Understanding TCP / IP

TCP/IP is the language of the Internet, and is a cross-platform protocol despite its long association with Unix. Because of the rise in importance of the Internet and connectivity, it makes sense to consider using IP in your intranet or LAN - or at least being aware of its implications.

By Liam Proven
IT Journalist

TCP / IP (Transmission Control Protocol / Internet Protocol, or IP for short) is the name given to a whole group of related protocols which comprise the language of the Internet. Although there's nothing intrinsically better about TCP / IP relative to better-known LAN protocols such as Novell's IPX / SPX or Microsoft's NetBEUI, it is rapidly becoming the *de facto* standard network protocol for one simple reason - the Internet.

IP has gone through multiple versions since its original development. Currently, version 4 is by far the most widely used. However, there are later revisions. Version 5 was never released, but its successor, once termed IPng (IP next generation) but now ratified as IPv6, is out there and will gradually replace the current version (IPv4). This is undoubtedly going to be a tremendous pain for everyone involved, as the changes are major, but it will be necessary. IPv4 uses 32-bit addresses, allowing for a theoretical maximum of 4,294,967,295 unique addresses. In October 1999 the number of human beings passed six billion, and the number of computers probably isn't too far behind - and one day, they might all need to be connected.

Clearly, 32-bit addresses won't be enough for very much longer, and this is the driving reason for IPv6, which uses 128-bit addresses, allowing a startlingly vast range of addresses: approximately 3.402824×10^{38}. Estimates vary, but this should be rather more than enough to allow every atom in the universe a unique IP address. Although the other changes between these versions are mostly minor and internal, the two protocols are not directly compatible; though they can share a network, IPv4 nodes and IPv6 nodes cannot directly communicate. Changing from one to the other is therefore a substantial task, and the transition will cause a comparable amount of trouble to the Y2K bug - and will come only a few years later. Right now, however, it is IPv4 that we must deal with, and that's what we will look at here.

not true

IP And Your LAN

Because the Internet is becoming so widespread as to be nearly universal, it is also becoming more useful in business. As more companies get connected, the viability of the Internet for business-to-business communication increases. Similarly, as more people use the Internet for personal or leisure purposes, its value as a way of reaching customers grows. Finally, even if neither of these appeals, the standardisation on Internet communications protocols and the fact that much Internet software is free means that, even for purely internal systems, businesses can reap significant cost savings by using Internet technologies.

As the Internet runs over IP, so do Internet-based applications. Whereas proprietary email systems such as Microsoft Mail use other, protocol-independent means of communication (such as shared file systems), Internet-based email programs communicate over IP, so client machines need an IP-based connection to the

Address Class	First Octet	Network Mask
A	1. to 127.	255.0.0.0
B	128. to 191.	255.255.0.0
C	192. to 233.	255.255.255.0
D	224. to 239.	None

Figure 1 - Summary of Internet address classes.

server. For systems which require other protocols, such as older versions of Novell NetWare, it is possible to "tunnel" IP over other protocols - for example, by encapsulating IP packets inside IPX packets. If the client machine's network stack hides this behind a standard API, such as Windows' WINSOCK, IP-based applications can run unmodified. As all major client and server OSes today support IP natively, even alongside other protocols, there's little reason to do this, although it may be used for making secure, encrypted connections over public networks.

How It Works

The snag is that building an IP network requires significantly more planning than when using most other protocols. IP was developed in the 1960s for linking disparate networks - separate in both a geographical sense and in the sense of running different, incompatible systems. Protocols such as IPX and AppleTalk, intended for small LANs, are inherently simpler.

Addresses

The first issue is IP addressing. Each device on an IP network requires a unique address. Unlike in other protocols, this is not automatically generated from the hardware (MAC) address; it must be manually assigned. The word "device" here is important. It does not mean each computer; IP addresses go by network port. For example, a server with two Ethernet cards (such as a firewall) would need two addresses, one per interface. Similarly, a machine with both a network card and a modem (or terminal adapter) requires addresses for both. To make matters even worse, it's possible to give one port multiple addresses, a technique called "multihoming". For instance, this allows a single machine to host several separate Web sites; each hostname points to a different address, but all refer to the same machine.

The address is divided into two parts: the network number and the host (or machine) number. All hosts on the same IP network must share the same network number, and no two hosts may share the same host number.

Subnet Masks

Alongside the address, each port requires a subnet mask. This value is used to split the complete address into network and host parts; in other words, to determine whether other IP addresses are on the local network or a remote one. These two values are the absolute minimum. Using these, a machine will be able to communicate with others on the local network if the other machine's IP address is known. Additional information is usually required, though, to be able to access nodes on other networks, to access machines by name rather than number, and so on.

"The standardisation on Internet communications protocols and the fact that much Internet software is free means that, even for purely internal systems, businesses can reap significant cost savings by using Internet technologies."

Gateways

For direct access to networks beyond the current one (which isn't always required), each machine must be told the IP address of the router (or gateway) that connects the local network with the wider world.

Name Servers

For a small, server-based network with only one or two servers, access to them by their numeric IP address may be sufficient, but usually it's desirable to use names instead. The most basic way of doing this is via a local configuration file called hosts. As a minimum, this contains a pair of entries per line, separated by spaces; first the address, then the corresponding name. However, for all but the

	8	16	24	32
Class A	Network Number 1.-127.	Host number 0.-255.	Host number 0.-255.	Host number 0.-255
Class B	Network Number 128.-191.	Network Number 0.-255.	Host number 0.-255.	Host number 0.-255
Class C	Network Number 192.-223.	Network Number 0.-255.	Network Number 0.-255.	Host number 0.-255

Figure 2 - Network and host numbers by class.

most trivial of networks, keeping all the local files updated rapidly becomes a logistical nightmare, and it is desirable to set up a central server to resolve names to addresses. For this, one or more name servers must be set up, and each client machine configured with the name servers' addresses. Name servers accept requests from the clients containing the name of a machine, such as www.cix.co.uk, and return the matching IP address. The industry standard system for this is the Domain Name Service (DNS).

Although IP was designed to be a cross-platform protocol, for many years it was mainly used on Unix, while mainframes, minicomputers and PCs used proprietary protocols (such as SNA, DECnet and NetBEUI respectively). IP was thus sometimes perceived as the Unix protocol. On Unix, the *de facto* standard package for providing DNS is the Berkeley Internet Name Daemon (BIND). Because, on Unix, DNS and BIND go hand-in-hand, the two abbreviations are occasionally and incorrectly used interchangeably. As it is such a fundamental part of an IP network, both functionally and as a performance bottleneck, most IP stacks expect to be supplied with the addresses of at least two DNS servers - a primary and a secondary.

However, DNS configuration is complex and the full functionality is not usually needed for a small LAN. Also, traditional DNS is static and does not cope gracefully with addresses that may change. For this reason, in Windows NT Server (both versions 3 and 4), Microsoft implemented its own proprietary system to deliver basic name-resolution services: the Windows Internet Name Service (WINS). WINS only works with Windows clients, but is far easier to configure than BIND. It automatically builds a table of machine names using NetBIOS broadcasts and, with a simple GUI, allows static addresses - for instance, of Unix servers - to be added to the database. Versions of Windows since Windows NT therefore expect WINS. Windows for Workgroups pre-dated Windows NT, but the additional 32-bit IP stack for Windows for Workgroups 3.11 came later; this and subsequent versions (such as Windows 95, Windows 98 and Windows NT Workstation) have fields in the configuration dialog for WINS servers. Windows NT even complains if you click the OK button and these fields are left blank.

Implementation: Address Ranges And Subnets

The first step in building an IP network - or adding IP to an existing system - is to determine the address range to be used. Many administrators unfamiliar with IP get this critical step wrong, and it can cause great problems later. In IPv4, addresses consist of a set of four eight-bit values. As each individual bit can be significant, rather than the value represented by each set of eight, these are strictly speaking not bytes but octets. Nonetheless, the four octets of an address or mask are usually written as decimal values, separated by full stops - the dotted-quad notation, such as 193.54.7.18. The problem is that these numbers are meaningful. Firstly, certain values are reserved and may not be used. 0 refers to an entire network; for example, 192.168.24.0 means the range of addresses from 192.168.24.1 to 192.168.24.254, and 192.0.0.0 refers to the 192.0.0.1 to 192.255.255.255 range. A machine therefore may not be given an address ending in 0. Similarly, 255 is the "broadcast address": a packet sent to 192.169.24.255 will be picked up by all machines in the 192.168.24.0 network. Thus, 255 may not be used in the address.

"Each device on an IP network requires a unique address. The word "device" here is important. It does not mean each computer; IP addresses go by network port."

Secondly, every port on every device on the Internet must have a unique number. Addresses are regulated, with blocks being allocated to organisations by controlling authorities - the InterNICs. It is, therefore, "illegal" to just pick numbers out of the air. You should apply to the NIC (or your ISP), giving them an estimate of the future size of your network, and they will allocate a block (or blocks) of addresses to you. These blocks come in three sizes: class A, class B and class C, in diminishing order of size. Think of the class as determining how many octets of each address are fixed.

Class A ranges use only the first octet to identify the network, and this lies in the range 1 to 126 (ie, 1.0.0.0 to 126.0.0.0); the matching subnet mask is 255.0.0.0 (see Figures 1 and 2). There are 224 (16,777,216) addresses in a class A network. Note that the 127.0.0.0 range is reserved for loopback (the internal logical IP network via which any machine running IP may address itself). All 125 of the class A ranges have been allocated. Class B ranges use the first two octets for the network

2^{24}

number, and the first octet must be in the range 128 to 191; the subnet mask is 255.255.0.0. There are 216 (65,536) addresses in a class B network. Most of the 16,382 class B ranges have been allocated. Class C ranges use the first three octets for the network number, and the first octet must be between 192 and 223. There are 28 (256) addresses in a class C range. There are also two special classes which are not normally assigned. The class D range (between 224.0.0.0 and 239.0.0.0) is used for IP multicast, a form of broadcasting. Finally, class E (Experimental) reserves values from 240.0.0.0 to 255.0.0.0, which currently are not used.

The most common size is a class C address. This fixes the first three octets, leaving only the last mutable; for instance, 193.54.7.x. As the .0 and .255 host addresses are reserved, this allows 254 addresses, from 193.54.7.1 to 193.54.7.254. The corresponding subnet mask is 255.255.255.0. Bitwise, it works as shown in Figure 3. The subnet mask "blanks out" the fixed part of the address (the network number), leaving just the local part (the host number). This, the simplest form of subnet mask, uses all ones or zeros within each octet; thus, subnet boundaries are also octet boundaries. However, a network can also be split into sub-units within an octet - so, for instance, dividing a single class C range into two parts. This is where subnet masks can become really useful - and really difficult to understand, at least in decimal notation. The example in Figure 4 translates to a subnet mask of 255.255.255.192 and two address ranges: 192.54.7.64 to 192.54.7.127, and 192.54.7.128 to 192.54.7.254. For historical reasons, which no longer strictly apply, subnets should always use at least two bits out of an octet.

Private Ranges

The next step is to choose the range of addresses you will use. The "official" way to do this, mentioned earlier, is to apply to a NIC for a range. In practice, it's now more common for you to be allocated one by your ISP, which has already purchased a whole set of ranges. Unfortunately, many people implementing IP don't know this and just make up a range, such as 100.100.100.0. This will work as long as the network isn't directly connected to the Internet. However, if - or when - it is, a working configuration suddenly goes wrong. As this range isn't private, there may be real hosts out there somewhere on the Internet using these addresses, and a local server address of 100.100.100.54 suddenly also points to another machine somewhere else in the world. Depending on how the Internet connection works, things start to fail. At best, when the link is open, machines on the internal network can no longer access that server - an intermittent fault, and those are always the hardest to trace. At worst, the server itself may detect a clash of IP addresses and fail.

Happily, it is not strictly necessary to reserve a range. The designers of IP anticipated this problem and set aside blocks of addresses for internal networks - the private ranges. There are three private ranges: one class A, one class B and one class C (see Figure 5). All you need to do is choose the one of appropriate size for your network. For most small LANs of under 255 machines, the private class C range is the best, even though the private class A range of 10.x.x.x is easier to remember. As these addresses are reserved as private, no hosts on the Internet will ever use addresses in any of these ranges. Similarly, the main routers on the Internet backbone will not pass packets with such addresses. There will be many other networks using the same ranges, but they can never clash with one another.

"On Unix, the standard package for providing DNS is the Berkeley Internet Name Daemon (BIND). Because, on Unix, DNS and BIND go hand-in-hand, the two abbreviations are occasionally and incorrectly used interchangeably."

	Octet 1	Octet 2	Octet 3	Octet 4
Bit number	12345678.	12345678.	12345678.	12345678
Minimum address	11000001.	00110110.	00000111.	00000000
Maximum address	11000001.	00110110.	00000111.	11111111
Subnet mask	11111111.	11111111.	11111111.	00000000
Significant bits (in the subnet)	00000000.	00000000.	00000000.	11111111

Figure 3 - Bitwise representation of a class C address.

If an illegal range is used, it's not necessarily the end of the world. There are ways around it - either avoiding a routed connection between the network and the Internet, or using a smart router which can translate on-the-fly between illegal internal addresses and legal external ones, a technique called Network Address Translation (NAT). Use of NAT is actually commonplace, although usually for security reasons rather than to repair earlier mistakes.

Today, intermediate networks (ones of between a few hundred to a few thousand hosts) are being allocated multiple class C (256-address) ranges rather than single class B (65,536-address) ones. This is because the total IPv4 address space is rapidly filling up. In the early days, companies were readily assigned class A ranges - in other words, their own first octet. Although there are less than 255 possible class A ranges, there probably aren't that many companies in existence which really require sixteen million machines visible on the Internet! Thus, vast ranges of potential addresses were effectively wasted, and efforts are afoot to make best use of the remaining space. Similarly, if your network is unlikely to ever exceed 255 machines, don't use the private class A or class B ranges unnecessarily. If you need to link up multiple networks into a WAN and you are using private ranges, you don't need a single big range to embrace them all unless there are more than 255 of them. It's preferable to use multiple private class C ranges and alter the third octet - for instance, the London office might use 192.168.1.0 and Edinburgh 192.168.2.0.

It's not usually a good idea to link private company LANs over the public Internet, for obvious reasons. For simple point-to-point links, either over ISDN or permanent leased lines, it doesn't matter what ranges you are using. However, if you wish to make a routed connection between a private network and the Internet, you will need to use routers that support NAT. For security and performance, in any case, it's generally preferable to use proxy servers, firewalls, or both.

Address Allocation

Once you have chosen the address range (or ranges) that you will use, the next job is allocating them - doling out addresses to individual machines. The simplest way to do this is just to go to each machine and configure it with its address - which is fine if there are only a handful of machines to set up. However, most server-centric networks are larger than this, with only a few machines that are accessed by all the rest. For such purposes, the addresses of the servers must be known to all machines, but those of individual workstations are irrelevant, as other machines will not routinely be connecting to them. This means the servers need to have static addresses (ones which are permanent) but workstations need not: their addresses can be given to them when they boot up, by a program running on a server. When a workstation shuts down or reboots, its address can then be released back into a pool of available addresses, and may later be given out to another machine when it boots.

"Addresses are regulated, with blocks being allocated to organisations by controlling authorities - the InterNICs. You should apply to the NIC (or your ISP), giving them an estimate of the future size of your network, and they will allocate a block (or blocks) of addresses to you."

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet mask	11111111.	11111111.	11111111.	11000000
Subnet 1	11000001.	00110110.	00000111.	01xxxxxx
Subnet 2	11000001.	00110110.	00000111.	11xxxxxx

Figure 4 - Dividing a single class C range into two parts.

Class	Start of Range	End of Range	Subnet Mask
A	10.0.0.0	10.255.255.255	255.0.0.0
B	172.16.0.0	172.31.255.255	255.255.0.0
C	192.168.0.0	192.168.255.255	255.255.255.0

Figure 5 - The private address ranges.

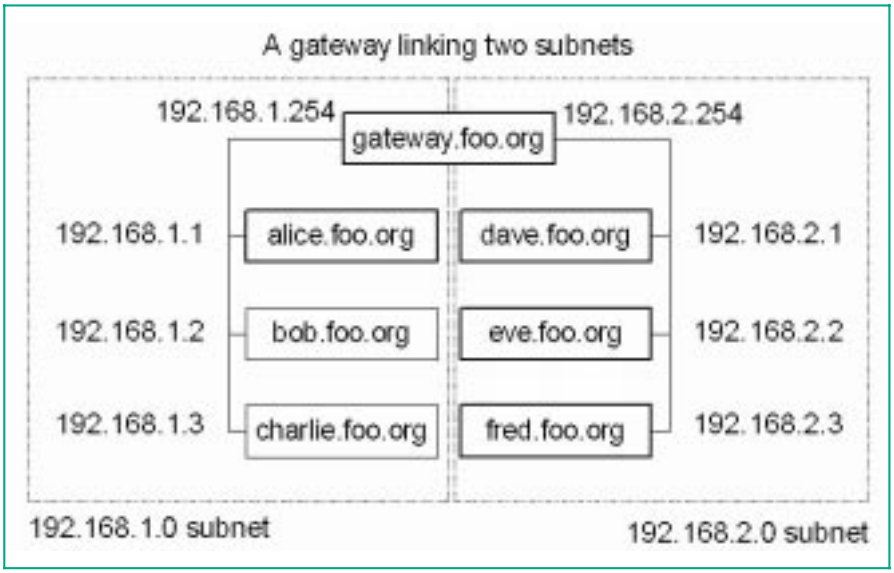

Figure 6 - A gateway linking two subnets.

This system relieves a great deal of the administrative burden. Rather than maintaining a list of all the addresses on the network and visiting each machine to set its address, you need only set a few fixed addresses, then set up a server to dynamically allocate addresses to workstations from a predefined range. Similarly, if workstations won't be accessed from other machines, they don't need to have individual entries in the name server. Although the operating system running on them may want a node name, no other machine need know it.

Allocation Protocols

Once again, modern PC operating systems start to diverge from traditional Unix systems here. For many years, Unix used a basic system for allocating IP addresses at system startup: the Boot Protocol (BOOTP). Like the simplified Windows-only name service, WINS, there's a simpler system, the Dynamic Host Configuration Protocol (DHCP), originally included with Windows NT Server. Unlike WINS, though, this isn't a Microsoft-only standard, and it is being widely adopted by other PC operating systems such as NetWare, MacOS, Linux and BeOS. DHCP is a superset of the older BOOTP system, which in time it will probably replace.

With DHCP, configuration is reduced to more or less the minimum level currently possible. The server needs only to be told the address range or ranges to put in the pool, and the client only that it should use DHCP to find its address. There's no need to tell the server the MAC addresses of the clients it will handle, or to tell the clients the address of the server; everything else happens automatically. DHCP doesn't only allocate addresses and subnet masks: it can also be used to inform clients of the location of name servers (both WINS and DNS) and gateways. DHCP servers are included with Windows NT Server, Linux and recent versions of NetWare. However, Windows NT Workstation, Windows 95 and 98 and MacOS do not, although third-party ones are available.

The hostnames would then be:

alice.london.foo.org
bob.london.foo.org
charlie.london.foo.org

and

dave.edinburgh.foo.org
eve.edinburgh.foo.org
fred.edinburgh.foo.org.

Figure 7 - See main text: giving subnets separate names.

Domain Names

Like IP addresses themselves, complete IP node names are divided into two parts: the name of the local network (or domain - not to be confused with Windows NT security domains), and a unique host name. For example, a simple two-node network called foo.org might contain two machines, alice and bob. These two machines' node names would therefore be alice.foo.org and bob.foo.org. When setting up a name server, then, the first thing to determine is the domain name of the network. Domain names (such as foo.org) must be purchased and, additionally, an annual fee is usually demanded from the name registrar or ISP to keep it active. Many ISPs also charge an additional fee for mail forwarding - capturing emails sent to the domain and redirecting them to the account-holder's mailbox. Recently, some UK ISPs have started offering free domain name registrations, but

the domain must be hosted with that ISP and transferring the domain to another ISP is costly. If you wish to use a unique company domain, therefore, you will have to purchase it, and should first investigate how much it will cost.

Alternatively, if you already have an account with an ISP and do not wish to purchase a domain name, you can use the name provided by your ISP, if it provides you with your own virtual subdomain. This is typically the part after the @ - for example, mail@liamp.cix.co.uk. Beware that some ISPs do not provide virtual domains (look out for email addresses in the form id@ispname.com), or may charge extra for using a network rather than a single machine on certain account tariffs. However, using such a subdomain will cause extra work if you later wish to change ISPs. If you don't plan to attach your network to the Internet, to prevent future problems you should still purchase a domain from a name registrar, so that someone else cannot use it instead of you.

Setting Up Name Servers

There are currently two main standards for IP name resolution: DNS, which is cross-platform, and WINS, which is Windows-only. However, this changes with the advent of Windows 2000, which subsumes WINS into an enhanced dynamic DNS-compatible system - something which may prove to be a significant driver towards adoption of Windows 2000. On Windows NT Server versions 3 and 4 WINS integrates closely with DHCP, and DNS is peripheral. Configuring a WINS server is almost as simple as configuring a DHCP one: all that needs to be done is to tell the server the domain name, add entries for any fixed addresses, and the server does the rest, automatically building a database by "scavenging" traffic for machine node names and their associated addresses.

DNS servers are complex and difficult, and describing the setup and configuration of them needs an article - or possibly book - to itself. There is only space here for the bare essentials. On Unix systems, DNS is usually implemented using the open source BIND program, but others are available, including DNS servers for NetWare and NT. The most basic kind of DNS server is a DNS proxy. This simply takes DNS requests from the local network and forwards them onto the ISP's name server; it maintains no database of its own whatsoever, so repeated requests for the same address will generate repeated lookups, including bringing up the

""With DHCP, configuration is reduced to more or less the minimum level currently possible. The server needs only to be told the address range or ranges to put in the pool, and the client only that it should use DHCP to find its address.

connection if it is not already open. Proxy servers such as Wingate (**www.wingate.net**) and Mailgate (**www.mailgate.com**), often used to provide external Web access for a LAN, frequently include a simple DNS proxy.

Since even requests for local nodes will cause a DNS proxy to query the ISP's servers, it is often desirable to run a more capable DNS server as well, to handle internal requests. A basic but nonetheless useful DNS server for Windows is SimpleDNS by Jesper Hoy (**www.jhsoft.com**). This looks up client requests in the local hosts file, which reduces administration to maintaining a single version; clients need only be told the address of the machine running the DNS server, either via local configuration or DHCP. If the proxy server itself uses the ISP to resolve external addresses, this is all that's needed, and such a server can significantly reduce the number of connections to the ISP. Better performance can be achieved by running a DNS caching proxy. This has no local database, but when an address is resolved using the ISP's servers, the name and address are kept in memory. After a period, all commonly-used addresses can be supplied locally without recourse to the ISP, improving response time and reducing the number of calls. A low-specification machine running Linux and BIND is ideal for this.

After this, DNS configuration gets more complex, as servers maintain part of a database and also refer to higher-level servers - up to the top-level master servers maintained by Network Solutions in the USA which control the top-level domains (TLDs) such as .com.

Gateways

It is not always necessary to provide a routed connection between a LAN and the outside world. For single-site networks, a proxy and email server can provide Web access, ftp access and email forwarding without routing. Here, the proxy is the only machine connected to the Internet, and IP packets never travel between the LAN and the Internet. However, for multi-site WANs or direct Internet access, a gateway machine must be set up to route packets from the LAN to elsewhere. This may be a dedicated router, or a machine with a server OS (such as NT Server, NetWare or Linux) running a software router as a process. In the example shown in Figure 6 there are two separate sub-networks, 192.168.1.0 and 192.168.2.0. The gateway (**gateway.foo.org**) has two network connections, attaching it to both networks; in the 192.168.1.0 subnet it appears as 192.168.1.254, and in the 192.168.2.0 subnet it appears as 192.168.2.254. Some choices are arbitrary, or purely matters of convenience. For instance, in this example the subnets are not given separate names, although they could be; one might be london.foo.org and the other edinburgh.foo.org. The hostnames would then be as shown in Figure 7.

Gateways are often (but by no means always) given address 0.0.0.254 in each of their networks. They may be given hostnames, but as they are usually referred to by address, this is not necessary.

"For multi-site WANs or direct Internet access, a gateway machine must be set up to route packets from the LAN to elsewhere. This may be a dedicated router, or a machine with a server OS running a software router as a process."

Summary

1 Unlike most other network protocols, TCP/IP addresses are user-defined.
2 Addresses are assigned to each network interface rather than to host machines.
3 Network and host addresses must be unique and are allocated by a central authority, unless certain predefined private ranges are used.
4 Sub-networks are defined by bit patterns in the network address.
5 Addresses can be allocated automatically via the BOOTP or DHCP protocols.
6 The mapping between names and numbers is secondary, purely for user convenience, and is performed by different software. It has no effect on the underlying protocol, which always uses numeric addresses.
7 Name resolution generally uses DNS, but Windows systems may use the Microsoft proprietary WINS instead, or as well.

PCNA