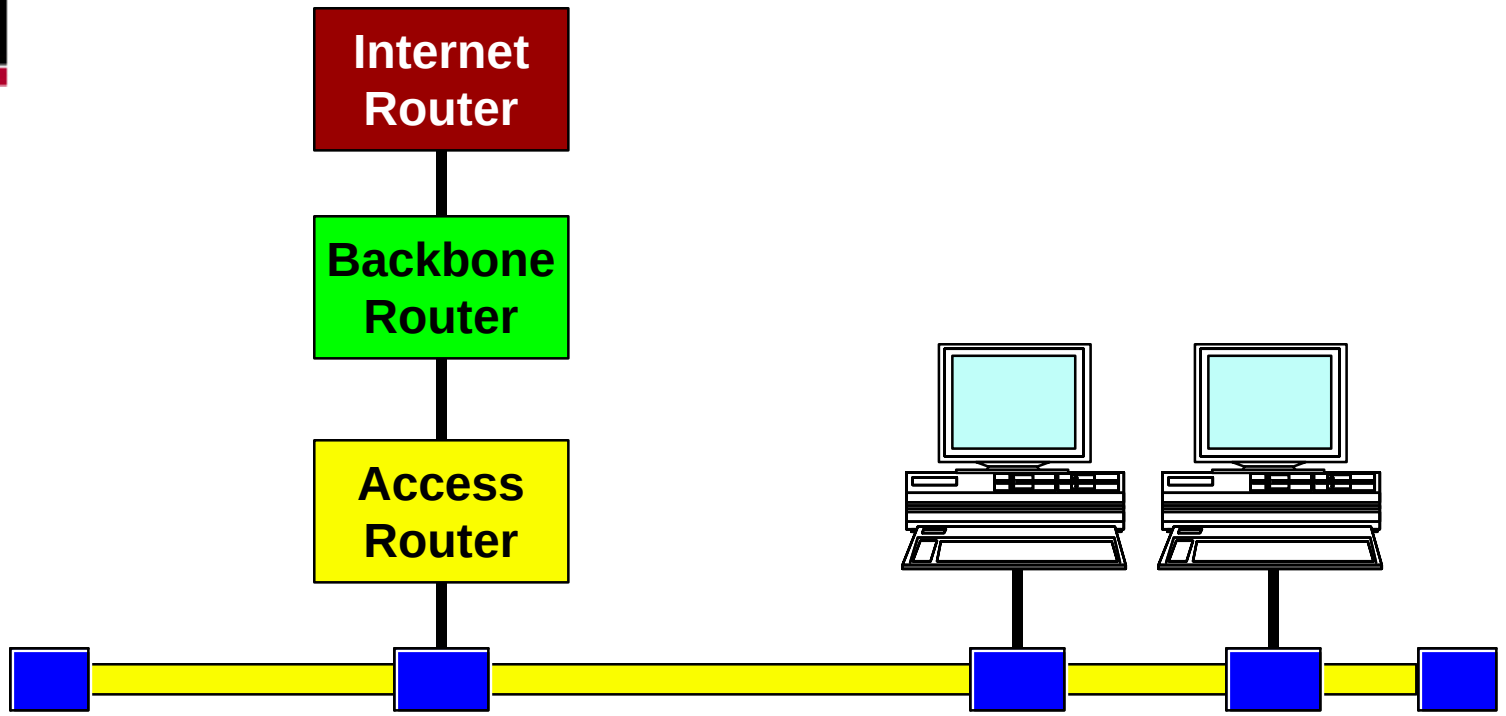
The most obvious evidence of address hierarchy is that used in the Internet for protecting one subscriber from routing errors made by another subscriber. This concept is, of course, the *Autonomous System* (AS).

Internet-attached routers must recognise AS concepts, and must terminate local routing protocol updates such as RIP or OSPF.



The next level in the hierarchy are the Backbone Routers, used to build the Corporate Backbone.

Backbone routers operate within an AS, but may need to maintain large routing tables depending on the size of the individual organisation.



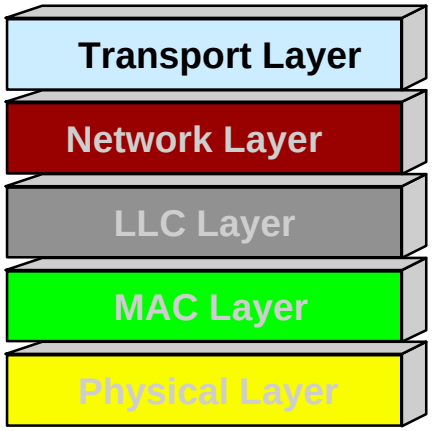
At the lowest level of router hierarchy are the Access Routers. These devices are much smaller, less powerful, and cheaper than their more complex cousins.

Access Routers may be used to connect a single LAN workgroup into the Corporate Backbone, and don't need to maintain complex routing tables.

Regardless of the type of router used, all of these devices make their switching decisions based on the Network Layer addressing I have just described.

FORE SYSTEMS

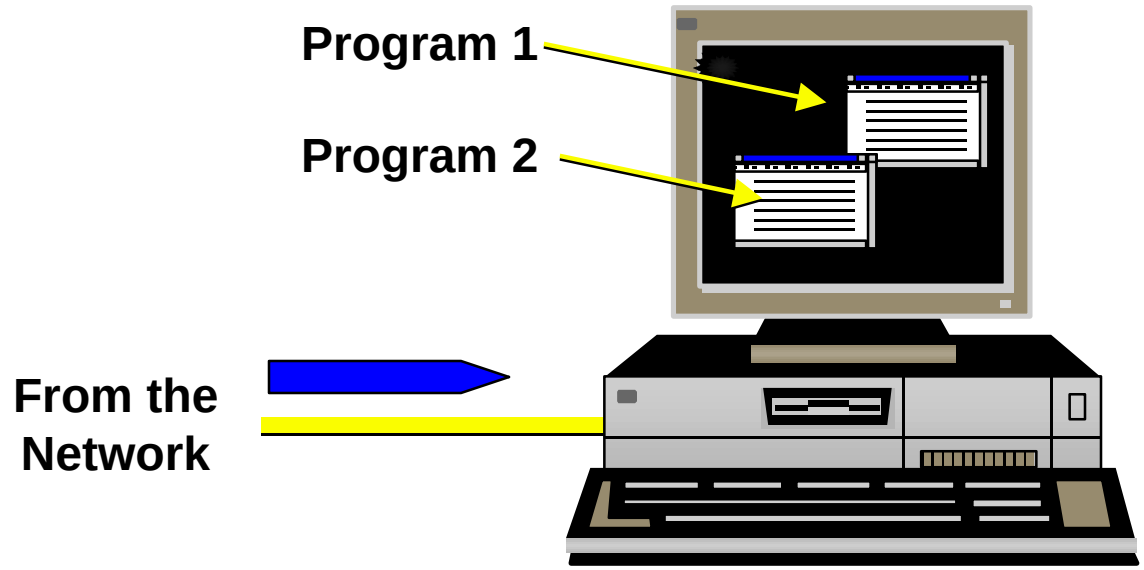
Transport Layer Addressing



Finally I'd like to look at Transport Layer addressing.

Just to recap, we can say that MAC Layer addressing allows us to transfer messages between two hosts on the same cable.

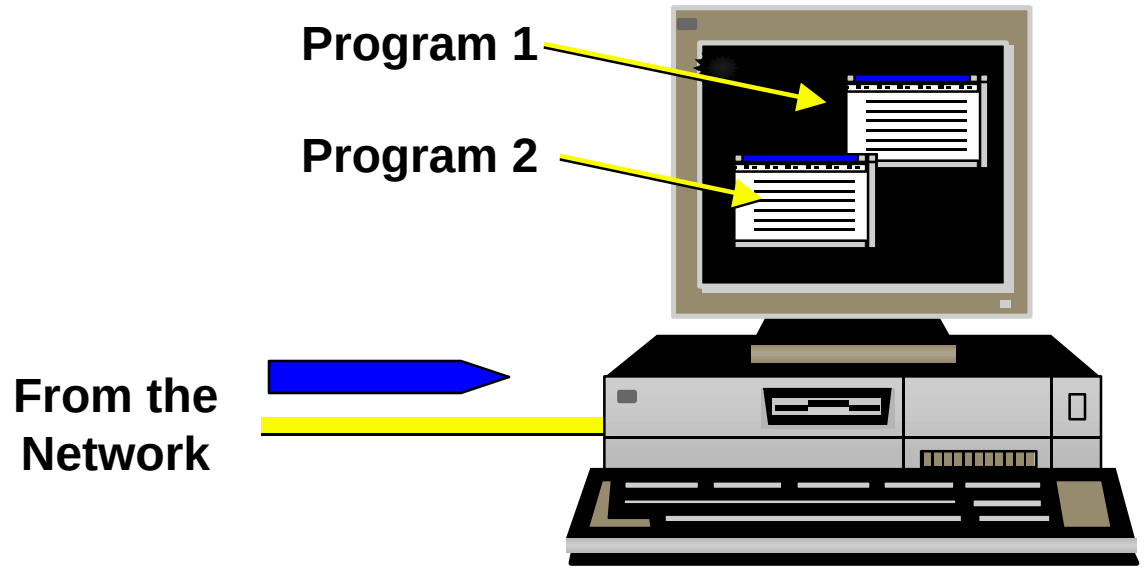
Network Layer addressing extends this communication ability so that we can cross multiple intermediate networks to get from one host to another. Network Layer addressing is also scaleable because the network designer can choose the addresses in a hierarchical way.



Here we see a LAN frame heading towards a PC from the network. MAC and Network Layer addressing have got the frame this far, but now there's a problem.

There are two possible communication programs running in the PC - Program 1 and Program 2.

The MAC and IP addresses on the PC only identify the machine itself, not the program to which the packet should be sent.



To differentiate between these programs, we use Transport Layer addressing.

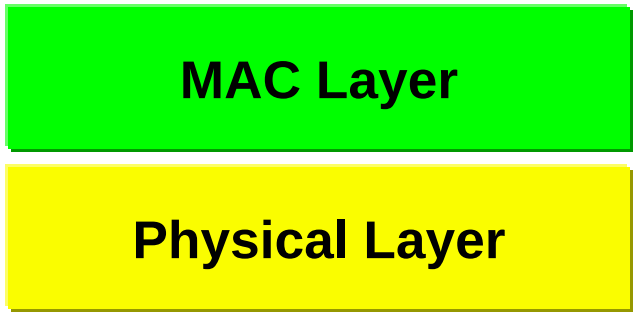
Note that it's not really practical to use IP addresses on a per-program basis for a couple of good reasons. First of all, you'd have to register each program with an IP address when it started. Because IP addresses are often assigned manually, the Network Administrator would have to limit the number of programs you can run from a machine so she would be able to pre-assign your IP addresses.

More critically, there are too few IP addresses to really do this in practice.

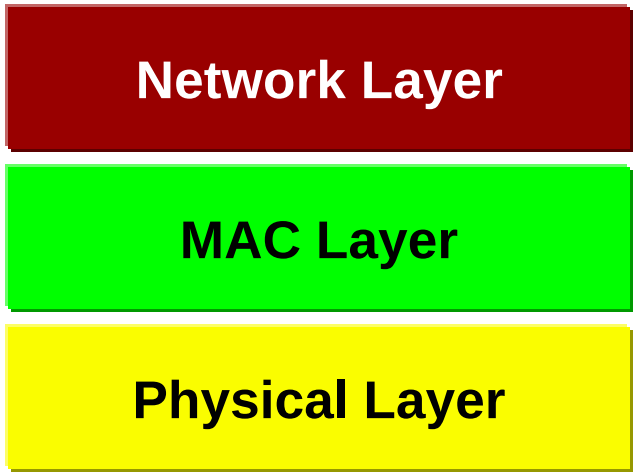
FORE SYSTEMS

In Summary...

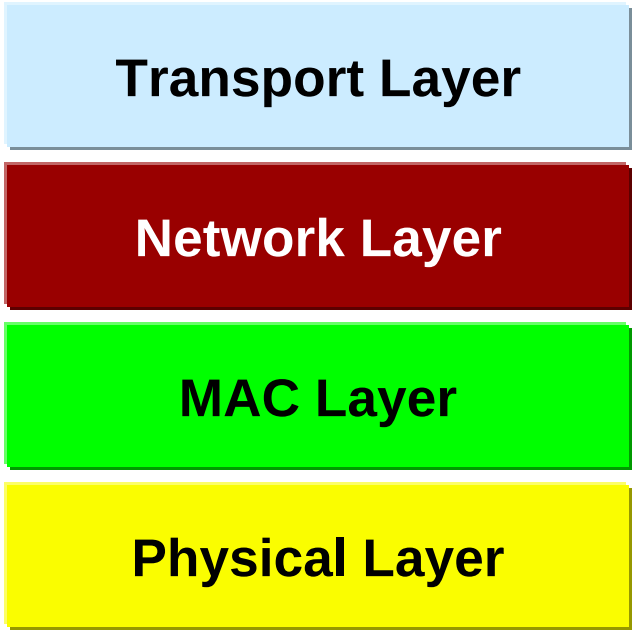
FORE SYSTEMS



MAC Layer Addresses are used to allow private communication between specific hosts, even though they share the same communication channel with many other systems.



Network Layer Addressing allows communication between hosts regardless of the type of network (or networks) that are used to connect the hosts.



Transport Layer Addressing allows a specific application process running in a host computer to communicate with an equivalent process running in another host.

The End

This concludes the tutorial.

If you aren't viewing this tutorial on the FORE Systems' ATM Academy Site, then you can find additional tutorials at:

http://academy.fore.com