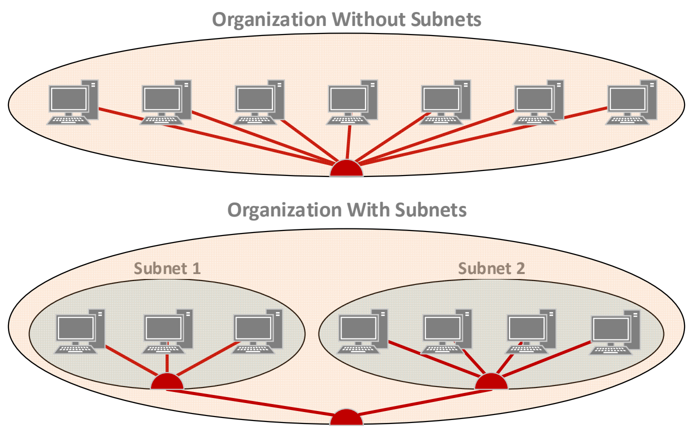
Organizations oftentimes subdivide their networks into smaller networks, and these smaller networks are termed *subnets*. By using subnets, organizations control the scope of broadcast traffic across their computers and devices, enhance the manageability of their network, and gain significant flexibility for their overall network design. This concept is illustrated in the figures below.



Notice that an organization without subnets connects all computers and devices directly to each other, while an organization with subnets divides the network traffic. Without the use of subnets, each organization has only one network with a potentially large number of hosts, resulting in unmanageable broadcast traffic. With subnets, organizations can segregate network traffic intelligently.

The conceptual distinction between a subnet and a network is based upon perspective. Network professionals, academics, and texts often associate the term “network” with a single overarching entity that belongs to an organization and represents the organization’s entire network, and they often associate the term “subnet” to refer to one specific sub-network within the organization’s overall network. However, this distinction is somewhat arbitrary. There is nothing to stop an organization from subdividing its overall network into multiple networks, and then assigning subnets to each of these networks. And there is also nothing to stop an Internet Service Provider (ISP) from doing the same across multiple organizations with any public IP addresses it has in its possession.

In this lab, you learn to understand subnets in more detail, and also learn how to perform many important subnet calculations used by networking professionals.

LAB OBJECTIVES

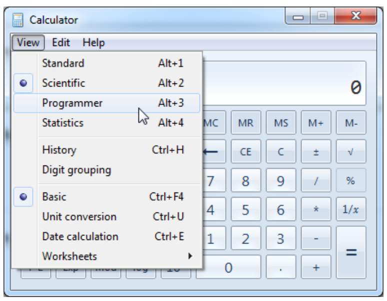
The objectives of this lab are:

- to understand the concepts of a subnet and subnet identifier.
- to learn to calculate IPv4 subnet directed broadcast addresses.
- to learn to calculate the number of assignable host addresses on a network.
- to learn to calculate network addresses for all subnets on a network.

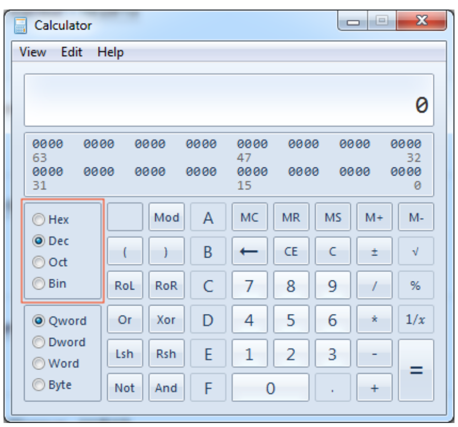
BINARY NUMBERS FROM THIS LAB FORWARD

In prior labs you have been required to manually convert each decimal number into a binary number and vice versa. It is important that you know how to perform these calculations by hand. However, when working in the real world and in academia, we often use tools, most commonly a calculator, to do many conversions for us. Think of the process you have gone through as akin to learning how to perform basic addition, subtraction, multiplication, and division by hand so that you understand the concepts and can do so when no tool is available, but after you understand how to do so you often use a calculator to save yourself some time.

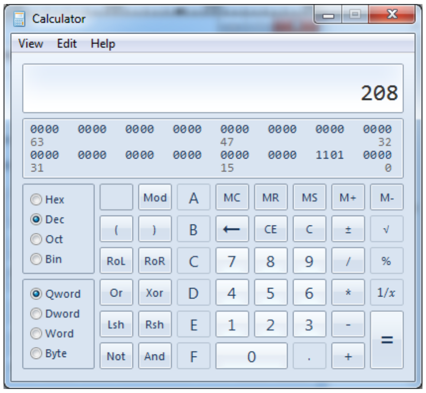
For example, in modern versions of Windows, one can switch the built-in calculator into “Programmer” mode like so.

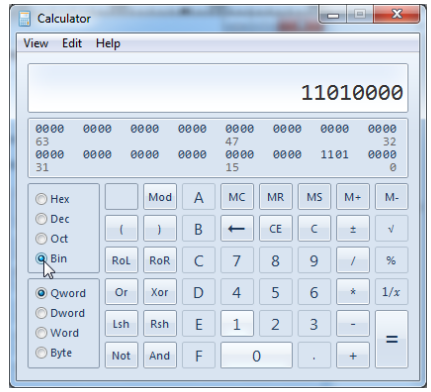


After doing so, you will notice that there are options to switch the calculator to a base other than base 10, including Hexadecimal (Hex), Octal (Oct), and Binary (Bin).

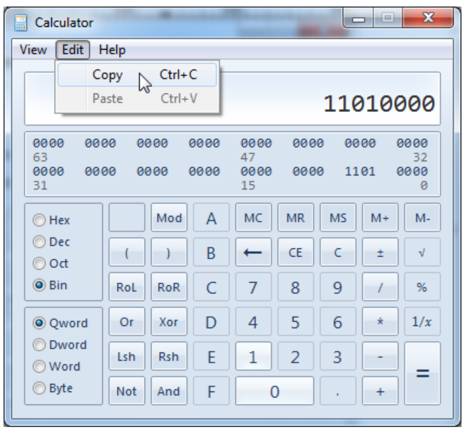


If you enter a number in one base, then switch to the other base, it will automatically convert the number for you. For example, to convert the first octet of the IPv4 address 208.64.121.161, we would enter 208 as decimal, then click “Bin”, to show its binary representation.





We can then simply use the Edit/Copy feature to paste it into our document.



Keep in mind while doing so that *the calculator removes any leading 0 bits*, so you will need to manually add leading 0 bits yourself. In other words, the calculator does not know we are specifically working with octets, and so it will not bother putting any leading 0 bits since leading 0 bits do not change the number's numeric value.

In this lab and in future labs, you are free to use a tool to perform these binary to decimal or decimal to binary conversions for you, unless you are specifically asked to perform the conversion by hand.

LAB SUBMISSION

Use the submission template provided in the assignment inbox to perform the steps requested by this lab. Return to the assignment inbox to submit your lab.

Section One – Subnet Calculations

STEPS

1. Now that we understand what a subnet is conceptually, we can learn to perform our first subnet calculation, which is determining the number of usable IPv4 addresses per subnet. Sometimes this calculation is described as determining the number of hosts per subnet. We learned that a subnet is a sub-network of an overall network; however, a subnet is still a network in its own right with all the same properties as a network. Hence a subnet has a network identifier and a host identifier which are conceptually the same as in any other network.

   At first glance, you might think that all we need to do is raise 2 to the power of the number of bits available in the host identifier to obtain the number of assignable IPv4 addresses. This is indeed the starting point, but we must also take into account that no host may be assigned the network address itself, and no host may be assigned the network broadcast address. We will learn more about network broadcast addresses in later steps, but for now it suffices you to know that all networks have a network address, and a broadcast address, and neither can be assigned to a computer/device. So to determine the number of allowable hosts in the network, we raise 2 to the power of the number of bits and subtract 2 from that number.

   Let us work through CIDR entry 96.45.82.5/20 as an example. We know that the network identifier spans 20 bits. That leaves 12 bits for the host identifier (32-20). To calculate the number of possible hosts for that network (96.45.80.0), we raise 2 to the power of 12, which comes to 4,096, then subtract 2 from that number, resulting in 4,094, to account for the network address and the broadcast address. So we conclude that the network represented by 96.45.82.5/20 has 4,094 usable IPv4 addresses. An alternative way to state this is that the network represented by 96.45.82.5/20 can have up to 4,094 hosts.

2. Now you give this calculation a try with the following subnet entry:

   66.147.242.171/15

   Using the methodology listed in step 1, determine the number of hosts (usable IPv4 addresses) for this subnet.

3. Another important subnet calculation is determining its broadcast address. While IPv4 packets are most usually addressed to exactly one recipient, there is sometimes the need to send a packet to all nodes on a network. When a computer sends an IP packet to one recipient, this is termed a unicast packet, and the recipient's IP address is termed a unicast address. When a computer sends an IP packet to all nodes on the

network, this is termed a broadcast packet, and it is addressed to the network's broadcast address.

A *subnet directed broadcast* is a broadcast addressed to all network adapters on a specific subnet, and is perhaps the most significant type of broadcast address. Almost all IPv4 communications are intended for a single recipient, but some protocols require the use of these broadcasts. For example, the Dynamic Host Configuration Protocol (DHCP) is used to dynamically assign an IPv4 address and a subnet mask to a network adapter. Before the network adapter has been assigned its IPv4 address, it is not possible for the DHCP server to send a unicast packet to the network adapter, so it must broadcast the packet to all nodes on the subnet, relying on the fact that the network adapter in need of the IPv4 address will receive the message.

A subnet directed broadcast address is always the *highest* address in the subnet. To compute the highest address in a subnet, we extract the network identifier from the IPv4 address, then append 1 bits to it until we fill out the full 32-bits. In other words, a broadcast address has a network identifier which uniquely identifies the appropriate subnet, and a host identifier consisting of all 1 bits to indicate that the address is not intended for a single recipient, but for all recipients on that subnet.

For example, if we are given a network address of 33.19.128.0 and are told that the network identifier spans 17 bits, we can calculate that network's broadcast address by filling the host identifier with all 1 bits. The network address for 33.19.128.0 in binary is:

00100001 00010011 10000000 00000000

If we fill the host identifier with all 1 bits, we arrive at a broadcast address of:

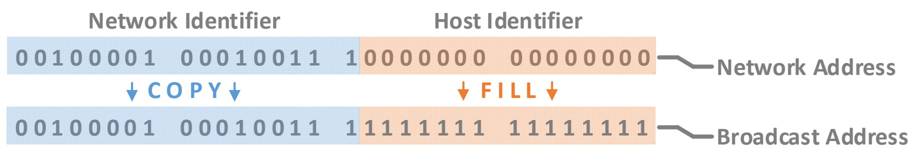
00100001 00010011 11111111 11111111

This address is the highest possible address for that network. We can represent the broadcast address in dotted decimal notation as well:

33.19.255.255

This example is illustrated in the following figure.

Determining the Broadcast Address for Network 33.19.128.0/17



Let us work through more complete example, the CIDR entry 96.45.82.5/20, for which we are not given the network address. To begin, we first need determine the network address. We know that 20 bits fully spans the first two octets and does not span the last octet at all, so the network identifier is 96.45.X.0. We then convert the value in the third octet, 82 in decimal, to binary, which is 01010010. The /20 indicates that 4 bits are spanned in this third octet, so extracting the first 4 bits and zeroing out the remainder leaves us with 01010000, which is 80 in decimal. So we now know that the network address for 96.45.82.5/20 is 96.45.80.0.

To calculate the broadcast address in binary, all we now need to do is fill the host bits with 1 bits. If the network identifier spans 20 bits, how many bits does the host identifier span? The answer is 12, calculated as, 32 - 20 = 12. The difference is that we start counting the bits for a network identifier starting from the left, and we start counting bits for the host identifier from the right.

We can use two shortcuts when calculating the broadcast address. Namely, if the host identifier fully spans an octet, it will consist of all 1s, and the octet is represented with 255 in decimal. If the host identifier does not span an octet at all, then the octet is simply copied from the network address. The only octet that needs a calculation is one that is partially spanned by the network identifier and by the host identifier.

Thus far we have calculated the network address as 96.45.80.0. If we want to now calculate that network's broadcast address, we see that the rightmost octet is fully spanned by the host identifier, and the third octet is partially spanned, so we end up with 96.45.X.255. Now we just need to calculate X. Recall that 80 is 01010000 in binary, and that 0101 is the portion spanned by the network identifier. Thus, we simply fill the last 4 bits in the octet with 1 bits, with a result of 01011111, which is 95 in decimal. Thus, the broadcast address is 96.45.95.255, and this is the highest possible address in the 96.45.80.0 network.

In summary, we calculate the network address in binary, set the host identifier bits to all 1s, then convert the result back to dotted decimal notation. We can use the shortcuts that any octet fully spanned by the host identifier is 255 in decimal, and that any octets not spanned at all by the host identifier are simply copied from the network address.

4. Now you give it a try by calculating the broadcast address for the following CIDR entry using the methodology listed in step 3. Represent your answers in dotted decimal notation, and make sure to show your work.

   64.152.0.21/25

5. Determining a subnet's assignable IPv4 address range is the next calculation we explore. Not only do we want to know the number of possible IPv4 addresses for a given subnet, as well as its broadcast address, but we also want its range of assignable IPv4 addresses. Once we know the broadcast and network address for a subnet, determining this range is straightforward. We start with the address one higher than the network address, and end with the address one lower than the broadcast address.

   Continuing with the example entry 96.45.82.5/20 from step 3, we know that the network address is 96.45.80.0. One address higher is 96.45.80.1, so that is the bottom of the range. We know that the broadcast address is 96.45.95.255, and the next lower address is 96.45.95.254, which is the top of the range. So the range of assignable addresses for 96.45.82.5/20 is 96.45.80.1 through 96.45.95.254.

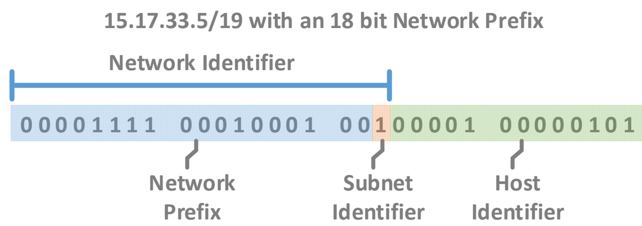
6. Now, you give this a try with the CIDR entry below. Give the range of assignable IPv4 addresses for that subnet, using the methodology listed in step 5. Make sure to show your work.

   66.147.242.171/15

7. Imagine that an organization is assigned the block of addresses represented by the network identifier 193.39.64.0/18, that is, all IP addresses beginning with 11000001 00100111 01. This assignment does not mean that the organization must use this network prefix atomically to define only one single network. If the organization were to do so, it would have one single network consisting of 2^14 -2, or 16,382, hosts. However, it is uncommon to have a single network consisting of such a large number of hosts due to the inefficiencies and management complexity that would create. Instead, the organization would subdivide its networks into smaller networks, each consisting of a smaller number of hosts. These smaller networks are subnets. We now begin to discern the meaningful distinction between a network and a subnet.

   From a technical perspective, a subnet is an IPv4 network whose network identifier consists of a network prefix and a subnet identifier. The network prefix is used to identify the network on which the subnet resides, and the subnet identifier is used to uniquely identify the subnet in the context of that network.

For example, imagine that there is a subnet which has a network identifier of 15.17.33.5/19, or 00001111 00010001 001 in binary. Further imagine that 18 bits are assigned as the network prefix, and 1 bit is assigned as the subnet identifier. That would mean that the network prefix is 00001111 00010001 00 and the subnet identifier is simply 1. This example is illustrated in the following figure.



Notice that the network identifier of the subnet spans 19 bits, and it is composed of the 18-bit network prefix and the 1-bit subnet identifier. In this example, we specify that the network prefix spans 18 bits. This does not mean that all subnet's network identifiers are required to use an 18-bit network prefix, but only means that in this specific example the network prefix spans 18 bits. Typically, an organization is assigned a public IP address block from an ISP, which mandates a certain length network prefix, or in the case of an internal (private) subnets, the organization chooses one of the legal private IP address blocks which also mandate a certain length network prefix.

We just reviewed many new terms and concepts with this example, and for absolute clarity, let us review them again. A subnet is literally a network in its own right, which means it has its own network identifier. What makes the network a subnet rather than a straight network is the fact that its network identifier has a network prefix common across all subnets in the same network, along with a subnet identifier which uniquely identifies that subnet in the context of that network. Conceptually, a subnet is a smaller network on an organization's larger network. So in the aforementioned example, the subnet's network identifier of 00001111 00010001 001 has a network prefix of 00001111 00010001 00 (the first 18 bits) and a subnet identifier of 1 (the 19th bit). This means that the subnet is a sub-network in the overarching 00001111 00010001 00 network.

It is important to understand that while routers using specific routing protocols can be configured with subnet information, the other computers and devices that participate in the network have no knowledge of a network prefix and a subnet identifier, and are only aware of a network identifier and host identifier. To state this another way, these computers and devices have nothing in their configuration to subdivide the network identifier. They always use the full network identifier as an atomic number, and do not have a finer-grained distinction beyond that of a network.

If these network adapters are not configured to use subnet identifiers, how are they used? Simple! Subnet identifiers are used when assigning network addresses to subnets. Hence, these network adapters are unaware of subnet identifiers during operation, but network designers use them to assign network addresses to subnets. Within the organization, the network prefix will be the same across all subnets, so the subnets are logically grouped as belonging to the same network. Now you understand that any network whose identifier can be separated into a network prefix and a subnet identifier is a subnet, and conceptually we view a subnet as a sub-network that exists within the context of an overarching network.

8. Imagine that a network adapter on a subnet is represented with the CIDR entry 103.147.213.34/17, and that its subnet identifier spans 2 bits. Represent its network prefix and subnet identifier in binary using the methodology listed in step 7. Recall that from the perspective of a subnet, its network prefix and subnet identifier are both contained within its network identifier.

9. We have now come upon a new concept, which is that of *the number of subnets per network.* Continuing with the example in step 7, which is an organization that is given the block of addresses beginning with 193.39.64.0/18, suppose the organization decides that it wants to divide its overall network into 8 smaller networks (subnets). How many bits would it need to do this? 3, because 2^3 is 8! Thus, the organization would use 11000001 00100111 01 as its network prefix, then the next three bits to uniquely identify each subnet. The subnet identifiers would be 000, 001, 010, 011, 100, 101, 110, and 111. And the overall network identifiers for these subnets would be:

```
First subnet:  11000001 00100111 01000
Second subnet: 11000001 00100111 01001
Third subnet:  11000001 00100111 01010
Fourth subnet: 11000001 00100111 01011
Fifth subnet:  11000001 00100111 01100
Sixth subnet:  11000001 00100111 01101
Seventh subnet:11000001 00100111 01110
Eighth subnet: 11000001 00100111 01111
```

Did you catch all this? The organization is assigned a network prefix. Rather than using this prefix to create a single, large network, the organization subdivides that network into 8 subnets, and those subnets are identified by an additional 3 bits appended to the end of the network prefix. Thus, *from the perspective of the organization, the network identifier consists of 18 bits, but from the perspective of each subnet, the network identifier consists of 21 bits*. Think about the last statement carefully until you are sure you understand it. Understanding this perspective is crucial understanding the content in the remainder of this lab.

10. Imagine that an organization is given a block of addresses represented with 139.93.40.0/22, and that it intends to use 4 subnets on its network. Give the number of bits needed for the subnet identifier for 4 subnets, and list the network identifiers for all four subnets, in binary, using the methodology listed in step 9.

11. Wow! Up to now you have learned a lot about working with IPv4 address. You have learned how to represent them in dotted decimal notation and in binary, how to specify the length of the network identifier in CIDR notation, how to work with subnet masks, how to calculate their network and broadcast addresses, and how to determine the number and range of assignable addresses on a network. You have also learned how to subdivide a network into subnets and calculate the corresponding subnet identifiers. In doing so, you have learned many of the addressing concepts integral to the operation of local area and backbone networks throughout the world. Congratulations!

    We can now practice with much of what we learned up to this point all together.

    If an organization is given the range of addresses corresponding to the CIDR entry 209.15.0.0/17, and the organization decides to use 2 subnets, we can answer the following series of questions.

    a. How many bits are needed for the subnet identifier, to distinguish one subnet from another on this network?
    Because there are 2 subnets, we would use 1 bit.

    b. What is the subnet mask for the 2 subnets in binary and dotted decimal?
    Since the network identifier for the organization is 17 bits, and the subnet identifier is 1 bit, the length of the network identifiers for both of the subnets is 18 bits. So we add 18 1 bits to the mask, followed by 14 0 bits, like so, to arrive at our subnet mask:

    11111111 11111111 11000000 00000000 or 255.255.192.0 in dotted decimal

    c. How many hosts are there per subnet?
    Because the network identifiers are 18 bits, that leaves 14 bits for the hosts. 2^14 = 16,384, and subtracting 2 from that gives us 16,382 possible hosts.

    d. What are the network identifiers for each subnet, in binary?
    First, we need to represent the network identifier for the organization, in binary, which is 11010001 00001111 0. We obtained this by converting the first three octets to binary, and only selecting the first bit from the third octet. Next, we append the two combinations of bits together to calculate the network identifier for each subnet.

    `First subnet:  11010001 00001111 00`

`Second subnet: 11010001 00001111 01`

e. What are the network addresses for each subnet in dotted decimal notation?
We have already represented the network prefixes for each subnet in binary, so we simply convert them to decimal and fill out the remaining bits with 0 bits.

First subnet: 209.15.0.0
Second subnet: 209.15.64.0

f. What are the broadcast addresses for each subnet in dotted decimal notation?
For the first subnet, we know that its network address is 209.15.0.0. When we set all of the 14 host bits to 1, we know that we have 209.15.X.255 because the third octet is partially spanned. To determine X, we list out the third octet with the host bits set to 1, which comes to 00111111. When we convert that to decimal, we see that it is 63. Thus the broadcast address for the first subnet is 209.15.63.255. We can sanity check this because the next address is 209.15.64.0, which starts the next subnet.

For the second subnet, we follow the same steps to arrive at 209.15.X.255. When we list out the third octet, with the host bits set to 1, we arrive at 011111111, which is 127 in decimal.

Notice that if we perform this conversion by hand, it would look as follows.

01111111 ➔ 127

Power of 2		128	64	32	16	8	4	2	1
Bit		0	1	1	1	1	1	1	1
Cumulative Amount	0	0	64	96	112	120	124	126	127

By hand we could have taken a shortcut by simply subtracting the highest bit value, 128, from 255 since the first bit is a 0 and the remaining bits are all 1s. That is, if we see an octet that has a single 0 bit coupled with seven 1 bits, we simply subtract that bit's value from 255 to convert it to decimal. Sometimes starting from 0, then adding each 1 bit's value is faster, and sometimes starting from 255, and subtracting each 0 bit's value, is faster.

So by substituting 127 for X, we arrive at a broadcast address of 209.15.127.255 for the second subnet.

g. What are the assignable IP addresses for each subnet?
First subnet: The first address that can be assigned is 1 higher than the network address. Since the network address is 209.15.0.0, the first assignable IP address is 209.15.0.1. The last address that can be assigned is 1 lower than the broadcast address. Since the broadcast address is 209.15.63.255, the highest assignable address

is 209.15.63.254. Putting these two together, we can say that all addresses between 209.15.0.1 and 209.15.63.254 are assignable.

Second subnet: Since the network address is 209.15.64.0, the first assignable address is 209.15.64.1. Since the broadcast address is 209.15.127.255, the last assignable address is 209.15.127.254. We can thus say that all addresses between 209.15.64.1 and 209.15.127.254 are assignable.

12. What will we do now? You guessed it! Now it's your turn to try it out.

    Imagine that an organization is assigned the range of addresses identified by CIDR entry 66.216.112.0/21, and that the organization has decided to use 8 subnets. Answer the following series of questions about this scenario. Make sure to show your work, using a tool to convert from binary to decimal and vice versa when necessary.

    a. How many bits are needed for the subnet identifier, to distinguish one subnet from another on this network?
    b. What is the subnet mask for the 8 subnets in binary and in dotted decimal?
    c. How many hosts are there per subnet?
    d. What are the network identifiers for each subnet, in binary?
    e. What are the network addresses for each subnet in dotted decimal notation?
    f. What are the broadcast addresses for each subnet in dotted decimal notation?
    g. What are the assignable IP addresses for each subnet?

13. In your own words, explain what a subnet is from a conceptual and a technical perspective, and also explain the advantages of using subnets versus using a single, monolithic computer network.

You are now familiar with many of the addressing concepts integral to the operation of local area and backbone networks throughout the world. Congratulations! This is significant milestone in your quest to understand and intelligently work with modern computer networks.