TCP/IP Tutorial

This tutorial is intended to supply a brief overview of TCP/IP protocol. Explanations of IP addresses, classes, netmasks, subnetting, and routing are provided, and several example networks are considered.

The IP Address and Classes

Hosts and networks

IP addressing is based on the concept of hosts and networks. A **host** is essentially anything on the network that is capable of receiving and transmitting IP packets on the network, such as a workstation or a router. It is not to be confused with a server: servers and client workstations are all IP hosts.

The hosts are connected together by one or more **networks**. The IP address of any host consists of its network address plus its own host address on the network. IP addressing, unlike, say, IPX addressing, uses one address containing both network and host address.

How much of the address is used for the network portion and how much for the host portion varies from network to network.

IP addressing

An IP address is 32 bits wide, and as discussed, it is composed of two parts: the **network number**, and the **host number** [1, 2, 3]. By convention, it is expressed as four decimal numbers separated by periods, such as "200.1.2.3" representing the decimal value of each of the four bytes. Valid addresses thus range from 0.0.0.0 to 255.255.255.255, a total of about 4.3 billion addresses. The first few bits of the address indicate the Class that the address belongs to:

Class	Prefix	Network Number	Host Number
A	0	Bits 1-7	Bits 8-31
B	10	Bits 2-15	Bits 16-31
C	110	Bits 3-23	Bits 24-31
D	1110	N/A	
E	1111	N/A	

The bits are labeled in network order, so that the first bit is bit 0 and the last is bit 31, reading from left to right. Class D addresses are multicast, and Class E are reserved. The range of network numbers and host numbers may then be derived:

Class	Range of Net Numbers	Range of Host Numbers
A	0 to 126	0.0.1 to

		255.255.254
B	128.0 to 191.255	0.1 to 255.254
C	192.0.0 to 233.255.255	1 to 254

Any address starting with 127 is a loopback address and should never be used for addressing outside the host. A host number of all binary 1's indicates a directed broadcast over the specific network. For example, 200.1.2.255 would indicate a broadcast over the 200.1.2 network. If the host number is 0, it indicates "this host". If the network number is 0, it indicates "this network" [2].

All the reserved bits and reserved addresses severely reduce the available IP addresses from the 4.3 billion theoretical maximum. Most users connected to the Internet will be assigned addresses within Class C, as space is becoming very limited. This is the primary reason for the development of IPv6, which will have 128 bits of address space.

Basic IP Routing

Classed IP Addressing and the Use of ARP

Consider a small internal TCP/IP network consisting of one Ethernet segment and three nodes. The IP network number of this Ethernet segment is 200.1.2. The host numbers for A, B, and C are 1, 2, and 3 respectively. These are Class C addresses, and therefore allow for up to 254 nodes on this network segment.

Each of these nodes have corresponding Ethernet addresses, which are six bytes long. They are normally written in hexadecimal form separated by dashes (02-FE-87-4A-8C-A9 for example).

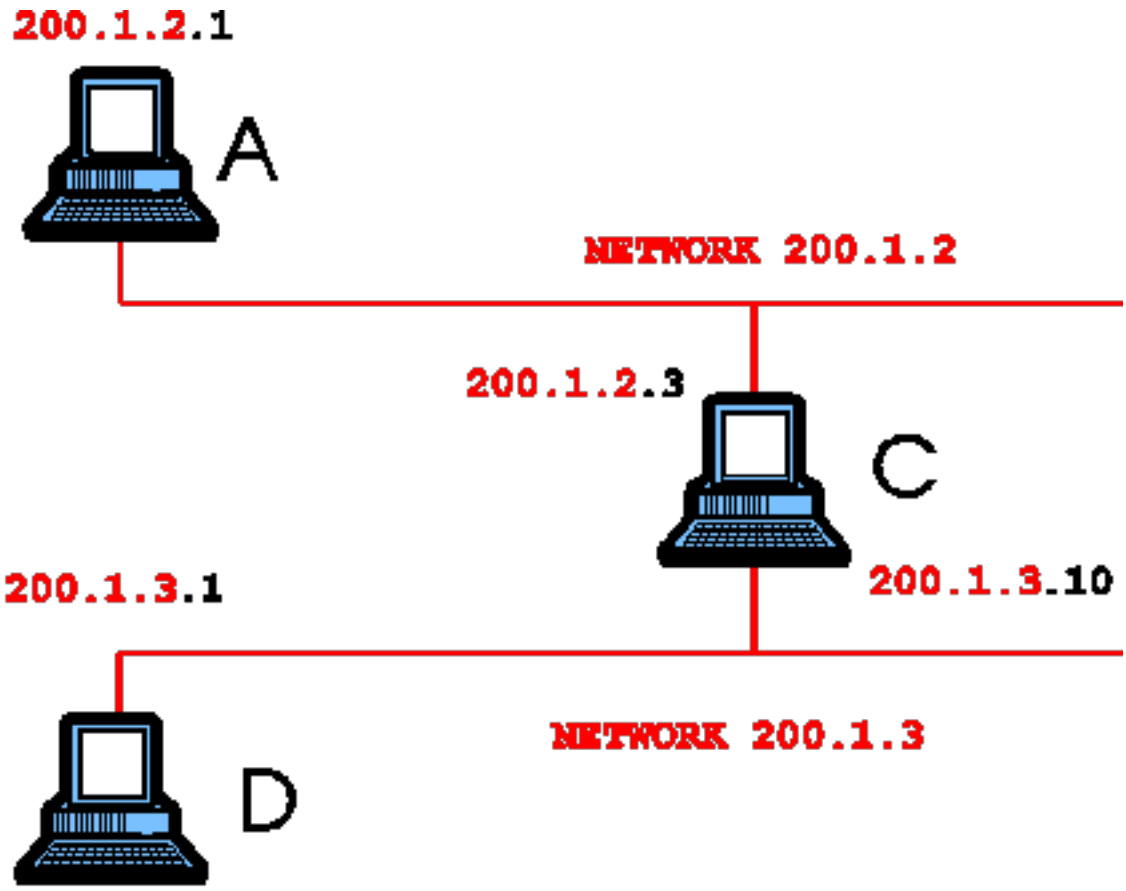
In the diagram above and subsequent diagrams, we have emphasized the network number portion of the IP address.

Suppose that A wanted to send a packet to C for the first time, and that it knows C's IP address. To send this packet over Ethernet, A would need to know C's Ethernet address. The **Address Resolution Protocol (ARP)** is used for the dynamic discovery of these addresses [1].

ARP keeps an internal table of IP address and corresponding Ethernet address. When A

attempts to send the IP packet destined to C, the ARP module does a lookup in its table on C's IP address and will discover no entry. ARP will then broadcast a special request packet over the Ethernet segment, which all nodes will receive. If the receiving node has the specified IP address, which in this case is C, it will return its Ethernet address in a reply packet back to A. Once A receives this reply packet, it updates its table and uses the Ethernet address to direct A's packet to C. ARP table entries may be stored statically in some cases, or it keeps entries in its table until they are "stale" in which case they are flushed.

Consider now two separate Ethernet networks that are joined by a PC, C, acting as an IP router (for instance, if you have two Ethernet segments on your server).



Device C is acting as a **router** between these two networks. A router is a device that chooses different paths for the network packets, based on the addressing of the IP frame it is handling. Different routes connect to different networks. The router will have more than one address as each route is part of a different network.

Since there are two separate Ethernet segments, each network has its own Class C network number. This is necessary because the router must know which network interface to use to reach a specific node, and each interface is assigned a network number. If A wants to send a packet to E, it must first send it to C who can then forward the packet to E. This is accomplished by having A use C's Ethernet address, but E's IP address. C will receive a packet destined to E and will then forward it using E's Ethernet address. These Ethernet addresses are obtained using ARP as described earlier.

If E was assigned the same network number as A, 200.1.2, A would then try to reach E in the

same way it reached C in the previous example - by sending an ARP request and hoping for a reply. However, because E is on a different physical wire, it will never see the ARP request and so the packet cannot be delivered. By specifying that E is on a different network, the IP module in A will know that E cannot be reached without having it forwarded by some node on the same network as A.

Direct vs. Indirect Routing

Direct routing was observed in the first example when A communicated with C. It is also used in the last example for A to communicate with C. If the packet does not need to be forwarded, i.e. both the source and destination addresses have the same network number, direct routing is used.

Indirect routing is used **when the network numbers of the source and destination do not match**. This is the case where the packet must be forwarded by a node that knows how to reach the destination (a router).

In the last example, A wanted to send a packet to E. For A to know how to reach E, it must be given routing information that tells it who to send the packet to in order to reach E. This special node is the "gateway" or router between the two networks. A Unix-style method for adding a routing entry to A is

```
route add [destination_ip] [gateway] [metric]
```

Where the metric value is the number of hops to the destination. In this case,

```
route add 200.1.3.2 200.1.2.3 1
```

will tell A to use C as the gateway to reach E. Similarly, for E to reach A,

```
route add 200.1.2.1 200.1.3.10 1
```

will be used to tell E to use C as the gateway to reach A.

It is necessary that C have two IP addresses - one for each network interface. This way, A knows from C's IP address that it is on its own network, and similarly for E. Within C, the routing module will know from the network number of each interface which one to use for forwarding IP packets.

In most cases it will not be necessary to manually add this routing entry. It would normally be sufficient to set up C as the **default gateway** for all other nodes on both networks. The default gateway is the IP address of the machine to send all packets to that are not destined to a node on the directly-connected network. The routing table in the default gateway will be set up to forward the packets properly, which will be discussed in detail later.

Static vs. Dynamic Routing

Static routing is performed using a preconfigured routing table which remains in effect indefinitely, unless it is changed manually by the user. This is the most basic form of routing, and it usually requires that all machines have statically configured addresses, and definitely requires that all machines remain on their respective networks. Otherwise, the user must manually alter the routing tables on one or more machines to reflect the change in network

topology or addressing. Usually at least one static entry exists for the network interface, and is normally created automatically when the interface is configured.

Dynamic routing uses special routing information protocols to automatically update the routing table with routes known by peer routers. These protocols are grouped according to whether they are Interior Gateway Protocols (IGPs) or Exterior Gateway Protocols. Interior gateway protocols are used to distribute routing information inside of an Autonomous System (AS). An AS is a set of routers inside the domain administered by one authority. Examples of interior gateway protocols are OSPF and RIP. Exterior gateway protocols are used for inter-AS routing, so that each AS may be aware of how to reach others throughout the Internet. Examples of exterior gateway protocols are EGP and BGP. See RFC 1716 [11] for more information on IP router operations.

WAN Routing

Our **WAN Cards** provide a network interface, and do not actually route packets according to IP address, or maintain IP routing information. Packet routing between interfaces is accomplished by the protocol stack, which can send IP based dynamic routing protocols over WAN card. The information and protocols needed for dynamic routing are handled by the protocol stack. **In practice, it is almost always better to use explicit static routing table entries rather than relying on dynamic routing**.

Advanced IP Routing

The Netmask

When setting up each node with its IP address, the netmask must also be specified. This mask is used to specify which part of the address is the network number part, and which is the host part. This is accomplished by a logical bitwise-AND between the netmask and the IP address. The result specifies the network number. For Class C, the netmask will always be 255.255.255.0; for Class B, the netmask will always be 255.255.0.0; and so on. When A sent a packet to E in the last example, A knew that E wasn't on its network segment by comparing A's network number 200.1.2 to the value resulting from the bitwise-AND between the netmask 255.255.255.0 and the IP address of E, 200.1.3.2, which is 200.1.3.

The netmask becomes very important, and more complicated, when "classless" addressing is used.

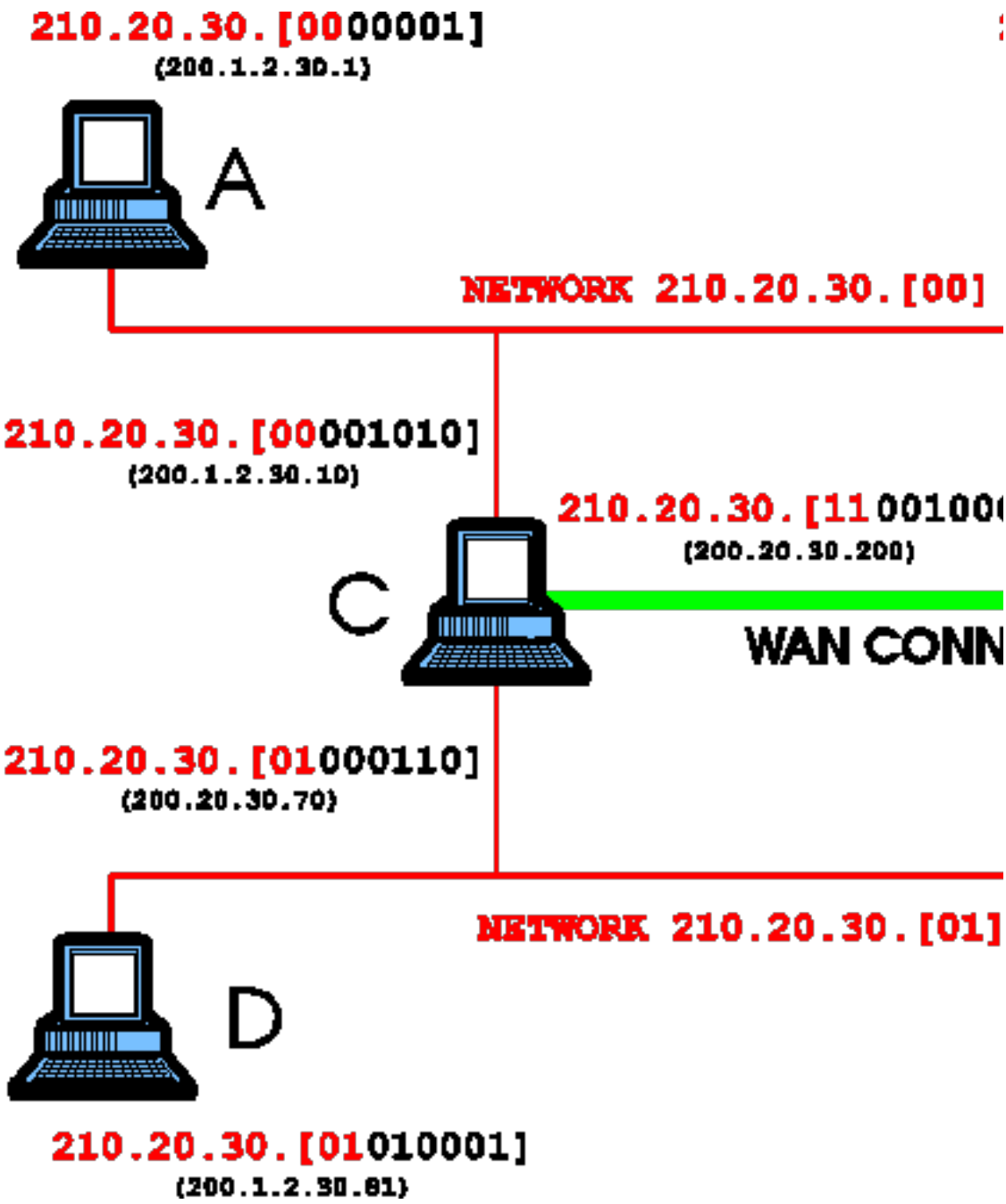
Hierarchical Sub-Allocation of Class C Addresses

To make more efficient use of Class C addresses in the Internet community, these addresses are subnetted hierarchically from the service provider to the organization. They are allocated bitmask-oriented subsets of the provider's address space [4, 5]. These are **classless addresses**.

Consider the following example of a small organization consisting of two Ethernet segments connecting to an Internet service provider using a WAN router that emulates an additional network segment, such as provided by our **Wan Cards**. The service provider has been allocated several different Class C addresses to be used for its clients. This particular organization has been allocated the network number 210.20.30, and the gateway address at the provider end is 210.20.30.254.

Networks 210.20.30.0->63
210.20.30.64->127
210.20.30.192->255

Netmask 255.255.255.192



We have expanded the last byte of the IP address so that we can show the network subaddressing. The standard IP address nomeclature is shown below this expanded version.

If the organization happened to have just one computer, C, and the entire Class C address is available for use, then the IP address for C may be anything in the range 210.20.30.1 to 210.20.30.253, and its default gateway would be 210.20.30.254 with netmask 255.255.255.0.

However, with two networks plus our Gateways, which must also be on its own network, the Class C address must somehow be **subnetted**. This is accomplished by using one or more of the bits that are normally allocated to the host number as part of the Class C address, in order to extend the size of the network number. In this case, 210.20.30 has been extended to include four networks, and the netmask has changed to 255.255.255.192 to reflect the additional use of two bits for the network number in the IP address.

Writing the netmask 255.255.255.192 in binary (from FFFFFFC0 in hex) is 11111111/11111111/11111111/11000000, with /' separating the bytes for clarity. Since the organization is allocated all of 210.20.30 (D2141E hex), it has the use of the four following network numbers (in binary):

Net #	IP Network Number
0	11010010/00010100/00011110/00
1	11010010/00010100/00011110/01
2	11010010/00010100/00011110/10
3	11010010/00010100/00011110/11

This leaves 6 bits at the end to use for host number, leaving space for 62 host nodes per network (all 0's and all 1's are reserved). The following addresses are therefore valid for hosts to use:

Net #	Address Range
0	210.20.30.1 to 210.20.30.62
1	210.20.30.65 to 210.20.30.126
2	210.20.30.129 to 210.20.30.190
3	210.20.30.193 to 210.20.30.254

In this example, Net#2 is reserved for future use.
The IP addresses and netmasks for each interface are:

Interface	IP Addressnbsp	Netmask
Node A	210.20.30.1	255.255.255.192
Node B	210.20.30.2	255.255.255.192
Node C (AB)	210.20.30.10	255.255.255.192
Node C (DE)	210.20.30.70	255.255.255.192
Node C (WAN)	210.20.30.200	255.255.255.192
Node D	210.20.30.81	255.255.255.192
Node E	210.20.30.82	255.255.255.192

The routing tables will be set for each node as follows. The destination address 0.0.0.0 indicates the default destination, if no other specific routes are configured for the given packet destination. This default destination is where all packets will be sent, and it is assumed that this destination is capable of forwarding these packets to the ultimate destination, or to another router along the appropriate path.

Node A Network Address	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	210.20.30.10	210.20.30.1
210.20.30.0	255.255.255.192	210.20.30.1	210.20.30.1
Node B Network Address	**Netmask**	**Gateway**	**Interface**
0.0.0.0	0.0.0.0	210.20.30.10	210.20.30.2
210.20.30.0	255.255.255.192	210.20.30.2	210.20.30.2
Node C Network	**Netmask**	**Gateway**	**Interface**

Address			
0.0.0.0	0.0.0.0	210.20.30.254	210.20.30.200
210.20.30.0	255.255.255.192	210.20.30.10	210.20.30.10
210.20.30.64	255.255.255.192	210.20.30.70	210.20.30.70
210.20.30.192	255.255.255.192	210.20.30.200	210.20.30.200

Node D

Network Address	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	210.20.30.70	210.20.30.81
210.20.30.64	255.255.255.192	210.20.30.81	210.20.30.81

Node E

Network Address	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	210.20.30.70	210.20.30.82
210.20.30.64	255.255.255.192	210.20.30.82	210.20.30.82

Node G

Network Address	Netmask	Gateway	Interface
210.20.30.0	255.255.255.0	210.20.30.200	210.20.30.254

(Plus all other pertinent entries)

The metric value, or hop count, is optional, but would be 0 for all gateways that are the same as the hosts, and greater than 0 if the destination is reached via one or more gateways. The metric for the default routes are indeterminate, but would always be at least 1.

For example, if D sent an ICMP echo request packet out onto the Internet, let's say to address 140.51.120.30, then first D would AND the netmask 255.255.255.192 with 140.51.120.30 to determine the network number. It would then find that it does not match the network number 210.20.30.64, and so it chooses the default route which points to the gateway 210.20.30.70. It then uses the Ethernet address of Node C (DE) to forward the IP packet to the gateway.

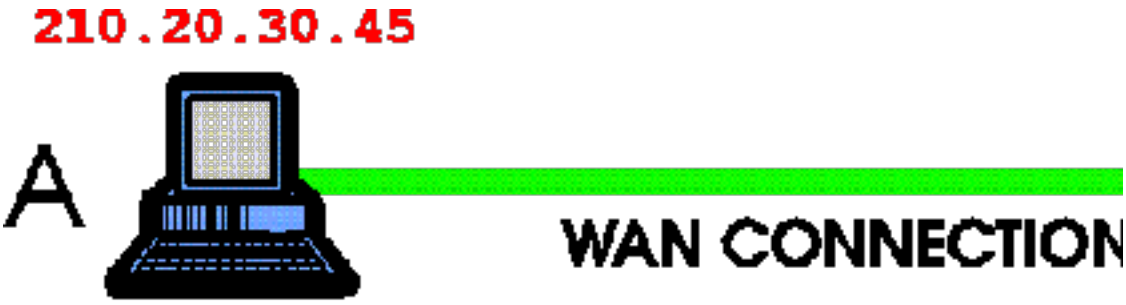
When C receives this packet, it will see that it is destined to 140.51.120.30. It checks all the routes in its table and determines that this address is not located on any of the listed networks in the routing table, and so it chooses the default route. It uses the WAN interface, of IP address 210.20.30.200 to send the packet to 210.20.30.254 (G). From then on, the packet will propagate from gateway to gateway until it reaches 140.51.120.30. When this node replies, the packet will be inbound on interface 210.20.30.200 (C) with destination address 210.20.30.81 (D). Node C will discover that 210.20.30.81 is on the 210.20.30.64 network and uses the interface 210.20.30.70 to send the packet back home to D.

TCP/IP Setup Examples by Protocol Stack and Platform

Two examples will be presented to explain how to set up the IP addressing and routing information when connecting to an Internet service provider using the **G1000**. The first case is when only one machine will be connected, and the other case describes the connection of a LAN to the Internet. The third example briefly illustrates the addressing and routing techniques for connecting two LANs over a point-to-point WAN connection.

Example 1: Single Node Connection to WAN Gateway

Assume that the node A is the G1000 and is assigned the IP address 210.20.30.45, and that the gateway address is 199.99.88.77.



The netmask for A may be set to 255.255.255.255, indicating no other nodes on the local network, and the gateway is set to 199.99.88.77. A default route must be set up at Node A as well, which provides the route for all packets whose destination does not correspond to any specific routing entries.

Node A			
Network Address	**Netmask**	**Gateway**	**Interface**
0.0.0.0	0.0.0.0	19.99.88.77	210.20.30.45
Node G			
Network Address	**Netmask**	**Gateway**	**Interface**
210.20.30.45	25.255.255.255	199.99.88.77	199.99.88.77

(Plus all other pertinent entries)

The routing for Node G is highly dependent on the context, and the above entry only serves as an example. The netmask of all 1's in this case is used to only allow packets destined to 210.20.30.45 to be forwarded to Node A, as there may be 253 other nodes connected in a similar way under this Class C network 199.99.88.0.

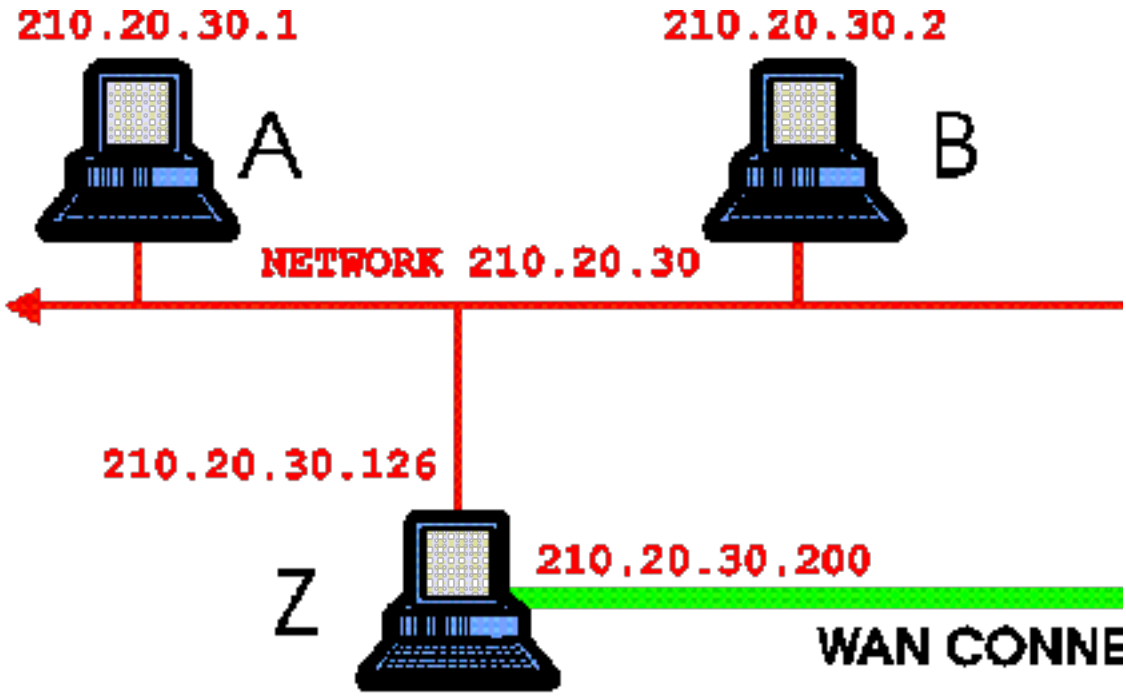
When the protocol stack's configuration asks for a default gateway, specifying 199.99.88.77 will cause the default routing entry 0.0.0.0 to be added automatically. It must be added manually if for some reason the stack does not ask for it.

The specific methods of configuring each protocol stack will be explained in detail in Example 2.

Example 2: LAN Connection to WAN Gateway

The following network topology will be used as an example, where one LAN is connected to the Internet for simplicity. This will also demonstrate the use of a different netmask for creating two Class C subnets. Note however that the remote WAN gateway may have an IP address outside the local Class C network, in which case the local WAN gateway interface will usually have an IP address on the same network as the remote WAN gateway. If this is the case, subnetting as shown below may not be necessary, unless more than one local network segment is involved.

Networks 210.20.30.0->127,210.20.30.128->255 **Netmask** 255.255.255.128



Node A is one of the many workstations on the Ethernet segment Net 0. Node Z is a JBM unit and the gateway from this Ethernet to the Internet service provider's gateway machine G. Some of the other workstations have been labeled as B to Y for illustration, but will not be referred to in this example as their setup will be the same as for A.

In this case, since only two subnets were needed, only one bit from the host address space need be sacrificed. Writing the netmask 255.255.255.128 in binary (from FFFFFF80 in hex) is 11111111/11111111/11111111/10000000, with /' separating the bytes for clarity. Since the organization is allocated all of 210.20.30 (D2141E hex), it has the use of the two following network numbers (in binary):

Net #	IP Network Number
0	11010010/00010100/00011110/0
1	11010010/00010100/00011110/1

This leaves 7 bits at the end to use for host number, leaving space for 126 host nodes per network (all 0's and all 1's are reserved). The following addresses are therefore valid for hosts to use:

Net #	Address Range
0	210.20.30.1 to 210.20.30.126
1	210.20.30.129 to 210.20.30.254

The IP addresses and netmasks for each interface are:

Interface	IP Address	Netmask
Node A	210.20.30.1	255.255.255.128
Node Z	210.20.30.126	255.255.255.128
Node Z	210.20.30.200	255.255.255.128

The routing tables will be set for each node as follows. Note that the destination address 0.0.0.0 indicates the default destination, if no other specific routes are indicated.

Node A			
Network Address	**Netmask**	**Gateway**	**Interface**
0.0.0.0	0.0.0.0	210.20.30.126	210.20.30.1
210.20.30.0	255.255.255.128	210.20.30.1	210.20.30.1
Node Z			
Network Address	**Netmask**	**Gateway**	**Interface**
0.0.0.0	0.0.0.0	210.20.30.254	210.20.30.200
210.20.30.0	255.255.255.128	210.20.30.126	210.20.30.126
210.20.30.128	255.255.255.128	210.20.30.200	210.20.30.200
Node G			
Network Address	**Netmask**	**Gateway**	**Interface**
210.20.30.0	255.255.255.0	210.20.30.200	210.20.30.254

(Plus all other pertinent entries)

If you need further information on TCP/IP or our implementation and support for this protocol, please e-mail us.

Information in this document is provided by Sangoma Technologies.