Private Subnetting One (PS1), an Algorithm for Private IPv4 Address Allocation

Nabhan Hamadneh
Department of Computer Science
Hashemite University
Zarqa, JORDAN
+962 5 3903 333 – Ex. 4950
nabhan@hu.edu.jo

David Murray
School of Engineering and
Information Technology
Murdoch University
WA, Australia
+61 8 9360 2723
D.Murray@murdoch.edu.au

ABSTRACT
Network hosts in private enterprises require local connectivity without any external connections. This kind of networks is operated using TCP/IP protocol and require a private address space. This article proposes a novel algorithm for IPv4 address allocation for this type of networks. This algorithm can be used by educators of subnetting, ISPs, university students and researchers to simplify the process of IPv4 address allocation. A tool called Private Subnetting One (PS1) programmed under Java is introduced in this article. This tool can be used by network engineers or university tutors to explain complex subnetting problems to students in introductory networking courses, such as CCNA.

CCS Concepts
• **Network architectures**→ **Network design principles** → **Naming and addressing.**

Keywords
Subnetting; Private networks; IPv4; Address allocation; IP calculator.

1. INTRODUCTION
The internet is growing rapidly and the number of subscribers has greatly increased depleting the limited IPv4 addresses. To overcome this address exhaustion; the Internet Engineering Task Force (IETF) deployed the IPv6 protocol to replace the IPv4. Despite the depletion of some IPv4 pools [1, 2], the use of the IPv4 protocol is likely to continue for the next two decades, especially in developing countries [3, 4] due to the tremendous cost and efforts for converting to IPv6 [5].

Hosts in private networks do not require access to external hosts in other enterprises. Even if they require external access, it will be to limited services such as e-Mail, FTP or remote login. Hence, organizations could have IP addresses for local use without authority of internet registry. However, the organizations still need to change to global IPs for Internet connectivity. Therefore, the Internet Assigned Number Authority (IANA) has reserved the IP address blocks in Table 1 for private use only.

The first block in Table 1 is called the *24-bit* block. The second block is referred to as the *20-bit* block, and the third is referred to as the *16-bit* block. It is noticeable that the first block is a whole class A CIDR block, whereas the second block is a composite of

16 contiguous class B CIDR block and the third block is constructed of 256 contiguous C CIDR blocks [6].

Subnetting can be complex and confusing task and any erroneous configuration can lead to overall malfunction of the network. In this article we revisit the IPv4 address allocation and provide an algorithm to simplify this process for private networks; especially for block 3 which is the *16-bit* block. In future work we plan to expand this algorithm to include the other two blocks or even generalize the algorithm for the whole CIDR blocks.

Table 1. Private IP address blocks by IANA.

Block #	First address	Last address	Prefix
1	10.0.0.0	10.255.255.255	10/8
2	172.16.0.0	172.31.255.255	172.16/12
3	192.168.0.0	192.168.255.255	192.168/16

The rest of this article is organized as the following: Section 2 presents the previous work in this topic. Section 3 provides a novel algorithm for private IP allocation. The implementation using a Java application is presented in Section 4. The conclusion and future works are presented in Section 5.

2. RELATED WORK
There are many tools for address allocation on the web. All of them take an IP address and a subnet mask to calculate the range of IP addresses for hosts in subnets. These tools are used as teaching tools for the subnetting topic in computer networking courses. Examples of these tools are the IpCal [7] and IpOnline [8] calculators.

Another interested tool which is worthy to be studied in this article is the Phase Teaching Model for Subnetting IPv4 (PTMS-IP4) [4]. What makes this tool special is the way it reacts to user inputs. Like any tool with GUI, IPv4 calculators have inputs, processing and outputs. The input and output are shown on each tool but the methods they process the input to generate the output is not fully documented or shared by the authors. The authors of the PTMS-IP4 published there method on handling the IP distributions in [4]. They convert the provided IP address into its binary format and use the equation $2^n - 2$ to make the PTMS-IP4 calculations. This process is illustrated in the next paragraphs.

Subnetting with PTMS-IP4 goes through three phases as shown in Figure 1. In phase one, the tool asks the user for the IP address and subnet mask. When the user hits the *run subnetting* button, the tool goes to phase two in which it calculates the number of

subnets, number of hosts per each subnet and the mask in binary code. In addition, phase two produces two more important parameters which are the *subnet factor* and *subnet permutations*.
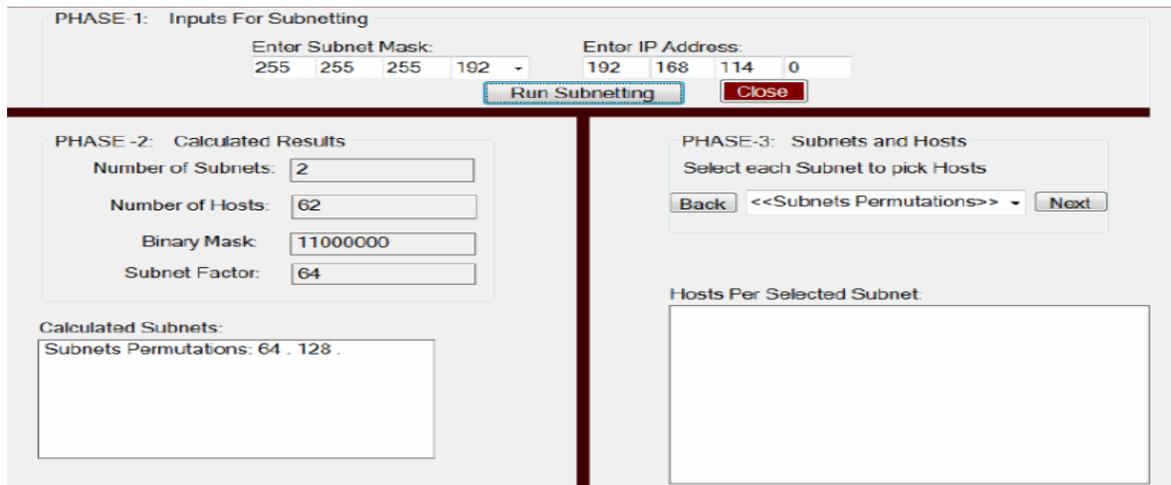

Figure 1: PTMS-IP4, three phases interface [4].

The *subnet factor* is the number of hosts in each subnet. For example, if the user intended to use the IP address 192.168.114.0 with the subnet mask 255.255.255.192 and hits the *Run Subnetting* button, then the resulted binary mask is 1100 0000 which is the number 192 depending on the last octet of the subnet mask. Using the $2^n - 2$ method, the number of networks will be $2^2 - 2$. Here, n equals 2 which is the number of 1s in the binary mask. The tool uses the same method $2^2 - 2$ to calculate the *subnet factor* which is the number of hosts to move from subnet border to another. Here, n equals six 0s in the binary mask. Therefore the *subnet factor* will be 64 and the number of hosts per network is 62 which is the *subnet factor* minus 2 by excluding the network address and broadcast address.

The *subnet permutation* in this example represents the forth octet of the beginning of each subnet. In other words, they represent the network ID of each subnet. Here, the subnet permutations are 64 and 128. In phase 3, the tool provides the user with the ability to select any permutation to show all available network address in this range.

A comparison between the PTMS-IP4, Ping-Scanner Pro 4.5 and IP Subnet Calculator IPv4/IPv6 is presented in [4]. However, all the algorithms of the IP calculators including the PTMS-IP4 depend on converting the IP address and subnet mask into their binary or hexadecimal forms to do the subnet calculations. Then they distribute the provided space into subnets with an equal number of addresses which causes wasting on the provided IP space. This comes into inconvenient to user especially when she/he requires dividing the space into different number of subnets with different number of IPs per each subnet.

The size of subnets is increasing but the private address space is limited [9]. Well managed IP Address subnetting minimizes the wastage of address space [10, 11].Hence, network distribution tools need to be more accurate to preserve the available private addresses space. In this article, we present a tool called the Private Subnetting One (PS1) which takes an IP address from the *16-bit* block and uses the decimal format of this IP to make the subnet calculations. The user of this tool is free to distribute the IP space to any number of subnets with any number of IPs per each

subnet. This will help in preserving the private IP address space from wasting. It is also designed to solve complex subnetting scenarios that are not solvable by other tools described in this section.

Due to using the decimal format of the IP address we suppose to find our algorithm more understandable by students in a class room. It also could be considered as alternative subnetting algorithm for network programming labs. The satisfaction of these two points will be testified in future work.

3. THE PROPOSED ALGORITHM

This section presents the proposed algorithm and the differences between it and the other algorithms in the literature. Figure 2 shows the flowchart of our algorithm and shows the process of calculating the IP address for a specific subnet group.

Table 2: The IP calculation process

octet1.occ2.acc.first1 …….. octet1.occ2.acc1.last1

octet1.occ2.acc2.first2 …….. octet1.occ2.acc2.last2

$$acc2 = acc1 + \frac{nob * apb}{256} - 1 \qquad (1)$$

$$last = (weight - flag) * apb - 1 \qquad (3)$$

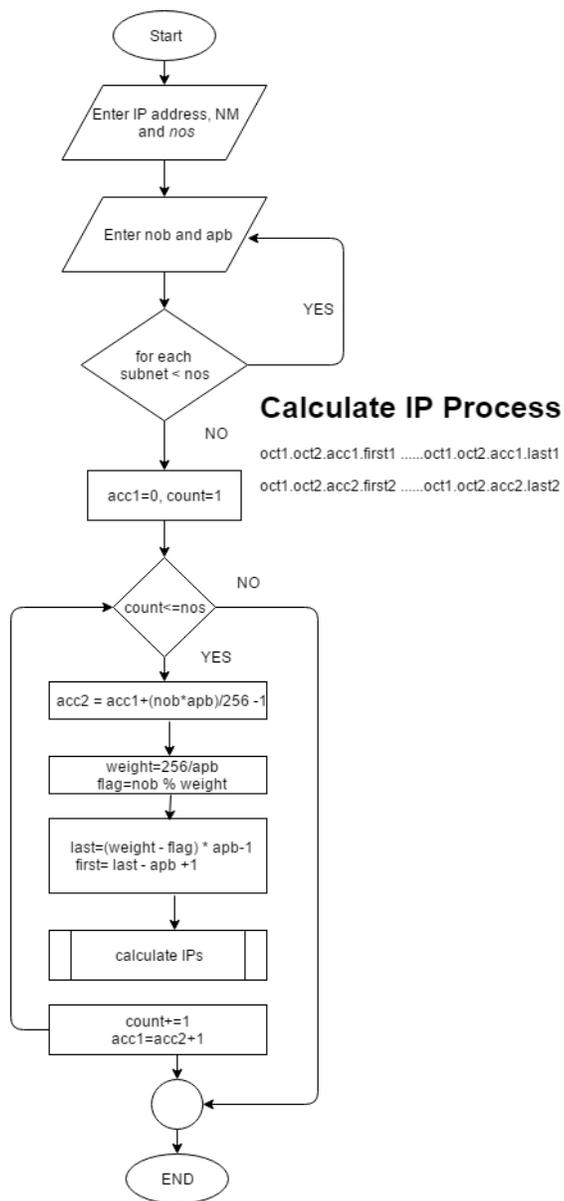$$first = last - apb + 1 \qquad (2)$$

Start

Enter IP address, NM and *nos*

Enter nob and apb

for each subnet < nos — YES

NO

acc1=0, count=1

count<=nos — NO

YES

acc2 = acc1+(nob*apb)/256 -1

weight=256/apb
flag=nob % weight

last=(weight - flag) * apb-1
first= last - apb +1

calculate IPs

count+=1
acc1=acc2+1

END

Calculate IP Process

oct1.oct2.acc1.first1 ......oct1.oct2.acc1.last1

oct1.oct2.acc2.first2 ......oct1.oct2.acc2.last2

Figure 2: The flowchart of the proposed algorithm.

Subnets ✕

ⓘ Subnet: 1
192.168.0.0........... 192.168.0.255

192.168.63.0........... 192.168.63.255

Subnet: 2
192.168.64.0........... 192.168.64.127

192.168.127.128........... 192.168.127.255

Subnet: 3
192.168.128.0........... 192.168.128.63

192.168.159.192........... 192.168.159.255

OK
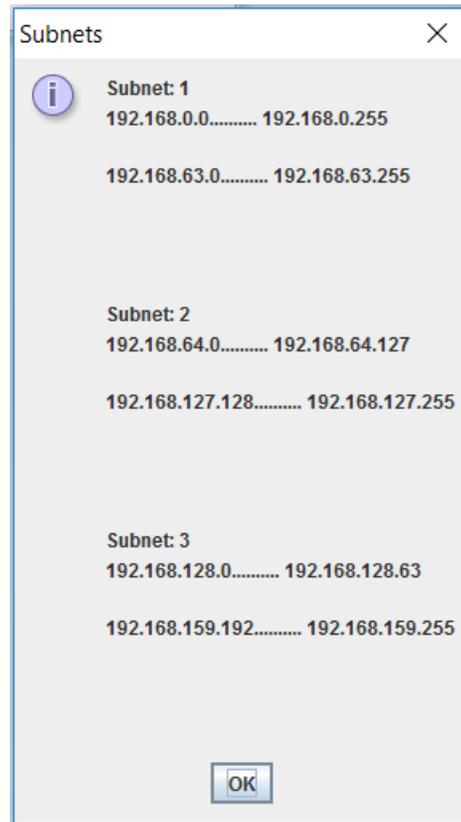
Figure 4: The results of example 1.

IPv4 address Allocation — ☐ ✕

Enter IP Address and subnet mask [192] . [168] . [0] . [0] / [16] ▼

Enter the number of subnets then press Enter: [3]

Enter the number of blocks and addresses for each subnet

	Number of Blocks	Addresses Per Block
Subnet1	64	256
Subnet2	128	128
Subnet3	128	64

Allocate IPs

Figure 3: Snapshot of example 1.
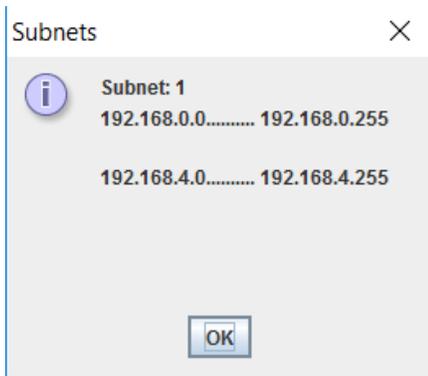
Figure 5: The results of example 2.

Figure 6: IP distribuation for example 2.

Since we use the *16-bit* block, the user does not have to enter the number of bits or network mask *NM*. The user has to enter an IP address from the required IP space and the number of required subnet groups denoted by *nos* in Figure 2. The user has also to enter the number of blocks in each group denoted by *nob* and the number of IPs per each block denoted by *apb*. The parameters *octet1* and *octet2* in Table 2 will be the same as in the provided IP address. Hence, IP calculations start from the third octet. Always for the first group the *acc1* octet is initialized to zero. For the last IP address *acc2* octet is calculated using Equation (1). These are the required information by the calculator to generate the right distribution for the user. The *weight* parameter represents the number of permutations for the last octet to make a full circle from *0-255*. The last and first octets are calculated using Equations (3) and (2) respectively.

4. THE IMPLEMENTATION

In this section, we implement our novel algorithm using a Java application. Figure 3 shows the GUI of our application. The user can enter an IP address from the required space. The *16-bit* block option is activated by default. For example, we use the IP address 192.168.0.0 to make three groups of subnets. The first group has 64 blocks with 256 IPs in each subnet. The second group has 128 subnets with 128 addresses in each subnet. The third group has 128 subnets with 64 addresses per each block. Our tool has a distinctive feature in solving this type of addressing problems. Figure 4 shows the result of the IPs distribution over subnets resulted by pressing the button *Allocate IPs* in Figure 3.

This is an example of complex subnetting scenarios. To test our tool for simple subnetting scenarios which are normally achievable by any other IP calculator in the literature; we use a second example with simple requirements. Here we will ask the tool to make only five subnets with the same IP address in the first scenario including 256 IPs in each subnet. By hitting the *Allocate IPs* button the tool generates the result shown in Figure 5. The distribution of the IP addresses generated by the second scenario will look like Figure 6 in real private network.
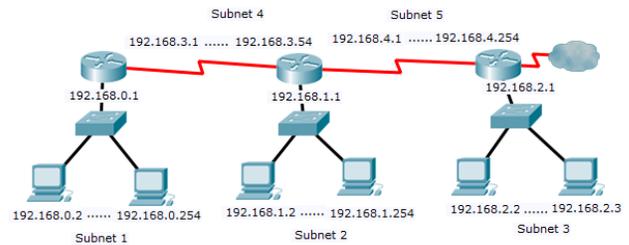
5. CONCLUSION

There exist a wide variety of software products that help network system administrators distribute private IPv4 addresses. However, they distribute these addresses for equal number of subnets and equal number of addresses per each subnet which waists the provided IP space. This article proposed a novel IPv4 private addressing algorithm for different number of groups of subnets and different number of addresses per each group. This provides the network administrator with flexible and easier way to distribute private IPs and preserves the IP space for future use.

As a future work we plan to expand this tool to a comprehensive IP calculator including *the 24-bit* block and the *20-bit* block. We also plan to replace the Java application used in this article by a Java applet and publish it online for public use.

6. REFERENCES

[1] Arin.net. *ARIN IPv4 Countdown Plan*. https://www.arin.net/resources/request/ipv4_countdown.html, 2014.

[2] LACNIC *No more IPv4 addresses in Latin America and the Caribbean*. http://www.lacnic.net/en/web/anuncios/2014-no-hay-mas-direcciones-ipv4-en-lac, 2016.

[3] Hogg, S. *Techniques for Prolonging the Lifespan of IPv4*. Network World, http://www.networkworld.com/article/2221076/cisco-subnet/cisco-subnet-techniques-for-prolonging-the-lifespan-of-ipv4.html, 2016.

[4] Jameson, M. and Kauna, M. Phase teaching model for subnetting IPv4. *International Journal of Technology and Secured Transactions*, 3, 11 (2011).

[5] Cheon S, Jin D and C., K. A VLSM Address Management Method for Variable IP Subnetting. In *Proceedings of the Computational Science and Its Applications -ICCSA* (2006).

[6] Merit, E. G. *Guidlines for Management of IP Address Space*. 1366, 1992.

[7] Fenski, B., Hainsworth, D. A., Hu, F., Quotschalla, F. et. al. *IP Calculator*. http://jodies.de/ipcalc, 2005.

[8] Online network tools. *IP Calculator online*. http://ipcalc.nmonitoring.com/, 2017.

[9] TechNet Microsoft *Public and Private Addresses* https://technet.microsoft.com/en-us/library/cc958825.aspx, 2017.

[10] Afwarman M. and Bataona, D. S. Analysis and Design of Subnetting Methods : Hybrid Fixed Length Subnet Masking (HFLSM). In *Proceedings of the International Conference on*

Electrical Engineering and Computer Science (Bali, Indonesia, 2014). IEEE.

[11] Sabir, M.R., Mian, M.S, Sattar, K. and M. A. Fahiem. IP Address Space Management using Aggregated Fixed Length

Subnet Masking. In *Proceedings of the International Conference on Electrical Engineering ICEE '07.* (Lahore, Pakistan, 2007).

Columns on Last Page Should Be Made As Close As Possible to Equal Length

Authors' background

Your Name	Title*	Research Field	Personal website
Nabhan Hamadneh	assistant professor	Networking	http://staff.hu.edu.jo/CV_E.aspx?ID=rdb5rdUEdvg=
David Murray	senior lecture	Networking	http://profiles.murdoch.edu.au/myprofile/david-murray/

*This form helps us to understand your paper better; the form itself will not be published.

*Title can be chosen from: master student, Phd candidate, assistant professor, lecture, senior lecture, associate professor, full professor