UTnet Guide to Internet Protocol (IP) Addresses and Routing

The Network Information Center
of the University of Texas at Austin (UT NIC)

Document version 1.0

Introduction

This guide serves as a brief primer on Internet Protocol (IP) addressing concepts and IP routing mechanisms. It is intended to be a short introduction to a few of the important concepts in IP addressing and routing so that the reader will have some idea of what the concepts mean and of how the routing mechanisms function.

There is a large amount of information presented here in a short space. The primer begins with the origins of the ARPANET and rapidly covers some highlights of IP addressing and routing. Unfamiliar terms may be found in the UTnet Network Glossary. Much more information about TCP/IP may be found in Douglas Comer's excellent text, **Internetworking With TCP/IP, Principles, Protocols and Architecture,** Prentice-Hall, 1988, ISBN 0-13-470154-2.

1. Development of TCP/IP

The development of the TCP/IP protocol suite was funded by the Defense Advanced Research Projects Agency (DARPA, commonly referred to by its old acronym, ARPA) which is a part of the U.S. Department of Defense. For many years ARPA has been involved in the development of computer network technology, and has sponsored work in this field at research centers and universities across the country.

1.1. The ARPANET

As a result of this work, the first major high speed computer network spanning the entire country, the ARPA Internet, was developed. A series of network protocols were created for what was generally called the ARPANET, culminating in the TCP/IP protocol suite. The use of public funds in the development of this protocol suite caused it to be placed in the public domain. The inclusion of the TCP/IP protocols in Berkeley UNIX has since caused this set of protocols to become widely adopted as the de facto protocol suite for all manner of computers attached to high speed networks.

The original ARPANET has since been largely replaced by the nationwide National Science Foundation network (NSFnet), whose operation is also based on the TCP/IP protocols.

IP stands for Internet Protocol, and refers to the fact that the Internet is made up of many individual networks. The IP portion of the protocol suite manages the task of making all of the individual networks look like one big network to every attached computer.

TCP stands for Transmission Control Protocol. This portion of the protocol suite provides a set of mechanisms for establishing reliable communications between software applications that run on computers linked by the network.

1.2. IP Datagrams

Information is sent over the Internet in IP datagrams. A datagram is a set of bits transmitted over the Internet that are formed up in a specified manner, according to the protocol in use. IP datagrams consist of an IP header, which carries the IP addresses, followed by the data being sent between applications running on the computer systems engaged in sending and receiving the datagram.

Another term in common use is ‘‘packet.’’ Sometimes the words packet and datagram are used interchangeably. We prefer to use ‘‘packet’’ to refer the unit of data sent across a packet switching network. In the latter case, a given IP datagram might end up occupying several packets. The IP protocol contains mechanisms for the fragmentation and reassembly of IP datagrams as they wend their way across the Internet system.

In this primer, a packet is the unit of data used on the physical network system, while a datagram is the basic unit of information passed across the Internet. As noted above, the Internet is composed of a number of different physical network transmission systems. The exact usage isn't that critical, except that it helps keep the boundary between a given physical network and the logical Internet (made up of many physical networks) a little clearer.

1.3. IP Addresses

The Internet Protocol has the task of finding a route between any two computers on a given internetwork. Each TCP/IP datagram has an IP portion that contains the unique IP address of both the sending computer and the destination system.

These addresses make it possible for the computer data from a given system to travel a network of unknown complexity and constantly changing topology and still arrive at the correct destination.

IP addresses must be unique

Each system attached to the network **MUST** have a unique IP address for this mechanism to work properly. This is why it is essential to request an IP address from the UT Network Information Center (UT NIC) before attaching a computer to the campus network. This allows the UT NIC to ensure that there are no duplicate IP addresses assigned on campus.

1.4. Structure of an IP Address

Each IP datagram contains a pair of IP addresses. The IP addresses are 32 bits long, and there is one for the destination address of the datagram and one for the source address of the host sending the datagram. The IP addresses are placed at the beginning of the datagram, in a section of the datagram called the ‘‘IP header.’’

Division into network and local

The IP address is divided internally into a network part and a local part. As such, each IP address is really a pair of identifiers. The first identifier refers to a given network and the second refers to the local addressing conventions used on that network to identify specific host computers. This portion is also known as the host address or host-id.

Address Classes.

The next major division of IP addresses results in a set of address classes. The structure of the address classes is set so that the first several (high-order) bits of the address determine which class the address falls into. There are three primary forms of IP addresses:

Class A.

The first type of address, Class A, is divided up into 7 bits of network number and 24 bits of local address. The highest-order bit is set to 0. This allows only 128 class A networks, but each Class A network can accommodate almost 17 million hosts.

1	7	24
0	Network	Host

Figure A-1. The Class A Internet Address Format

Class B.

The second type of address, Class B, is divided up into 14 bits of network number and 16 bits of local address. The two highest-order bits are set to 1,0. This allows over 16,000 class B networks to be assigned with approximately 65,000 hosts per network.

1	1	14	16
1	0	Network	Host

Figure A-2. The Class B Internet Address Format

Class C.

The third type of address, Class C, is divided up into 21 bits of network number and 8 bits of local address. The three highest-order bits are set to 1,1,0. This allows over 2 million class C networks, with up to 254 hosts per network. (Addresses 0 and 255 are reserved.)

1	1	1	21	8
1	1	0	Network	Host

Figure A-3. The Class C Internet Address Format

There are two further classes of addresses. Class D addresses are reserved for multi-cast experiments. The four highest-order bits of the Class D address are set to 1,1,1,0. Class E is reserved for future use.

1.5. Dotted Decimal Notation

The commonly used notation system for IP addresses divides the address up into four 8 bit fields separated by periods. Each field value is expressed as a decimal number. For example, the address of the computer called sri-nic.arpa is 026.000.000.073 which is shortened to 26.0.0.73 for clarity. This represents a class A address. The class A addresses include the range of addresses from 1.0.0.0 through 126.0.0.0.

The address for emx.utexas.edu, a computer located on UTnet, is 128.83.1.33. This is a class B address. The class B addresses include the range of addresses from 128.1.0.0 through 191.254.0.0. Class C addresses begin with 192.0.1.0 and continue to 223.255.254.0.

The largest possible value for a field is an octet of all ones expressed as 255, so that an IP address of all ones would be 255.255.255.255.

1.6. Loopback Address

Address 127.nnn.nnn.nnn (where ‘‘nnn’’ stands for any local address) is a special address dedicated to the software loopback function. It is there for the use of higher level protocol software as a means for sending data between applications co-resident on a single host. This address should **never** be assigned to a host system. Data sent to a 127 address should never be seen on the network.

The default loopback address in UNIX is 127.0.0.1.

1.7. Address Conventions

There are several IP address conventions currently in use.

Referring to a network:

If the bits of the **host** portion of an IP address are all zero, then the address refers to the network specified in the network portion of that address. For example, the Class B address **128.83.0.0** refers to the particular network known as UTnet.

Note that there is a *coincidental* similarity between the old form of IP broadcast used on UTnet **(128.83.0.0),** and the convention used to refer to the UTnet network as a whole.

It’s unfortunate that the old form of IP broadcast address and the convention for referring to an IP network look identical in this instance, but it’s fairly easy to tell them apart by their usage. If **128.83.0.0** is used as a *destination address* in a datagram, it’s the UTnet IP broadcast address. If it’s used as an entry in a routing table or in writing about a network, then it’s simply the conventional way of identifying UTnet by its Class B IP network number.

Referring to a host:

If the bits of the **network** portion of an IP address are all zero, then that address refers to the host specified in the host portion of the address. For example, **0.0.1.33** would refer to a particular host.

Addresses of this form are sometimes seen when diskless computers are first booting and do not yet know their own IP address. One approach is for the host to make up an address, perhaps based on its own serial number, and then use this form of IP address briefly while finding out its correct IP address. Since the host does not yet know the number of the network it is attached to, it uses zeros for the network part of the address.

Referring to all hosts:

If the bits in the host portion of an address are all ones, then the address refers to all hosts attached to the network specified in the network portion of the address. For example, the Class B address **128.83.255.255** refers to all hosts on network **128.83.0.0. NOTE!** UTnet is currently using the older, non-standard, form for broadcasting to all hosts on a network. Hosts attached to UTnet must be set to use this form and not the convention just described. See the section on **IP Broadcast Addresses** below.

1.8. Address Resolution Protocol (ARP)

ARP is a protocol separate from the TCP/IP suite that is used to discover a host’s physical network address. It’s not enough to have an IP address for a host, since each host actually communicates using low level network hardware that has no knowledge of IP addresses. The network hardware system has its own set of unique addresses. In the case of Ethernet* hardware, the most common Local Area Network (LAN) on campus, the network address is a 48 bit number assigned

---

*Ethernet is a registered trademark of Xerox, Inc.

by the manufacturer to the Ethernet interface hardware installed in the host. ARP provides a mechanism for finding out what the host's Ethernet hardware address is, given a host's IP address.

ARP Example

For example, if host A has the IP address for host B and wishes to send a datagram to B, how does host A discover host B's Ethernet address? Host A accomplishes this task by sending out an ARP packet encapsulated in an Ethernet packet sent to the Ethernet broadcast address, so that every host attached to the Ethernet system will be forced to accept the packet. The ARP packet essentially says ''Who out there owns the following target IP address?'' Because of the use of an Ethernet broadcast packet, every single host on the Ethernet reads in the packet and looks at the target IP address it contains.

The host whose own IP address matches the target IP address in the ARP packet sends a reply to the originator of the ARP (host A), containing its Ethernet hardware address. The ARP originator stores the IP and Ethernet address pair in a cache of ARP replies, and proceeds to send the TCP/IP datagram in question out over the Ethernet to the target host.

ARP Broadcasts

ARP relies on the broadcast ability of the Ethernet hardware system to ensure that every host on the Ethernet hears the ARP request. By caching the ARP replies in a table, the host need not send out a second request for the same address since it can consult its table for the information it needs. The Ethernet broadcast address is fixed as part of the Ethernet LAN system, so that every host equipped with Ethernet hardware always knows the correct Ethernet broadcast address to use.

1.9. Ethernet and TCP/IP

Note that there are two independent but co-operating systems being described here. The Ethernet LAN functions as the delivery service for data being sent between computers. Ethernet packets can carry data from any protocol suite in the Ethernet packet's data field and are not limited to TCP/IP. The TCP/IP protocol suite, in turn, uses Ethernet packets to send data between cooperating hosts. Other LAN systems can be used as well. TCP/IP network software implementations can cooperate with all manner of LANs as well as wide area network technologies.

The following figure illustrates how a TCP/IP datagram is carried within the data field of an Ethernet frame.

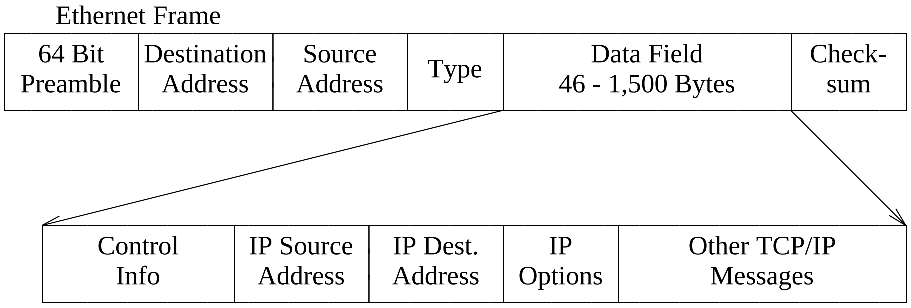

Figure A-4. IP Header And Other Messages In Data Field Of Ethernet Frame

The Ethernet frame begins with a preamble and set of addresses. These are followed by an Ethernet type field which is used to identify what type of protocols the Ethernet frame is carrying. For instance, Ethernet type 0800 (hex) indicates that the Ethernet frame is carrying TCP/IP protocols. Following the type field is the data field which ranges from a minimum size of 46 bytes to a maximum of 1,500 bytes. In this example, the Ethernet data field is filled with the TCP/IP information beginning with the IP header.

The IP header is composed of various pieces of control information, the IP source and destination addresses, and the IP options. The IP header is followed by data that contains other elements of the TCP/IP protocol suite, such as TCP messages, ICMP messages, and so on.

The data field of an Ethernet frame can carry many different network protocols since the Ethernet system itself is unconcerned with the format of the bits carried in its data field. Ethernet frames also carry DECnet protocols and XNS protocols for example. The same Ethernet system can deliver data from all of these protocols with impunity. This way DECnet hosts and TCP/IP hosts, among others, can communicate between themselves using the same Ethernet network.

Broadcast in Ethernet and TCP/IP

The Ethernet system is low cost, high speed and easy to use and has become the default campus LAN. As noted above, the broadcast address is fixed in Ethernet systems and is never in question. When a TCP/IP datagram is addressed to the TCP/IP broadcast address, it will be sent over the Ethernet in an Ethernet packet addressed to the Ethernet broadcast address, so that every host attached to the Ethernet system will be forced to read it.

The IP-level broadcast address is not fixed, however, and must be set correctly or severe problems can result. The correct IP broadcast address settings are described below.

1.10. IP Broadcast Addresses.

The standard form of IP broadcast address has changed over time. As a result there are two IP broadcast addresses currently in use. It is essential that all host systems on a given network such as UTnet use the same IP broadcast address. Otherwise, packet avalanches known as ‘‘broadcast storms’’ can cause the network to fail due to transient overloads in the packet traffic rate.

Berkeley Broadcasts

Early versions of Berkeley UNIX used zeros to indicate the IP broadcast address. Since then the standard practice for IP has converged on the use of all ones for broadcast. Because the early Berkeley UNIX distributions did not include a way to change the broadcast address as a configuration parameter, the older hosts are forced to continue using zeros for the broadcast address, despite the new standard.

Problems with multiple broadcast addresses

This means that when sharing a network with computers using zeros for IP broadcast (sometimes called ‘‘Old Broadcastians’’), all hosts must set their broadcast address to conform to the old system of broadcast. If a host sharing a network with Old Broadcastians is incorrectly set to use the new broadcast address of all ones, then this can cause packet avalanches.

The avalanche of packets will come from all the other hosts on the network that are correctly set for the old style broadcast address. These hosts will all hear the Ethernet broadcast frame containing an IP address using all ones. The higher level TCP/IP software will interpret the all ones address as an unknown IP address. If the host’s network software attempts to forward the packet, the simultaneous output of ARP requests for the unknown IP address by hundreds of hosts generates a packet avalanche known as a ‘‘broadcast storm.’’ The effect of broadcast storms can range from periodic short term network outages to complete network ‘‘meltdowns’’ in which the network ceases to be useful for relatively long periods of time.

UTnet broadcast usage

Until all hosts sharing the network system have been upgraded and are capable of using the standard of all ones, it will be necessary for all hosts to conform with the Old Broadcastians in our midst. This means that all older hosts attached to UTnet must use zeros for the broadcast address, as shown in the configuration settings for UTnet hosts.

The most recent network software implementations based on BSD 4.3 have been improved so that packet forwarding does not occur, and so that the host makes no other response to packets sent tot he broadcast address. This is fortunate since some vendors of recent network software have also removed the ability to change the IP broadcast address and have set their hosts to always use ones. If your host cannot be changed from the all ones IP address, then it is still possible to attach the host to UTnet as long as all other configuration settings are strictly followed. In this case you should contact the Network Information Center for the latest instructions.

1.11. Internet Routing

The structure of an Internet address is designed to facilitate routing datagrams in the Internet. The official Internet (with a capital ‘‘I’’) is the set of interconnected networks using TCP/IP protocols that make up the worldwide TCP/IP network system. The Internet depends on gateways to route datagrams between the various networks that make up the Internet system. The Internet engineers use the term gateway to refer to devices that are also known as routers.

The routing decision

All hosts must make a decision when sending a datagram out onto the network. They must develop an answer for the question ‘‘Is this datagram addressed to a host on the network system to which I am directly attached, or must it be sent on to a gateway for further routing?’’ This activity on the part of the host is referred to as ‘‘making the routing decision.’’

TCP/IP routing algorithm

Given what we’ve seen so far, the routing algorithm may be described as follows (adapted from Comer, **‘‘Internetworking With TCP/IP,’’ p. 84.)**

TCP/IP Routing Algorithm:

Route_IP_Datagram(datagram, routing_table)

Extract destination IP address, Idest, from datagram
Compute IP address of destination network, Inetwork
    (This involves using the first few bits to
    discover the address class, or using
    subnet masks as described in the
    section on IP subnets below.)
if Inetwork matches any directly connected network address
    send datagram to destination over that network
    (This involves resolving the IP address into a physical
    address using ARP, and then placing the datagram in an
    appropriately addressed Ethernet packet, for example.)
else if Inetwork appears in host’s routing table
    route datagram as specified in table;
else if a default route has been specified
    route datagram to default gateway;
else declare a routing error;

Figure A-5. TCP/IP Routing Algorithm

1.12. IP Subnets

Most early network protocol suites have suffered from the unexpected and explosive growth of networks that has occurred in recent years, and TCP/IP is no exception.

Not too many years ago, when TCP/IP was first being developed, the number of networked computers was small and computer networking was considered a somewhat exotic research project. The rapid growth of UNIX systems running TCP/IP protocols and dependent on networking for critical computer resources has been a recent phenomenon.

The high levels of growth have been compounded by the fact that most networked sites have complex internal network topologies of their own, that they would like to manage by dividing everything up into separately addressed networks as is done on the Internet. Campus networks are a textbook example of local internets (with a small ‘‘i’’) springing up and growing steadily more complex every year.

Network design using subnets

A widely adopted design on campuses today is to treat a complex multi-building campus network as a small internet divided up into a number of individual IP networks. This makes network monitoring, management and troubleshooting more straightforward. This also allows for the installation of internal gateways that help manage network traffic, protect the network segments from excessive broadcasts, make it possible to interconnect multiple media systems, and so on.

As a result of these concerns, the IP addressing scheme came under pressure as hundreds of these complex computer networks sprang up. Registering all of the local nets as part of the entire Internet would have led to administrative headaches as well as to enormous routing tables in the Internet gateways. Some mechanism was needed for hiding the local complexity from the external Internet while preserving the ability to communicate across the entire system.

IP subnets and masks

To deal with the explosion of network numbers an approach called IP subnetting was developed. The idea is simply one of borrowing some bits from the local part of the IP address to form a special sub-network field for identifying the local networks in use. Or in other words, to form subnets of the IP network in question.

IP subnets function by imposing another level of structure onto the basic 32 bit IP address. Instead of encoding this new network structure in the first several bits of the address as in the case of IP network classes, IP subnets are defined with the use of a 32 bit subnet mask that is logically ANDed with the 32 bit IP address to extract the subnet address.

Local subnet architecture

The use of a subnet mask makes it possible for the local network designer to define a subnet field of whatever length the local network topology requires, within reason. In other words, the local network designer can apportion the subnet field according to the number of local networks and local hosts that must be accommodated.

In practice, the design of most Class B campus networks is accommodated with the use of 8 bit long subnet fields. This leaves an 8 bit field for the host address, with the result that the campus internet may have up to 255 local networks, with up to 254 hosts each.

Subnet address in a Class B internet address

The subnet address is carved out of the local part of the Internet address. Subnets are used to address the individual physical networks that make up the campus internet. The following figure illustrates the division of the local part of a Class B Internet address into equally sized subnet and host address fields.

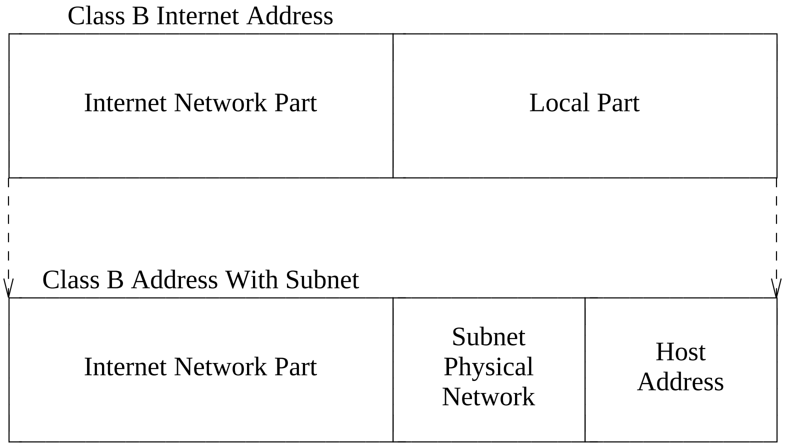

Figure A-6. IP Subnet As New Network Field In Local Part Of Internet Address

Default Class B subnet mask

For example, a default Class B address has a network part of 16 bits and a local part of 16 bits. A subnet mask that reflects the default Class B structure would be **255.255.0.0.** When a host uses this mask to derive the destination network address, it logically ANDs the mask with the destination IP address of the datagram the host is trying to route. The result is a 16 bit network number that a host on a default Class B network system uses to make the routing decision for this datagram. Once the routing decision has been made, the host sends the complete datagram on to its destination.

Example: Default Class B Subnet Mask ANDed With Destination IP Address

Subnet mask 255.255.0.0:
11111111 11111111 00000000 00000000

ANDed with Internet address 128.83.134.7:
10000000 01010011 10000110 00000111

Results in a network address of 128.83.0.0:
10000000 01010011 00000000 00000000

In this example, there is no local subnet structure in use, so the subnet mask simply reflects the default Class B network address structure. This demonstrates that the subnet mechanism actually encompasses the original Internet address class structure. Subnet masks can be used to derive standard IP network classes as well as for defining local subnetworks of those classes.

Subnetted Class B example

Suppose a campus has decided to divide their network system up into a series of departmental networks with gateways for each departmental network. One widely adopted approach is to split the local part of a Class B address into 8 bits of subnet and 8 bits of host address, allowing the Class B address to potentially provide up to 255 networks of 256 hosts each. (In practice the all ones and all zeros host addresses are reserved for compatibility with the old and new broadcast standards.)

The subnet mask for this example network would be **255.255.255.0.** When ANDed with a destination IP address, this mask results in a network and subnet address for routing purposes.

For example, the destination host address **128.83.134.7** when ANDed with the subnet mask **255.255.255.0** results in a network address of **128.83.134.0** for routing purposes. The host would then work through the routing algorithm to discover whether this matched the address of the attached network or whether the datagram needed to be sent to a local sub-network gateway for routing within the campus internet, or sent to the default gateway for routing out into the larger Internet.

Example: Class B With 8 Bit Subnet Mask ANDed With Destination IP Address

Subnet mask 255.255.255.0:
11111111 11111111 11111111 00000000

ANDed with Internet address 128.83.134.7:
10000000 01010011 10000110 00000111

Results in a network address of 128.83.134.0:
10000000 01010011 10000110 00000000

Datagrams from external hosts

Hosts located outside the campus need not know about the local subnet structure to send datagrams to a local host. This is because the gateway that is located between the local network system and the external Internet knows about the local subnet structure. It can translate the destination address from a datagram sent to the Class B address assigned to the entire local network into a destination address that reflects the local subnet structure.

For example, a host at some other point on the Internet need only route based on the IP address class. To send a datagram to host **128.83.134.7** the external host need only send it on to network **128.83.0.0** by whatever set of gateways handle the routes to that network.

The datagram will arrive at the gateway that connects the Internet with the local campus network system. At that point, the campus gateway knows what subnet structure is in use and can apply the correct mask to the datagram to extract the local subnet address from the ‘‘local part’’ of the IP destination address, and then route the datagram to the local subnet as required.

1.13. UTnet Subnet Architecture

UTnet is currently using the default Class B address mask of **255.255.0.0.** The campus backbone network operates today as one large network that travels to every building on campus, and there are no subnetwork gateways in place.

Nonetheless, UTnet host addresses are assigned *as though* an 8 bit subnet architecture is in use locally. This approach is generally referred to as using ‘‘pseudo-subnets.’’ There are several advantages to the use of pseudo-subnets on a growing campus internet.

Advantages of pseudo-subnets

The chief advantage is that using pseudo-subnets makes it possible to more easily accommodate future changes in the UTnet system. Currently the network relies on LAN Bridge 100s for traffic control and rudimentary routing. As the host population increases and the media system becomes more complex, it is reasonable to expect that the use of internal subnet gateways will begin to occur.

When a subnet gateway is installed, the host addresses will easily accommodate the change to true subnets due to the presence of the pseudo-subnet number in the host address field. If the hosts had not been addressed with pseudo-subnets, the installation of a subnet gateway would have required assigning new addresses to every host on the departmental subnet.

Pseudo-subnets make present day troubleshooting much easier as well, since the network managers can determine the location of a failing or misconfigured host by means of its pseudo-subnet number.

UTnet uses default Class B subnet mask

Despite the presence of pseudo-subnet addresses, the actual operation of the UTnet system is based on the default Class B network address. This means that even though you may have heard that UTnet hosts are given subnet addresses, there is no local subnet architecture in use on UTnet today. The subnet mask for hosts on UTnet must be set to reflect the default IP Class B address structure **(255.255.0.0).**

Even when building networks are converted to the use of subnet gateways it turns out that keeping a standard Class B subnet mask for UTnet hosts works well. That way the host's configuration can be kept as simple as possible. Upon installation at a building network the subnet gateway uses the actual subnet routing mask for routing purposes, but the hosts simply use static defaults or ARP-based routing to discover the gateway. ARP-based routing can help negate any side effects of incorrect host netmasks.

This means that hosts with different pseudo-subnet addresses that are currently sharing the same network wire can continue to share a network segment after the gateway is installed. Keeping the host configuration as simple as possible results in the maximum flexibility when it comes to converting from simple bridging of datagrams on UTnet to routing through subnet gateways.