CS118 Discussion, Week 6

Taqi
Outline

• Network Layer
  • IP
  • NAT
  • DHCP
• Project 2 spec
Network layer: overview

- Basic functions for network layer
  - Routing
  - Forwarding
- Connection v.s. connection-less delivery
  - circuit switch
  - packet switch
- Network layer protocols
  - Addressing and fragmentation: IPv4, IPv6
  - Routing: RIP, OSPF, BGP, DVMRP, PIM
  - Others: DHCP, ICMP, NAT
IPv4 Header

- **Header length**: 4-byte unit
- **Length**: 1-byte unit
- **Fragmentation**: id + MF/DF + offset (8-byte unit)
  - Why do we have to fragment packets?
- **TTL**: why do we need it?
- **Checksum**
  - Since TCP/UDP already has checksum, is it redundant?
  - Why is it just header checksum (not including data)?
- **Protocol**: identifies the upper layer protocol
- **Source and destination IP addresses**

---

<table>
<thead>
<tr>
<th>ver</th>
<th>header len</th>
<th>type of service</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16-bit identifier</th>
<th>flgs</th>
<th>fragment offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>time to live</th>
<th>protocol</th>
<th>header checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

32 bit source IP address

32 bit destination IP address

options (if any)

data
(variable length, typically a TCP or UDP segment)
IP fragmentation and reassembly

- MTU: maximum transmission unit
- identifier
- flag bit: MF = 0?
- offset
IP address

- Globally recognizable identifier
- IPv4: 0.0.0.0~255.255.255.255
  - Most IP addresses are globally unique
  - Exception — why?
- Network id, host id
- CIDR address
IP address classes


<table>
<thead>
<tr>
<th>Class</th>
<th>1st Octet Decimal Range</th>
<th>1st Octet High Order Bits</th>
<th>Network/Host ID (N=Network, H=Host)</th>
<th>Default Subnet Mask</th>
<th>Number of Networks</th>
<th>Hosts per Network (Usable Addresses)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1 – 126*</td>
<td>0</td>
<td>N.H.H.H</td>
<td>255.0.0.0</td>
<td>126 (2^7 – 2)</td>
<td>16,777,214 (2^24 – 2)</td>
</tr>
<tr>
<td>B</td>
<td>128 – 191</td>
<td>10</td>
<td>N.N.H.H</td>
<td>255.255.0.0</td>
<td>16,382 (2^14 – 2)</td>
<td>65,534 (2^16 – 2)</td>
</tr>
<tr>
<td>C</td>
<td>192 – 223</td>
<td>110</td>
<td>N.N.N.H</td>
<td>255.255.255.0</td>
<td>2,097,150 (2^21 – 2)</td>
<td>254 (2^8 – 2)</td>
</tr>
<tr>
<td>D</td>
<td>224 – 239</td>
<td>1110</td>
<td>Reserved for Multicasting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>240 – 254</td>
<td>1111</td>
<td>Experimental; used for research</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Private Networks</th>
<th>Subnet Mask</th>
<th>Address Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10.0.0.0</td>
<td>255.0.0.0</td>
<td>10.0.0.0 - 10.255.255.255</td>
</tr>
<tr>
<td>B</td>
<td>172.16.0.0 - 172.31.0.0</td>
<td>255.240.0.0</td>
<td>172.16.0.0 - 172.31.255.255</td>
</tr>
<tr>
<td>C</td>
<td>192.168.0.0</td>
<td>255.255.0.0</td>
<td>192.168.0.0 - 192.168.255.255</td>
</tr>
</tbody>
</table>
CIDR address

• a.b.c.d/x
  • x: # bits in network ID portion of the address
  • address: a.b.c.d, network mask: $2^{32} - 2^{(32-x)}$

CIDR  11001000 00010111 000100000 000000000

IP prefix  200.23.16.0/23

netmask  11111111 11111111 11111110 00000000

255.255.254.0
Hierarchical addressing

- subnet: a portion of addressing space
  - extend bits from the network id
  - `<network address>/<subnet mask>`
- route aggregation
Forwarding

- Longest prefix matching

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Link interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00011000 **********</td>
<td>0</td>
</tr>
<tr>
<td>11001000 00010111 00010*** **********</td>
<td>1</td>
</tr>
<tr>
<td>11001000 00010111 0001**** **********</td>
<td>2</td>
</tr>
<tr>
<td>********** ********** ********** **********</td>
<td>3</td>
</tr>
</tbody>
</table>

- Linear lookup
More on Forwarding and Routing

- Patricia tree
- MPLS
- Routing:
  - $ netstat -f inet -rn
  - Simple heuristic
NAT (network address translation)

- Depletion of IPv4 addresses — short-term solution
  - IP tunneling?
- Use private IP addresses
- Side-benefit: security
- How to achieve?
  - <public IP:port> — <private IP:port> trick
NAT: detail

- outgoing packets:
  - replace (source IP address, source port #) of every outgoing packet to (NAT IP address, new port #)
  - remote clients/servers will respond using (NAT IP address, new port #) as destination address
  - remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair

- incoming packets:
  - replace (destination NAT IP address, destination port #) of every incoming packet with corresponding (source IP address, port #) stored in NAT table
NAT: downside

- Increased complexity
- Single point of failure
- Cannot run services inside a NAT box
DHCP: Dynamic Host Configuration Protocol

- Dynamically allocates the following info to a host
  - IP address for the host
  - IP address for default router
  - Subnet mask
  - IP address for DNS caching resolver
- Allows address reuse
DHCP: operations

- Host broadcasts “DHCP discovery” msg
- DHCP server responds with “DHCP offer” msg
- Host requests IP address: “DHCP request” msg
- DHCP server sends address: “DHCP ack” msg
IP Fragmentation
MTU

- Maximum Transmission Unit (MTU)
  - Largest IP packet a network will accept
  - Arriving IP packet may be larger
IP Fragmentation

• If IP packet is longer than the MTU, the router breaks packet into smaller packets
  – Called IP fragments
  – Fragments are still IP packets
  – Earlier in Mod A, fragmentation in TCP
IP Fragmentation

- What is Fragmented?
  - Only the original data field
  - New headers are created
IP Fragmentation

• What Does the Fragmentation?
  – The router
  – Not the subnet
Multiple Fragmentations

• Original packet may be fragmented multiple times along its route
Defragmentation

- Internet layer process on destination host defragments, restoring the original packet
- IP Defragmentation only occurs once
Fragmentation and IP Fields

- **More Fragments** field (1 bit)
  - 1 if more fragments
  - 0 if not
  - Source host internet process sets to 0
  - If router fragments, sets More Fragments field in last fragment to 0
  - In all other fragments, sets to 1
**Identification Field**

- IP packet has a 16-bit *Identification* field

<table>
<thead>
<tr>
<th>Version (4)</th>
<th>Hdr Len (4)</th>
<th>TOS (8)</th>
<th>Total Length in bytes (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>Indication (16 bits)</strong></td>
<td>Flags (3)</td>
</tr>
<tr>
<td>Time to Live (8)</td>
<td>Protocol (8)</td>
<td><strong>Source IP Address</strong></td>
<td><strong>Destination IP Address</strong></td>
</tr>
<tr>
<td><strong>Options (if any)</strong></td>
<td><strong>PAD</strong></td>
<td><strong>Data Field</strong></td>
<td></td>
</tr>
</tbody>
</table>
Identification Field

- IP packet has a 16-bit *Identification* field
  - Source host internet process places a random number in the Identification field
  - Different for each IP packet

<table>
<thead>
<tr>
<th>Version (4)</th>
<th>Hdr Len (4)</th>
<th>TOS (8)</th>
<th>Total Length in bytes (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Indication (16 bits)</td>
<td>Flags (3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time to Live (8)</td>
<td>Protocol (8)</td>
</tr>
</tbody>
</table>
Identification Field

• IP packet has a 16-bit Identification field
  – If router fragments, places the original Identification field value in the Identification field of each fragment
Identification Field

• Purpose
  – Allows receiving host’s internet layer process know what fragments belong to each original packet
  – Works even if an IP packet is fragmented several times
**Fragment Offset Field**

- Fragment offset field (13 bits) is used to reorder fragments with the same Identification field
- Contains the data field’s starting point (in octets) from the start of the data field in the original IP packet

<table>
<thead>
<tr>
<th>Version (4)</th>
<th>Hdr Len (4)</th>
<th>TOS (8)</th>
<th>Total Length in bytes (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indication (16 bits)</td>
<td>Flags (3)</td>
<td>Fragment Offset (13)</td>
<td></td>
</tr>
</tbody>
</table>
Fragment Offset Field

- Receiving host’s internet layer process assembles fragments in order of increasing fragment offset field value
- This works even if fragments arrive out of order!
- Works even if fragmentation occurs multiple times
Fragmentation: Recap

• IP Fragmentation
  – Data field of a large IP packet is fragmented
  – The fragments are sent into a series of smaller IP packets fitting a network’s MTU
  – Fragmentation is *done by routers*
  – Fragmentation may be done multiple times along the route
Defragmentation: Recap

• IP Defragmentation

  – *Defragmentation* (reassembly) is *done once*, by destination host’s internet layer process
Defragmentation: Recap

- All IP packets resulting from the fragmentation of the same original IP packet have the same Identification field value

- Destination host internet process orders all IP packets from the same original on the basis of their Fragment Offset field values

- More Fragments field tells whether there are no more fragments coming
IP Addressing & Subnetting Made Easy

Developed by Peter Smith
peter.joseph.smith@tafensw.edu.au
Modified by taqi for class discussion
Working with IP Addresses

Peter Smith
Introduction

- You can probably work with **decimal numbers** much easier than with the **binary numbers** needed by the computer.
- Working with binary numbers is time-consuming & error-prone.
Octets

- The 32-bit IP address is broken up into 4 octets, which are arranged into a dotted-decimal notation scheme.
- An octet is a set of 8 bits & not a musical instrument.
- Example of an IP version 4: 
  172.64.126.52
Thinking in Binary

- The binary system uses only 2 values “0 & 1” to represent numbers in positions representing increasing powers of 2.
- We all are accustomed to thinking & working in the decimal system, which is based on the number 10.
Thinking in Binary (Cont.)

- To most humans, the number 124 represents $100 + 20 + 4$.

- To the computer, this number is 1111100, which is $64 \times 2^6 + 32 \times 2^5 + 16 \times 2^4 + 8 \times 2^3 + 4 \times 2^2 + 0 + 0$. 
Each position in a binary number represents, right to left, a power of two beginning with $2^0$ & increasing by one power as it moves left: $2^0, 2^1, 2^2, 2^4$, etc.
Converting to Decimal

- You’ll need to convert binary to decimal & vice versa to compute subnets & hosts.
- So, it’s time for a quick review lesson in binary-to-decimal conversion.
- There are 8 bits in an octet & each bit can only be a 1 or a 0.
What then do you suppose is the largest decimal number that can be expressed in an octet?

Eight 1’s (1111 1111)
Converting to Decimal (Cont.)

Now, for double the money, what is its equivalent decimal value?

<table>
<thead>
<tr>
<th></th>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

The binary number 1111 1111 converts into the decimal number:
128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255
Converting to Decimal (Cont.)

- Therefore, the largest decimal number that can be stored in an IP address octet is 255.
- The significance of this should become evident later in this presentation.
IP Address Classes

- IP addresses are divided into 5 classes, each of which is designated with the alphabetic letters A to E.
- Class D addresses are used for multicasting.
- Class E addresses are reserved for testing & some mysterious future use.
The 5 IP classes are split up based on the value in the 1st octet:

<table>
<thead>
<tr>
<th>Class</th>
<th>First Octet Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>0 ~ 127</td>
</tr>
<tr>
<td>Class B</td>
<td>128 ~ 191</td>
</tr>
<tr>
<td>Class C</td>
<td>192 ~ 223</td>
</tr>
<tr>
<td>Class D</td>
<td>224 ~ 239</td>
</tr>
<tr>
<td>Class E</td>
<td>240 ~ 255</td>
</tr>
</tbody>
</table>
IP Address Classes (Cont.)

- Using the ranges, you can determine the class of an address from its 1\textsuperscript{st} octet value.
- An address beginning with 120 is a Class A address, 155 is a Class B address & 220 is a Class C address.
Are You the Host or the Network?

The 32 bits of the IP address are divided into Network & Host portions, with the octets assigned as a part of one or the other.

<table>
<thead>
<tr>
<th>Class</th>
<th>Octet1</th>
<th>Octet2</th>
<th>Octet3</th>
<th>Octet4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>Network</td>
<td>Host</td>
<td>Host</td>
<td>Host</td>
</tr>
<tr>
<td>Class B</td>
<td>Network</td>
<td>Network</td>
<td>Host</td>
<td>Host</td>
</tr>
<tr>
<td>Class C</td>
<td>Network</td>
<td>Network</td>
<td>Network</td>
<td>Host</td>
</tr>
</tbody>
</table>
Each Network is assigned a network address & every device or interface (such as a router port) on the network is assigned a host address.

There are only 2 specific rules that govern the value of the address.
Are You the Host or the Network? (Cont.)

- A host address cannot be designated by all zeros or all ones.
- These are special addresses that are reserved for special purposes.
Class A Addresses

- Class A IP addresses use the 1\(^{st}\) 8 bits (1\(^{st}\) Octet) to designate the Network address.
- The 1\(^{st}\) bit which is always a 0, is used to indicate the address as a Class A address & the remaining 7 bits are used to designate the Network.
- The other 3 octets contain the Host address.
There are 128 Class A Network Addresses, but because addresses with all zeros aren’t used & address 127 is a special purpose address, 126 Class A Networks are available.
Class A Addresses (Cont.)

- There are 16,777,214 Host addresses available in a Class A address.
- Rather than remembering this number exactly, you can use the following formula to compute the number of hosts available in any of the class addresses, where “$n$” represents the number of bits in the host portion:

  \[(2^n - 2) = \text{Number of available hosts}\]
For a Class A network, there are: \(2^{24} - 2\) or 16,777,214 hosts.

Half of all IP addresses are Class A addresses.

You can use the same formula to determine the number of Networks in an address class.

Eg., a Class A address uses 7 bits to designate the network, so \((2^7 - 2) = 126\) or there can be 126 Class A Networks.
Class B IP Addresses

- Class B addresses use the 1\textsuperscript{st} 16 bits (two octets) for the Network address.
- The last 2 octets are used for the Host address.
- The 1\textsuperscript{st} 2 bit, which are always 10, designate the address as a Class B address & 14 bits are used to designate the Network. This leaves 16 bits (two octets) to designate the Hosts.
Class B IP Addresses (Cont.)

- So how many Class B Networks can there be?
- Using our formula, \((2^{14} - 2)\) there can be 16,382 Class B Networks & each Network can have \((2^{16} - 2)\) Hosts, or 65,534 Hosts.
Class C IP Addresses

- Class C addresses use the 1\textsuperscript{st} 24 bits (three octets) for the Network address & only the last octet for Host addresses. The 1\textsuperscript{st} 3 bits of all class C addresses are set to 110, leaving 21 bits for the Network address, which means there can be $2,097,150 \ (2^{21} - 2)$ Class C Networks, but only $254 \ (2^8 - 2)$ Hosts per Network.
### Class C IP Addresses (Cont.)

<table>
<thead>
<tr>
<th>Class</th>
<th>Address Range</th>
<th>Identify Bits (binary value)</th>
<th>Bits in Network ID</th>
<th>Number of Networks</th>
<th>Bits in Host ID</th>
<th>Number of Hosts/Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0 ~ 127</td>
<td>1 (0)</td>
<td>7</td>
<td>126</td>
<td>24</td>
<td>16,777,214</td>
</tr>
<tr>
<td>B</td>
<td>128~191</td>
<td>2 (10)</td>
<td>14</td>
<td>16,382</td>
<td>16</td>
<td>5,534</td>
</tr>
<tr>
<td>C</td>
<td>192~223</td>
<td>3 (110)</td>
<td>21</td>
<td>2,097,150</td>
<td>8</td>
<td>254</td>
</tr>
</tbody>
</table>
Special Addresses

- A few addresses are set aside for specific purposes.
- Network addresses that are all binary zeros, all binary ones & Network addresses beginning with 127 are special Network addresses.
### Special IP Addresses

<table>
<thead>
<tr>
<th>Network Address</th>
<th>Host Address</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>0's</td>
<td>0's</td>
<td>Default Cisco Route</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>0's</td>
<td>Host Address</td>
<td>Local Network Hosts</td>
<td>0.0.0.115</td>
</tr>
<tr>
<td>1's</td>
<td>1's</td>
<td>Broadcast to Local Network</td>
<td>255.255.255.255</td>
</tr>
<tr>
<td>Network Address</td>
<td>1's</td>
<td>Broadcast to Network Address</td>
<td>192.21.12.255</td>
</tr>
<tr>
<td>127</td>
<td>Anything</td>
<td>Loopback Testing</td>
<td>127.0.0.1</td>
</tr>
</tbody>
</table>
Special Addresses (Cont.)

- Within each address class is a set of addresses that are set aside for use in local networks sitting behind a firewall or NAT (Network Address Translation) device or Networks not connected to the Internet.
Special Addresses (Cont.)

- A list of these addresses for each IP address class:

<table>
<thead>
<tr>
<th>IP Class</th>
<th>Address Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>10.0.0.0 ~ 10.255.255.255</td>
</tr>
<tr>
<td>Class B</td>
<td>172.16.0.0 ~ 172.31.255.255</td>
</tr>
<tr>
<td>Class C</td>
<td>192.168.0.0 ~ 192.168.255.255</td>
</tr>
</tbody>
</table>
Subnet Mask

- An IP address has 2 parts:
  - The Network identification.
  - The Host identification.
- Frequently, the Network & Host portions of the address need to be separately extracted.
- In most cases, if you know the address class, it’s easy to separate the 2 portions.
Subnet Mask (Cont.)

- With the rapid growth of the internet & the ever-increasing demand for new addresses, the standard address class structure has been expanded by borrowing bits from the Host portion to allow for more Networks.

- Under this addressing scheme, called Subnetting, separating the Network & Host requires a special process called Subnet Masking.
The subnet masking process was developed to identify & extract the Network part of the address.

A subnet mask, which contains a binary bit pattern of ones & zeros, is applied to an address to determine whether the address is on the local Network.

If it is not, the process of routing it to an outside network begins.
Subnet Mask (Cont.)

- The function of a subnet mask is to determine whether an IP address exists on the local network or whether it must be routed outside the local network.
- It is applied to a message’s destination address to extract the network address.
- If the extracted network address matches the local network ID, the destination is located on the local network.
Subnet Mask (Cont.)

- However, if they don’t match, the message must be routed outside the local network.
- The process used to apply the subnet mask involves Boolean Algebra to filter out non-matching bits to identify the network address.
Boolean Algebra

- **Boolean Algebra** is a process that applies binary logic to yield binary results.
- Working with subnet masks, you need only 4 basic principles of Boolean Algebra:
  - 1 and 1 = 1
  - 1 and 0 = 0
  - 0 and 1 = 0
  - 0 and 0 = 0
In another words, the only way you can get a result of a 1 is to combine 1 & 1. Everything else will end up as a 0.

The process of combining binary values with Boolean Algebra is called **Anding**.
Default Standard Subnet Masks

There are default standard subnet masks for Class A, B and C addresses:

<table>
<thead>
<tr>
<th>Address Class</th>
<th>Subnet Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>255.0.0.0</td>
</tr>
<tr>
<td>Class B</td>
<td>255.255.0.0</td>
</tr>
<tr>
<td>Class C</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>
Subnet masks apply only to Class A, B or C IP addresses.

The subnet mask is like a filter that is applied to a message’s destination IP address.

Its objective is to determine if the local network is the destination network.
A Trial Separation (Cont.)

- The subnet mask goes like this:
  1. If a destination IP address is 206.175.162.21, we know that it is a Class C address & that its binary equivalent is: 11001110.10101111.10100010.00010101
A Trial Separation (Cont.)

2. We also know that the default standard Class C subnet mask is: 255.255.255.0 and that its binary equivalent is:
   11111111.11111111.11111111.00000000
A Trial Separation (Cont.)

3. When these two binary numbers (the IP address & the subnet mask) are combined using Boolean Algebra, the Network ID of the destination network is the result:

<table>
<thead>
<tr>
<th>206.175.162.21</th>
<th>11001110.10101111.10100010.00010101</th>
</tr>
</thead>
<tbody>
<tr>
<td>and</td>
<td></td>
</tr>
<tr>
<td>255.255.255.0</td>
<td>11111111.11111111.11111111.00000000</td>
</tr>
</tbody>
</table>

yields

| 11001110.10101111.10100010.00000000 |
A Trial Separation (Cont.)

4. The result is the IP address of the network which in this case is the same as the local network & means that the message is for a node on the local network.
Subnetting
Introduction

- Subnetting is the foundation underlying the expansion of both Local Networks & the Internet in today’s world.
- Subnetting has become essential knowledge for the Administrator of any network.
- There are 2 fundamental reasons why subnetting has so much importance in today’s networking environment:
1) The world is running out of available IP addresses. There just isn’t an unlimited number of IP addresses available & subnetting helps extend the existing addresses until either the next version of IP is rolled out or some other technology charges on the scene.
Introduction (Cont.)

2) Subnetting reduces the size of the routing tables stored in routers. Subnetting extends the existing IP address base & restructures the IP address. As a result, routers must have a way to extract from a IP address both the Network address & the Host address.
There are only 3 usable IP address classes:
- Class A
- Class B
- Class C

Class A networks have the highest number of available hosts.
Class C networks have the fewest number of hosts.
Subnetting Networks ID

- A 3-step example of how the default Class A subnet mask is applied to a Class A address:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Address</td>
<td>123.123.123.001</td>
</tr>
<tr>
<td></td>
<td>01111011.01111011.01111011.00000001</td>
</tr>
<tr>
<td>Subnet Mask</td>
<td>255.0.0.0</td>
</tr>
<tr>
<td></td>
<td>11111111.00000000.00000000.00000000</td>
</tr>
<tr>
<td>Network ID</td>
<td>123.0.0.0</td>
</tr>
<tr>
<td></td>
<td>01111011.00000000.00000000.00000000</td>
</tr>
</tbody>
</table>
In the previous slide, the default Class A subnet mask (255.0.0.0) is AND’d with the Class A address (123.123.123.001) using Boolean Algebra, which results in the Network ID (123.0.0.0) being revealed.

The default Class B subnet mask (255.255.0.0) strips out the 16-bit network ID & the default Class C subnet mask (255.255.255.0) strips out the 24-bit network ID.
Subnetting, Subnet & Subnet Mask

- Subnetting, a subnet & a subnet mask are all different.
- In fact, the 1st creates the 2nd & is identified by the 3rd.
- Subnetting is the process of dividing a network & its IP addresses into segments, each of which is called a subnetwork or subnet.
Subnetting, Subnet & Subnet Mask (Cont.)

- The subnet mask is the 32-bit number that the router uses to cover up the network address to show which bits are being used to identify the subnet.
Subnetting

- A network has its own unique address, such as a Class B network with the address **172.20.0.0** which has all zeroes in the host portion of the address.

- From the basic definitions of a Class B network & the default Class B subnet mask, you know that this network can be created as a single network that contains **65,534** individual hosts.
Subnetting (Cont.)

- Through the use of subnetting, the network from the previous slide can be logically divided into subnets with fewer hosts on each subnetwork.
- It does not improve the available shared bandwidth only, but it cuts down on the amount of broadcast traffic generated over the entire network as well.
The 2 primary benefits of subnetting are:

1. Fewer IP addresses, often as few as one, are needed to provide addressing to a network & subnetting.
2. Subnetting usually results in smaller routing tables in routers beyond the local internetwork.
Subnetting (Cont.)

Example of subnetting: when the network administrator divides the 172.20.0.0 network into 5 smaller networks – 172.20.1.0, 172.20.2.0, 172.20.3.0, 172.20.4.0 & 172.20.5.0 – the outside world still sees the network as 172.20.0.0, but the internal routers now break the network addressing into the 5 smaller subnetworks.
Subnetting (Cont.)

- In the example, only a single IP address is used to reference the network & instead of 5 network addresses, only one network reference is included in the routing tables of routers on other networks.
Borrowing Bits to Grow a Subnet

- The key concept in subnetting is borrowing bits from the host portion of the network to create a subnetwork.
- Rules govern this borrowing, ensuring that some bits are left for a Host ID.
- The rules require that two bits remain available to use for the Host ID & that all of the subnet bits cannot be all 1s or 0s at the same time.
Borrowing Bits to Grow a Subnet (Cont.)

- For each IP address class, only a certain number of bits can be borrowed from the host portion for use in the subnet mask.
## Borrowing Bits to Grow a Subnet (Cont.)

<table>
<thead>
<tr>
<th>Address Class</th>
<th>Host Bits</th>
<th>Bits Available for Subnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>24</td>
<td>22</td>
</tr>
<tr>
<td>B</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>C</td>
<td>8</td>
<td>6</td>
</tr>
</tbody>
</table>
Subnetting a Class A Network

- The default subnet mask for a class A network is 255.0.0.0 which allows for more than 16,000,000 hosts on a single network.
- The default subnet mask uses only 8 bits to identify the network, leaving 24 bits for host addressing.
Subnetting a Class A Network (Cont.)

- To subnet a Class A network, you need to borrow a sufficient number of bits from the 24-bit host portion of the mask to allow for the number of subnets you plan to create, now & in the future.

- Example: To create 2 subnets with more than 4 millions hosts per subnet, you must borrow 2 bits from the 2nd octet & use 10 masked (value equals one) bits for the subnet mask (11111111.11000000) or 255.192 in decimal.
Subnetting a Class A Network (Cont.)

- Keep in mind that each of the 8-bit octets has binary place values.
- When you borrow bits from the Host ID portion of the standard mask, you don’t change the value of the bits, only how they are grouped & used.
## Subnetting a Class A Network (Cont.)

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Power of 2</th>
<th>Decimal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>$2^7$</td>
<td>128</td>
</tr>
<tr>
<td>6</td>
<td>$2^6$</td>
<td>64</td>
</tr>
<tr>
<td>6</td>
<td>$2^5$</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>$2^4$</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>$2^3$</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>$2^2$</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>$2^1$</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>$2^0$</td>
<td>1</td>
</tr>
</tbody>
</table>
A sample of subnet mask options available for Class A addresses.

<table>
<thead>
<tr>
<th>Subnet Mask</th>
<th>Number of 1 Bits in Mask</th>
<th>Number of Subnets</th>
<th>Number of Hosts per Subnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>255.0.0.0</td>
<td>8</td>
<td>0</td>
<td>16,777,214</td>
</tr>
<tr>
<td>255.192.0.0</td>
<td>10</td>
<td>2</td>
<td>4,194,302</td>
</tr>
<tr>
<td>255.240.0.0</td>
<td>12</td>
<td>14</td>
<td>1,048,574</td>
</tr>
<tr>
<td>255.255.0.0</td>
<td>16</td>
<td>254</td>
<td>65,534</td>
</tr>
<tr>
<td>255.255.128.0</td>
<td>17</td>
<td>510</td>
<td>32,766</td>
</tr>
<tr>
<td>255.255.240.0</td>
<td>20</td>
<td>4,094</td>
<td>4,094</td>
</tr>
<tr>
<td>255.255.255.128</td>
<td>25</td>
<td>131,070</td>
<td>126</td>
</tr>
<tr>
<td>255.255.255.240</td>
<td>28</td>
<td>1,048,574</td>
<td>14</td>
</tr>
<tr>
<td>255.255.255.252</td>
<td>30</td>
<td>4,192,302</td>
<td>2</td>
</tr>
</tbody>
</table>
Class A Subnet Masks (Cont.)

- All subnet masks contain 32 bits; no more, no less.

- However a subnet mask cannot filter more than 30 bits. This means 2 things:
  - One, that there cannot be more than 30 ones bits in the subnet mask.
  - Two, that there must always be at least 2 bits available for the Host ID.
The subnet mask with the highest value (255.255.255.252) has a binary representation of:

11111111.11111111.11111111.11111100

The 2 zeroes in this subnet mask represent the 2 positions set aside for the Host address portion of the address.
Remember that the addresses with all ones (broadcast address) & all zeroes (local network) cannot be used as they have special meanings.
Subnetting Class B & Class C

- The table on slide 76 “Class A Subnet Masks” is similar to the tables used for Class B & Class C IP addresses & subnet masks.

- The only differences are that you have fewer options (due to a fewer number of bits available) & that you’re much more likely to work with Class B & Class C networks in real life.
Subnetting Class B & Class C (Cont.)

A sample of the subnet masks available for Class B networks.

<table>
<thead>
<tr>
<th>Subnet Mask</th>
<th>No. of 1 Bits in Mask</th>
<th>No. of Subnets</th>
<th>No. of Hosts per Subnet</th>
<th>Binary Subnet Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>255.255.0.0</td>
<td>16</td>
<td>0</td>
<td>65,534</td>
<td>11111111.11111111.00000000.00000000</td>
</tr>
<tr>
<td>255.255.192.0</td>
<td>18</td>
<td>2</td>
<td>16,382</td>
<td>11111111.11111111.11000000.00000000</td>
</tr>
<tr>
<td>255.255.240.0</td>
<td>20</td>
<td>14</td>
<td>4,094</td>
<td>11111111.11111111.11100000.00000000</td>
</tr>
<tr>
<td>255.255.255.0</td>
<td>24</td>
<td>254</td>
<td>254</td>
<td>11111111.11111111.11110000.00000000</td>
</tr>
<tr>
<td>255.255.255.240</td>
<td>28</td>
<td>4,094</td>
<td>14</td>
<td>11111111.11111111.11110000.00000000</td>
</tr>
<tr>
<td>255.255.255.252</td>
<td>30</td>
<td>16,382</td>
<td>2</td>
<td>11111111.11111111.11111111.11111111.11111111.11111100</td>
</tr>
</tbody>
</table>
Subnetting Class B & Class C (Cont.)

A list of the subnet masks available for Class C networks.

<table>
<thead>
<tr>
<th>Subnet Mask</th>
<th>No. of 1 Bits in Mask</th>
<th>No. of Subnets</th>
<th>No. of Hosts per Subnet</th>
<th>Binary Subnet Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>255.255.255.0</td>
<td>24</td>
<td>0</td>
<td>254</td>
<td>11111111.11111111.11111111.00000000000</td>
</tr>
<tr>
<td>255.255.255.192</td>
<td>26</td>
<td>2</td>
<td>62</td>
<td>11111111.11111111.11111111.11000000000</td>
</tr>
<tr>
<td>255.255.255.224</td>
<td>27</td>
<td>6</td>
<td>30</td>
<td>11111111.11111111.11111111.11000000000</td>
</tr>
<tr>
<td>255.255.255.240</td>
<td>28</td>
<td>14</td>
<td>14</td>
<td>11111111.11111111.11111111.11100000000</td>
</tr>
<tr>
<td>255.255.255.248</td>
<td>29</td>
<td>30</td>
<td>6</td>
<td>11111111.11111111.11111111.11110000000</td>
</tr>
<tr>
<td>255.255.255.252</td>
<td>30</td>
<td>62</td>
<td>2</td>
<td>11111111.11111111.11111111.11111000000</td>
</tr>
</tbody>
</table>
Knowing How to Calculate Subnets

To determine the number of subnets & hosts per subnet available for any of the available subnet masks, 2 simple formulas to calculate these numbers:

Number of Hosts per Subnet = \((2^{\text{number of bits used for Host}}) - 2\)

and

Number of Subnets = \((2^{\text{number of bits used for Subnets}}) - 2\)
Knowing How to Calculate Subnets (Cont.)

- Although the 2 formulas look identical, the key is to remember the number you’re trying to calculate, hosts or subnets.

- Eg., suppose you are asked to determine the number of subnets available & the number of hosts available on each subnet on the network 192.168.1.0
Knowing How to Calculate Subnets (Cont.)

- Using the subnet & hosts formulas, the answers are easily calculated. Of course, you must know your powers of 2 to calculate the answers.
Class C Subnets

Knowing the relationships in this table will significantly reduce the time you spend calculating subnetting problems.

<table>
<thead>
<tr>
<th>Mask</th>
<th>Subnet Bits Used</th>
<th>Number of Subnets</th>
<th>Number of Hosts/Subnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>255.255.255.252</td>
<td>6</td>
<td>62</td>
<td>2</td>
</tr>
<tr>
<td>255.255.255.248</td>
<td>5</td>
<td>30</td>
<td>6</td>
</tr>
<tr>
<td>255.255.255.240</td>
<td>4</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>255.255.255.224</td>
<td>3</td>
<td>6</td>
<td>30</td>
</tr>
<tr>
<td>255.255.255.192</td>
<td>2</td>
<td>2</td>
<td>62</td>
</tr>
<tr>
<td>255.255.255.128</td>
<td>1</td>
<td>Not a Legal Subnet Value</td>
<td></td>
</tr>
<tr>
<td>255.255.255.0</td>
<td>0 (Default)</td>
<td>1</td>
<td>254</td>
</tr>
</tbody>
</table>
Class C Subnets (Cont.)

- To determine the total length of the subnet mask, add 24 to the number of borrowed (subnet) bits.
Class B Subnets

- To calculate the number of subnets & hosts available from a Class B subnet mask, you use the same host & subnet formulas described for calculating Class C values.

- Using these formulas I have constructed a table that contains the Class B subnet & host values.
### Class B Subnets (Cont.)

<table>
<thead>
<tr>
<th>Mask</th>
<th>Subnet Bits Used</th>
<th>Number of Subnets</th>
<th>Number of Hosts/Subnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>255.255.255.252</td>
<td>14</td>
<td>16,382</td>
<td>2</td>
</tr>
<tr>
<td>255.255.255.248</td>
<td>13</td>
<td>8,190</td>
<td>6</td>
</tr>
<tr>
<td>255.255.255.240</td>
<td>12</td>
<td>4,094</td>
<td>14</td>
</tr>
<tr>
<td>255.255.255.224</td>
<td>11</td>
<td>2,046</td>
<td>30</td>
</tr>
<tr>
<td>255.255.255.192</td>
<td>10</td>
<td>1,022</td>
<td>62</td>
</tr>
<tr>
<td>255.255.255.128</td>
<td>9</td>
<td>510</td>
<td>126</td>
</tr>
<tr>
<td>255.255.255.0</td>
<td>8</td>
<td>254</td>
<td>254</td>
</tr>
<tr>
<td>255.255.254.0</td>
<td>7</td>
<td>126</td>
<td>510</td>
</tr>
<tr>
<td>255.255.252.0</td>
<td>6</td>
<td>62</td>
<td>1,022</td>
</tr>
<tr>
<td>255.255.248.0</td>
<td>5</td>
<td>30</td>
<td>2,046</td>
</tr>
<tr>
<td>255.255.240.0</td>
<td>4</td>
<td>14</td>
<td>4,094</td>
</tr>
<tr>
<td>255.255.224.0</td>
<td>3</td>
<td>6</td>
<td>8,190</td>
</tr>
<tr>
<td>255.255.192.0</td>
<td>2</td>
<td>2</td>
<td>16,382</td>
</tr>
<tr>
<td>255.255.128.0</td>
<td>1</td>
<td>Not a Legal Subnet Value</td>
<td></td>
</tr>
<tr>
<td>255.255.0.0</td>
<td>0 (Default)</td>
<td>1</td>
<td>65,534</td>
</tr>
</tbody>
</table>
A Short Broadcast

- A broadcast is a message that every node on a network or subnetwork receives & examines.
- Cisco IOS supports 2 different types of broadcast messages:
  - Flooded
  - Directed
A Short Broadcast (Cont.)

- Generally speaking, routers do not propagate broadcasts, which is one of the benefits of installing a router in the first place.
A Short Broadcast (Cont.)

- Flooded broadcasts (those with the nominal broadcast address of 255.255.255.255) are not forwarded by the router & are considered local traffic only.

- Directed broadcasts, which contain all 1’s in the Host portion of the IP address, are addressed to a specific subnetwork & are allowed to pass.