TCP/IP FAQ; Frequently Asked Questions (1999-09) Part 1 of 2

Part1 - Part2

TCP/IP FAQ; Frequently Asked Questions (1999-09) Part 1 of 2

From: tcp-ip-faq@eng.sun.com (TCP/IP FAQ Maintainer)
Newsgroups: comp.protocols.tcp-ip
Subject: TCP/IP FAQ; Frequently Asked Questions (1999-09) Part 1 of 2
Date: 7 Sep 1999 03:36:53 GMT
Message-ID: <tcp-ip-faq-1.1999-09@eng.sun.com>
Summary: Part 1 of a 2-part informational posting that contains
responses to common questions on basic TCP/IP network
protocols and applications.
X-Disclaimer: Approval for postings in *.answers is based on form, not content.

Archive-name: internet/tcp-ip/tcp-ip-faq/part1
Version: 5.15
Last-modified: 1999-09-06 20:11:43
Posting-Frequency: monthly (first Friday)
Maintainer: tcp-ip-faq@eng.sun.com (Mike Oliver)
URL: http://www.itprc.com/tcpipfaq/default.htm

TCP/IP Frequently Asked Questions

Part 1: Introduction and Fundamental Protocols

This is Part 1 of the Frequently Asked Questions (FAQ) list for the comp.protocols.tcp-ip Usenet newsgroup. The FAQ provides answers to a selection of common questions on the various protocols (IP, TCP, UDP, ICMP and others) that make up the TCP/IP protocol suite. It is posted to the news.answers, comp.answers and comp.protocols.tcp-ip newsgroups on or about the first Friday of every month.

The FAQ is posted in two parts. Part 1 contains answers to general questions and questions that concern the fundamental components of the suite. Part 2 contains answers to questions concerning common applications that depend on the TCP/IP suite for their network

connectivity.

Comments on this document can be emailed to the FAQ maintainer at <tcp-ip-faq@eng.sun.com>.

- -

Table of Contents

Further Sources of Information

- -

Administrivia

1. Where can I find an up-to-date copy of this FAQ?

You can browse a hyperlinked version of this FAQ on the World Wide Web at <http://www.itprc.com/tcpipfaq/default.htm> in the US (thanks to Irwin Lazar) and at <http://t2.technion.ac.il/~s2845543/tcpip-faq/default.htm> in Israel (thanks to Uri Raz). Links to RFC's from Irwin's site refer to the ISI RFC repository in the US, while links to RFC's from Uri's site refer to the RFC repository at Imperial College in the UK. Use whichever gives you better response time.

The current version of this FAQ is posted on a monthly basis to the news.answers, comp.answers and comp.protocols.tcp-ip newsgroups.

A plaintext copy of the most recently posted version of the FAQ is available by anonymous FTP from <ftp://rtfm.mit.edu/pub/faqs/internet/tcp-ip/tcp-ip-faq/>.

2. Who wrote this FAQ?

This FAQ was compiled from Usenet postings and email contributions made by many people, including: Rui Duarte Tavares Bastos, Mark Bergman, Stephane Bortzmeyer, Rodney Brown, Dr. Charles E. Campbell Jr., James Carlson, Phill Conrad, Alan Cox, Michael Hunter, Jay Kreibrich, William Manning, Barry Margolin, Vic Metcalfe, Jim Muchow, George V. Neville-Neil, Dang Thanh Ngan, Subu Rama, Uri Raz, and W. Richard Stevens.

The FAQ is currently maintained by Mike Oliver. Comments, criticisms and contributions should be mailed to <tcp-ip-faq@eng.sun.com>. Please do not send TCP/IP questions to this address; it is intended only for FAQ issues. If you have a question that is not already answered by the material in this FAQ you will get a much faster (and probably more accurate) response by posting the question to the comp.protocols.tcp-ip newsgroup than you will by sending it to the FAQ maintainer.

- -

About TCP/IP

1. What is TCP/IP?

TCP/IP is a name given to the collection (or suite) of networking protocols that have been used to construct the global Internet. The protocols are also referred to as the DoD (dee-oh-dee) or Arpanet protocol suite because their early development was funded by the Advanced Research Projects Agency (ARPA) of the US Department of Defense (DoD).

The TCP/IP name is taken from two of the fundamental protocols in the collection, IP and TCP. Other core protocols in the suite are UDP and ICMP. These protocols work together to provide a basic networking framework that is used by many different application protocols, each tuned to achieving a particular goal.

TCP/IP protocols are not used only on the Internet. They are also widely used to build private networks, called internets (spelled with a small 'i'), that may or may not be connected to the global Internet (spelled with a capital 'I'). An internet that is used exclusively by one organization is sometimes called an intranet.

2. How is TCP/IP defined?

All of the protocols in the TCP/IP suite are defined by documents called Requests For Comments (RFC's). An important difference between TCP/IP RFC's and other (say, IEEE or ITU) networking standards is that RFC's are freely available online.

RFC's can be composed and submitted for approval by anyone. Standards RFC's are often the product of many weeks or months of discussion between interested parties designated as working groups, during which time drafts of the proposed RFC are continually updated and made available for comment. These discussions typically take place on open mailing lists which welcome input from all quarters. The RFC approval process is managed by the Internet Engineering Steering Group (IESG) based on recommendations from the Internet Engineering Task Force (IETF) which is a prime mover in the formation of working groups focused on strategic TCP/IP issues. You can find out more about IESG and IETF activities from the IETF home page at <http://www.ietf.org/>.

Not all RFC's specify TCP/IP standards. Some RFC's contain background information, some provide hints for managing an internet, some document protocol weaknesses in the hope that they might be addressed by future standards, and some are entirely humorous.

3. Where can I find RFC's?

The Definitive RFC Repository

The official and definitive RFC repository is the anonymous FTP archive maintained by the Information Sciences Institute of the University of Southern California at <ftp://ftp.isi.edu/in-notes>. It is reachable via the Web at <http://www.rfc-editor.org/>.

RFC Repository Mirror Sites

The RFC repository is mirrored at many sites on the Internet, and you may get a faster response from a local archive than you would from the often-overworked ISI site. Primary mirrors are updated at the same time as the ISI site. Secondary mirrors may lag by a few hours or days. The current primary mirror sites are:

In the USA ...

Missouri:
<ftp://wuarchive.wustl.edu/doc/rfc>
New Jersey:
<ftp://nisc.jvnc.net/>
North Carolina:
<ftp://ftp.ncren.net/rfc>
Texas:
<ftp://ftp.sesqui.net/pub/>

In Europe ...

France:
<ftp://ftp.imag.fr/pub/archive/IETF/rfc>
Italy:

<ftp://ftp.nic.it/rfc>

UK:

<ftp://src.doc.ic.ac.uk/rfc>

Secondary mirror sites are listed in a document named rfc-retrieval.txt which can be found alongside the RFC's themselves at any of the above sites.

RFC's by Email

If you don't have direct access to the Internet but are able to send and receive email then you can still get RFC's through various email-to-ftp gateways. For instructions on how to do this, send email containing the text:

help: ways_to_get_rfcs

to <rfc-info@isi.edu>.

4. How do I find the right RFC?

There are over 2500 RFC's. Each RFC is known by a number. For instance, RFC 1180 presents a tutorial on TCP/IP, RFC 1920 lists the current standards RFC's and explains the RFC standards process, and RFC 1941 is a FAQ list on the topic of Internet deployment in educational establishments. RFC numbers are assigned in ascending order as each RFC is approved.

The RFC files in the archive are named rfcNNNN.txt where NNNN is the number of the RFC. For instance, the text of RFC 822 is contained in the file named rfc822.txt. A small number of RFC's are also available in PostScript format, in which case a file named rfcNNNN.ps will exist in addition to the .txt file.

Basic information (number, title, author, publication date and so on) on all of the RFC's is contained in the RFC index document named rfc-index.txt which you can find alongside the RFC's at any of the RFC archive sites. If you don't know which RFC's you need, the index is a good place to start. The index also indicates the current status of each RFC. The content of an RFC does not change once the RFC has been published, but since TCP/IP is in a constant state of evolution the information in one RFC is often revised, extended, clarified and in some cases completely superseded by later RFC's. Annotations in the index indicate when this is the case.

If you find yourself using the index a lot then you might find it convenient to create your own HTML version of the index. Wayne Mesard has published a Perl script that takes the plaintext index file as input and produces an HTML version with hyperlinks to your chosen RFC FTP repository or to your own local RFC archive. The script is available at
<ftp://ftp.ibnets.com/pub/wmesard/>.

If you don't want to wade through the index, some sites provide the ability to search the RFC catalogue by keyword:

Keyword Searches on the Web
<http://www.faqs.org/rfcs/> lets you search on RFC content.
<http://web.nexor.co.uk/public/rfc/index/rfc.html> and
<http://www.csl.sony.co.jp/rfc/> let you search on words in the RFC title.
Keyword Searches via gopher
<gopher://r2d2.jvnc.net/11/Internet%20Resources/RFC> or
<gopher://muspin.gsfc.nasa.gov:4320/1g2go4%20ds.internic.net%2070%201%201/.ds/.internetdocs>
RFC Keyword Searches via WAIS

- -

About IP

1. What is IP?

Internet Protocol (IP) is the central, unifying protocol in the TCP/IP suite. It provides the basic delivery mechanism for packets of data sent between all systems on an internet, regardless of whether the systems are in the same room or on opposite sides of the world. All other protocols in the TCP/IP suite depend on IP to carry out the fundamental function of moving packets across the internet.

In terms of the OSI networking model, IP provides a Connectionless Unacknowledged Network Service, which means that its attitude to data packets can be characterised as "send and forget". IP does not guarantee to actually deliver the data to the destination, nor does it guarantee that the data will be delivered undamaged, nor does it guarantee that data packets will be delivered to the destination in the order in which they were sent by the source, nor does it guarantee that only one copy of the data will be delivered to the destination.

Because it makes so few guarantees, IP is a very simple protocol. This means that it can be implemented fairly easily and can run on systems that have modest processing power and small amounts of memory. It also means that IP demands only minimal functionality from the underlying medium (the physical network that carries packets on behalf of IP) and can be deployed on a wide variety of networking technologies.

The no-promises type of service offered by IP is not directly useful to many applications. Applications usually depend on TCP or UDP to provide assurances of of data integrity and (in TCP's case) ordered and complete data delivery.

The fundamentals of IP are defined in RFC 791. RFC 1122 summarises the requirements that must be met by an IP implementation in an Internet host, and RFC 1812 summarises the IP requirements for an Internet router.

2. How Is IP Carried On A Network?

IP really isn't very fussy about how its packets are transported. The details of how an IP packet is carried over a particular kind of network are usually chosen to be convenient for the network itself. As long as the transmitter and receiver observe some convention that allows IP packets to be differentiated from any other data that might be seen by the receiver, then IP can be used to carry data between those stations.

On a LAN, IP is carried in the data portion of the LAN frame and the frame header contains additional information that identifies the frame an an IP frame. Different LAN's have different conventions for carrying that additional information. On an Ethernet the Ethertype field is used; a value of 0x0800 identifies a frame that contains IP data. FDDI and Token Ring use frames that conform to IEEE 802 Logical Link Control, and on those LAN's IP is carried in Unnumbered Information frames with Source and Destination LSAP's of 0xAA and a SNAP header of 00-00-00-08-00.

The only thing that IP cares strongly about is the maximum size of a frame that can be carried on the medium. This controls whether,

and to what extent, IP must break down large data packets into a train of smaller packets before arranging for them to be transmitted on the medium. This activity is called "fragmentation" and the resulting smaller and incomplete packets are called "fragments". The final destination is responsible for rebuilding the original IP packet from its fragments, an activity called "fragment reassembly".

3. Does IP Protect Data On The Network?

IP itself does not guarantee to deliver data correctly. It leaves all issues of data protection to the transport protocol. Both TCP and UDP have mechanisms that guarantee that the data they deliver to an application is correct.

IP does try to protect the packet's IP header, the relatively small part of each packet that controls how the packet is moved through the network. It does this by calculating a checksum on the header fields and including that checksum in the transmitted packet. The receiver verifies the IP header checksum before processing the packet. Packets whose checksums no longer match have been damaged in some way and are simply discarded.

The IP checksum is discussed in detail in RFC 1071, which also includes sample code for calculating the checksum. RFC 1141 and RFC 1624 describe incremental modification of an existing checksum, which can be useful in machines such as routers which modify fields in the IP header while forwarding a packet and therefore need to compute a new header checksum.

The same checksum algorithm is used by TCP and UDP, although they include the data portion of the packet (not just the header) in their calculations.

4. What is ARP?

Address Resolution Protocol (ARP) is a mechanism that can be used by IP to find the link-layer station address that corresponds to a particular IP address. It defines a method that is used to ask, and answer, the question "what MAC address corresponds to a given IP address?". ARP sends broadcast frames to obtain this information dynamically, so it can only be used on media that support broadcast frames. Most LAN's (including Ethernet, FDDI, and Token Ring) have a broadcast capability and ARP is used when IP is running on those media. ARP is defined in RFC 826. That definition assumes an Ethernet LAN. Additional details for ARP on networks that use IEEE 802.2 frame formats (IEEE 802.3 CSMA/CD, IEEE 802.4, IEEE 802.5 Token Ring) are in RFC 1042. ARP on FDDI is described in RFC 1390.

When IP is runnning over non-broadcast media (say, X.25 or ATM) some other mechanism is used to match IP addresses to media addresses. IP really doesn't care how the media address is obtained.

RFC 903 defines Reverse ARP (RARP) which lets a station ask the question "which IP address corresponds to a given MAC address?". RARP is typically used to let a piece of diskless equipment discover its own IP address as part of its boot procedure. RARP is rarely used by modern equipment; it has been supplanted by the Boot Protocol (BOOTP) defined in RFC 1542. BOOTP in turn is being supplanted by the Dynamic Host Configuration Protocol (DHCP).

5. What is IPv6?

IP Version 6 (IPv6) is the newest version of IP, sometimes called

IPng for "IP, Next Generation". IPv6 is fairly well defined but is not yet widely deployed. The main differences between IPv6 and the current widely-deployed version of IP (which is IPv4) are:

o IPv6 uses larger addresses (128 bits instead of 32 bits in IPv4) and so can support many more devices on the network, and

o IPv6 includes features like authentication and multicasting that had been bolted on to IPv4 in a piecemeal fashion over the years.

Information on IPv6 can be found on the IPv6 home page at <http://playground.sun.com/pub/ipng/html/ipng-main.html>

6. What happened to IPv5?

Or, ""Why are we skipping from IPv4 to IPv6?"

IPv5 never existed. The version number "5" in the IP header was assigned to identify packets carrying an experimental non-IP real-time stream protocol called ST. ST was never widely used, but since the version number 5 had already been allocated the new version of IP was given its own unique identifying number, 6. ST is described in RFC 1819.

7. What is the 6bone?

The 6bone is the experimental IPv6 backbone being developed using IPv6-in-IPv4 tunnels. This is intended for early experimentation with IPv6 and is not a production service.

8. What is the MBONE?

The Multicast backBONE (MBONE) is a multicast-capable portion of the Internet backbone. Multicast support over IP is provided by a protocol called IGMP (Internet Group Management Protocol) which is defined in RFC 1112. The MBONE is still a research prototype, but it extends through most of the core of the Internet (including North America, Europe, and Australia). It is typically used to relay multimedia (audio and low bandwidth video) presentations from a single source to multiple receiving sites dispersed over the Internet.

A slightly dated MBONE FAQ is available by anonymous FTP from <ftp://ftp.isi.edu/mbone/faq.txt>.

9. What is IPsec?

IPsec stands for "IP Security". The IPsec working group of the IETF is developing standards for cryptographic authentication and for encryption within IP. The base specifications are defined in RFC's 1825, 1826 and 1827. Products that implement these are beginning to appear.

A freely distributable implementation of IPsec for IPv4 and IPsec for IPv6 is included in the NRL IPv6/IPsec distribution for 4.4-Lite BSD. The NRL software is available from <http://web.mit.edu/network/isakmp/> (for distribution within the US only), from <http://www.cisco.com/public/library/isakmp/ipsec.html> (for distribution within the US and Canada), and from <ftp://ftp.ripe.net/ipv6/nrl/> (for unrestricted distribution).

(Some countries consider encryption software to have military significance and so restrict the export and import of such

software, which is why there are geographical restrictions on the areas served by the above sites.)

- -

About TCP

1. What is TCP?

Transmission Control Protocol (TCP) provides a reliable byte-stream transfer service between two endpoints on an internet. TCP depends on IP to move packets around the network on its behalf. IP is inherently unreliable, so TCP protects against data loss, data corruption, packet reordering and data duplication by adding checksums and sequence numbers to transmitted data and, on the receiving side, sending back packets that acknowledge the receipt of data.

Before sending data across the network, TCP establishes a connection with the destination via an exchange of management packets. The connection is destroyed, again via an exchange of management packets, when the application that was using TCP indicates that no more data will be transferred. In OSI terms, TCP is a Connection-Oriented Acknowledged Transport protocol.

TCP has a multi-stage flow-control mechanism which continuously adjusts the sender's data rate in an attempt to achieve maximum data throughput while avoiding congestion and subsequent packet losses in the network. It also attempts to make the best use of network resources by packing as much data as possible into a single IP packet, although this behaviour can be overridden by applications that demand immediate data transfer and don't care about the inefficiencies of small network packets.

The fundamentals of TCP are defined in RFC 793, and later RFC's refine the protocol. RFC 1122 catalogues these refinements as of October 1989 and summarises the requirements that a TCP implementation must meet.

TCP is still being developed. For instance, RFC 1323 introduces a TCP option that can be useful when traffic is being carried over high-capacity links. It is important that such developments are backwards-compatible. That is, a TCP implementation that supports a new feature must continue to work with older TCP implementations that do not support that feature.

2. How does TCP try to avoid network meltdown?

TCP includes several mechanisms that attempt to sustain good data transfer rates while avoiding placing excessive load on the network. TCP's "Slow Start", "Congestion Avoidance", "Fast Retransmit" and "Fast Recovery" algorithms are summarised in RFC 2001. TCP also mandates an algorithm that avoids "Silly Window Syndrome" (SWS), an undesirable condition that results in very small chunks of data being transferred between sender and receiver. SWS Avoidance is discussed in RFC 813. The "Nagle Algorithm", which prevents the sending side of TCP from flooding the network with a train of small frames, is described in RFC 896.

Van Jacobson has done significant work on this aspect of TCP's behaviour. The FAQ used to contain a couple of pieces of historically interesting pieces of Van's email concerning an early implementation of congestion avoidance, but in the interests of saving space they've been removed and can instead be obtained by anonymous FTP from the end-to-end mailing list archive at

<ftp://ftp.isi.edu/end2end/end2end-1990.mail>. PostScript slides of a presentation on this implementation of congestion avoidance can be obtained by anonymous FTP from <ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z>.

That directory contains several other interesting TCP-related papers, including one (<ftp://ftp.ee.lbl.gov/papers/fastretrans.ps>) by Sally Floyd that discusses a algorithm that attempts to give TCP the ability to recover quickly from packet loss in a network.

3. How do applications coexist over TCP and UDP?

Each application running over TCP or UDP distinguishes itself from other applications using the service by reserving and using a 16-bit port number. Destination and source port numbers are placed in the UDP and TCP headers by the originator of the packet before it is given to IP, and the destination port number allows the packet to be delivered to the intended recipient at the destination system.

So, a system may have a Telnet server listening for packets on TCP port 23 while an FTP server listens for packets on TCP port 21 and a DNS server listens for packets on port 53. TCP examines the port number in each received frame and uses it to figure out which server gets the data. UDP has its own similar set of port numbers.

Many servers, like the ones in this example, always listen on the same well-known port number. The actual port number is arbitrary, but is fixed by tradition and by an official allocation or "assignment" of the number by the Internet Assigned Numbers Authority (IANA).

4. Where do I find assigned port numbers?

The IANA allocates and keeps track of all kinds of arbitrary numbers used by TCP/IP, including well-known port numbers. The entire collection is published periodically in an RFC called the Assigned Numbers RFC, each of which supersedes the previous one in the series. The current Assigned Numbers RFC is RFC 1700.

The Assigned Numbers document can also be obtained directly by FTP from the IANA at <ftp://ftp.isi.edu/in-notes/iana/assignments>.

- -

About UDP

1. What is UDP?

User Datagram Protocol (UDP) provides an unreliable packetized data transfer service between endpoints on an internet. UDP depends on IP to move packets around the network on its behalf.

UDP does not guarantee to actually deliver the data to the destination, nor does it guarantee that data packets will be delivered to the destination in the order in which they were sent by the source, nor does it guarantee that only one copy of the data will be delivered to the destination. UDP does guarantee data integrity, and it does this by adding a checksum to the data before transmission. (Some machines run with UDP checksum generation disabled, in which case data corruption or truncation can go undetected. Very few people think this is a good idea.)

The fundamentals of UDP are defined in RFC 768. RFC 1122

summarises the requirements that a UDP implementation must meet.

- -

About ICMP

1. What is ICMP?

Internet Control Message Protocol (ICMP) defines a small number of messages used for diagnostic and management purposes. ICMP depends on IP to move packets around the network on its behalf.

The fundamentals of ICMP are defined in RFC 792. RFC 1122 summarises the requirements that must be met by an ICMP implementation in an Internet host, and RFC 1812 summarises the ICMP requirements for an Internet router.

ICMP is basically IP's internal network management protocol and is not intended for use by applications. Two well known exceptions are the ping and traceroute diagnostic utilities:

o ping sends and receives ICMP "ECHO" packets, where the response packet can be taken as evidence that the target host is at least minimally active on the network, and

o traceroute sends UDP packets and infers the route taken to the target from ICMP "TIME-TO-LIVE EXCEEDED" or "PORT UNREACHABLE" packets returned by the network. (Microsoft's TRACERT sends ICMP "ECHO" packets rather than UDP packets, and so receives ICMP "TIME-TO-LIVE EXCEEDED" or "ECHO RESPONSE" packets in return.)

- -

TCP/IP Network Operations

1. How can I measure the performance of an IP link?

You can get a quick approximation by timing how long it takes to FTP or RCP a large file over the link, but bear in mind that that measurement will be skewed by the time spent in dealing with the local and remote filesystems, not simply with the network itself. And remember to measure the time it takes to receive a file, not the time it takes to send it; the sender can report completion even though large amounts of data are still buffered locally by TCP and have not yet been delivered to the destination.

Two well-known open-source programs that measure and report throughput over an IP link without involving the filesystem are:

o TTCP, available for anonymous ftp from the Silicon Graphics FTP archive at < ftp://ftp.sgi.com/sgi/src/ttcp/>.

o Rick Jones' NETPERF, available on the Web at <http://www.cup.hp.com/netperf/NetperfPage.html>.

If neither of those tools does what you want then you might find something that meets your needs in CAIDA's measurement tools list at <http://www.caida.org/Tools/meastools.html>.

2. What IP addresses should I assign to machines on a private internet?

You shouldn't use IP addresses that have been assigned to some other organisation, because if knowledge of your network ever gets leaked onto the Internet they may disrupt that innocent

organisation's activity. RFC 1918 provides a solution for this problem by allocating several IP address ranges specifically for use on private networks. These addresses will never be assigned to any organisation and are never supposed to appear on the Internet. The ranges are:

Class A: 10.0.0.0 through 10.255.255.255
Class B: 172.16.0.0 through 172.31.255.255
Class C: 192.168.0.0 through 192.168.255.255

3. Can I set up a gateway to the Internet that translates IP addresses, so that I don't have to change all our internal addresses to an official network?

This is called Network Address Translation, or NAT. In general it is a difficult thing to do properly because many applications embed IP addresses in the application-level data (FTP's "PORT" command is a notable example) so NAT isn't simply a matter of translating addresses in the IP header and recalculating header checksums. Also, if the network number(s) you're using match those assigned to another organisation, your gateway may not be able to communicate with that organisation. As noted above, RFC 1918 proposes network numbers that are reserved for private use, to avoid such conflicts, but if you're already using a different network number this won't help you.

However, there are several products that do attempt to provide this kind of on-the-fly NAT. Linux provides NAT through its "IP Masquerading" feature, and many firewall and router vendors offer NAT capabilities in their products -- look for "Network Address Translation" in your favourite Web search engine to get a list of vendors. Proxy servers developed for firewalls can also sometimes be used as a substitute for an address-translating gateway for particular application protocols. This is discussed in more detail in the FAQ for the comp.security.firewalls newsgroup. That FAQ can be viewed on the Web at <http://www.clark.net/pub/mjr/pubs/fwfaq/>.

4. Can I use a single bit subnet?

The answer used to be a straightforward "no", because a 1-bit subnet can only have a subnet part of all-ones or all-zeroes, both of which were assigned special meanings when the subnetting concept was originally defined. (All-ones meant "broadcast, all subnets of this net" and all-zeroes meant "this subnet, regardless of its actual subnet number".)

However, the old definition of subnetting has been superseded by the concept of Classless Inter-Domain Routing (CIDR, pronounced 'cider'). Under CIDR the subnet doesn't really have an existence of its own and the subnet mask simply provides the mechanism for isolating an arbitrarily-sized network portion of an IP address from the remaining host part. As CIDR-aware equipment is deployed it becomes increasingly like that you will be able to use a 1-bit subnet with at least some particular combinations of networking equipment. However, it's still not safe to assume that a 1-bit subnet will work properly with all kinds of equipment.

As Steinar Haug explains:

From RFC 1122:

> 3.3.6 Broadcasts
>
> Section 3.2.1.3 defined the four standard IP broadcast address
> forms:

```
>   Limited Broadcast:                {-1, -1}
>   Directed Broadcast:               {<Network-number>, -1}
>   Subnet Directed Broadcast:        {<Network-number>, <Subnet-number>, -1}
>   All-Subnets Directed Broadcast:   {<Network-number>, -1, -1}
```

All-Subnets Directed broadcasts are being deprecated in favor of IP multicast, but were very much defined at the time RFC1122 was written. Thus a Subnet Directed Broadcast to a subnet of all ones is not distinguishable from an All-Subnets Directed Broadcast.

For those old systems that used all zeros for broadcast in IP addresses, a similar argument can be made against the subnet of all zeros.

Also, for old routing protocols like RIP, a route to subnet zero is not distinguishable from the route to the entire network number (except possibly by context).

Most of today's systems don't support variable length subnet masks (VLSM), and for such systems the above is true. However, all the major router vendors and *some* Unix systems (BSD 4.4 based ones) support VLSMs, and in that case the situation is more complicated :-)

With VLSMs (necessary to support CIDR, see RFC 1519), you can utilize the address space more efficiently. Routing lookups are based on *longest* match, and this means that you can for instance subnet the class C net with a mask of 255.255.255.224 (27 bits) in addition to the subnet mask of 255.255.255.192 (26 bits) given above. You will then be able to use the addresses x.x.x.33 through x.x.x.62 (first three bits 001) and the addresses x.x.x.193 through x.x.x.222 (first three bits 110) with this new subnet mask. And you can continue with a subnet mask of 28 bits, etc. (Note also, by the way, that non-contiguous subnet masks are deprecated.)

This is all very nicely covered in the paper by Havard Eidnes:

Practical Considerations for Network Address using a
CIDR Block Allocation
Proceedings of INET '93

This paper is available with anonymous FTP from
aun.uninett.no:pub/misc/eidnes-cidr.ps.

The same paper, with minor revisions, is one of the articles in the special Internetworking issue of Communications of the ACM (last month, I believe).

Steinar Haug, SINTEF RUNIT, University of Trondheim, NORWAY
Email: Steinar.Haug@runit.sintef.no

- -

TCP/IP Protocol Implementations

1. Where can I find TCP/IP source code?

Code used in the venerable Net-2 version of Berkeley Unix is available by anonymous FTP from
<ftp://ftp.uu.net/systems/unix/bsd-sources/sys/netinet> (at UUNet in Virginia, US) and
<ftp://gatekeeper.dec.com/pub/BSD/net2/sys/> (at Compaq in California, US).

Source code for the TCP/IP implementations in the current dialects of BSD Unix is available. Instructions on how to access the sources through FTP and other methods is detailed on their

respective websites: FreeBSD at <http://www.freebsd.org/>; NetBSD at <http://www.netbsd.org/>; and OpenBSD at <http://www.openbsd.org/>.

Source for the Unix-like Linux operating system is at <http://www.kernel.org/pub/linux/>.

Source for the TCP/IP implementation of the Xinu operating system discussed in Comer & D. L. Stevens' "Internetworking with TCP/IP Volume II" is at <ftp://ftp.cs.purdue.edu/pub/Xinu/>.

WATTCP is a DOS TCP/IP stack derived from the NCSA Telnet program and much enhanced. It is available from many DOS software archive sites as WATTCP.ZIP. This file includes some example programs and complete source code. The interface isn't BSD sockets but is well suited to PC type work.

KA9Q is Phil Karn's network operating system for PC's. It includes a TCP/IP implementation originally developed for use over packet radio. Source is available from Phil's website at <http://people.qualcomm.com/karn/code/ka9qos/>.

2. Where can I find TCP/IP application source code?

Source code for the various TCP/IP applications delivered with the current BSD Unix flavours is available through the FreeBSD, NetBSD and OpenBSD websites noted in the previous section.

Linux application source is at <http://www.kernel.org/pub/linux/>. Much of the application source used by Linux was originally developed by the GNU Project whose website is at <http://www.gnu.org/>.

Code from Comer & D. L. Stevens' "Internetworking with TCP/IP Volume III" is available by anonymous FTP from <ftp://ftp.cs.purdue.edu/pub/dls/>.

Code from W. R. Stevens' "TCP/IP Illustrated, Volume 1" is available from <ftp://ftp.uu.net/published/books/stevens.tcpipiv1.tar.Z>.

Source code for some well-known cross-system TCP/IP applications (BIND, sendmail, Apache and so on) is available from the various organisations that sponsor the applications. See Part 2 of the FAQ for details.

3. Where can I find IPv6 source code?

There are several freely distributable implementations of IPv6, particularly for BSD and Linux. You can find detailed information at <http://playground.sun.com/pub/ipng/html/ipng-implementations.html>, part of the IPv6 home site mentioned above.

- -

Further Sources of Information

1. TCP/IP-related newsgroups and FAQ lists

Collections of newsgroup FAQ documents are archived at many locations including <http://www.faqs.org/> and <ftp://rtfm.mit.edu/pub/faqs/>. The following newsgroups are particularly relevant to TCP/IP:

comp.protocols.tcp-ip covers all of the TCP/IP suite.

comp.protocols.dns.bind covers the BIND suite, which contains server and client implementations of DNS.

comp.protocols.tcp-ip.domains discusses DNS global administration and politics.

comp.protocols.nfs covers NFS protocol, implementation, and administration.

comp.protocols.snmp covers SNMP definition, implementation, and administration.

comp.protocols.time.ntp covers NTP definition, implementation, and administration.

comp.protocols.tcp-ip.ibmpc discusses TCP/IP for IBM(-like) personal computers. The group's FAQ is available at <ftp://ftp.netcom.com/pub/ma/mailcom/IBMTCP/>.

comp.os.ms-windows.networking.tcp-ip discusses TCP/IP on Microsoft Windows machines.

comp.os.ms-windows.programmer.tools.winsock covers the Winsock sockets API on Microsoft Windows machines. The group's FAQ is available at < http://www.cyberport.com/~tangent/programming/winsock/>.

comp.os.os2.networking.tcp-ip discusses TCP/IP on OS/2.

comp.dcom.lans.ethernet covers Ethernet and IEEE 802.3 LAN's. The group's FAQ is available at < ftp://dorm.rutgers.edu/pub/novell/info_and_docs/Ethernet.FAQ>.

comp.dcom.lans.fddi covers FDDI LAN's.

comp.dcom.lans.token-ring covers IBM Token Ring and IEEE 802.5 LAN's.

comp.dcom.lans.misc covers all other types of LAN.

comp.protocols.ppp covers PPP and SLIP. The group's FAQ is available at < http://cs.uni-bonn.de/ppp/part1.html>.

comp.dcom.sys.cisco discusses cisco products.

comp.dcom.sys.wellfleet discusses Wellfleet (now Bay Networks) products.

2. Are there any good books on TCP/IP?

Yes, lots of them, far too many to list here. Uri Raz maintains a TCP/IP bibliography (the "TCP/IP Resources List") that is posted to the comp.protocols.tcp-ip newsgroup on a monthly basis. It is available on the Web at <http://www.qnx.com/%7Emphunter/tcpip_resources.html> and <http://www.faqs.org/faqs/internet/tcp-ip/resource-list/index.html> or can be retrieved by anonymous FTP from <ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/comp/protocols/tcp-ip/>.

However, a couple of books that always head the list of recommended reading are:

"Internetworking with TCP/IP Volume I (Principles, Protocols, and Architecture)" by Douglas E. Comer, published by Prentice Hall, 1991 (ISBN 0-13-468505-9). This is an introductory book which covers all of the fundamental protocols, including IP,

UDP, TCP, and the gateway protocols. It also discusses some higher level protocols such as FTP, Telnet, and NFS.

"TCP/IP Illustrated, Volume 1: The Protocols" by W. Richard Stevens, published by Addison-Wesley, 1994 (ISBN 0-201-63346-9). This book explains the TCP/IP protocols and several application protocols in exquisite detail. It contains many real-life traces of the protocols in action, which is especially valuable for people who need to understand the protocols in depth.

If you're writing programs that use TCP/IP then the classic text is "Unix Network Programming" by W. Richard Stevens, published by Prentice Hall, 1990 (ISBN 0-13-949876-1). It's now being rewritten as a three volume set. The first volume "Unix Network Programming: Networking APIs: Sockets and Xti" published by Prentice Hall, 1997 (ISBN 013490012X), contains just about everything you need to know about using TCP/IP and includes material on topics (eg IPv6, multicasting, threads) that are not covered in the original UNP.

- -

Part1 - Part2

[Usenet FAQs | Web FAQs | Documents | RFC Index]

Send corrections/additions to the FAQ Maintainer:
tcp-ip-faq@eng.sun.com (TCP/IP FAQ Maintainer)

Last Update June 29 2010 @ 07:58 AM