This submission template is a convenient document for you to provide your work and your answers for Lab 4. This submission template is intended to be used in conjunction with the Lab 4 Instructions document. The instructions document illustrates how to correctly derive the answers, explains important theoretical and practical details, and contains the complete set of instructions for this lab.

Name:

Donghang He

Date:

2019/11/24

**Section One – Variable Length Subnet Masks**

1. Explanation regarding fixed lengths subnet masks.   
  
2. Calculations and results for the given scenarios. Make sure to show your work, but you do not need to draw a diagram unless it assists you in answering the question.   
  
*Scenario A*

* First we determine the number of hosts bits:

32 bits – 22 bits = 10 bits

* Then we raise our base of 2 to the exponent represented by our host bits:

210 = 1024

* Finally, we subtract the network and broadcast addresses, as well as the host addresses:

1024 – 2 – 217 = 805

* So the number of assignable addresses are 1022 and the number of unused addresses are 805.

*Scenario B*

* first we determine the number of hosts bits:

32 bits – 25 bits = 7 bits

* then we raise our base of 2 to the exponent represented by our host bits:

27 = 128

* finally, we subtract the network and broadcast addresses, as well as the host addresses:

128 – 2 – 119 = 7

* so the number of assignable addresses are 126 and the number of unused addresses are 7

*Scenario C*

* first we determine the number of hosts bits:

32 bits – 28 bits = 4 bits

* then we raise our base of 2 to the exponent represented by our host bits:

24 = 16

* finally, we subtract the network and broadcast addresses, as well as the host addresses:

16 – 2 – 1 = 13

* so the number of assignable address are 14 and the number of unused addresses are 13

4. Explanation regarding re-allocation of subnets.

6. Calculations and results for the given scenarios. Make sure to show your work.

*Scenario A*

1. we start by converting the first subnet’s network address into binary:

10101100.11100010.00000000.00000000

Now isolating our 26 bits network identifier we arrive at:

10101100.11100010.00000000.00

Now because we are using fixed length subnet masks, we know that the second subnet will use the same number of bits for its network identifier, but we must increment the last bit in the first subnet’s network identifier in order to arrive at the second subnet’s network identifier.

10101100.11100010.00000000.01

So, our second subnet has a network address represented by the CIDR entry:

172.226.0.64/26

1. first we need to determine how many host bits we need to support the 27 hosts needed in our second subnet

We would need 5 hosts bits as 25 = 32

Now looking at the network identifier for our first subnet

10101100.11100010.00000000.00

We need to increment the network identifier of the first subnet to arrive at the next available location for our second subnet.

10101100.11100010.00000000.01

To use VLSM, we now need to add 2 additional bits to our subnet identifier:

10101100.11100010.00000000.0100

We convert this to dotted decimal to arrive at our network address in CIDR entry:

172.226.0.64/27

1. part a: 32 – 26 = 6 26 = 64 64 – 2 – 27 = 35 assignable addresses: 62, unused addresses: 35

part b: 25 = 32 32 – 2 – 27 = 3 assignable addresses: 30, unused addresses: 3

*Scenario B*

1. we start by converting the first subnet’s network address into binary:

01000000.01011011.11100000.00000000

Now isolating our 23 bits network identifier we arrive at:

01000000.01011011.1110000

Now because we are using fixed length subnet masks, we know that the second subnet will use the same number of bits for its network identifier, but we must increment the last bit in the first subnet’s network identifier in order to arrive at the second subnet’s network identifier.

01000000.01011011.1110001

So, our second subnet has a network address represented by the CIDR entry:

64.91.226.0/23

1. first, we need to determine how many host bits we need to support the 75 hosts needed in our second subnet

We would need 7 hosts bits as 27 = 128

Now looking at the network identifier for our first subnet

01000000.01011011.1110000

We need to increment the network identifier of the first subnet to arrive at the next available location for our second subnet.

01000000.01011011.1110001

To use VLSM, we now need to add 2 additional bits to our subnet identifier:

01000000.01011011.11100010.0

We convert this to dotted decimal to arrive at our network address in CIDR entry:

64.91.226.0/25

1. part a: 32 – 23 = 9 29 = 512 512 – 2 – 390 = 120 assignable addresses: 510, unused addresses: 120

part b: 27 = 128 128 – 2 – 75 = 51 assignable addresses: 126, unused addresses: 51

*Scenario C*

1. we start by converting the first subnet’s network address into binary:

01100010.10001000.11110001.10000000

Now isolating our 27 bits network identifier we arrive at:

01100010.10001000.11110001.100

Now because we are using fixed length subnet masks, we know that the second subnet will use the same number of bits for its network identifier, but we must increment the last bit in the first subnet’s network identifier in order to arrive at the second subnet’s network identifier.

01100010.10001000.11110001.101

So, our second subnet has a network address represented by the CIDR entry:

98.136.241.160/27

1. first, we need to determine how many host bits we need to support the 18 hosts needed in our second subnet

We would need 5 hosts bits as 25 = 32

Now looking at the network identifier for our first subnet

01100010.10001000.11110001.100

We need to increment the network identifier of the first subnet to arrive at the next available location for our second subnet.

01100010.10001000.11110001.101

To use VLSM, we now don’t need to add additional bits to our subnet identifier:

01100010.10001000.11110001.101

We convert this to dotted decimal to arrive at our network address in CIDR entry:

98.136.241.160/27

1. Part a: 32 – 27 = 5 25 = 32 32 – 2 – 18 = 12 assignable addresses: 30, unused addresses: 12

Part b: 25 = 32 32 – 2 – 18 = 12 assignable addresses: 30, unused addresses: 12

7. Explanation of the effectiveness of using VLSMs for scenarios A, B, and C in number 6.

*Scenario A*  
Yes, use the VLSM significantly reduce the number of unused addresses. Because the second subnet need 27 hosts and the first need 45 hosts, means 5 bits and 6 bits. So use the VLSM can reduce the number of unused addresses.

*Scenario B*  
Yes, use the VLSM significantly reduce the number of unused addresses. Because the second subnet need 75 hosts and the first need 390 hosts, means 7 bits and 9 bits. So use the VLSM can reduce the number of unused addresses.  
  
*Scenario C*  
No, use the VLSM doesn’t reduce the number of unused addresses. Because the first subnet needs 25 host and the second need 18 hosts, means 5 bits and 5 bits. So use each method can cause the same result.

9. Calculations and results of the allocations using VLSMs. Make sure to show your work.

First subnet: 29 = 512 32 – 15 – 9 = 8

Second subnet: 27 = 128 32 – 15 – 7 = 10

b.

first subnet: 15 + 8 = 23 11111111 11111111 11111110 00000000

second subnet: 15 + 10 = 25 11111111 11111111 11111111 10000000

c.

first subnet: 29 - 2 = 510

second subnet: 27 – 2 = 126

d.

first subnet: 237.118.0.0

second subnet: 237.118.2.0

e.

first subnet: 237.118.1.255

second subnet: 237.118.2.127

f.

first subnet: 237.118.0.1 – 237.118.1.254

second subnet: 237.118.2.1 – 237.118.2.126

Your lab submission will be evaluated according to the following rubric.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Letter Grade** | **Qualities Demonstrated by the Lab Submission** | **Grade Assigned** |
| **Answers and Methodology**  **Measures the correctness and completeness of the answers and methodology used for lab steps** | A+ 🡺 100 | The answers, and answer justifications where required, are entirely complete and correct for all steps. The methodologies used to derive the answers are entirely applicable to the given problems, and are implemented correctly, for all steps. There are absolutely no technical or other errors present. |  |
| A 🡺 96 | One insignificant technical or other error is present, but otherwise the answers, and answer justifications where required, are entirely complete and correct for all steps. Excluding the insignificant error, the methodologies used to derive the answers are entirely applicable to the given problems, and are implemented correctly, for all steps. |
| A- 🡺 92 | One or two technical or other errors are present, but otherwise the answers, and answer justifications where required, are entirely complete and correct for all steps. Excluding the one or two errors, the methodologies used to derive the answers are entirely applicable to the given problems, and are implemented correctly, for all steps. |
| B+ 🡺 88 | The answers, and answer justifications where required, are complete and correct for most steps. Likewise, the methodologies used to derive the answers are applicable to the given problems, and are implemented correctly, for most steps. |
| B 🡺 85 | The answers are correct or almost correct for most steps. Some answer justifications may be missing or incorrect, but most are present and correct where required. The methodologies used to derive the answers are applicable and implemented correctly for most steps. |
| B- 🡺 82 | The answers, and answer justifications where required, are complete and correct for about ¾ of the steps. Likewise, the methodologies used to derive the answers are applicable to the given problems, and are implemented correctly, for about ¾ of the steps. |
| C+ 🡺 78 | The answers are correct or almost correct for about ¾ of the steps. Some answer justifications may be missing or incorrect. The methodologies used to derive the answers are applicable to the given problems, and are implemented correctly, for about ¾ of the steps. |
| C 🡺 75 | The answers for about half of the steps are either missing or incorrect. Likewise, the methodologies used for about half of the steps are either inapplicable to the given problem, or are implemented incorrectly. Some answer justifications are missing or incorrect where required. |
| C- 🡺 72 | The answers for most of the steps are either missing or incorrect. Likewise, the methodologies used for most of the steps are either inapplicable to the given problem, or are implemented incorrectly. Some answer justifications are missing or incorrect where required. |
| D 🡺 67 | The answers for almost all of the steps are either missing or incorrect. Likewise, the methodologies used for almost all of the steps are either inapplicable to the given problem, or are implemented incorrectly. Some answer justifications are missing or incorrect where required. |
| F 🡺 0 | The answers for virtually all of the steps are either missing or incorrect. Likewise, the methodologies used for virtually all of the steps are either inapplicable to the given problem, or are implemented incorrectly. Some or all answer justifications are missing or incorrect where required. |

Use the **Ask the Facilitators Discussion Board** if you have any questions regarding how to approach this lab.

Save your assignment as ***lastnameFirstname\_lab4.doc*** and submit it in the *Assignments* section of the course.

For help uploading files please refer to the *Technical Support* page in the syllabus.

* + Use ALTER TABLE statements to add the foreign key constraints to the tables after all of them have been created.