# APPLICATION NOTE

**THIS INFORMATION PROVIDED BY AUTOMATIONDIRECT.COM TECHNICAL SUPPORT IS PROVIDED "AS IS" WITHOUT A GUARANTEE OF ANY KIND.**
These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

**Product Family: Miscellaneous**

**Number: AN-MISC-034**

**Subject: Ethernet Fundamentals**

**Date Issued: 10-15-2013**

**Revision:**

# Ethernet Fundamentals 2013

# I. Various numbers/addresses used for Ethernet communications

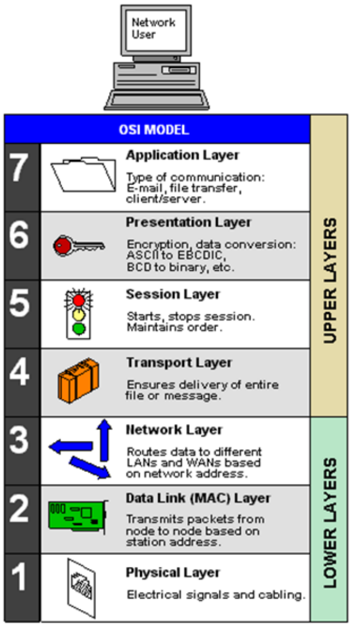
There are 3 primary values or addresses that we will discuss in this document to help explain how Ethernet messages get from 1 device to another:

- MAC ID
- IP Address (along with Subnet Mask and Default Gateway address)
- UDP/TCP Port number

This is not to say that these are the only important numbers used in Ethernet communications but they are key in trying to understand how a device is routed across a network.

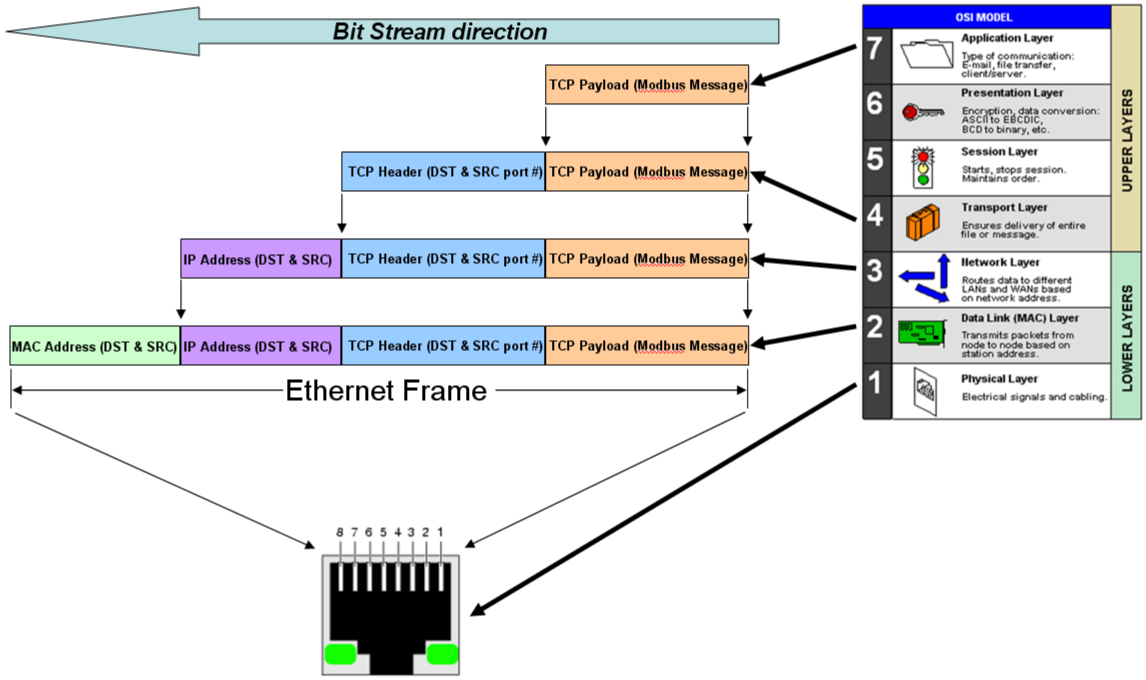
# II. Review of the OSI 7 layer model

It is important to understand the OSI 7 layer model for Ethernet in order to understand how Ethernet packets move across a typical network. Many of the devices that an Ethernet packet will go through determine where to send the packet based upon one or more of the Ethernet layers. Many Ethernet switches and routers are classified as “layer 2” or “layer 3” devices.


Network User
OSI MODEL
7
Application Layer
Type of communication: E-mail, file transfer, client/server.
6
Presentation Layer
Encryption, data conversion: ASCII to EBCDIC, BCD to binary, etc.
5
Session Layer
Starts, stops session. Maintains order.
4
Transport Layer
Ensures delivery of entire file or message.
3
Network Layer
Routes data to different LANs and WANs based on network address.
2
Data Link (MAC) Layer
Transmits packets from node to node based on station address.
1
Physical Layer
Electrical signals and cabling.
UPPER LAYERS
LOWER LAYERS

The diagram above is a good chart explaining the different layers in the OSI model. It is important to note that not all Ethernet messages will include all of these layers. Most, in fact, will only contain 4 or 5 layers.

The image below shows the structure of a Modbus TCP message and how it corresponds to the OSI model.


Bit Stream direction
TCP Payload (Modbus Message)
TCP Header (DST & SRC port #)
IP Address (DST & SRC)
MAC Address (DST & SRC)
Ethernet Frame
8 7 6 5 4 3 2 1
OSI MODEL
7 Application Layer
Type of communication: E-mail, file transfer, client/server.
6 Presentation Layer
Encryption, data conversion: ASCII to EBCDIC, BCD to binary, etc.
5 Session Layer
Starts, stops session. Maintains order.
4 Transport Layer
Ensures delivery of entire file or message.
3 Network Layer
Routes data to different LANs and WANs based on network address.
2 Data Link (MAC) Layer
Transmits packets from node to node based on station address.
1 Physical Layer
Electrical signals and cabling.
UPPER LAYERS
LOWER LAYERS

When an Ethernet message is created in an application, typically the first step is to form the application message which in the case show above would be a Modbus message.

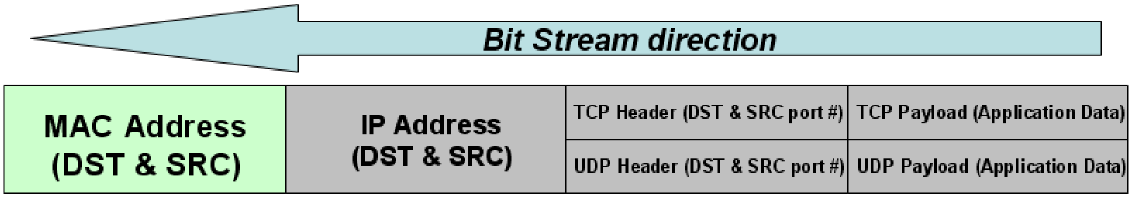
The application message would then be sent to what it typically referred to as the "stack". The "stack" is the commonly used name for the code that exists in the Operating System of most devices that actually takes care of forming the TCP/UDP packet and then adding on the proper IP header, MAC header and sending this entire Ethernet frame to the physical chip (PHY) that actually sends the electrical signals on the wire.

As you can see from the image above, "layer 2" corresponds to the MAC address layer and "layer 3" corresponds to the IP address layer.

# III. MAC ID Communication

## A. What is a MAC ID?

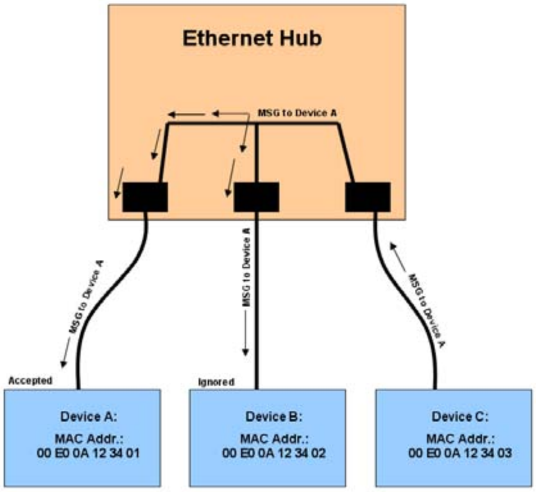
The MAC ID value is the physical hardware address of a device. It is the first value that comes in to the Ethernet port. This value is typically non-configurable as opposed to the IP address or port number.

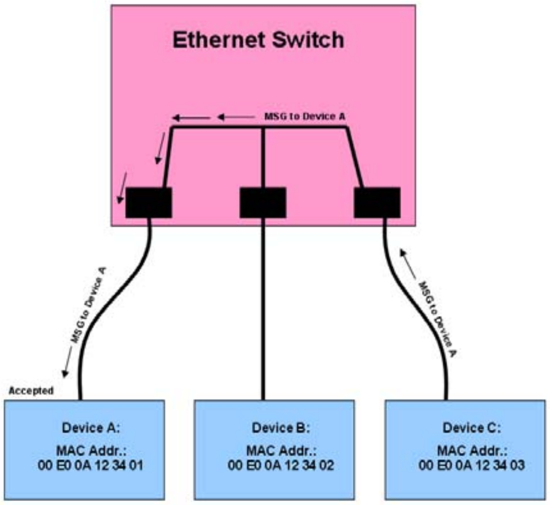

Bit Stream direction
MAC Address (DST & SRC)
IP Address (DST & SRC)
TCP Header (DST & SRC port #)
TCP Payload (Application Data)
UDP Header (DST & SRC port #)
UDP Payload (Application Data)

## B. MAC ID role with Ethernet switches and hubs

Standard off the shelf unmanaged switches use, what is called, “layer 2” switching. In other words, the switch uses the Data Link layer that contains the MAC address to decide which physical port to send a packet that it has received. So an unmanaged switch only needs to read the first 6 bytes of an Ethernet packet to decide where to send it. This prevents the unnecessary collisions that occur in an Ethernet hub.

The 2 diagrams below show the difference between an Ethernet hub versus an Ethernet switch. The Switch uses the MAC address to intelligently decide which ports to send out the messages versus a hub that always sends every message received from a port out to all of its ports.





### C. How MAC address is derived from IP address (ARP)

Even though the value needed to communicate to a device is the MAC address, the user almost never actually specifies the MAC address when sending a message to a given device. Usually the IP address of the target is specified when sending a message to another device in an application so how does the sending device know what the MAC address is if only given the IP address? The answer is through ARP (Address Resolution Protocol). Whenever a sending device wants to send a message to another device, it will send a broadcast message (MAC address = FF FF FF FF FF FF and IP address = FF.FF.FF.FF) using the ARP protocol asking for the MAC address of a specified IP address. The device that has the specified IP address will reply with its MAC address so that the sending device can now form a proper message with the destination MAC address.

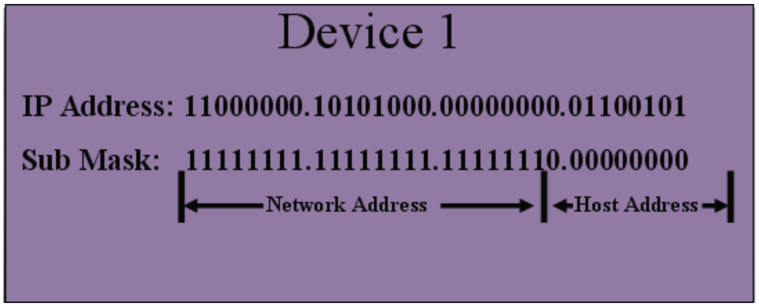
Most devices will typically keep an ARP table that contains a pre-defined list of MAC addresses to IP addresses to reduce the amount of traffic on the network. Instead of sending an ARP message every time the sending device needs to send a message, it will first check its ARP table to see if this entry already exists.

Another method commonly used is for a device to send a "gratuitous ARP" message when it is powered up or joins the network that announces its IP address and MAC address and any device listening can then add this entry to its ARP table.

## IV. IP Addressing

The next obvious question is: If I have to use a MAC address to communicate to a device why do I need to bother with an IP address?

The answer is: for network routing. The IP address is not just a device number (also called "Host Address") but also contains a network number (see the image below).


Device 1
IP Address: 11000000.10101000.00000000.01100101
Sub Mask: 11111111.11111111.11111110.00000000
Network Address
Host Address

As you can see from the image above, the Subnet Mask determines what part of the IP address is the Network Address and what part is the Host Address.

### A. How IP addresses work with subnet masks

The most common reason that 2 IP (UDP or TCP) devices are unable to communicate to each other is that the IP addresses and Subnet masks are not configured properly.

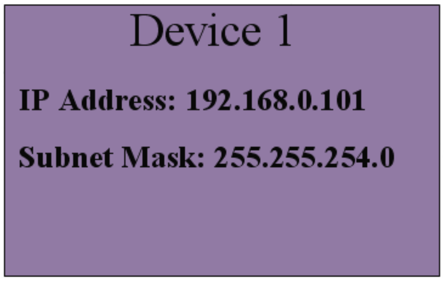
The IP address and Subnet masks are used in conjunction to create a “Network Address” and “Host Address” as mentioned above.
This allows more networks and more devices per network and reduced traffic per network in most cases.
IP addresses and Subnet masks are usually displayed in decimal notation:
IP Address:  192.168.10.1
Subnet mask:  255.255.255.0

In order to better understand how these 2 addresses are used together, it is necessary to view them in binary form:
IP Address:    11000000.10101000.00001010.00000001
Subnet mask: 11111111.11111111.11111111.00000000

Wherever there are 1’s in the Subnet mask, this is the Network Address used.  Where there are 0’s, this indicates the Host Address:
IP Address:    11000000.10101000.00001010.00000001
Subnet mask: 11111111.11111111.11111111.00000000
|←        Network Address        →|←Host→|

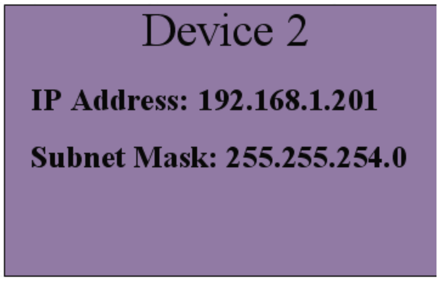
- Network Address = 192.168.10.0
- Host Address = 0.0.0.1

The key point for 2 devices to be able to communicate with each other is:  Their Network Addresses must be the same and their Host Addresses must be different.


Device 1

**IP Address: 192.168.0.101**

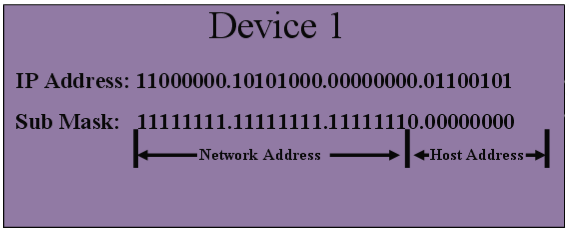
**Subnet Mask: 255.255.254.0**


Device 2

**IP Address: 192.168.1.201**

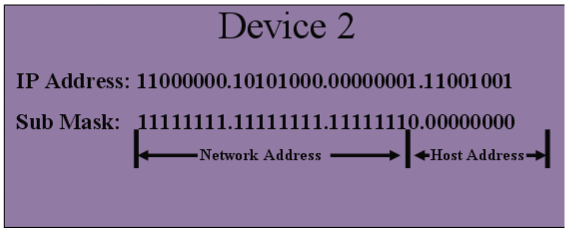
**Subnet Mask: 255.255.254.0**

Although it may ‘appear’ that these 2 devices shown above cannot talk to each other they are, in fact, in the same network and can communicate to each other.
This is more clearly seen when you look at the addresses in binary form (shown below).  Note that they have the same network address but different Host addresses.


Device 1

**IP Address: 11000000.10101000.00000000.01100101**

**Sub Mask:   11111111.11111111.11111110.00000000**


Device 2

**IP Address: 11000000.10101000.00000001.11001001**

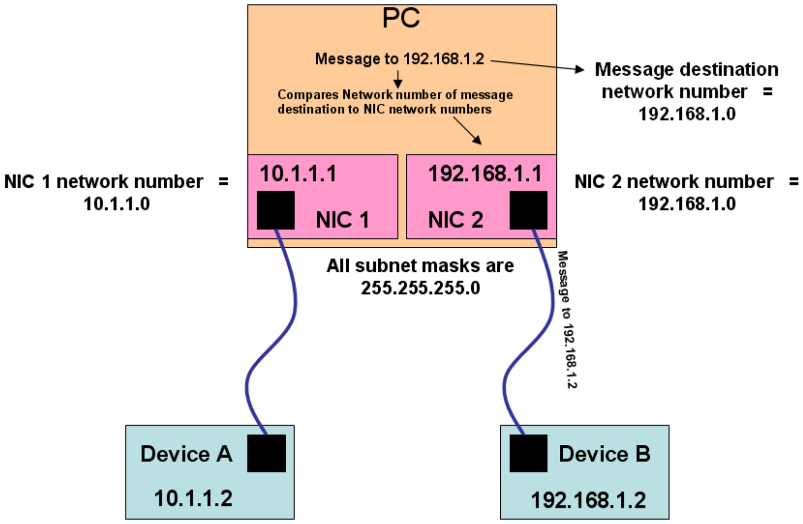
**Sub Mask:   11111111.11111111.11111110.00000000**

If one of the Devices shown above had been configured with a subnet mask of 255.255.255.0, they would then be in different networks and not be able to communicate with each other.

## B. IP address are used to route messages to the correct "path"

A device can use the IP address (in conjunction with the subnet mask) in several different manners to decide which way to send the message.

A very common situation where a PC needs to decipher the network address from the IP and subnet mask to route is when there is multiple network interface cards installed.


PC
Message to 192.168.1.2
Compares Network number of message destination to NIC network numbers
Message destination network number = 192.168.1.0
NIC 1 network number = 10.1.1.0
10.1.1.1
NIC 1
192.168.1.1
NIC 2
NIC 2 network number = 192.168.1.0
All subnet masks are 255.255.255.0
Message to 192.168.1.2
Device A
10.1.1.2
Device B
192.168.1.2

So in the example shown above, the PC has a message that it needs to send to Device B. It has to decide which Network Interface Card (NIC) to send the message out. The PC looks at the network number for the message and the 2 NICs, finds that the network number matches NIC 2 so decides to send the message out that port.

This is a fairly straight forward concept. The next question is: what if the message destination network number doesn't match either of the NICs network numbers?

This brings us to the next topic of Gateway addressing.

## C. Gateway address is used to reach Router for packets intended for other subnet

The default gateway address is typically the internal IP address of the local side of the router. Interestingly enough, the default gateway IP address does not actually appear in the message being sent to the router. The device actually uses the default gateway IP address to resolve to the MAC address of the router and the router MAC address is used as the destination MAC address in the message being sent. An illustration of this is shown below:

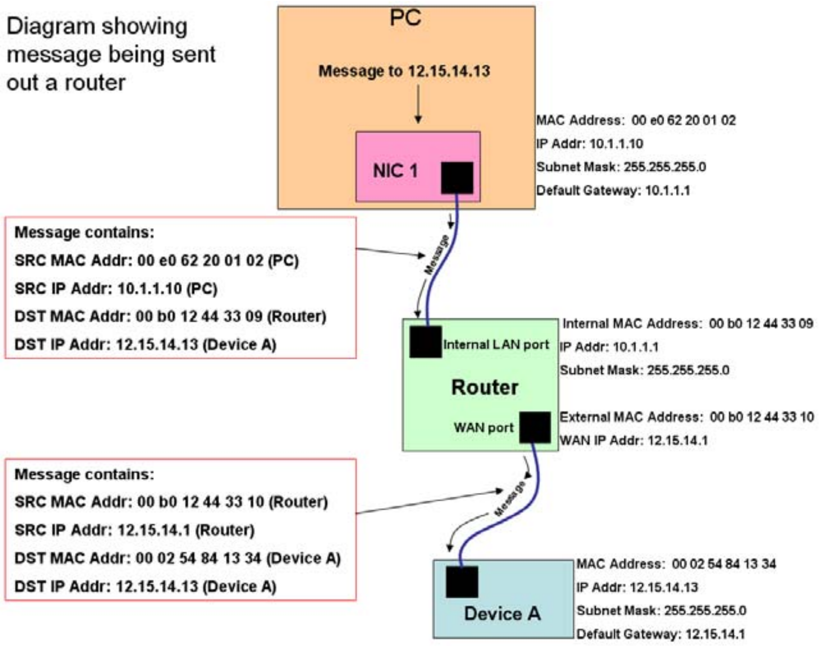


Diagram showing message being sent out a router

MAC Address: 00 e0 62 20 01 02
IP Addr: 10.1.1.10
Subnet Mask: 255.255.255.0
Default Gateway: 10.1.1.1

Message contains:
SRC MAC Addr: 00 e0 62 20 01 02 (PC)
SRC IP Addr: 10.1.1.10 (PC)
DST MAC Addr: 00 b0 12 44 33 09 (Router)
DST IP Addr: 12.15.14.13 (Device A)

Internal MAC Address: 00 b0 12 44 33 09
IP Addr: 10.1.1.1
Subnet Mask: 255.255.255.0

External MAC Address: 00 b0 12 44 33 10
WAN IP Addr: 12.15.14.1

Message contains:
SRC MAC Addr: 00 b0 12 44 33 10 (Router)
SRC IP Addr: 12.15.14.1 (Router)
DST MAC Addr: 00 02 54 84 13 34 (Device A)
DST IP Addr: 12.15.14.13 (Device A)

MAC Address: 00 02 54 84 13 34
IP Addr: 12.15.14.13
Subnet Mask: 255.255.255.0
Default Gateway: 12.15.14.1

Following the diagram above, we will go through the steps of a device sending an Ethernet message out through a router.

1. PC sends a message to Device A. Since the PC and Device A are in different subnets, the PCs message uses the Router's MAC address as its destination MAC address.
2. The Router takes the message and changes the Source MAC address and source IP address to its own MAC and IP. It uses Device A's MAC address and IP address for its destination addresses.

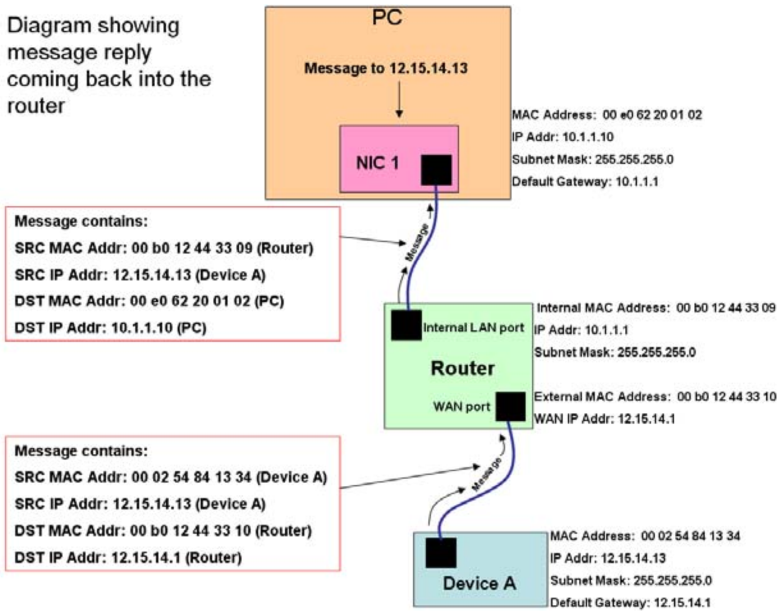


Diagram showing message reply coming back into the router

MAC Address: 00 e0 62 20 01 02
IP Addr: 10.1.1.10
Subnet Mask: 255.255.255.0
Default Gateway: 10.1.1.1

Message contains:
SRC MAC Addr: 00 b0 12 44 33 09 (Router)
SRC IP Addr: 12.15.14.13 (Device A)
DST MAC Addr: 00 e0 62 20 01 02 (PC)
DST IP Addr: 10.1.1.10 (PC)

Internal MAC Address: 00 b0 12 44 33 09
IP Addr: 10.1.1.1
Subnet Mask: 255.255.255.0

External MAC Address: 00 b0 12 44 33 10
WAN IP Addr: 12.15.14.1

Message contains:
SRC MAC Addr: 00 02 54 84 13 34 (Device A)
SRC IP Addr: 12.15.14.13 (Device A)
DST MAC Addr: 00 b0 12 44 33 10 (Router)
DST IP Addr: 12.15.14.1 (Router)

MAC Address: 00 02 54 84 13 34
IP Addr: 12.15.14.13
Subnet Mask: 255.255.255.0
Default Gateway: 12.15.14.1

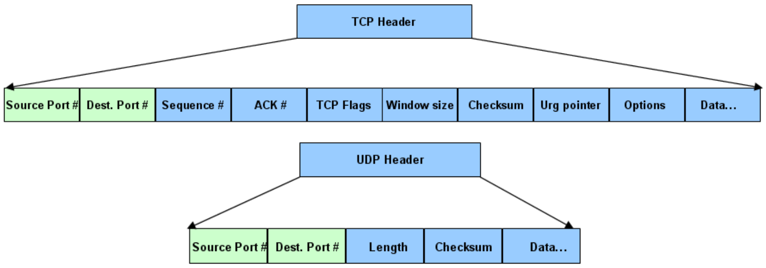
The steps for the reply message back into the router are shown below:

1. Device A sends a reply back to the Routers MAC and IP using its own MAC and IP as the source addresses.
2. The Router changes the destination MAC and IP addresses to the PCs MAC and IP. The Router changes the Source MAC address to the internal Router MAC Address but uses Device A's source IP address.

# V. TCP and UDP Port numbers

TCP and UDP port numbers are used by applications or processes when sending and receiving Ethernet packets.

## A. Port numbers are used for application and connection management



As you can see in the images above, the TCP and UDP port numbers are the first two fields in the TCP and UDP header. There is a Source Port number that is 16 bits and a Destination Port number that is 16 bits.

When a client sends an Ethernet packet, the Destination Port number in its message needs to correspond to a "listening" port number in the device that the message is being sent to.
It is typically applications that are running on a device that "listens" to a specific port number.

For example: If a device is running a Web Server application, it is most often listening on port 80. Port 80 is the normal or default port number for HTTP. If another PC opens up a Web browser (such as Internet Explorer) it will typically send a message with a destination port of 80.

The Source port number is typically less important as a specific value. Generally, the Client device will select a Source port number from a pool of numbers referred to as the "Ephemeral" port number range.

There are 3 types and/or ranges of Port numbers specified by the IANA (Internet Assigned Numbers Authority):

- The "Well Known Ports" = 0 – 1023
- The "Registered Ports" = 1024 – 49151
- The "Dynamic/Private Ports" (used as Ephemeral typically) = 49152 – 65535

The Well Known ports are the ports that are very often used such as HTTP: 80, FTP: 21, SMTP: 25, ModbusTCP: 502, etc… The port numbers in this range should only be used by an application if they are registered with the IANA.

The Registered ports can be used as listening ports by applications as well and are very often used for smaller applications and processes that don't feel the need to register with the IANA.

And, as mentioned above, the Dynamic ports should be used as the Ephemeral port for Clients sending TCP and UDP messages.

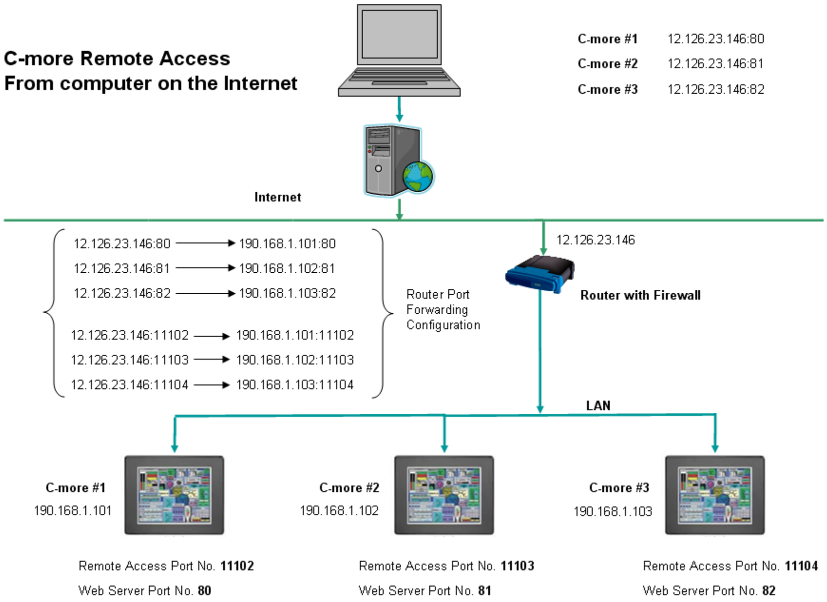
Port numbers are also used by the applications to 'bind' themselves to a socket, which is an internal software structure. This helps with maintaining robust communications amongst multiple devices.

## B. Port numbers are used for Firewall access and direction

Firewalls use the Destination port number of an Ethernet message targeted at the router to determine whether to accept or reject the message.

By default, most firewalls will have some of the Well Known ports enabled such as HTTP (80) and SMTP (25) but most anything else will have to be added to a table typically called a "Port Forwarding" table.

The Port Forwarding table not only tells the firewall which Port numbers to allow inside of the router, it also tells the router to which internal device (IP address) to direct the messages coming in with the specified port numbers. So in using this methodology, thousands of internal devices with private IP addresses can receive messages through only 1 public IP address

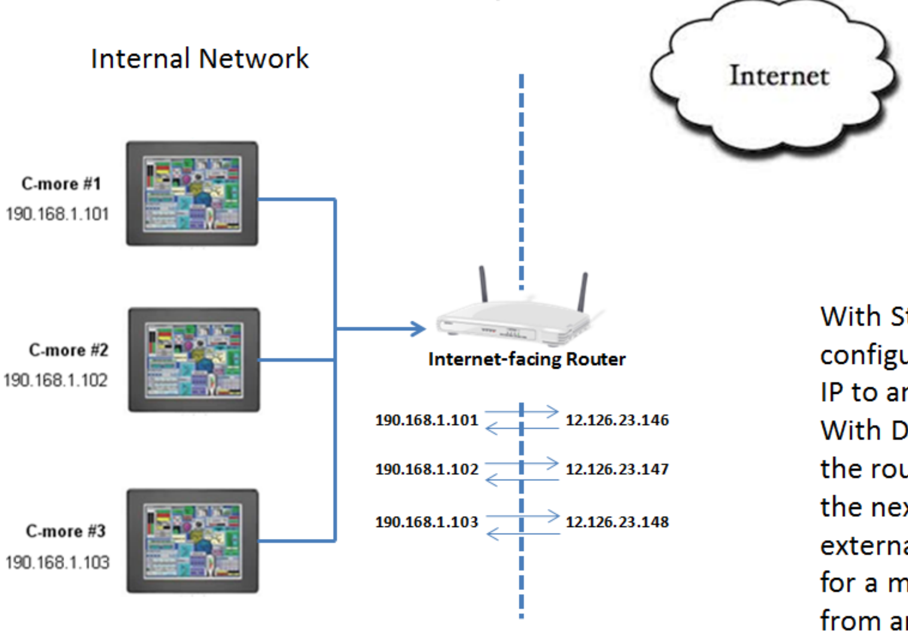


In the image above (taken from the C-more Remote Access Help file), the Router's Port Forwarding table has been setup to allow Ports 80 - 82 (for the C-more web server) and Ports 11102 – 11104 (for the C-more Remote Console Application). You can see that by using only 1 public IP address (12.126.23.146) but using multiple TCP Port numbers, we can access 3 different C-mores inside of the router.

## C. NAT (Network Address Translation) and PAT (Port Address Translation)

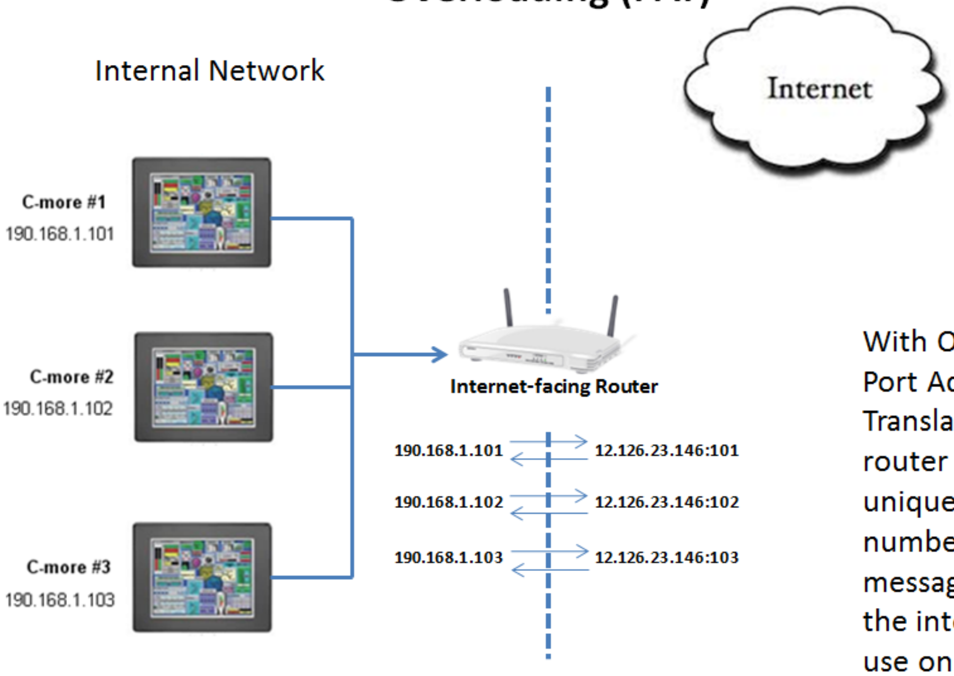
One of the principle benefits of using an Internet facing router is the ability to give large amounts of devices access to the Internet without requiring a unique IANA address. Usually each public IP address that you require has a monthly cost attributed to it.
NAT and PAT allow multiple internal, non-registered IP addresses to be 'translated' to a smaller amount (or 1) of registered, public IP address(es).

## Static/Dynamic NAT



With Static NAT, you configure an internal IP to an external IP. With Dynamic NAT, the router will grab the next available external IP address for a message sent from an Internal IP.

## Overloading (PAT)

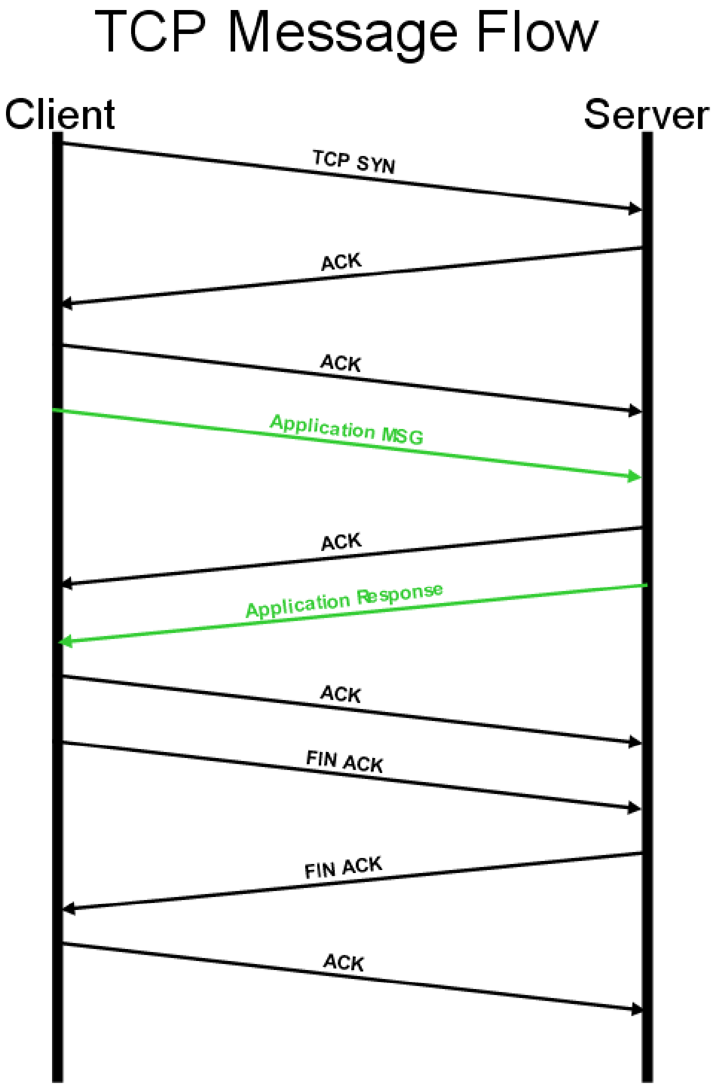


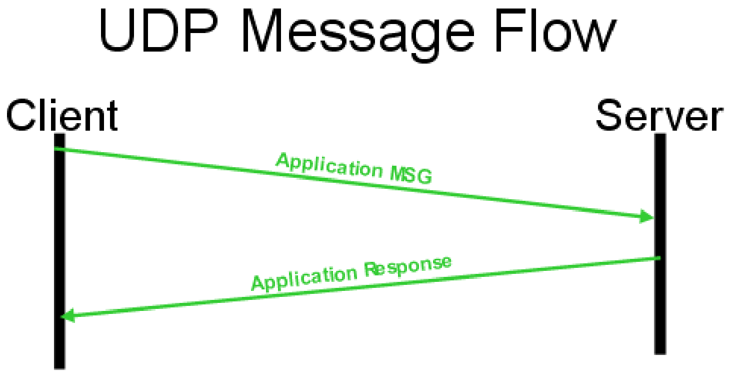
With Overloading or Port Address Translation (PAT), the router will tag a unique source port number for each message sent from the internal side but use only 1 external IP address.

## VI. TCP versus UDP

Throughout this document there is a lot of mention of UDP and TCP. This next section will explain the major differences between these 2 protocol layers.

The diagrams below illustrate the actual message flow in TCP and UDP for a Client application to send a message to a Server and for the Server application to send a response.





As you can see from the diagrams above, it takes many more Ethernet messages to send and receive an application request within the TCP transport mechanism than it does in the UDP transport mechanism.

With that being said, there are many Pros and Cons to using TCP versus using UDP.

The purpose of TCP is to guarantee reliability when sending a message from one device to another. TCP accomplishes this by first creating a ‘connection’ between a Client and Server. After the connection has been established, the Client and Server can send data back and forth and the TCP connection mechanism ‘guarantees’ that the data will get there by means of Acknowledgement messages and sequencing (to put the messages in the correct order). If a Client sends a message to a Server and does not receive an Acknowledgement (ACK) message from that device, it will then re-transmit the message.
The real beauty of this method is that the user application does not have to perform all of this complicated reliability routines, this is embedded in what is called the ‘stack code’ that typically comes with an Operating System. The user application only needs to specify which device it wishes to communicate with and send the message and everything else is handled underneath in the ‘stack’.

The downfall to this is there is a performance hit to the system in sending all these messages and performing all the calculations needed to maintain the reliability. It also adds significant network traffic.

Conversely, there is no ‘connection’ involved with UDP messaging. In UDP, the Client and Server simply send out the message to the device and hope that the message gets there. If reliability is desired, then the message flow timeout and retry mechanisms have to be written into the user’s application on both the Client side and Server side.

Because of this lack of connection handling, UDP performance is superior to TCP. There is much less traffic on the network and the ‘stack’ has to perform far fewer calculations.

Does this mean that UDP is better than TCP or vice versa? The answer is: it depends on the application.

If UDP were used to move large data files between one device to another over a busy network through many switches and routers, the complexity of the application to handle ‘missed’ packets and re-transmissions, getting the packets in the correct order, etc… would be very daunting. TCP would make much more sense for an application such as this.

On the other hand, if an application wanted to send streaming video or sound across this same network, UDP may make more sense. If a packet were lost in this situation, it would be better to keep going and ignore the lost packet since it would make little sense to try and re-introduce old sound or video later in the stream. This type of application would do better with faster throughput and less reliability.

Some example applications of these 2 protocols in our products are shown below:

UDP – Data View in DirectSoft to a Direct Logic PLC: 90% of the time, DirectSoft is just reading values, so if 1 packet is missed, the data is getting read so fast it will be almost imperceptible. DirectSoft uses a simple method of writing a packet and getting a reply, so if a write is missed, it will simply send again or give a warning. It is not trying to put together and re-assemble large files.

TCP – C-more Programming Software project transfer: The C-more Programming Software sends a project file to the panel using TCP. So in the case that 1 packet is missed it will be re-transmitted and the file is still re-assembled correctly.

## VII. DNS

Another address field that you will often see when setting up Ethernet devices is the DNS address. In most cases, you will see a “Preferred DNS server” or “Primary DNS server” and an “Alternate DNS server” or “Secondary DNS server”.

DNS stands for Domain Name System. It is the mechanism that allows applications to use a more-easily remembered name for a remote device versus a cryptic address or value. For example: when bringing up an Internet browsing software such as Internet Explorer, Chrome or Firefox, most of the time you do not type in an IP address in the URL field to access a remote web server. You would typically type in something like: www.google.com. However, this does not follow the rules of Ethernet so an IP address must be derived from this name.

When you enter in a Preferred DNS server address and a Secondary DNS server address, this gives a path for your PC to ‘find’ the IP address of the specified web site name. The Internet browser application will take the name entered in (www.google.com in this case) and send that name to the IP address specified in the Preferred DNS server field. The Preferred DNS server will then respond with the IP address of the name specified. Then the Internet browser application can send a request to connect to the web server specified. If the first attempt at sending this name to the Preferred DNS server fails, the Internet browser application will then attempt to send the name to the Alternate DNS server.

The DNS server itself has a complicated method of keeping up its table of IP addresses versus names.  Another benefit of this mechanism is that devices with Internet facing IP addresses may choose to use a pool of IP addresses and this can be ‘obscured’ by using a DNS lookup.

**Technical Assistance:** If you have questions regarding this Application Note, please contact us at 770-844-4200 for further assistance.