# IP Subnets

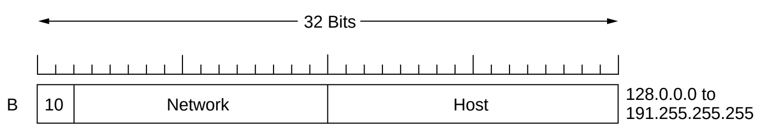
CSC 343·643

Wake Forest University
Department of Computer Science

Fall 2013

## IP Networks

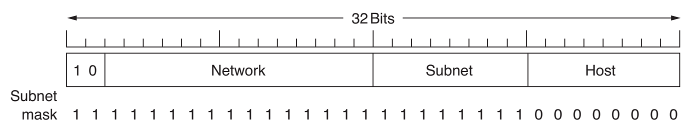
Consider a company with one class B network address `162.1.0.0/16`




- This allows up to $2^{16} - 2 = 65534$ hosts in *one* network
- We know all hosts must have the same network number
  - IP classes are hierarchical, but a single network may be flat
  - This may cause problems as the network grows
- Similarly, assume the company has multiple distinct departments
  - Would like a separate network per department
  - Localize traffic to department

## IP Subnets

- A solution to the proceeding problem is to use **subnets** [RFC950]
  - Internally create a new network hierarchy
  - Externally appears like a single (flat) network
- To subnet, take original network address and divide **host** portion into **subnet** and **host** (*in that order*)



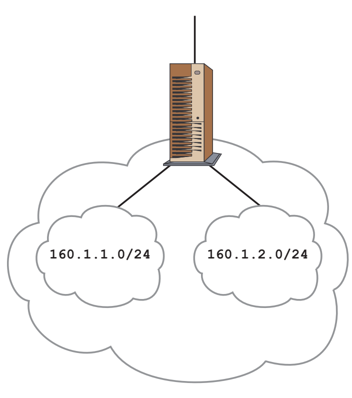

  - From the original 16 bits for the host, 8 bits have been reserved for the subnet, remaining bits are for the host

    *How many subnets, how many hosts per subnet?*

- N.B. Subnet space does **not** have to stop on byte boundaries!



- Now the network `160.1.0.0/16` can be divided into subnets
  - Assume we use the subnet division from the previous slide
  - `160.1.1.0/24`, `160.1.2.0/24`, `160.1.3.0/24`, ...
    are all subnets of `160.1.0.0/16`




  - For each subnet, the network number is /24

    *Could these subnets be subnetted?*

# IP Subnet Mask

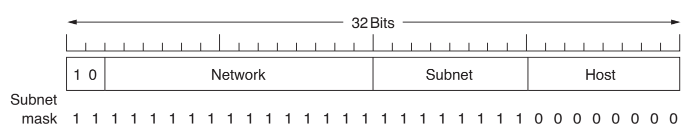
- Subnet is internal to network, other (external) networks do not care how network is divided
  - Subnet routing must be handled internally
- With classful routing we could easily determine the hierarchy

  *How is this done?*

- Must distinguish host from network and subnet numbers
  - This is done using a **subnet mask**
- Subnet mask is a 32 bit value containing one bits for the network and subnet numbers, and zero bits for the host number



- For our previous example,




- The subnet mask is [`11111111 11111111 11111111 00000000`]
  - `255.255.255.0` in dotted-decimal notation; however, hexadecimal is more commonly used... yippee
  - Now you know the real meaning of /24 vs /16
- The *local* subnet mask **must** be known by each host
  - This is how a host can determine if an IP address is *local* (same subnet)

- Assume your IP address is `140.1.1.1` and the subnet mask `255.255.255.0`
  - Take the IP address and AND (logical) it with the subnet mask

$$\begin{array}{rl} & \texttt{10001100 00000001 0000001 00000001} \\ \oplus & \texttt{11111111 11111111 1111111 00000000} \\ \hline & \texttt{10001100 00000001 0000001 00000000} \end{array}$$

  - Assume the destination address is `140.1.1.2`, is it local?

$$\begin{array}{rl} & \texttt{10001100 00000001 0000001 00000010} \\ \oplus & \texttt{11111111 11111111 1111111 00000000} \\ \hline & \texttt{10001100 00000001 0000001 00000000} \end{array}$$

  - Assume the destination address is `140.1.2.1`, is it local?

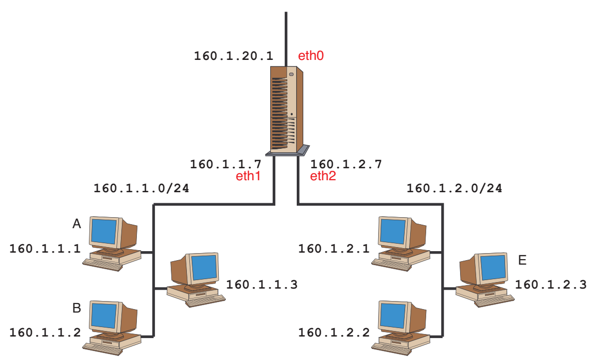
$$\begin{array}{rl} & \texttt{10001100 00000001 0000010 00000001} \\ \oplus & \texttt{11111111 11111111 1111111 00000000} \\ \hline & \texttt{10001100 00000001 0000010 00000000} \end{array}$$

*So what is compared to determine if an address is local?*



## Example IP Network with Subnets

- Assume a network has a class B address `160.1.0.0/16`




- Network is divided into subnets, with the mask `255.255.255.0`
- There are two subnets `160.1.1.0/24` and `160.1.2.0/24`

  *How must the machines be connected if no subnets?*

## IP Routing Tables

- We must add a new column in the previous routing table
  - New column will record the network (subnet) mask
- The routing table for host A would be

| Routing Table for Host A | | | |
|---|---|---|---|
| **Destination** | **Net Mask** | **Next Hop** | **Interface** |
| 160.1.1.0/24 | 255.255.255.0 | | eth0 |
| 160.1.2.0/24 | 255.255.255.0 | 160.1.1.7 | eth0 |
| *default* | | 160.1.1.7 | eth0 |

  - First entry is the local network, if we AND the mask with the IP destination address and get the destination entry, it is local!
  - Similar action required for the second entry
  - Added a default entry: send to router if previous entries fail



*What would an host entry for 160.1.2.2 consist of?*

- The routing table for the router would be

| Routing Table for Router | | | |
|---|---|---|---|
| **Destination** | **Net Mask** | **Next Hop** | **Interface** |
| 160.1.1.0/24 | 255.255.255.0 | | eth1 |
| 160.1.2.0/24 | 255.255.255.0 | | eth2 |
| *default* | | ? | eth0 |

  - Default entry would send packets to the Internet, next-hop would be the first router outside the 160.1.0.0/16 network

## IP Routing with Subnets

- Routing datagrams is slightly different with subnets
- The routing algorithm is

$I$ = IP datagram destination address
$D_i$ = $i$th destination (host or network) see table
$M_i$ = $i$th subnet mask
**for** each table entry $i$
    **if** $I \oplus M_i == D_i$
        route to host or network, depending on entry
        exit algorithm
    **endif**
**endfor**
**if** no match, route to default router

- Same algorithm run by hosts and routers

  *What is the difference between the two?*



## Routing Examples

| Routing Table for Host A | | | |
|---|---|---|---|
| **Destination** | **Net Mask** | **Next Hop** | **Interface** |
| 160.1.1.0/24 | 255.255.255.0 | | eth0 |
| 160.1.2.0/24 | 255.255.255.0 | 160.1.1.7 | eth0 |
| *default* | | 160.1.1.7 | eth0 |

- Host A (160.1.1.1) send a datagram to host B (160.1.1.2)
  - Host A first entry comparison

$$\begin{array}{rl} & 10100000\ 00000001\ 0000001\ 00000001 \\ \oplus & 11111111\ 11111111\ 1111111\ 00000000 \\ \hline & 10001100\ 00000001\ 0000001\ 00000000 \end{array}$$

    Result is 160.1.1.0 which matches the routing table destination entry; therefore, destination is local

  - Host A creates link-layer frame addressed to link-layer address of 160.1.1.2
  - Frame sent and received by host B

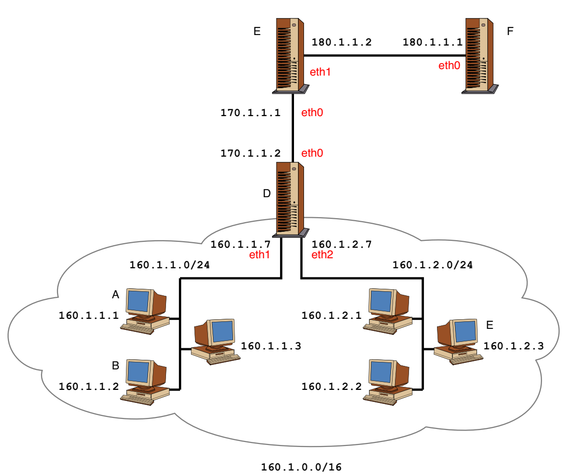
| Routing Table for Host A | | | |
|---|---|---|---|
| **Destination** | **Net Mask** | **Next Hop** | **Interface** |
| 160.1.1.0/24 | 255.255.255.0 | | eth0 |
| 160.1.2.0/24 | 255.255.255.0 | 160.1.1.7 | eth0 |
| *default* | | 160.1.1.7 | eth0 |

- Host A (160.1.1.1) send a datagram to host E (160.1.2.3)
  - Host A first entry comparison
    160.1.2.3 $\oplus$ 255.255.255.0 = 160.1.2.0
    which is not equal to 160.1.1.0
  - Host A second entry comparison
    160.1.2.3 $\oplus$ 255.255.255.0 = 160.1.2.0
    which is equal to table entry 160.1.2.0
  - Host A creates link-layer frame addressed to link-layer address of 160.1.1.7 (router)
  - Frame sent and received by router, and process repeats...



## Another Routing Example

- Assume D is the *gateway* for the 160.1.0.0/16 network




*Do E and F need to know about the 160.1.0.0/16 subnets?*
*What are the routing tables for E and F?*