

Athit Vue

Date: May 6, 2017

This lab was pretty interesting. It took a lot of research but learning about it was pretty interesting. There were many things that was new to me, thus it was pretty difficult to began the lab. So after a few long hours of research I finally had enough information to get started.

Menu of program:

```
Terminal - tee@TEE-DESKTOP: ~/344/lab13_14
File Edit View Terminal Tabs Help
tee@TEE-DESKTOP:~/344/lab13_14$ main.py
Please enter the number of the job you would like to perform:
1. For a maximum number of hosts on a subnet, determine the mask needed.
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host IP address.
5. Given a host IP and mask, determine the valid host IP address range.
6. Given a subnet IP and mask, determine if it is a valid subnet IP.
7. Exit.
Response: 1
10001100.10110011.11011100.11001000 140.179.220.200 IP IP
11111111.11111111.11100000.00000000 255.255.224.000 Subnet Mask
-----
10001100.10110011.11000000.00000000 140.179.192.000 Subnet IP
10001100.10110011.11011111.11111111 140.179.223.255 Broadcast IP

In this example there are 524,288 ( $2^{19}$ ) subnet IP's available with this size mask. Each sub
( $2^{13} - 2$ ) hosts. Each subnet can have hosts assigned to any address between the Subnet IP
Broadcast IP address.

To calculate the number of subnets or hosts, we used use the formula ( $2^n - 2$ ) where n = nu
host field, and  $2^n$  represents 2 raised to the  $n^{th}$  power. Note that although subnet masks wit
mask bits are allowed; they are not recommended.

Deliverables: Write a Ruby, Bash, or Python script that accomplishes the following 6 type
class/Lab we will do many examples. The best strategy is to first get a good understanding
works, determine a pattern, and finally translate the design into scripting code. This lab wi
performing the following:

1. If asked for a maximum number of hosts required on a subnet, determine the mask need
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host
5. Given a host IP and mask, determine the valid host IP address
6. Given a subnet IP and a mask, determine if it is a valid subne
is not valid since its host IP address is not all 0's.

As mentioned in previous work, either bash or ruby scripting ca
less familiar in order to gain your knowledge plus be better pre
```

First job with results:

```
Terminal - tee@TEE-DESKTOP: ~/344/lab13_14
File Edit View Terminal Tabs Help
tee@TEE-DESKTOP:~/344/lab13_14$ main.py
Please enter the number of the job you would like to perform:
1. For a maximum number of hosts on a subnet, determine the mask needed.
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host IP address.
5. Given a host IP and mask, determine the valid host IP address range.
6. Given a subnet IP and mask, determine if it is a valid subnet IP.
7. Exit.
Response: 1
Please enter the maximum number of hosts or quit(q): 6
Your network mask should be set to: 255.255.255.248
tee@TEE-DESKTOP:~/344/lab13_14$
```

Host Ip and subnet mask, output subnet IP:

```
Terminal - tee@TEE-DESKTOP: ~/344/lab13_14
File Edit View Terminal Tabs Help
tee@TEE-DESKTOP:~/344/lab13_14$ main.py
Please enter the number of the job you would like to perform:
1. For a maximum number of hosts on a subnet, determine the mask needed.
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host IP address.
5. Given a host IP and mask, determine the valid host IP address range.
6. Given a subnet IP and mask, determine if it is a valid subnet IP.
7. Exit.
Response: 1
Please enter the maximum number of hosts or quit(q): 6
Your network mask should be set to: 255.255.255.248
tee@TEE-DESKTOP:~/344/lab13_14$ main.py
Please enter the number of the job you would like to perform:
1. For a maximum number of hosts on a subnet, determine the mask needed.
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host IP address.
5. Given a host IP and mask, determine the valid host IP address range.
6. Given a subnet IP and mask, determine if it is a valid subnet IP.
7. Exit.
Response: 2
Please enter a host IP or quit(q): 140.179.220.200
Please enter a subnet mask or quit(q): 255.255.224.000
Your subnet IP should be: 140.179.192.0
tee@TEE-DESKTOP:~/344/lab13_14$
```

address and the subnet mask, then set
Address can be calculated by first tog
OR operation between the IP Address

In this example there are 524,288 (2^{19}
($2^{13} - 2$) hosts. Each subnet can have 1
Broadcast IP address.

To calculate the number of subnets or
host field, and 2^n represents 2 raised to
mask bits are allowed; they are not re

Deliverables: Write a Ruby, Bash, or
class/Lab we will do many examples.
works, determine a pattern, and finally
performing the following:

1. If asked for a maximum number of

Getting the broadcast ip given subnet ip and subnet mask:

```
Terminal - tee@TEE-DESKTOP: ~/344/lab13_14
File Edit View Terminal Tabs Help
tee@TEE-DESKTOP:~/344/lab13_14$ main.py
Please enter the number of the job you would like to perform:
1. For a maximum number of hosts on a subnet, determine the mask needed.
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host IP address.
5. Given a host IP and mask, determine the valid host IP address range.
6. Given a subnet IP and mask, determine if it is a valid subnet IP.
7. Exit.
Response: 3
Please enter a subnet IP or quit(q): 140.179.192.000
Please enter a subnet mask or quit(q): 255.255.224.000
Your broadcast ip should be: 140.179.223.255
tee@TEE-DESKTOP:~/344/lab13_14$
```

lab13_14 - File Manager

DEVICES

- File System
- 91 GB Volume
- VM

PLACES

Advanced Mode

2⁷

2⁶

256-224

32+256

10001100.10110011.11011100.11011111.11111111.11111111.11100000.00000000

10001100.10110011.11000000.00000000

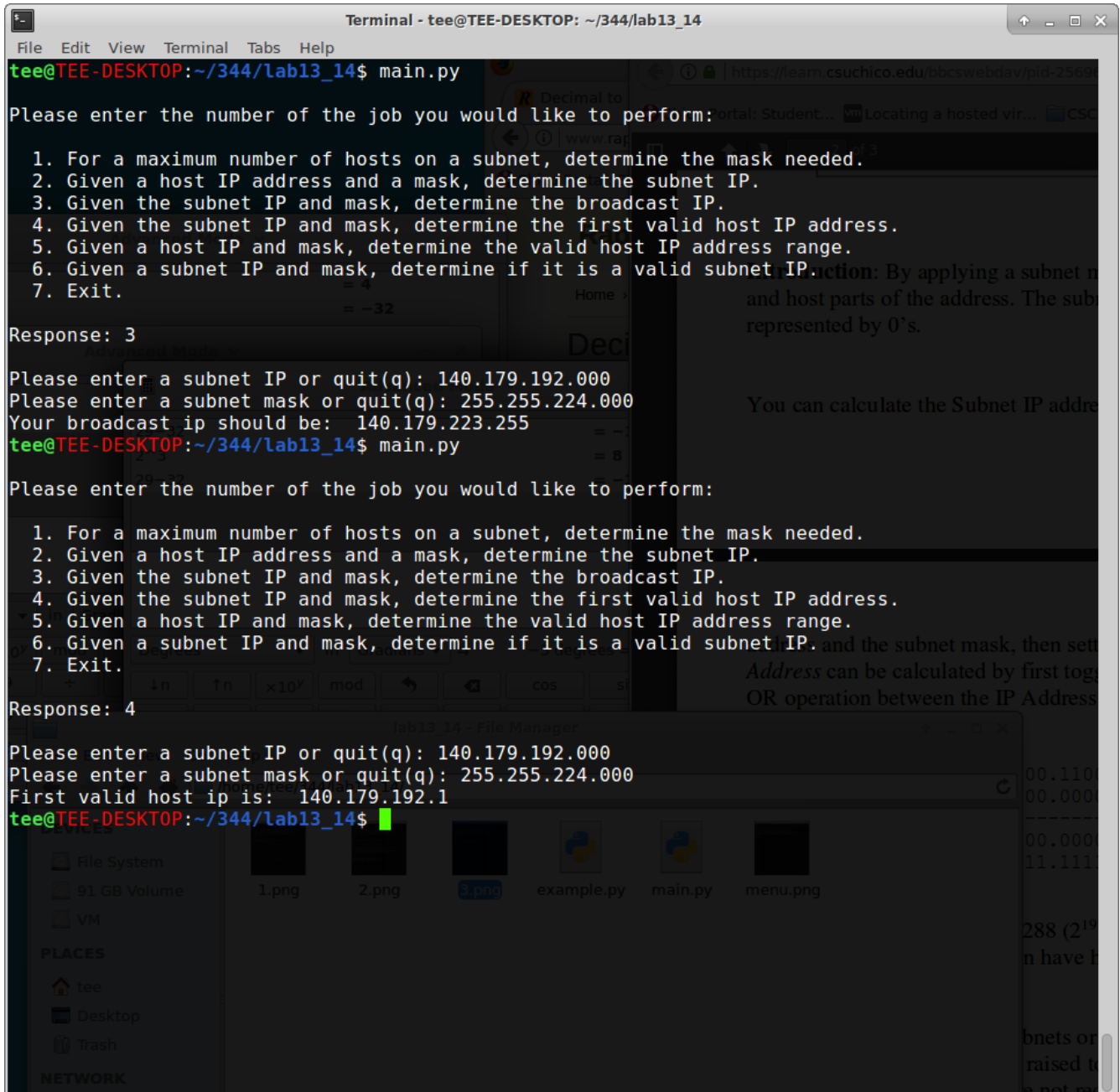
10001100.10110011.11011111.11111111

In this example there are 524,288 (2¹⁹ - 2) hosts. Each subnet can have 1 Broadcast IP address.

To calculate the number of subnets or host field, and 2ⁿ represents 2 raised to the power of n. mask bits are allowed: they are not reserved.

First valid host IP address:

```
Terminal - tee@TEE-DESKTOP: ~/344/lab13_14
File Edit View Terminal Tabs Help
tee@TEE-DESKTOP:~/344/lab13_14$ main.py
Please enter the number of the job you would like to perform:
1. For a maximum number of hosts on a subnet, determine the mask needed.
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host IP address.
5. Given a host IP and mask, determine the valid host IP address range.
6. Given a subnet IP and mask, determine if it is a valid subnet IP.
7. Exit.
Response: 3
Please enter a subnet IP or quit(q): 140.179.192.000
Please enter a subnet mask or quit(q): 255.255.224.000
Your broadcast ip should be: 140.179.223.255
tee@TEE-DESKTOP:~/344/lab13_14$ main.py
Please enter the number of the job you would like to perform:
1. For a maximum number of hosts on a subnet, determine the mask needed.
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host IP address.
5. Given a host IP and mask, determine the valid host IP address range.
6. Given a subnet IP and mask, determine if it is a valid subnet IP.
7. Exit.
Response: 4
Please enter a subnet IP or quit(q): 140.179.192.000
Please enter a subnet mask or quit(q): 255.255.224.000
First valid host ip is: 140.179.192.1
tee@TEE-DESKTOP:~/344/lab13_14$
```



Determine valid host ip range given host ip and mask:

```
Terminal - tee@TEE-DESKTOP: ~/344/lab13_14
File Edit View Terminal Tabs Help
tee@TEE-DESKTOP:~/344/lab13_14$ main.py
Please enter the number of the job you would like to perform:
1. For a maximum number of hosts on a subnet, determine the mask needed.
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host IP address.
5. Given a host IP and mask, determine the valid host IP address range.
6. Given a subnet IP and mask, determine if it is a valid subnet IP.
7. Exit.
Response: 5
Please enter a host IP or quit(q): 140.179.220.200
Please enter a subnet mask or quit(q): 255.255.224.000
Range for valid host IP should be: 140.179.192.1 - 140.179.224.254
tee@TEE-DESKTOP:~/344/lab13_14$ vim main.py
tee@TEE-DESKTOP:~/344/lab13_14$
```

lab13_14 - File Manager

File Edit View Go Help

/home/tee/344/lab13_14/

DEVICES

- File System
- 91 GB Volume
- VM

PLACES

- tee
- Desktop
- Trash

NETWORK

1.png 2.png 3.png 4.png example.py main.py menu.png

Determine if subnet ip is valid (Throws an error to screen if not valid):

```
Terminal - tee@TEE-DESKTOP: ~/344/lab13_14
File Edit View Terminal Tabs Help
tee@TEE-DESKTOP:~/344/lab13_14$ main.py
Please enter the number of the job you would like to perform:
1. For a maximum number of hosts on a subnet, determine the mask needed.
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host IP address.
5. Given a host IP and mask, determine the valid host IP address range.
6. Given a subnet IP and mask, determine if it is a valid subnet IP.
7. Exit.
Response: 5
Please enter a host IP or quit(q): 140.179.220.200
Please enter a subnet mask or quit(q): 255.255.224.000
Range for valid host IP should be: 140.179.192.1 - 140.179.224.254
tee@TEE-DESKTOP:~/344/lab13_14$ vim main.py
tee@TEE-DESKTOP:~/344/lab13_14$ main.py
Please enter the number of the job you would like to perform:
1. For a maximum number of hosts on a subnet, determine the mask needed.
2. Given a host IP address and a mask, determine the subnet IP.
3. Given the subnet IP and mask, determine the broadcast IP.
4. Given the subnet IP and mask, determine the first valid host IP address.
5. Given a host IP and mask, determine the valid host IP address range.
6. Given a subnet IP and mask, determine if it is a valid subnet IP.
7. Exit.
Response: 6
Please enter a subnet IP or quit(q): 140.179.192.000
Please enter a subnet mask or quit(q): 255.255.224.000
Traceback (most recent call last):
  File "./main.py", line 126, in <module>
    status=ipaddress.ip_address(subnet_ip_string+"/"+subnet_mask_string)
  File "/usr/lib/python3.5/ipaddress.py", line 54, in ip_address
    address)
ValueError: '140.179.192.000/255.255.224.000' does not appear to be an IPv4 or IPv6 address
tee@TEE-DESKTOP:~/344/lab13_14$
```

```
Terminal - tee@TEE-DESKTOP: ~/344/lab13_14
1 #!/usr/bin/python3.5
2
3 import math
4 import sys
5 import ipaddress
6
7 response = input("\nPlease enter the number of the job you would like to perform:\n\n 1. For a maximum number of hosts o
n a subnet, determine the mask needed.\n 2. Given a host IP address and a mask, determine the subnet IP.\n 3. Given the
subnet IP and mask, determine the broadcast IP.\n 4. Given the subnet IP and mask, determine the first valid host IP ad
dress.\n 5. Given a host IP and mask, determine the valid host IP address range.\n 6. Given a subnet IP and mask, deter
mine if it is a valid subnet IP.\n 7. Exit.\n\nResponse: ")
8
9 # If user wants to determine mask needed
10 if response == "1":
11     max_host_string = input("\nPlease enter the maximum number of hosts or quit(q): ")
12     if max_host_string != "q":
13         max_hosts = int(max_host_string)
14         # 1. Take the max hosts and convert it to the nth power with base 2.
15         # Must account for network ip and broadcast ip so add 2 to max_hosts.
16         nth_power = math.log2(max_hosts+2)
17         n = math.ceil(nth_power)
18         # 2. Subtract that number from total CIDR (32..32-bits)
19         CIDR = 32 - n
20         # print(CIDR) **keep for debugging
21         # 3. Need a way to store and calculate subnet mask...use 4-dimension array
22         subnet_mask = [0, 0, 0, 0]
23         for i in range(CIDR):
24             subnet_mask[int(i/8)] = subnet_mask[int(i/8)] + (1<< (7 - i % 8))
25         # 4. Calculation is down print network mask, need delimiter "." and join()
26         print("Your network mask should be set to: ", ".".join(map(str, subnet_mask)), "\n")
27     else:
28         print("\nUser quit! Goodbye!\n")
29         exit(0)
30 # If user wants subnet IP
31 elif response == "2":
32     host_ip_string = input("\nPlease enter a host IP or quit(q): ")
33     if host_ip_string == "q":
34         print("User quit! Goodbye!\n")
35         exit(0)
36     subnet_mask_string = input("Please enter a subnet mask or quit(q): ")
37     if subnet_mask_string != "q":
38         host_ip = host_ip_string.split(".")
39         subnet_mask = subnet_mask_string.split(".")
40         subnet_ip = []
41         for i in range(4):
42             subnet_ip.append(int(host_ip[i]) & int(subnet_mask[i]))
43         print("Your subnet IP should be: ", ".".join(map(str, subnet_ip)))
44     else:
45         print("User quit! Goodbye!\n")
46         exit(0)
47
48 :set number
```

Line 7 is where I do all the asking user for input. I've decided to make a menu for the operations that users may want to render. On line 10 – 29 is where I do the determination of mask needed based on a subset. I use CIDR and some logical operations to get through, on lines 31 – 47 is where I execute the functionality for finding the broadcast IP


```
Terminal - tee@TEE-DESKTOP: ~/344/lab13_14
File Edit View Terminal Tabs Help
48 elif response == "3":
49     subnet_ip_string = input("\nPlease enter a subnet IP or quit(q): ")
50     if subnet_ip_string == "q":
51         print("User quit! Goodbye!\n")
52         exit(0)
53     subnet_mask_string = input("Please enter a subnet mask or quit(q): ")
54     if subnet_mask_string != "q":
55         ip = ipaddress.ip_network(subnet_ip_string+"/"+subnet_mask_string)
56         print("Your broadcast ip should be: ", ip.broadcast_address)
57     else:
58         print("User quit! Goodbye!\n")
59         exit(0)
60 # If user wants to determine the first valid host ip address:
61 elif response == "4":
62     subnet_ip_string = input("\nPlease enter a subnet IP or quit(q): ")
63     if subnet_ip_string == "q":
64         print("User quit! Goodbye!\n")
65         exit(0)
66     subnet_mask_string = input("Please enter a subnet mask or quit(q): ")
67     if subnet_mask_string != "q":
68         subnet_ip = subnet_ip_string.split(".")
69         subnet_mask = subnet_mask_string.split(".")
70         count=1
71         host_first = []
72         for i in range(4):
73             if count != 4 or subnet_mask[i] == "255":
74                 count+=1
75                 host_first.append(subnet_ip[i])
76             else:
77                 if count == 4:
78                     host_first.append(int(subnet_ip[i])+1)
79                     count+=1
80                 print("First valid host ip is: ", ".".join(map(str, host_first)))
81             else:
82                 print("User quit! Goodbye!\n")
83                 exit(0)
84 elif response == "5":
85     host_ip_string = input("\nPlease enter a host IP or quit(q): ")
86     if host_ip_string == "q":
87         print("User quit! Goodbye!\n")
88         exit(0)
89     subnet_mask_string = input("Please enter a subnet mask or quit(q): ")
90     if subnet_mask_string != "q":
91         host_ip = host_ip_string.split(".")
92         subnet_mask = subnet_mask_string.split(".")
93         subnet_ip = []
94         host_first = []
95         host_last = []
96         count=1
97         for i in range(4):
```

Introduction: By applying a subnet mask to an IP address it allows the identification of the network (subnet) represented by the 1's in the mask, and the host bits are represented by 0's.

You can calculate the Subnet IP address by performing a bitwise logical AND operation between the IP address and the subnet mask, then setting all the host bits to 0's. What about the broadcast IP? The Broadcast IP address can be calculated by first toggling all the bits for the Mask. Let's call it NotMask. Then performing an AND operation between the IP address and NotMask.

| IP | Mask | Subnet IP | Broadcast IP |
|-------------|-----------------|--------------|--------------|
| 00.11001000 | 140.179.220.200 | IP | IP |
| 00.00000000 | 255.255.224.000 | Subnet Mask | |
| 00.00000000 | 140.179.192.000 | Subnet IP | |
| 00.11111111 | 140.179.223.255 | Broadcast IP | |

288 (2¹⁹) subnet IP's available with this size mask. Each subnet has 8190 hosts assigned to any address between the Subnet IP address and the broadcast IP address.

For subnets or hosts, we used the formula (2ⁿ - 2) where n = number of bits in the subnet mask. Note that although subnet masks with non-contiguous 1's are not recommended.

97,1 55%

And the rest just follow suits. On the execution for number 4 was quite confusing at first. However I used some loop and knowledge that I read from my online researches and it seems to be not that hard now. What I had to do for the first one to just getting the subnet and the first valid host will always be after that one. The ending part was the tricky part. But after numerous of research, I finally was able to solve the riddle by nesting the loop functions.

```
Terminal - tee@TEE-DESKTOP: ~/344/lab13_14
File Edit View Terminal Tabs Help
86 if host_ip_string == "q":
87     print("User quit! Goodbye!\n")
88     exit(0)
89 subnet_mask_string = input("Please enter a subnet mask or quit(q): ")
90 if subnet_mask_string != "q":
91     host_ip = host_ip_string.split(".")
92     subnet_mask = subnet_mask_string.split(".")
93     subnet_ip = []
94     host_first = []
95     host_last = []
96     count=1
97     for i in range(4):
98         subnet_ip.append(int(host_ip[i]) & int(subnet_mask[i]))
99     for i in range(4):
100         if subnet_mask[i] == "255":
101             count+=1
102             host_first.append(subnet_ip[i])
103             host_last.append(subnet_ip[i])
104         else:
105             if count == 4:
106                 n = 256-int(subnet_mask[i])
107                 host_first.append(int(subnet_ip[i])+1)
108                 host_last.append(int(subnet_ip[i]+(n-2)))
109                 count+=1
110             else:
111                 n = 256-int(subnet_mask[i])
112                 host_first.append(subnet_ip[i])
113                 host_last.append(int(subnet_ip[i]+(n)))
114                 count+=1
115     print("Range for valid host IP should be: ", ".join(map(str, host_first)), ". ", ".join(map(str, host_last)), "\n")
116     else:
117         print("User quit! Goodbye!\n")
118         exit(0)
119 elif response == "6":
120     subnet_ip_string = input("\nPlease enter a subnet IP or quit(q): ")
121     if subnet_ip_string == "q":
122         print("User quit! Goodbye!\n")
123         exit(0)
124     subnet_mask_string = input("Please enter a subnet mask or quit(q): ")
125     if subnet_mask_string != "q":
126         status=ipaddress.subnet_ip_string+"/"+subnet_mask_string
127         print(status)
128     else:
129         print("\nUser exit! Goodbye!\n")
130         exit(0)
131
132 elif response == "7":
133     print("\nUser exit! Goodbye!\n")
134     exit(0)
```

Introduction: By applying a subnet mask to an IP address it allows the identification of the network (subnet) represented by the 1's in the mask, and the host bits are presented by 0's.

Subnet IP address by performing a bitwise logical AND operation between the IP address and the subnet mask, then setting all the host bits to 0's. What about the broadcast IP? The Broadcast IP address is calculated by first taking all the bits for the Mask, and inverting it. This is performing an OR operation between the IP Address and NotMask.

```
00.11001000 140.179.220.200 IP IP
00.00000000 255.255.224.000 Subnet Mask
00.00000000 140.179.192.000 Subnet IP
11.11111111 140.179.223.255 Broadcast IP
```

288 (219) subnet IP's available with this size mask. Each subnet has 8190 n have hosts assigned to any address between the Subnet IP address and the

bnets or hosts, we used use the formula $(2^n - 2)$ where n = number of bits in the raised to the n^{th} power. Note that although subnet masks with non-contiguous e not recommended.

134,1 Bot

In conclusion:

Although this lab was difficult at first, doing some research really helped pave the road. I got to learn a lot about IP addresses and its components. I am glad that we had this lab .