TCP/IP Cheat Sheet
A Free Study Guide by Boson Software, LLC

Table 1

<table>
<thead>
<tr>
<th>Class</th>
<th>Number of Subnets</th>
<th>Number of Hosts per Subnet</th>
<th>Range of Network IDs (First Octet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>126</td>
<td>16,777,214</td>
<td>1 – 126</td>
</tr>
<tr>
<td>Class B</td>
<td>16,384</td>
<td>65,534</td>
<td>128 – 191</td>
</tr>
<tr>
<td>Class C</td>
<td>2,097,152</td>
<td>254</td>
<td>192 – 223</td>
</tr>
</tbody>
</table>

The First-Octet Rule: High-Order Bits Determine the Address Class

An octet is a binary number of 8 bits, with the lowest possible number being 00000000 and the highest possible number being 11111111, or \(2^8\). The binary number 11111111, or \(2^8\), converts to 255 in decimal format.

The decimal number in the first octet, or the high-order bits, represents the network ID for the address, and that network ID remains the same for all hosts on that network segment. High-order bits, which are also called the most significant bits, are those in the left-most portion of the binary address. The remaining octets act as host IDs. Address classes are categorized based on the decimal number in the first octet, as seen in Table 1, the Address Class Summary table.

Table 2

<table>
<thead>
<tr>
<th>Class</th>
<th>Binary Start</th>
<th>Binary End</th>
<th>First Octet</th>
<th>Most Significant Bits</th>
<th>Special Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>00000001</td>
<td>01111111</td>
<td>1 – 126</td>
<td>0</td>
<td>Assignable</td>
</tr>
<tr>
<td>Class B</td>
<td>10000000</td>
<td>10111111</td>
<td>128 – 191</td>
<td>10</td>
<td>Assignable</td>
</tr>
<tr>
<td>Class C</td>
<td>11000000</td>
<td>11011111</td>
<td>192 – 223</td>
<td>110</td>
<td>Assignable</td>
</tr>
<tr>
<td>Class D</td>
<td>11100000</td>
<td>11101111</td>
<td>224 – 239</td>
<td>1110</td>
<td>Multicast</td>
</tr>
<tr>
<td>Class E</td>
<td>11110000</td>
<td>11110111</td>
<td>240 – 255</td>
<td>11110</td>
<td>InterNIC</td>
</tr>
</tbody>
</table>

Table 3

<table>
<thead>
<tr>
<th>Class</th>
<th>Total Number of Subnets Per Class</th>
<th>Equation</th>
<th>Subnets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>01111111.00000000.00000000.00000000</td>
<td>(2^7 - 2 = 126)</td>
<td>Net.H.H.H</td>
</tr>
<tr>
<td>Class B</td>
<td>10111111.11111111.00000000.00000000</td>
<td>(2^{10} - 2 = 16,384)</td>
<td>Net.Net.H.H</td>
</tr>
<tr>
<td>Class C</td>
<td>11011111.11111111.11111111.00000000</td>
<td>(2^{24} - 2 = 2,097,152)</td>
<td>Net.Net.Net.H</td>
</tr>
</tbody>
</table>

Table 4

<table>
<thead>
<tr>
<th>Class</th>
<th>Total Number of Hosts Per Subnet</th>
<th>Total Number of Hosts Per Subnet</th>
<th>Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>00000000.11111111.11111111.11111111</td>
<td>(2^{24} - 2 = 16,777,214)</td>
<td>x.255.255.255</td>
</tr>
<tr>
<td>Class B</td>
<td>00000000.00000000.11111111.11111111</td>
<td>(2^{16} - 2 = 65,534)</td>
<td>x.y.255.255</td>
</tr>
<tr>
<td>Class C</td>
<td>00000000.00000000.00000000.11111111</td>
<td>(2^8 - 2 = 254)</td>
<td>x.y.z.255</td>
</tr>
</tbody>
</table>
Table 5

<table>
<thead>
<tr>
<th>Netblock</th>
<th>Special Use</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.x.x.x</td>
<td>Private</td>
<td>RFC 1918</td>
</tr>
<tr>
<td>127.x.x.x</td>
<td>Loopback</td>
<td>Diagnostics</td>
</tr>
<tr>
<td>172.(16 – 31).x.x</td>
<td>Private</td>
<td>RFC 1918</td>
</tr>
<tr>
<td>192.0.0.x</td>
<td>Reserved</td>
<td>JBP</td>
</tr>
<tr>
<td>192.0.1.x</td>
<td>Backbone – Test – C</td>
<td>RH6</td>
</tr>
<tr>
<td>192.0.2.x</td>
<td>Internet – Test – C</td>
<td>JBP</td>
</tr>
<tr>
<td>192.0.(3 – 255).x</td>
<td>Unassigned</td>
<td>NIC</td>
</tr>
<tr>
<td>192.1.(0 – 1).x</td>
<td>Backbone Local Nets</td>
<td>SGC</td>
</tr>
<tr>
<td>192.1.3.x</td>
<td>Backbone Apollo Nets</td>
<td>SGC</td>
</tr>
<tr>
<td>192.168.x.x</td>
<td>Private</td>
<td>RFC 1918</td>
</tr>
</tbody>
</table>

Reserved Address Space
*RFC 1166 & 1918 = Private Address Space (Internal Use Only)*

Converting Decimal Into Binary

1. Create a chart with eight columns.
2. In the first row, enter 1 in the far right column. Moving to the left, double the preceding value to enter 2 in the next column, 4 in the next column, 8 in the next column, etc.
3. Continue in that manner until all columns in the first row contain a value. The column on the far left should contain the value 128. These values are called weighted values.
4. Beginning with the left-most weighted value, subtract each weighted value from the decimal being converted to binary until you reach the largest weighted value that can be subtracted without yielding a negative result. Enter a 1 as the binary value for that weighted value, and enter a 0 as the binary values for the weighted values that yielded negative results.
5. After reaching the largest weighted value that can be subtracted from the decimal without yielding a negative result, subtract the next weighted value from the remainder of the previous weighted value, not from the decimal. If a subsequent weighted value yields a negative result, enter a 0 as the binary value and subtract the next weighted value from the remainder yielded by the last successful subtraction (Note: See Weighted Values 2 and 1 in Table 6 below for an example).
6. Table 6 below illustrates the conversion of the decimal 29 into binary:

<table>
<thead>
<tr>
<th>Weighted Values</th>
<th>Subtraction from Decimal</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>29 – 128 = –99</td>
<td>0</td>
</tr>
<tr>
<td>64</td>
<td>29 – 64 = –35</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>29 – 32 = –3</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>29 – 16 = 13</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>13 – 8 = 5</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5 – 4 = 1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1 – 2 = –1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1 – 1 = 0</td>
<td>1</td>
</tr>
</tbody>
</table>

7. The decimal 29 converts into binary as 00011101.

Converting Binary Into Decimal

1. Create a chart with eight columns.
2. In the first row, enter 1 in the far right column. Moving to the left, double the preceding value to enter 2 in the next column, 4 in the next column, 8 in the next column, etc.
3. Continue in that manner until all columns in the first row contain a value. The column on the far left should contain the value 128. These values are called weighted values.
4. Enter the binary value in the chart below the weighted values so that there is one 0 or one 1 below each weighted value.
5. Add together all weighted values that have 1s below them.
6. Table 7 below illustrates the conversion of the binary 11000011 into decimal:

<table>
<thead>
<tr>
<th>Weighted Values</th>
<th>Most Significant Bit</th>
<th>Least Significant Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>128</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Binary</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

7. The binary 11000011 converts into the decimal 195 (128 + 64 + 2 + 1 = 195).

### Subnetting

Subnetting allows a large network block to be logically subdivided into multiple smaller networks, or subnets. The use of multiple smaller networks allows the use of varying physical networks, such as Ethernet or Token Ring, which could not otherwise be combined. Additionally, the smaller subnets can improve the speed of traffic and permit easier management.

Subnet masks provide the logical segmentation required by routers to be able to address logical subnets. A default gateway IP address on a router interface allows clients to access networks outside their local subnets. The number of bits used in a subnet mask determines the number of subnets available.

### Things to Remember

- Subnet masks cannot contain all 0s or all 1s in either the host portion or the network portion of the address.
- Subnet Zero, which contains all 0s, can be used sometimes. It is not recommended.
- All 1s Subnet, which contains all 1s, can never be used.
- The incremental value is the IP host address starting point for the subnet.
- Each incremental value represents another logical subnet.
- Routers only use the network ID and the subnet broadcast.
- Clients only use the IP address on the clients’ local subnets and the clients’ subnet broadcasts.
- Each bit represents a power of 2.

### Table 8

<table>
<thead>
<tr>
<th>Binary Mask</th>
<th>Decimal Mask</th>
<th>Binary Hosts</th>
<th>Bits Used</th>
<th>Number of Subnets</th>
<th>Valid Host Increments</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>0</td>
<td>00000000</td>
<td>0</td>
<td>Net ID</td>
<td>Not a subnet</td>
</tr>
<tr>
<td>10000000</td>
<td>128*</td>
<td>00000001</td>
<td>1</td>
<td>1</td>
<td>128*</td>
</tr>
<tr>
<td>11000000</td>
<td>192</td>
<td>00000111</td>
<td>2</td>
<td>2</td>
<td>64</td>
</tr>
<tr>
<td>11100000</td>
<td>224</td>
<td>00001111</td>
<td>3</td>
<td>6</td>
<td>32</td>
</tr>
<tr>
<td>11110000</td>
<td>240</td>
<td>00011111</td>
<td>4</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>11111000</td>
<td>248</td>
<td>00111111</td>
<td>5</td>
<td>30</td>
<td>8</td>
</tr>
<tr>
<td>11111100</td>
<td>252</td>
<td>01111111</td>
<td>6</td>
<td>62</td>
<td>4</td>
</tr>
<tr>
<td>11111110</td>
<td>254*</td>
<td>01111111</td>
<td>7</td>
<td>126</td>
<td>2*</td>
</tr>
<tr>
<td>11111111</td>
<td>255*</td>
<td>11111111</td>
<td>8</td>
<td>254</td>
<td>1*</td>
</tr>
</tbody>
</table>

* = Seven subnets are valid for Class A or B in the first octet. Class C has five valid subnets; the last two subnets are each a binary of 1.
Logical Addressing

All subnets can be categorized into seven ranges with three classes each (Class A, Class B, and Class C).

### Table 9: Class A

<table>
<thead>
<tr>
<th>Maximum Number of Subnets</th>
<th>Calculation for Host ID</th>
<th>Maximum Number of Hosts per Subnet</th>
<th>Subnet Mask</th>
<th>Subnet Bits Required</th>
<th>Host Bits Available</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(2^{24} - 2)</td>
<td>16,777,214</td>
<td>255.0.0.0</td>
<td>0</td>
<td>24</td>
<td>Begin Class A</td>
</tr>
<tr>
<td>1</td>
<td>(2^{23} - 2)</td>
<td>Invalid</td>
<td>128 = Subnet 0</td>
<td>1</td>
<td>23</td>
<td>Not Recommended</td>
</tr>
<tr>
<td>2</td>
<td>(2^{22} - 2)</td>
<td>4,194,302</td>
<td>255.192.0.0</td>
<td>2</td>
<td>22</td>
<td>Range 1/7</td>
</tr>
<tr>
<td>6</td>
<td>(2^{21} - 2)</td>
<td>2,097,150</td>
<td>255.224.0.0</td>
<td>3</td>
<td>21</td>
<td>Range 2/7</td>
</tr>
<tr>
<td>14</td>
<td>(2^{20} - 2)</td>
<td>1,048,574</td>
<td>255.240.0.0</td>
<td>4</td>
<td>20</td>
<td>Range 3/7</td>
</tr>
<tr>
<td>30</td>
<td>(2^{19} - 2)</td>
<td>524,286</td>
<td>255.248.0.0</td>
<td>5</td>
<td>19</td>
<td>Range 4/7</td>
</tr>
<tr>
<td>62</td>
<td>(2^{18} - 2)</td>
<td>262,142</td>
<td>255.252.0.0</td>
<td>6</td>
<td>18</td>
<td>Range 5/7</td>
</tr>
<tr>
<td>126</td>
<td>(2^{17} - 2)</td>
<td>131,070</td>
<td>255.254.0.0</td>
<td>7</td>
<td>17</td>
<td>Range 6/7</td>
</tr>
</tbody>
</table>

### Table 10: Class B

<table>
<thead>
<tr>
<th>Maximum Number of Subnets</th>
<th>Calculation for Host ID</th>
<th>Maximum Number of Hosts per Subnet</th>
<th>Subnet Mask</th>
<th>Subnet Bits Required</th>
<th>Host Bits Available</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(2^{16} - 2)</td>
<td>65,534</td>
<td>255.0.0.0</td>
<td>8</td>
<td>16</td>
<td>Class A Range 7/7 &amp; Begin Class B</td>
</tr>
<tr>
<td>1</td>
<td>(2^{15} - 2)</td>
<td>Invalid</td>
<td>128 = Subnet 0</td>
<td>9</td>
<td>15</td>
<td>Not Recommended</td>
</tr>
<tr>
<td>2</td>
<td>(2^{14} - 2)</td>
<td>16,382</td>
<td>255.255.192.0</td>
<td>10</td>
<td>14</td>
<td>Range 1/7</td>
</tr>
<tr>
<td>6</td>
<td>(2^{13} - 2)</td>
<td>8,190</td>
<td>255.255.224.0</td>
<td>11</td>
<td>13</td>
<td>Range 2/7</td>
</tr>
<tr>
<td>14</td>
<td>(2^{12} - 2)</td>
<td>4,094</td>
<td>255.255.240.0</td>
<td>12</td>
<td>12</td>
<td>Range 3/7</td>
</tr>
<tr>
<td>30</td>
<td>(2^{11} - 2)</td>
<td>2,046</td>
<td>255.255.240.0</td>
<td>13</td>
<td>11</td>
<td>Range 4/7</td>
</tr>
<tr>
<td>62</td>
<td>(2^{10} - 2)</td>
<td>1,022</td>
<td>255.255.254.0</td>
<td>14</td>
<td>10</td>
<td>Range 5/7</td>
</tr>
<tr>
<td>126</td>
<td>(2^9 - 2)</td>
<td>510</td>
<td>255.255.254.0</td>
<td>15</td>
<td>9</td>
<td>Range 6/7</td>
</tr>
</tbody>
</table>

### Table 11: Class C

<table>
<thead>
<tr>
<th>Maximum Number of Subnets</th>
<th>Calculation for Host ID</th>
<th>Maximum Number of Hosts per Subnet</th>
<th>Subnet Mask</th>
<th>Subnet Bits Required</th>
<th>Host Bits Available</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(2^8 - 2)</td>
<td>254</td>
<td>255.255.255.0</td>
<td>16</td>
<td>8</td>
<td>Class B Range 7/7 &amp; Begin Class C</td>
</tr>
<tr>
<td>1</td>
<td>(2^7 - 2)</td>
<td>Invalid</td>
<td>128 = Subnet 0</td>
<td>17</td>
<td>7</td>
<td>Not Recommended</td>
</tr>
<tr>
<td>2</td>
<td>(2^6 - 2)</td>
<td>62</td>
<td>255.255.255.192</td>
<td>18</td>
<td>6</td>
<td>Range 1/5</td>
</tr>
<tr>
<td>6</td>
<td>(2^5 - 2)</td>
<td>30</td>
<td>255.255.255.224</td>
<td>19</td>
<td>5</td>
<td>Range 2/5</td>
</tr>
<tr>
<td>14</td>
<td>(2^4 - 2)</td>
<td>14</td>
<td>255.255.255.240</td>
<td>20</td>
<td>4</td>
<td>Range 3/5</td>
</tr>
<tr>
<td>30</td>
<td>(2^3 - 2)</td>
<td>6</td>
<td>255.255.255.248</td>
<td>21</td>
<td>3</td>
<td>Range 4/5</td>
</tr>
<tr>
<td>62</td>
<td>(2^2 - 2)</td>
<td>2</td>
<td>255.255.255.252</td>
<td>22</td>
<td>2</td>
<td>Range 5/5</td>
</tr>
</tbody>
</table>
**Things to Remember**

- The octets that contain only 1s or only 0s do not need additional calculation.
- If there are only 1s, the decimal value is 255; if there are only 0s, the decimal value is 0.
- Calculation is only required for the octet that has been broken at the bit boundary.

### Class C Example Breakdown

<table>
<thead>
<tr>
<th>Subnet</th>
<th>Octet</th>
<th>Range</th>
<th>Incremental Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network (Subnet 0 – Do not use)</td>
<td>00/000000</td>
<td>0</td>
<td>64</td>
<td>Net ID (goes in routing table)</td>
</tr>
<tr>
<td></td>
<td>00/00001</td>
<td>1</td>
<td></td>
<td>first host ID in Subnet 0</td>
</tr>
<tr>
<td></td>
<td>00/11110</td>
<td>62</td>
<td></td>
<td>last host ID in Subnet 0</td>
</tr>
<tr>
<td></td>
<td>00/11111</td>
<td>63</td>
<td></td>
<td>broadcast for only Subnet 0</td>
</tr>
<tr>
<td>Subnet 1 of 4</td>
<td>01/000000</td>
<td>64</td>
<td>64</td>
<td>Net ID (goes in routing table)</td>
</tr>
<tr>
<td></td>
<td>01/00001</td>
<td>65</td>
<td></td>
<td>first host ID for Subnet 1</td>
</tr>
<tr>
<td></td>
<td>01/11110</td>
<td>126</td>
<td></td>
<td>last host ID for Subnet 1</td>
</tr>
<tr>
<td></td>
<td>01/11111</td>
<td>127</td>
<td></td>
<td>broadcast for only Subnet 1</td>
</tr>
<tr>
<td>Subnet 2 of 4</td>
<td>10/000000</td>
<td>128</td>
<td>64</td>
<td>Net ID (goes in routing table)</td>
</tr>
<tr>
<td></td>
<td>10/00001</td>
<td>129</td>
<td></td>
<td>first host ID for Subnet 2</td>
</tr>
<tr>
<td></td>
<td>10/11110</td>
<td>190</td>
<td></td>
<td>last host ID for Subnet 2</td>
</tr>
<tr>
<td></td>
<td>10/11111</td>
<td>191</td>
<td></td>
<td>broadcast for only Subnet 2</td>
</tr>
<tr>
<td>Broadcast Subnet (Do not use)</td>
<td>11/000000</td>
<td>192</td>
<td>64</td>
<td>Net ID (goes in routing table)</td>
</tr>
<tr>
<td></td>
<td>11/00001</td>
<td>193</td>
<td></td>
<td>first host ID in All 1s subnet</td>
</tr>
<tr>
<td></td>
<td>11/11110</td>
<td>254</td>
<td></td>
<td>last host ID in All 1s subnet</td>
</tr>
<tr>
<td></td>
<td>11/11111</td>
<td>255</td>
<td></td>
<td>Local wire (all subnets) broadcast</td>
</tr>
<tr>
<td>Network (Subnet 0 – Do not use)</td>
<td>000/000000</td>
<td>0</td>
<td>32</td>
<td>Net ID (goes in routing table)</td>
</tr>
<tr>
<td></td>
<td>000/00001</td>
<td>1</td>
<td></td>
<td>first host ID in Subnet 0</td>
</tr>
<tr>
<td></td>
<td>000/11110</td>
<td>30</td>
<td></td>
<td>last host ID in Subnet 0</td>
</tr>
<tr>
<td></td>
<td>000/11111</td>
<td>31</td>
<td></td>
<td>broadcast for only Subnet 0</td>
</tr>
<tr>
<td>Subnet 1 of 8</td>
<td>001/00000</td>
<td>32</td>
<td></td>
<td>Net ID (goes in routing table)</td>
</tr>
<tr>
<td></td>
<td>001/00001</td>
<td>33</td>
<td></td>
<td>first host ID for Subnet 1</td>
</tr>
<tr>
<td></td>
<td>001/11110</td>
<td>62</td>
<td></td>
<td>last host ID for Subnet 1</td>
</tr>
<tr>
<td></td>
<td>001/11111</td>
<td>63</td>
<td></td>
<td>broadcast for only Subnet 1</td>
</tr>
<tr>
<td>Subnet 2 of 8</td>
<td>010/00000</td>
<td>64</td>
<td>32</td>
<td>Net ID (goes in routing table)</td>
</tr>
<tr>
<td></td>
<td>010/00001</td>
<td>65</td>
<td></td>
<td>first host ID for Subnet 2</td>
</tr>
<tr>
<td></td>
<td>010/11110</td>
<td>94</td>
<td></td>
<td>last host ID for Subnet 2</td>
</tr>
<tr>
<td></td>
<td>010/11111</td>
<td>95</td>
<td></td>
<td>broadcast for only Subnet 2</td>
</tr>
<tr>
<td>Subnet 3 of 8</td>
<td>011/00000</td>
<td>96</td>
<td>32</td>
<td>Net ID (goes in routing table)</td>
</tr>
<tr>
<td></td>
<td>011/00001</td>
<td>97</td>
<td></td>
<td>first host ID for Subnet 3</td>
</tr>
<tr>
<td></td>
<td>011/11110</td>
<td>126</td>
<td></td>
<td>last host ID for Subnet 3</td>
</tr>
<tr>
<td></td>
<td>011/11111</td>
<td>127</td>
<td></td>
<td>broadcast for only Subnet 3</td>
</tr>
<tr>
<td>Subnet 4 of 8</td>
<td>100/00000</td>
<td>128</td>
<td>32</td>
<td>Net ID (goes in routing table)</td>
</tr>
<tr>
<td></td>
<td>100/00001</td>
<td>129</td>
<td></td>
<td>first host ID for Subnet 4</td>
</tr>
<tr>
<td></td>
<td>100/11110</td>
<td>158</td>
<td></td>
<td>last host ID for Subnet 4</td>
</tr>
<tr>
<td></td>
<td>100/11111</td>
<td>159</td>
<td></td>
<td>broadcast for only Subnet 4</td>
</tr>
<tr>
<td>Subnet 5 of 8</td>
<td>101/00000</td>
<td>160</td>
<td>32</td>
<td>Net ID (goes in routing table)</td>
</tr>
<tr>
<td></td>
<td>101/00001</td>
<td>161</td>
<td></td>
<td>first host ID for Subnet 5</td>
</tr>
<tr>
<td></td>
<td>101/11110</td>
<td>190</td>
<td></td>
<td>last host ID for Subnet 5</td>
</tr>
<tr>
<td></td>
<td>101/11111</td>
<td>191</td>
<td></td>
<td>broadcast for only Subnet 5</td>
</tr>
</tbody>
</table>
Subnet 6 of 8

<table>
<thead>
<tr>
<th>Subnet Number</th>
<th>Net ID (goes in routing table)</th>
<th>Host ID</th>
<th>Last Host ID</th>
<th>Broadcast for only Subnet 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>110/00000</td>
<td>192</td>
<td>192</td>
<td>192</td>
<td>192</td>
</tr>
<tr>
<td>110/00001</td>
<td>193</td>
<td>222</td>
<td>222</td>
<td>222</td>
</tr>
<tr>
<td>110/11110</td>
<td>222</td>
<td>223</td>
<td>223</td>
<td>223</td>
</tr>
<tr>
<td>110/11111</td>
<td>223</td>
<td>223</td>
<td>223</td>
<td>223</td>
</tr>
</tbody>
</table>

Broadcast Subnet (Do not use)

<table>
<thead>
<tr>
<th>Subnet Number</th>
<th>Net ID (goes in routing table)</th>
<th>Host ID</th>
<th>Last Host ID</th>
<th>Broadcast for all Subnet 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>111/00000</td>
<td>224</td>
<td>224</td>
<td>224</td>
<td>224</td>
</tr>
<tr>
<td>110/00001</td>
<td>225</td>
<td>225</td>
<td>225</td>
<td>225</td>
</tr>
<tr>
<td>110/11110</td>
<td>254</td>
<td>254</td>
<td>254</td>
<td>254</td>
</tr>
<tr>
<td>111/11111</td>
<td>255</td>
<td>255</td>
<td>255</td>
<td>255</td>
</tr>
</tbody>
</table>

**Power of 2 Shortcuts for Calculating Subnets**

**Networks Gained from Using Host Bits**

Where \( n \) is equal to the number of host bits used for subnets, \( 2^n - 2 \) = the total number of networks gained.

**Hosts Available after Subnetting**

Where \( h \) is equal to the number of host bits remaining after subnetting, \( 2^h - 2 \) = the total number of hosts available.

**Example**

You need to subnet the Class C network address 193.5.15.0 to obtain 5 additional networks, and you need to find out how many hosts will be available after subnetting.

There are 8 host bits for a Class C address. Beginning with 1 host bit, calculate the number of networks that would be gained until you find a number of host bits that yields at least five networks.

1 host bit: \( 2^1 - 2 = 1 \) network
2 host bits: \( 2^2 - 2 = 2 \) networks
3 host bits: \( 2^3 - 3 = 6 \) networks

From the calculations, you can see that 2 host bits will only yield 2 networks; you would not obtain the 5 networks that you need. However, 3 host bits will yield 6 networks, which will provide the 5 networks that you need.

Of the 8 host bits for a Class C address, you have allocated 3 of those for subnetting, thus leaving 5 host bits available for their original purpose. As a result, \( 2^5 - 2 \), or 30, hosts can be supported on each subnet.
### Class A
(24 bits available for subnetting)

<table>
<thead>
<tr>
<th>Bits Required (n)</th>
<th>Subnets ((2^n - 2))</th>
<th>Hosts Per Subnet ((2^h - 2))</th>
<th>Slash Notation</th>
<th>Mask</th>
<th>Subnet Slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(2^1 - 2 = 0)</td>
<td>(2^{15} - 2 = 8,388,606)</td>
<td>/9</td>
<td>255.128.0.0</td>
<td>Subnet 0</td>
</tr>
<tr>
<td>2</td>
<td>(2^2 - 2 = 2)</td>
<td>(2^{22} - 2 = 4,194,302)</td>
<td>/10</td>
<td>255.192.0.0</td>
<td>Subnet 1/7</td>
</tr>
<tr>
<td>3</td>
<td>(2^3 - 2 = 6)</td>
<td>(2^{21} - 2 = 2,097,150)</td>
<td>/11</td>
<td>255.224.0.0</td>
<td>Subnet 2/7</td>
</tr>
<tr>
<td>4</td>
<td>(2^4 - 2 = 14)</td>
<td>(2^{20} - 2 = 1,048,574)</td>
<td>/12</td>
<td>255.240.0.0</td>
<td>Subnet 3/7</td>
</tr>
<tr>
<td>5</td>
<td>(2^5 - 2 = 30)</td>
<td>(2^{19} - 2 = 524,286)</td>
<td>/13</td>
<td>255.248.0.0</td>
<td>Subnet 4/7</td>
</tr>
<tr>
<td>6</td>
<td>(2^6 - 2 = 62)</td>
<td>(2^{18} - 2 = 262,142)</td>
<td>/14</td>
<td>255.252.0.0</td>
<td>Subnet 5/7</td>
</tr>
<tr>
<td>7*</td>
<td>(2^7 - 2 = 126)</td>
<td>(2^{17} - 2 = 131,070)</td>
<td>/15</td>
<td>255.254.0.0</td>
<td>Subnet 6/7</td>
</tr>
<tr>
<td>8*</td>
<td>(2^8 - 2 = 254)</td>
<td>(2^{16} - 2 = 65,534)</td>
<td>/16</td>
<td>255.255.0.0</td>
<td>Subnet 7/7</td>
</tr>
</tbody>
</table>

* = 7 subnets valid for Class A

### Class B
(16 bits available for subnetting)

<table>
<thead>
<tr>
<th>Bits Required (n)</th>
<th>Subnets ((2^n - 2))</th>
<th>Hosts Per Subnet ((2^h - 2))</th>
<th>Slash Notation</th>
<th>Mask</th>
<th>Subnet Slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(2^1 - 2 = 0)</td>
<td>(2^{15} - 2 = 32,766)</td>
<td>/17</td>
<td>255.128.0.0</td>
<td>Subnet 0</td>
</tr>
<tr>
<td>2</td>
<td>(2^2 - 2 = 2)</td>
<td>(2^{14} - 2 = 16,382)</td>
<td>/18</td>
<td>255.192.0.0</td>
<td>Subnet 1/7</td>
</tr>
<tr>
<td>3</td>
<td>(2^3 - 2 = 6)</td>
<td>(2^{13} - 2 = 8,190)</td>
<td>/19</td>
<td>255.224.0.0</td>
<td>Subnet 2/7</td>
</tr>
<tr>
<td>4</td>
<td>(2^4 - 2 = 14)</td>
<td>(2^{12} - 2 = 4,094)</td>
<td>/20</td>
<td>255.240.0.0</td>
<td>Subnet 3/7</td>
</tr>
<tr>
<td>5</td>
<td>(2^5 - 2 = 30)</td>
<td>(2^{11} - 2 = 2,046)</td>
<td>/21</td>
<td>255.248.0.0</td>
<td>Subnet 4/7</td>
</tr>
<tr>
<td>6</td>
<td>(2^6 - 2 = 62)</td>
<td>(2^{10} - 2 = 1,022)</td>
<td>/22</td>
<td>255.252.0.0</td>
<td>Subnet 5/7</td>
</tr>
<tr>
<td>7*</td>
<td>(2^7 - 2 = 126)</td>
<td>(2^{9} - 2 = 510)</td>
<td>/23</td>
<td>255.254.0.0</td>
<td>Subnet 6/7</td>
</tr>
<tr>
<td>8*</td>
<td>(2^8 - 2 = 254)</td>
<td>(2^{8} - 2 = 254)</td>
<td>/24</td>
<td>255.255.0.0</td>
<td>Subnet 7/7</td>
</tr>
</tbody>
</table>

* = 7 subnets valid for Class B

### Class C
(8 bits available for subnetting)

<table>
<thead>
<tr>
<th>Bits Required (n)</th>
<th>Subnets ((2^n - 2))</th>
<th>Hosts Per Subnet ((2^h - 2))</th>
<th>Slash Notation</th>
<th>Mask</th>
<th>Subnet Slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(2^1 - 2 = 0)</td>
<td>(2^{7} - 2 = 126)</td>
<td>/25</td>
<td>255.255.255.128</td>
<td>Subnet 0</td>
</tr>
<tr>
<td>2</td>
<td>(2^2 - 2 = 2)</td>
<td>(2^{6} - 2 = 62)</td>
<td>/26</td>
<td>255.255.255.192</td>
<td>Subnet 1/5</td>
</tr>
<tr>
<td>3</td>
<td>(2^3 - 2 = 6)</td>
<td>(2^{5} - 2 = 30)</td>
<td>/27</td>
<td>255.255.255.224</td>
<td>Subnet 2/5</td>
</tr>
<tr>
<td>4</td>
<td>(2^4 - 2 = 14)</td>
<td>(2^{4} - 2 = 14)</td>
<td>/28</td>
<td>255.255.255.240</td>
<td>Subnet 3/5</td>
</tr>
<tr>
<td>5*</td>
<td>(2^5 - 2 = 30)</td>
<td>(2^{3} - 2 = 6)</td>
<td>/29</td>
<td>255.255.255.248</td>
<td>Subnet 4/5</td>
</tr>
<tr>
<td>6*</td>
<td>(2^6 - 2 = 62)</td>
<td>(2^{2} - 2 = 2)</td>
<td>/30</td>
<td>255.255.255.252</td>
<td>Subnet 5/5</td>
</tr>
</tbody>
</table>

* = 5 valid subnets for Class C; last 2 are binary all 1
Practice

You have the address 132.7.0.0. You need 5 subnets of equal size with 1,500 hosts each. Find the following information in the given order:

1. the number of host bits to obtain for the required number of subnets
2. the numbers of hosts available per subnet after subnetting
3. the decimal value of the new subnet mask
4. the incremental value of the subnets
5. the first host, broadcast, and last host of each subnet

1. Using powers of 2, determine the number of host bits to obtain for 5 subnets.
   • The first octet is 132, which is within the Class B range (128 – 191). Recall that each octet has 8 bits.
   • The first two octets in a Class B address may not be used for subnetting, so there are two octets, or 16 bits, available.
   • Beginning at the left side of the available bits and moving toward the right, remove one bit at a time from the host portion of the binary and calculate the number of networks gained with each bit.
     \[2^1 - 2 = 0\] (You will never use just 1 bit.)
     \[2^2 - 2 = 2\] (Not enough networks; keep trying.)
     \[2^3 - 2 = 6\] (3 bits will provide the 5 subnets and leave 1 available for future expansion.)
   • Enter those 3 bits into a weighted values chart, as shown in Table 16 below.

<table>
<thead>
<tr>
<th></th>
<th>Most Significant Bit</th>
<th>Least Significant Bit</th>
<th>Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighted Values</td>
<td>128</td>
<td>64</td>
<td>32</td>
</tr>
<tr>
<td>Binary</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

2. Using powers of 2, determine how many hosts will be available after subnetting.
   • Recall that there are 16 host bits available for a Class B address. You will use 3 of those for subnetting, thus leaving 13 host bits available to use for hosts.
     \[2^{13} - 2 = 8,190\] hosts available per subnet
   *If you prefer to have extra subnets available for future expansion over having extra host capacity, you can use additional host bits now.

3. Using the weighted values chart created in step 1, determine the new decimal value of the subnet mask.
   • Add together each weighted value with a 1 below it.
     \[128 + 64 + 32 = 224\]
   • Since this is a Class B address and the first two octets are reserved, the default subnet mask is 255.255.0.0.
   • The 3 host bits were removed from the third octet, so the new decimal mask is 255.255.224.0, or /19 in slash notation (8 bits + 8 bits + 3 bits = 19).
4. Using the decimal mask or the last subnet bit's weighted value, determine the incremental value of the subnets.
   • Subtract the decimal value calculated in step 3 from 256 to obtain the incremental value.
     
     \[ 256 - 224 = 32 \]
     
     OR
   
   • Look at Table 16 from step 1, and find the last weighted value toward the right with a 1 below it. That value, 32, is the incremental value.

5. Use the incremental value to find the first host IP, broadcast, and last host IP for each subnet
   • Begin with the incremental value, which is 32 in this scenario, and add 1 to obtain the first host IP for subnet 1.
     
     \[ 32 + 1 = 33 \] (first host IP for subnet 1)
     
     • Add the current subnet’s incremental value with the incremental value and subtract 1 to obtain the broadcast for subnet 1.
     
     \[ 32 + 32 - 1 = 63 \]
     
     • Subtract 1 from the subnet 1’s broadcast to obtain the last host IP for subnet 1.
     
     \[ 63 - 1 = 62 \]
     
     • Repeat this process for each subsequent subnet, as shown in Table 17 below.

<table>
<thead>
<tr>
<th>Range</th>
<th>Subnet</th>
<th>Incremental Value (subnet ID used by routing table)</th>
<th>First Host IP</th>
<th>Broadcast</th>
<th>Last Host IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 x 32</td>
<td>0</td>
<td>0</td>
<td>Invalid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 x 32</td>
<td>1</td>
<td>32</td>
<td>33</td>
<td>63</td>
<td>62</td>
</tr>
<tr>
<td>2 x 32</td>
<td>2</td>
<td>64</td>
<td>65</td>
<td>95</td>
<td>94</td>
</tr>
<tr>
<td>3 x 32</td>
<td>3</td>
<td>96</td>
<td>97</td>
<td>127</td>
<td>126</td>
</tr>
<tr>
<td>4 x 32</td>
<td>4</td>
<td>128</td>
<td>129</td>
<td>159</td>
<td>158</td>
</tr>
<tr>
<td>5 x 32</td>
<td>5</td>
<td>160</td>
<td>161</td>
<td>191</td>
<td>190</td>
</tr>
<tr>
<td>6 x 32</td>
<td>6</td>
<td>192</td>
<td>193</td>
<td>223</td>
<td>222</td>
</tr>
<tr>
<td>7 x 32</td>
<td>7</td>
<td>224</td>
<td>Broadcast Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes
• You cannot use either All 0s or All 1s subnets, which reduces the available subnets by 2 \((2^n - 2) = \) available subnets).
• The All 0s address is the network ID for that subnet and will be used by a router in its routing table.
• The All 1s address is the network broadcast for all subnets on that wire.
## Bit Conversions

<table>
<thead>
<tr>
<th>Bits</th>
<th>Binary</th>
<th>Decimal Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11000000</td>
<td>192</td>
</tr>
<tr>
<td>3</td>
<td>11100000</td>
<td>224</td>
</tr>
<tr>
<td>4</td>
<td>11110000</td>
<td>240</td>
</tr>
<tr>
<td>5</td>
<td>11111000</td>
<td>248</td>
</tr>
<tr>
<td>6</td>
<td>11111100</td>
<td>252</td>
</tr>
<tr>
<td>7*</td>
<td>11111110</td>
<td>254</td>
</tr>
<tr>
<td>8*</td>
<td>11111111</td>
<td>255</td>
</tr>
</tbody>
</table>

* Not valid for Class C addresses

## Subnets and Hosts Per Bit

<table>
<thead>
<tr>
<th>Bits</th>
<th>Subnets</th>
<th>Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2^2 – 2 = 2</td>
<td>2^6 – 2 = 62</td>
</tr>
<tr>
<td>3</td>
<td>2^3 – 2 = 6</td>
<td>2^5 – 2 = 30</td>
</tr>
<tr>
<td>4</td>
<td>2^4 – 2 = 14</td>
<td>2^4 – 2 = 14</td>
</tr>
<tr>
<td>5</td>
<td>2^5 – 2 = 30</td>
<td>2^3 – 2 = 6</td>
</tr>
<tr>
<td>6</td>
<td>2^6 – 2 = 62*</td>
<td>2^6 – 2 = 2</td>
</tr>
<tr>
<td>7</td>
<td>2^7 – 2 = 126*</td>
<td>2^1 – 1 = 1*</td>
</tr>
<tr>
<td>8</td>
<td>2^8 – 2 = 254*</td>
<td>2^0 – 1 = 0*</td>
</tr>
</tbody>
</table>

* Not valid for Class C addresses

## Incremental Value Per Mask

<table>
<thead>
<tr>
<th>Decimal Mask</th>
<th>Incremental Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>192</td>
<td>64</td>
</tr>
<tr>
<td>224</td>
<td>32</td>
</tr>
<tr>
<td>240</td>
<td>16</td>
</tr>
<tr>
<td>248</td>
<td>8</td>
</tr>
<tr>
<td>252</td>
<td>4</td>
</tr>
<tr>
<td>254*</td>
<td>2</td>
</tr>
<tr>
<td>255*</td>
<td>1</td>
</tr>
</tbody>
</table>

* Not valid for Class C addresses

## Address Class Ranges

<table>
<thead>
<tr>
<th>Class</th>
<th>Range</th>
<th>Network ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1 – 126</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>128 – 191</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>192 – 223</td>
<td>110</td>
</tr>
</tbody>
</table>

## Calculating Binary and Decimal Mask

<table>
<thead>
<tr>
<th>Subnets</th>
<th>Most Significant Bit</th>
<th>Least Significant Bit</th>
<th>Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighted Values</td>
<td>128</td>
<td>64</td>
<td>32</td>
</tr>
<tr>
<td>Binary</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The chart above depicts the decimal mask 255 in binary.

\[(128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255)\]

Warning: no warranty expressed or implied. Use at your own risk.

Copyright © 2010 Boson Software, LLC. All rights reserved.