Automation and Detailed Analysis of Subnetting Problems Using Mathematical Modelling

Deepank Dixit (Student), CIT, UPES, Dehradun
Abhineet Anand, Assistant Professor, Galgotia University, Greater Noida

deepankdixit0804@gmail.com
abhineet.mnnit@gmail.com

Abstract:

This paper aims to develop an application to automate the process of subnetting in the field of networking. One of the prime jobs of a network engineer in an organization is the execution of complex process of subnetting considering the number of required hosts per network or the number of networks that the organization may need. Subnetting basically is the process of division of a large overwhelming network into smaller sub-networks, each with limited broadcast domain, which proves healthy to the entire communicating network.[2] Dividing a network into sub-networks is a crucial step in an organization due to several reasons such as limiting traffic on network and the need for improved communication speed etc. The process of subnetting, although, seems like a moderately challenging task, has more to it than just seemingly trivial mathematical calculation. In this newly emerged age of automation of pretty much everything around us, it is only logical to adopt this accurate and time efficient approach to solving the subnetting problems. Such an approach obviates the, however unlikely, yet the distinct possibility of occurrence of a mathematical error and assures of no erroneous results. The prime objective of this paper is to display all the information of the network (pertaining to subnetting) that one may need to know. The development of the mathematical model of subnetting process is consummated.

Keywords: IPv4 Management, Network ranges, Subnetting, Broadcast/Collision Domains, Subnet masking

Objectives:

The objective is to develop an application to automate the process of by displaying all the information of the network (pertaining to subnetting) that one needs to know.

Methods/Statistical analysis:

The existing subnetting techniques require us to list all network ranges one by one by setting one of the bits to on or off. It is always a cumbersome task to represent the network ranges in binary and later converting it to decimal. Our 3- step approach is to list the network ranges in decimal in the first place, thereby eliminating the need for conversion later.

Findings:

The mathematical approach to solve subnetting problems discussed in this paper is one of the many out there. This approach, however, insures to bring solutions to more than 95% of the subnetting problems. There are a number of ways in which this particular approach can be moulded but the beauty is that all we have to care about is one unique outcome because that is what captures the essence of the network. Therefore, solution to any subnetting problem can be crosschecked through online resources.

No organization today is devoid of the impact of subnetting. It is inevitable, after all. Network traffic control, congestion avoidance, preservation of address space and enhanced network security are amongst the most remarkable advantages of segregating the network.

Application/Improvements:

The applications of subnetting are ubiquitous in the realm of networking. The mentioned approach presents much easier way of designing networks by outlining subnets through the network.

Introduction

Subnetting is one of the many vital parts of networking. It is logical subdivision of an IP network. *Subnetting is the practice of dividing a network into two or more networks.* Dividing a large network into subnets representing administration responsibility for each subnet can make administration of a large network easier. Subnetting has a rather vigorous impact in the realm of networking. Loosely writing, one simple way to look at the Internet could be that the Internet is a collection of routers, routers upon routers, a whole web of routers. The thing about router is that every interface of a router represents a unique network and an organization may contain from a few up to hundreds of routers within it. The design of TCP/IP IPv4 addressing scheme doesn't allow the authorities to assign these many public IP address for one organization because that would go far beyond our exploitation boundaries of IPv4 addresses.[1]

Subnetting brings with itself a myriad of advantages. Security, Limitations, Speed network transfers and improved performance in isolation are only a few of them[2]. Above mentioned almighty routers can require that a WAN link connecting two networks must itself form a separate subnet. Troubleshooting, diagnosing, and fixing problems in a TCP/IP internetwork typically require thorough familiarity with subnetting. Subnetting can improve network performance by splitting up collision and broadcast domains[3]. It reflects organizational structure and help support security policies. Subnets can be treated in a manner that they define administrative units and hence support the structuring and delegation of administrative tasks.

The process of subnetting involves a series of moderately complex mathematical steps, executing which with precision, demands the supervision of a network engineer. A major stumbling block to successful subnetting is often a lack of understanding of the underlying binary math. In fact, the principles of subnetting are difficult to grasp without artistic mastery of binary arithmetic, logic, and binary/decimal conversions. However skilled an engineer might be, there is always a slim possibility of calculation mistake. This paper proposes the idea of an automated system which calculates the subnetting information of a network. Such a system would not be prone to mathematical errors, thereby providing the reliable information that shapes our network.

Example Figure

Figure 1 is a trivial network diagram which shows how two absolutely resembling intra-networks do belong to different network prefix[4]. This also schematically expounds upon one innate property of the router which is having different network ID at its each interface.

The figure very elegantly explains the inevitability of subnetting even in this small network. We have two sub-networks here- LAN1 and LAN2. It is routers job to connect two different networks, which is what it does here. But for every such network in a vast network, unique public IP addresses cannot be assigned. Therefore, we'll have to do the subnetting.

Before getting to the root of subnetting, we must first revise the fundamentals of TCP/IP scheme.

<u>TCP/IP IPv4 Addressing Scheme:</u>

The IP addressed in network world is divided in 5 different sized classes[5]. Microsoft TCP/IP supports class A, B and C addresses to be assigned to hosts. Each IP address is of 32 bits. These bits are either set to 1 or set to 0.These classes are formed only to distinguish between the respective number of 0s and 1s in the IP address. 1 in turn represents the network bit and 0 denotes for the host bit. This also allows us to determine the total number of possible networks and the number of hosts.

Understanding of subnet masks comes in handy when it comes to explore the concepts of subnetting[6]. There are custom subnet masks which will be explored as the discussion becomes richer but each class has its own unique default subnet mask. The subnet mask, same as an IP address is 32 bits long. Even the rudimentary rule of 1 being considered as network bit and 0 as host bit is same. Under class-based network addressing, each class has a unique subnet mask which masks an IP address and divides it into network address and host address. The subnet mask is formed by setting all the network bits to 1s and host bits to 0s[7]. In TCP/IP addressing scheme, it is defined how many bits make up the network portion and how many the host portion.

For example: A class C address has 24 bits dedicated to define the network portion which gives us a network ID. Rest 8 bits are for determining the number of hosts. These 8 bits are what we call the host bits. In this 32 bit number that places itself in 4 octets, each octet, as the name explains itself, has 8 bits.

NNNNNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH
(*N shows the reserved place for a network bit and H for host bit.*)

By setting all 24 network bits to 1s and 8 host bits to 0s, we learn that the subnet mask is-

11111111.11111111.11111111.00000000

This, however, is the binary representation of subnet mask. The decimal representation will be 255.255.255.0.

Thus, the default subnet mask of a class C address is 255.255.255.0 which reveals that the first 3 octets have all of their bits set to 1. Last octet however, is the one with all bits set to 0. This is how a subnet mask determines the total number of network bits and host bits in a type /class of a network[8].

Module 1 IP

Subnetting (General)

Procedure:

To be able to have a machine capable of performing the mathematical calculations just as a network engineer or a student would do, except here with absolute accuracy, it is required to do the step wise analysis of the problem.

This module consists of three sub-modules:

I. Determines the number of total networks or the number of hosts required in a sub-network
II. Calculates the new subnet mask
III. Finds the increment

The increment defines the size of a block (sub-network) which allows us to find the network ranges and network ranges allow us to determine the network ID and the broadcast ID of the respective sub-network.

Proposed Mathematical Model:

A user is prompt to enter an IP address. The class of the entered IP address is first determined. In order to deduce the class of the IP, first octet is examined keeping in mind the predefined structure of classful IP addresses. Once the entered address is known to be belonging to a particular class, the corresponding subnet mask is figured with the help of the reference Table1.

From here on there are two ways flow of execution and respective mathematical calculation may go along (i.e. either on the basis of number of hosts per network or on the basis of required number of networks)[9].Either way, the network is being segmented into smaller ones. The number of networks is increasing, meaning the number of network bits are growing, which is only possible at the expanse of the available host bits[10].The number of predefined host bits and network bits in each class is mentioned in Table 2.

These host bits are the ones that are at the heart of subnetting process[11], which in the module, for the sake of avoiding ambiguity in calculation, are used as default bits (db). After all, these are the default number of bits that are going to be exploited in subnetting. Subnetting can be performed in two ways which originate from the fundamental network requirement of the organization. It can be the required number of hosts per network or the required number of networks that allows us to perform the further calculations.

Any one of these approaches to segment a network is not any less or more popular than the other[12]. In fact, it all depends upon the requirement of the network infrastructure. Therefore, a detailed analysis of both the approaches is in order.

I. (A) On the basis of required number of hosts per network:

The user is prompt to enter the number of hosts required per sub-network in his network scenario. At this point, the knowledge about the class of the IP, default subnet mask of corresponding class, number of host bits and number of network bits is gathered. All this information helps present the basic layout of the concerned network. The number of required hosts invokes the calculation of further mathematical problem which is to subnet the network based upon this latest entry made by the user, which throughout the module is denoted as **hpn**.

Algorithm1 (a) *(Calculates the number of host bits and network bits)*:

$$\boldsymbol{nhb} = \log(hpn+2) / \log(2)$$

nhb is the decimal value of number of host bits. The ceiling value of this obtained value is to be taken, which is calculated using *ceil ()* function.

Thus, number of host bits after subnetting = *ceil (nhb)*

Follows from it, instantly the number of total assignable hosts per sub-network using the formula $2^{ceil\ (nhb)}$.

Number of subnet bits after subnetting = *db - ceil (nhb)*

db here is the aforementioned number of default host bits. The newly defined number of hosts per sub-network by the user alters the default number of host bits and thus emerges a new number that we calculated to be *nhb*. Rest of the bits in the default host bits (0s) are then converted to network bits (1s). Therefore, subtracting of number of calculated host bits from the default host bits yields the number of new network bits with the formula *db - ceil (nhb).*

The number of total sub-networks is $2^{db-ceil\ (nhb)}$**.**

Evidently, subnetting is performed at the cost of the default number of host bits[13]. Hosts bits are sacrificed in order to obtain more networks (sub-networks).

(B) On the basis of required number of networks:

The approach to calculate the host bits and network bits is almost as same as the above one with a few differences in mathematical formulation. The formulation below is written considering the first and the last subnet as valid subnets.

Algorithm 1(b) *(Calculates the number of host bits and network bits)*:

$$nsb = \log(snw) / \log(2)$$

nsb is the fraction value of number of subnet bits.

Number of subnet bits = *ceil (nsb)*

These are the number of bits that needs to be changed to 1. These bits were originally in the default subnet mask as host bits (0s) but since the network scenario demands more number of networks, some host bits need to be sacrificed and converted into network bits (1s).

Total number of sub-networks = $2^{ceil\ (nsb)}$

Number of host bits = *db – ceil (nsb)*

Total number of valid hosts = $2^{(db-ceil\ (nsb))}-2$

For the sake of simplicity, let's say number of host bits and number of network bits are stored in *var1* and *var2* respectively (for algorithm 1(a) and 1(b) both).

II. Calculation of new Subnet Mask:

Nsm here is used as an abbreviation for New Subnet Mask or Custom Subnet Mask. A separate sub module for the nsm is designed as its calculation is comparatively tricky and needs to be carried out cautiously.

Algorithm 2*(Calculates the new subnet mask):*

(I)For Class C IP address:

db = 8

Change *var2* number of bits to 1 from most significant bit side in the increment octet (4th octet).These 1s are contiguous and followed by *var1* number of 0s. The new subnet mask will be calculated using the formula-

$$255.255.255.\ \sum = - 2$$

db= 8 shows that the IP address is of class C and first three octets (i.e.255.255.255) in the default subnet mask of class C must be preserved. Bits of octet four are to be altered. Therefore, change is made in the fourth octet of the default subnet mask.

(ii)For Class B IP address:

db = 16

There are two cases:

Case 1- 1 <= var2 <= 8

Change *var2* number of bits to 1 from most significant bit side in 3rd octet. The new subnet mask will be calculated using the formula-

$255.255.\sum = - 2\ .0$

Case 2- 9 <= var2 <= 15

Change *var2* number of bits to 1 from most significant bit side in 4th octet.

The new subnet mask will be calculated using the formula-

$255.255.255.\sum_{i=0}^{var2-9} 2^{7-i}$

(iii)For Class A IP address:

db = 24

There are three cases:

Case 1- 1 <= var2 <= 8

Change *var2* number of bits to 1 from most significant bit side in 2nd octet.

The new subnet mask will be calculated using the formula-

$255.\sum = - 2\ .0.0$

Case 2- 9 <= var2 <= 16

Change *var2* number of bits to 1 from most significant bit side in 3rd octet. The new subnet mask will be calculated using the formula-

$255.255.\sum_{i=7}^{-var2} 2^{i}\ .0$

Case 3- 17 <= var2 <= 23

Change *var2* number of bits to 1 from most significant bit side in 4th octet.

The new subnet mask will be calculated using the formula-

$255.255.255.\sum_{i=7}^{24-var2} 2^{i}$

III. Calculation of increment to find ranges of Sub-networks:

Increment is the important parameter to calculate ranges of the sub-networks. It allows us to determine the segmentation structure of sub-networks. After all, initially there was one network which is now segregated into several smaller ones. The important information to be gained is at what position in the original one range of the total host does the barrier need to be placed, so as to discover ranges of each individual sub-network. This sub-module computes this sensitive problem.

Every sub-network has its own network ID and a broadcast ID[14]. The first address among the total number of hosts per sub-network is used to denote the network identity and the last one for the broadcast ID. However, implementing logic of finding ranges through coding is comparatively arduous; calculation for ranges of sub-network is rather trivial. The complexity increases as we transit from class C to class A calculations. Increment Octet is another parameter which is crucial in penning down these ranges. Since there are four octets, it is important to know from which octets begins the ranging mechanism.

Mathematical formulation to find increment varies from class to class. Even there are several cases within a class depending upon the number of network/host bits that are needed to be altered. The value of increment octet follows directly from the values of var1 and var2 with little bit of mental calculation. Increment is denoted as *inc* and increment octet as *incOct*.

Algorithm 3*(calculates range of sub-networks):*

(I) Class C

$inc = 2^{var1}$

$incOct = 4$

(ii) Class B

Case 1- $1 \leq var2 \leq 8$

$inc = 2^{8-var2}$ $incOct = 3$

Case 2- $9 \leq var2 \leq 15$

$inc = 2^{16-var2}$ $incOct = 4$

(iii) Class A

Case 1- $1 \leq var2 \leq 8$

$inc = 2^{8-var2}$ $incOct = 2$

Case 2- $9 \leq var2 \leq 16$

$inc = 2^{16-var2}$

$incOct = 3$

Case 3-17 $\leq$ var2 $\leq$ 23

$inc = 2^{24-var2}$ $incOct = 4$

Module 2

Reverse Engineering Subnetting

Although, the subnetting style that seems to be mostly dealt with is the one above analysed, it is not the entirety of the situation. The most common style of subnetting is performed in reverse engineering manner[15]. It calls for working out our problems backwards.

This type of subnetting mostly proves useful to troubleshoot network configuration errors. In most cases we want to know what network a host belongs to or which network is this host a part of[16]. Determining whether a particular host is on the same network as the other or not, becomes easy with reverse engineering such problems. There are times when we want to check the validity of the IP address with a particular subnet mask[17]. Figuring out these problems with this approach is time efficient, accurate and reliable.

Procedure:

To reverse engineer a subnet problem it is crucial to find out the increment that was used to calculate the range of the sub-networks. The way, it is done it rather straightforward with the binary/decimal calculation. There are several ways to find the increment with the help of the entered custom subnet mask. Each octet of a valid subnet mask consists of one of three types of decimal values. Two of these are 255 and 0. Three of the four octets have the decimal value as either 255 or 0. The remaining one octet is what is known as non-255/non-zero octet. The job is to find out the non-255/non-zero octet value. Once the value is known, it is then converted into binary to find out the increment.

Mathematical Formulation:

Since the required value is the decimal value of the non-255/non-zero octet in binary, let's say the resulting binary 8 bit representation is-

R7	R6	R5	R4	R3	R2	R1	R0

Say, Q is the decimal value of non-255/non-zero octet.

$Q_n \in I$ and $R_n \in I$

$Q_0 = Q/2$ and $R_0 = Q - 2Q_0$

If $Q_0 = 0$

$R1=R2=R3=R4=R5=R6=R7 = 0$

If $Q_0 \neq 0$

$Q_1 = Q_0 / 2$ and $R_1 = Q_0 - 2Q_1$

If $Q_1 = 0$

$R2=R3=R4=R5=R6=R7 = 0$

If $Q_1 \neq 0$

$Q_2 = Q_1 / 2$ and $R_2 = Q_1 - 2Q_2$

If $Q_2 = 0$

$R3=R4=R5=R6=R7 = 0$

If $Q_2 \neq 0$

$Q_3 = Q_2 / 2$ and $R_3 = Q_2 - 2Q_3$

If $Q_3 = 0$

$R4=R5=R6=R7 = 0$

If $Q_3 \neq 0$

$Q_4 = Q_3 / 2$ and $R_4 = Q_2 - 2Q_3$

If $Q_4 = 0$

$R5=R6=R7 = 0$

If $Q_4 \neq 0$

$Q_5 = Q_4 / 2$ and $R_5 = Q_4 - 2Q_5$

If $Q_5 = 0$

$R6=R7 = 0$

If $Q_5 \neq 0$

$Q_6 = Q_5 / 2$ and $R_6 = Q_5 - 2Q_6$

If $Q_6 = 0$

$R7 = 0$

If $Q_6 \neq 0$

$Q_7 = Q_6 / 2$ and $R_7 = Q_6 - 2Q_7$

And $Q_8 = 0$

This way, all the values from R_0 to R_7 are known. These 8 bits when put together yield the binary representation of the decimal value. Thus, the binary value is found. Now, the increment is to be calculated. To find the increment, among these 8 bits (R_1 through R_7), the lowest network bit is to be translated to its decimal value[18]. This is the increment that the network designers used to build the concerned network.

The computation of the range can be done using Algorithm 3 Module 1. Thus, the sub-network which the entered IP address belongs to is displayed along with details of all other sub-networks.

Simulation and Results

The idea of subnet simulating can be implemented in various programming languages; however procedural languages are always a good choice to begin with the logic development for mathematical calculations. Once logic is built, one can initiate to write the codes in the preferred language. Figure 2 is the workflow process diagram followed by the snapshots of logic testing results from both modules which present an overview of how the flow of execution is driven in program

(a)Module 1

Module 1 solves IP subnetting problem by general approach. User is prompt to enter an IP address. Validity of this IP address is checked. Then, the user is prompt to choose the type of subnetting problem that needs to be solved. Once all the inputs are taken, with the help of algorithms mentioned in Module one: Subnetting procedure, the layout of the sub-netted network is made available to user on screen. Run time interface for Module 1 is attached in Figure 3.

(b) Module 2

This module solves the subnetting problem by reverse engineering method. Values of an IP address and a subnet mask is inputted. This subnet mask is the one that network designers decided to structure the network. Mathematical formulation mentioned in Module two helps to compute the corresponding sub-network ranges and display them to the user. Run time interface for Module 2 is attached in Figure 4.

Conclusion

No organization today is devoid of the impact of subnetting. It is inevitable, after all. Network traffic control, congestion avoidance, preservation of address space and enhanced network security are amongst the most remarkable advantages of segregating the network. The paper thoroughly explores the various features of this tool and presents the entire subnetting layout of the network. Not only does this tool help network engineers to crosscheck their performed operations while troubleshooting the network, it also can prove greatly useful for education purposes. Students can verify their sub-netting results to make sure whether or not they got the correct answer.

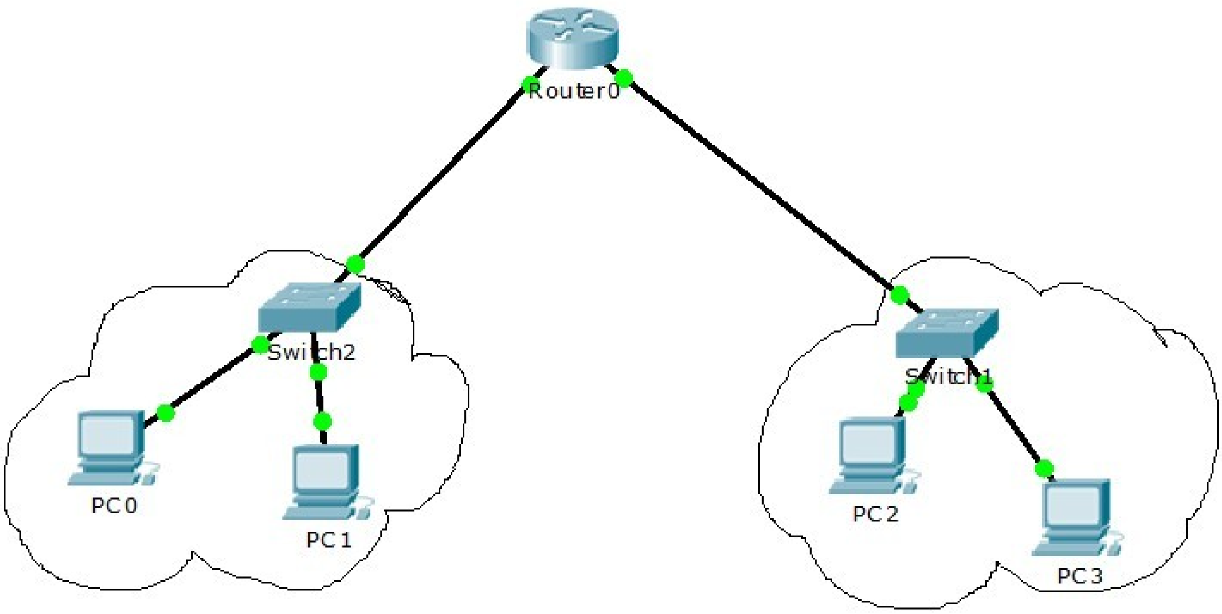
References

1. Todd Lammle Subnetting class C network address www.techrepublic.com/article/subnetting-aclass-c-network-address/, Date accessed: 24/05/2001.
2. Jeremy Cioara-Exam Prep CCNA 2nd Edition, ISBN-13: 978-0-7897-3713-7 https://www.amazon.com/CCNA-Exam-Prep-640-802-2nd/.../07897371
3. Andrew S. Tanenbaum-Computer Networks, Fifth Edition, ISBN-13: 978-0-13-212695-3.1944.
4. Basha S, AhmedN. Department of Quality Assurance, Mazoon University College, Muscat), Automation of Sub-Netting Problems- International Journal of Engineering Science Invention.2016.
5. Academy Delmar, Subnetting via TCP/IPhttp://academy.delmar.edu/Courses/ITSC1405/eBooks/Win2K03 (ServerSecretsBook)/Chapter07.pdf
6. IPv4 Addressing and Subnetting v1.41 – Aaron Balchunas http://www.routeralley.com/guides/ipv4.pdf,Date accessed: 2013.
7. Met M H, GunesG, Sarac K. —Inferring Subnets in Router-level Topology Collection

   Studies, 7th ACM conference on Internet measurement, USA, 2007, pp. 1-6.
8. Richker J .IP Subnetting for the Masses .2008.
9. Tech-faq, 127.0.0.1- what are its uses and why it is important http://www.tech-faq.com/1270-0-1.html.Date accessed: 11/5/2016.
10. Demand Media, Importance of subnetting, http://yourbusiness.azcentral.com/importancesubnetting-20038.html,Date accessed:2016.
11. BG Expert: IP addresses. www.bgpexpert.com/addressespercountry.php.Date accessed: 18/06/2016.
12. Brien M, Posey P. Subnetting to Increase Performance .2015.
13. IP Addressing and Subnetting for New Usershttp://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html
14. Solving the Mysteries of Subnetting
15. https://d12vzecr6ihe4p.cloudfront.net/media/966006/wp-solving-the-mysteries-ofsubnetting.pdf
16. Neural subnet design by direct polynomial mappinghttp://ieeexplore.ieee.org/abstract/document/165606/?reload=true
17. Advanced folding the paper subnetting concepts
18. http://www.ebrahma.com/2014/12/advanced-folding-paper-subnetting-concepts/

19. Math of IP Subnetting and Subnet Masks - DigiPro Digital Productions
20. http://www.digipro.com/Papers/IP_Subnetting.shtml

21. Subnetting Networks Research Paper
22. http://essayzoo.org/research-paper/apa/technology/subnetting-networks.php



LAN1
Subnet 1
192.168.10.32/28

LAN 2
Subnet 2
192.168.1.80/27

Figure 1: Two local area networks (on different subnets) connected by a router

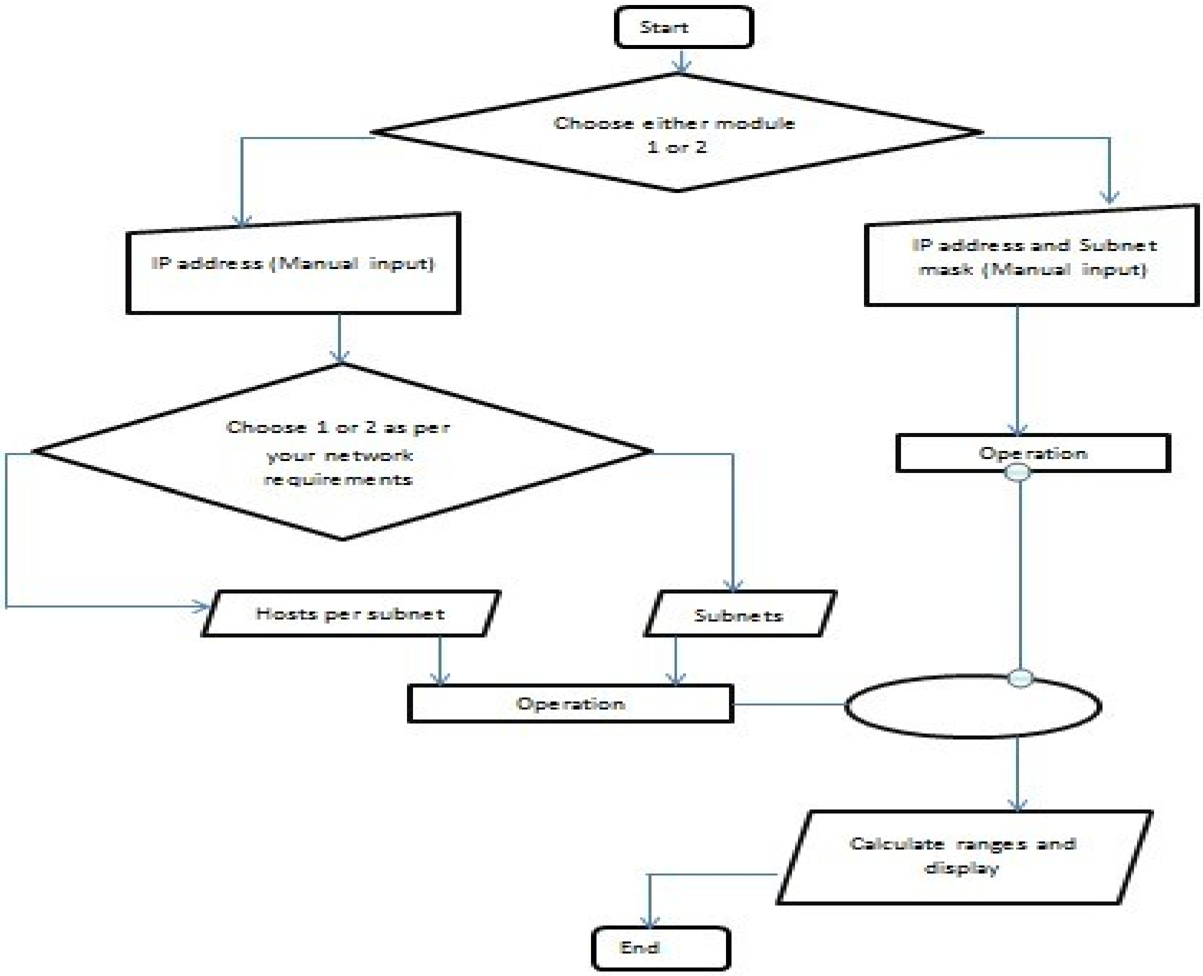


Figure 2: Workflow process diagram

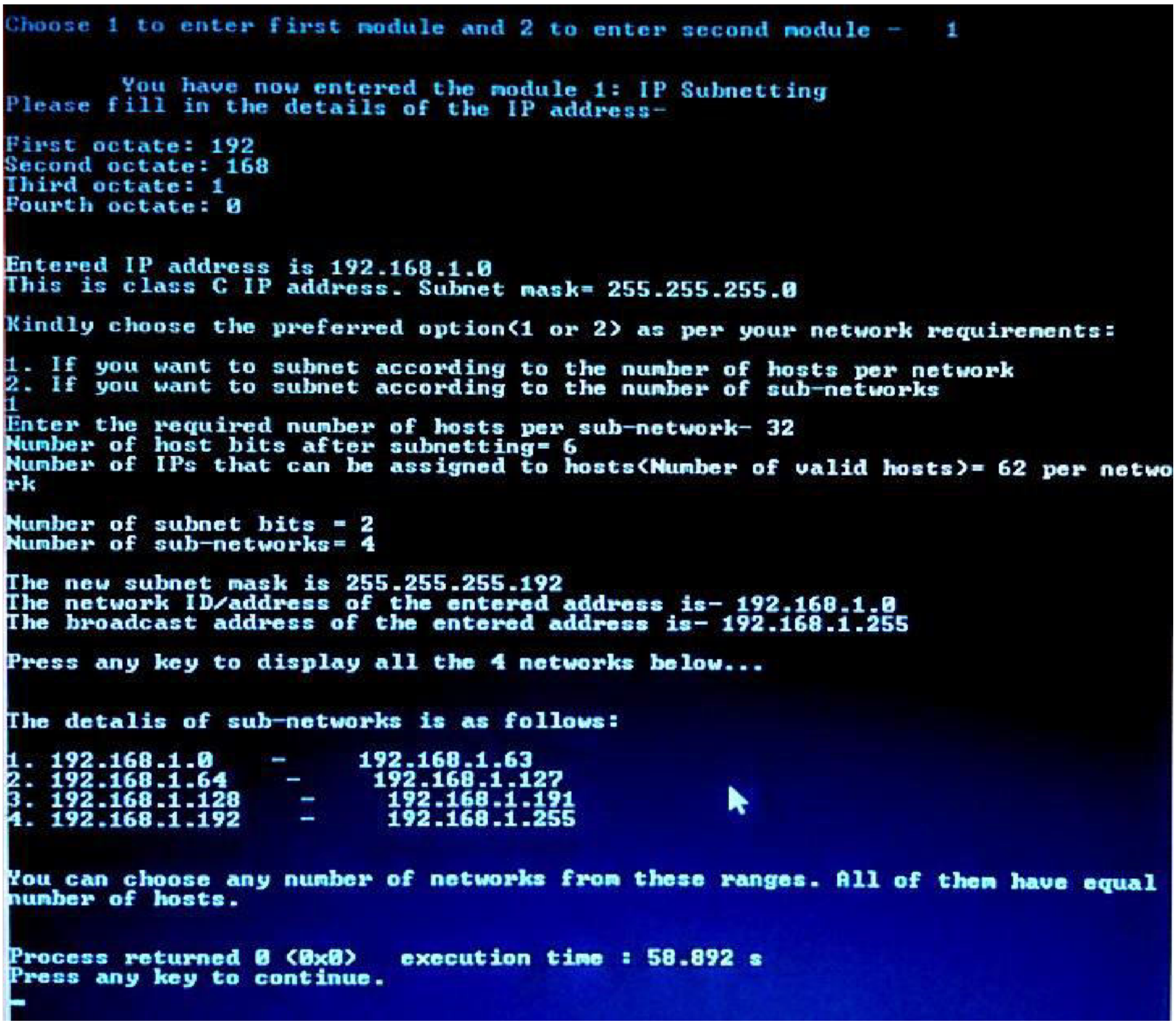


Figure 3: Customer interface for Module 1

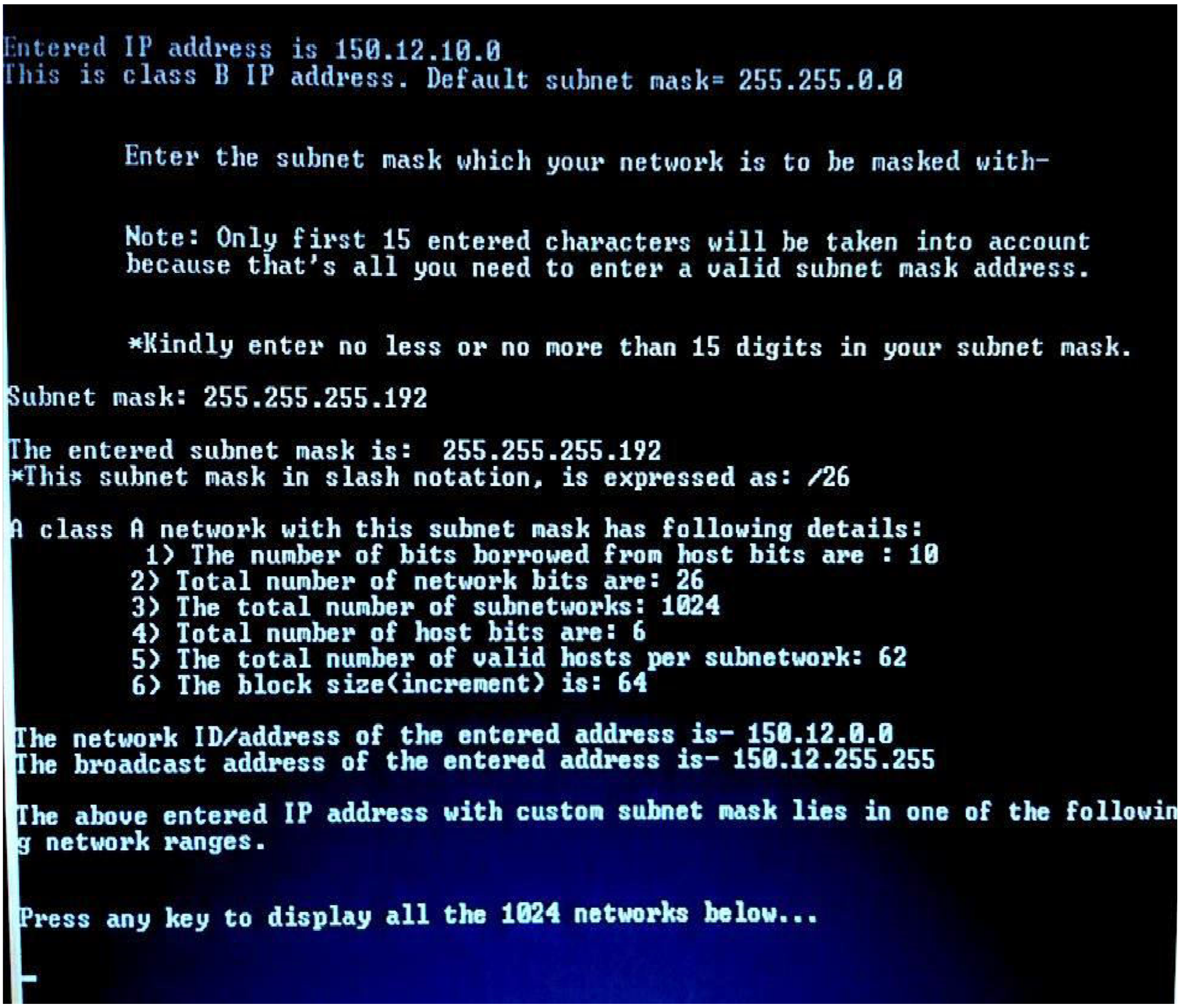


Figure 4: Customer interface for Module 2

Corresponding Class	Range of the First Octet	Subnet Mask
A	1-126	255.0.0.0 or /8
B	128-191	255.255.0.0 or /16
C	192-223	255.255.255.0 or/24

Table 1

Corresponding Class	Number of Host Bits	Number of Network Bits
A	24	8
B	16	16
C	8	24

Table 2