CS443 - Networks (pt 2)

Note Title 2/4/2013

Announcements

- Quiz today
- HW2 due
- Lab 3 up - due in 1 week

The Lab: IPTables

- Go to deterlab.net
  - Login
  - Go to "Begin an Experiment"

/share/education/PermissionsFirewalls_UCLA/permissions.ns

Enter info

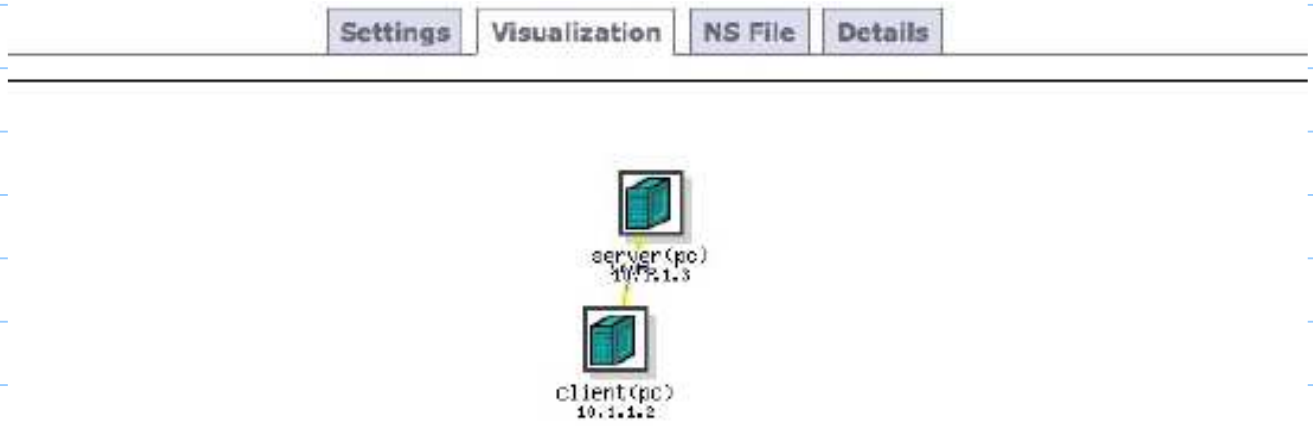
Begin a Testbed Experiment

- **If you have an NS file:**
  You may want to **syntax check it first**
- **If you do not have an NS file:**
  New GUI editor - An enhanced Java applet for editing topologies.
- **For manipulating your experiment, consider SEER.**

Select Project:	UP-CS448
Group:	Default Group (Must be default or correspond to selected project)
Name: (No blanks)	
Description: (A concise sentence)	
Your NS file: Syntax Check	Upload (500k max) Browse... or On Server (/proj, /users, /groups, /share)
Swapping:	Idle-Swap: Swap out this experiment after 4 hours idle. If not, why not? Max. Duration: Swap out after 12 hours, even if not idle.
Linktest Option:	Skip Linktest (What is this?)
Swap In Immediately	
Submit	

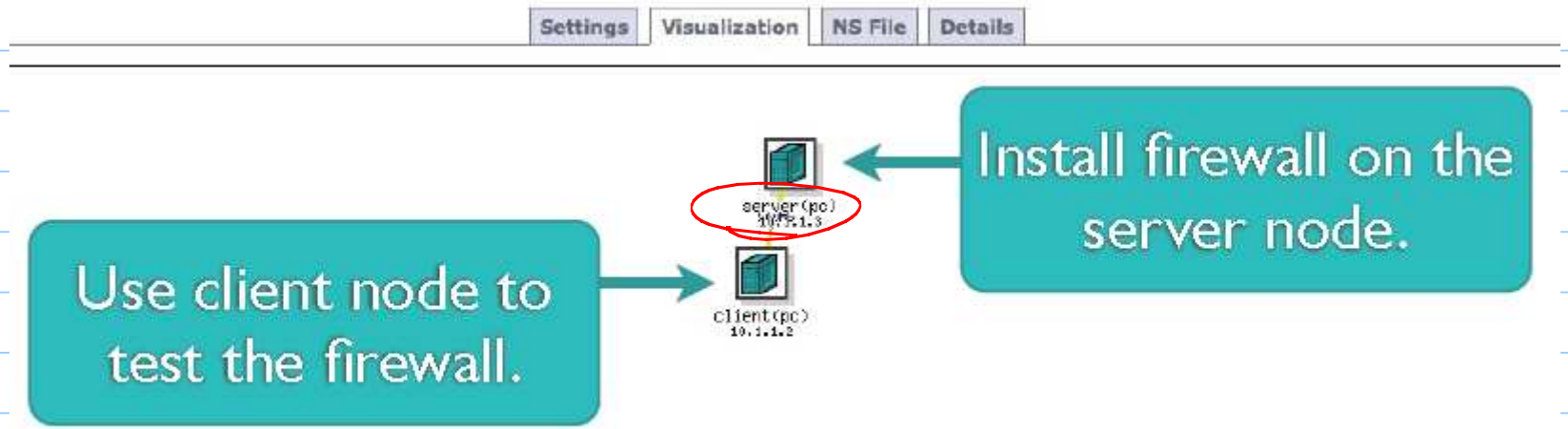
The visualization tab will show the network architecture of this experiment.

(will be useful in later labs, too.)



This lab:

Two goals:



Edit & run the firewall script

```
up448ar@server:~$ cd /root/firewall

up448ar@server:/root/firewall$ ls

extingui.sh  extingui.sh~  firewall.sh  firewall.sh~  iptables_reset
tester.sh
```

make youre edits

```
up448ar@server:/root/firewall$ sudo sh firewall.sh

Starting firewall: done.
```

Note:

Use extingui.sh to reset between attempts.

```
up448ar@server:~$ cd /root/firewall

up448ar@server:/root/firewall$ ls

extingui.sh  extingui.sh~  firewall.sh firewall.sh~  iptables_reset
tester.sh

up448ar@server:/root/firewall$ sudo sh extingui.sh

up448ar@server:/root/firewall$
```

IPtables : Quick intro

-L : list rules

```
% sudo iptables -L

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Adding a rule:

-A: Append a rule to the INPUT, OUTPUT or FORWARD chain.

-p: Specify a protocol.

-d: Specify a destination port.

-j: Specify a target:

ACCEPT: Accept the packet.

REJECT: Reject the packet, notify the sender.

DROP: Silently ignore the packet.

LOG: Log the packet.

Example:

Add rule to accept incoming ssh traffic:

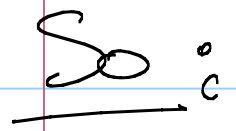
```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Also: restrict based on where it is from

-i: input interface.

-o: output interface.

-s: source, e.g., a machine name or an IP address.



```
up448ar@server:~$ ifconfig
eth4      Link encap:Ethernet  HWaddr 00:1b:21:1e:ac:14
          inet addr:10.1.1.3  Bcast:10.1.1.255  Mask:255.255.255.0
          inet6 addr: fe80::21b:21ff:fe1e:ac14/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:706 (706.0 B)

eth8      Link encap:Ethernet  HWaddr 00:13:72:4e:cc:6d
          inet addr:192.168.1.181  Bcast:192.168.3.255  Mask:255.255.252.0
          inet6 addr: fe80::213:72ff:fe4e:cc6d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14551 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2131 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18471062 (18.4 MB)  TX bytes:199954 (199.9 KB)
```

Then use this:

```
iptables -A INPUT -i eth4 -s server -j DROP
```

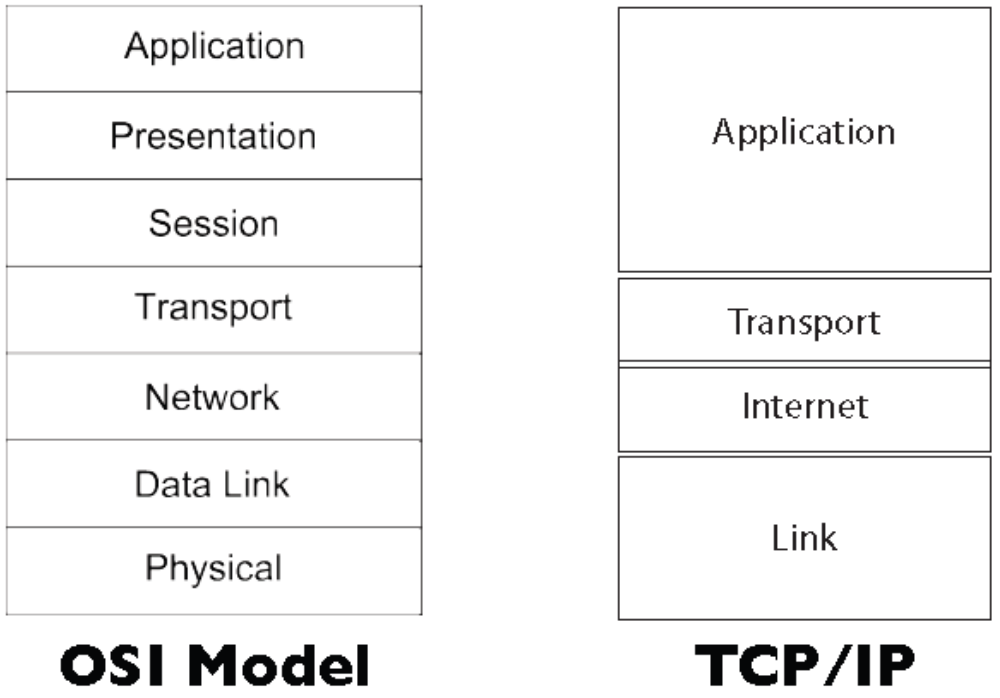
What does this rule do?

Drop all incoming traffic on eth4 from server

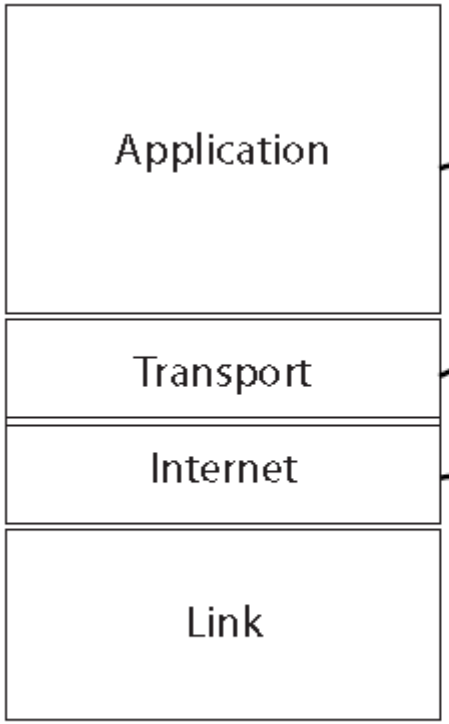
Networking:

The Internet Protocol Suite, informally called TCP/IP, is an implementation of the OSI model.

Recall:



TCP/IP Layers



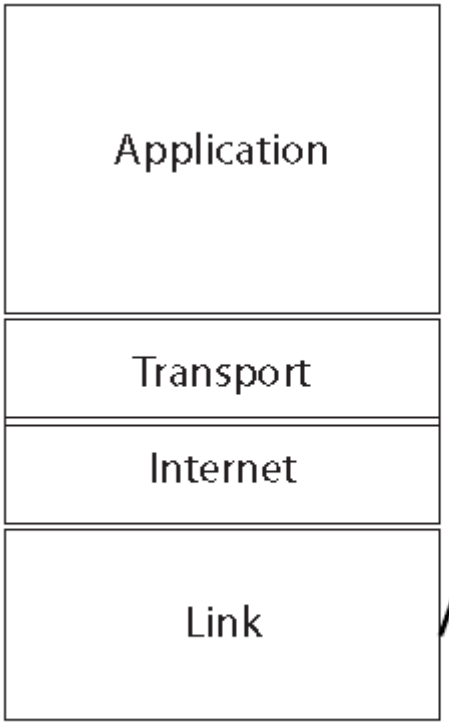
TCP/IP

Determines how an application should use the network.

Handles details of data transmission at the end-points of communication.

Expresses how packets should be forwarded and how packets should be addressed (also called **Network Layer**)

TCP/IP Layers



TCP/IP

Sometimes divided into Link Layer and Physical Layer.

Link Layer: Provides for synchronization and transfer of information. Defines how physical machines address each other.

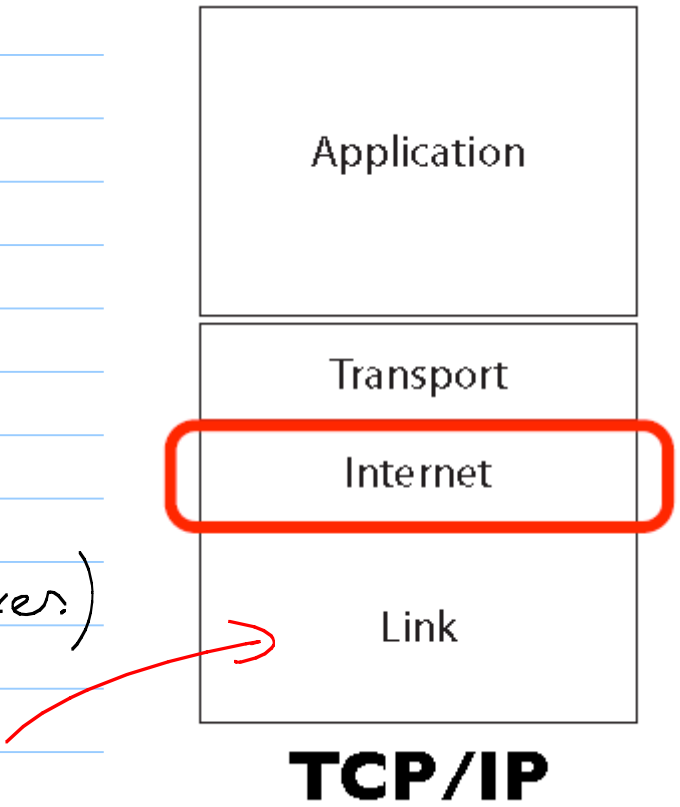
Physical Layer: Defines electrical aspects of sending signals along a wire or wirelessly. Also addresses switch and router hardware.

TCP/IP:

We'll focus on the Internet layer –

how packets are addressed and forwarded over the network.

(Also called Network layer)



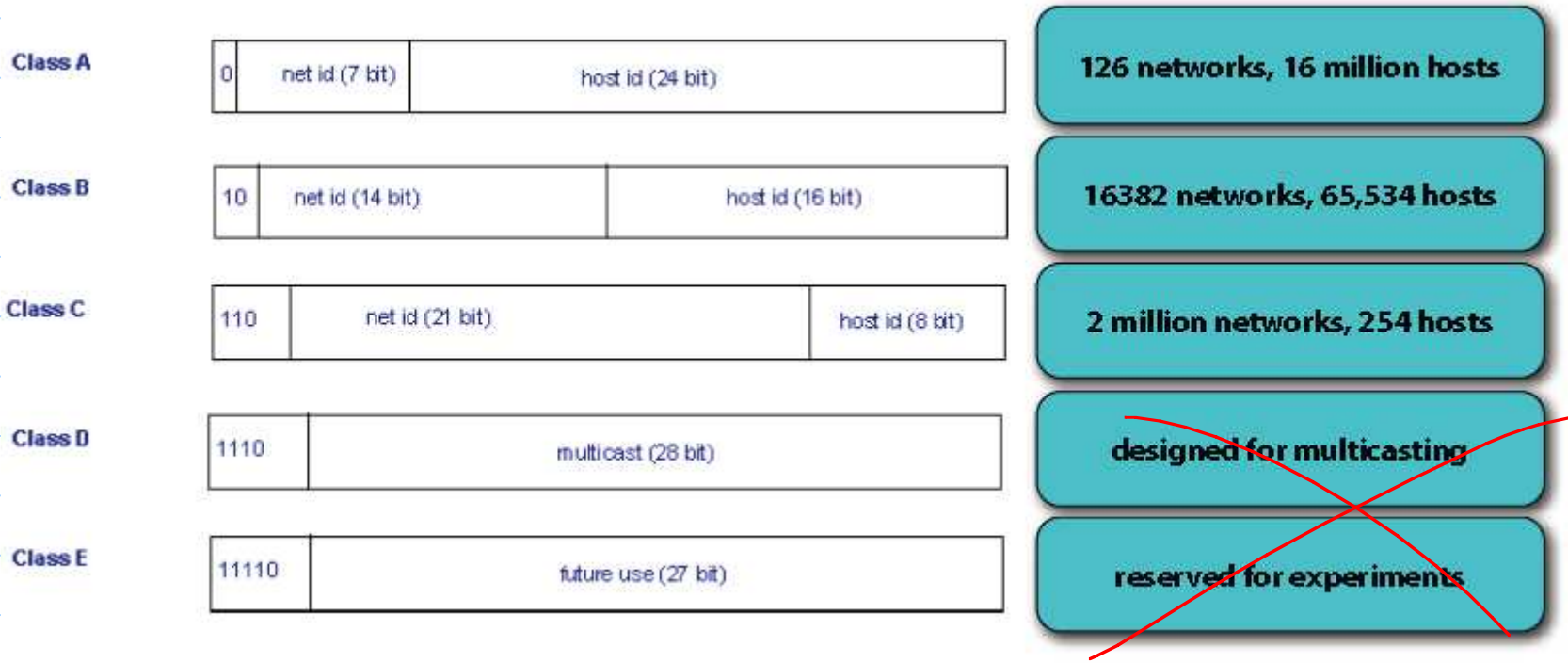
Headers in IPv4:

- Divided into 32-bit segments
- Headers are 5 segments long (usually), with data at the end

bit offset	0–3	4–7	8–13	14-15	16–18	19–31
0	Version	Header Length	Differentiated Services Code Point	Explicit Congestion Notification	Total Length	
32	Identification				Flags	Fragment Offset
64	Time to Live		Protocol		Header Checksum	
96	Source IP Address					
128	Destination IP Address					
160	Options (if Header Length > 5)					
160 or 192+	Data					

IPv4

10.1.6.3


Class A
0
net id (7 bit)
host id (24 bit)
126 networks, 16 million hosts
Class B
10
net id (14 bit)
host id (16 bit)
16382 networks, 65,534 hosts
Class C
110
net id (21 bit)
host id (8 bit)
2 million networks, 254 hosts
Class D
1110
multicast (28 bit)
designed for multicasting
Class E
11110
future use (27 bit)
reserved for experiments

Example: Consider the address:

10001000 11100101 11001001 0001000

Class? B

What IP? 136.229.201.8

Problem:

IPv4 was designed in 1981.

Classes A-C allow for under 4.3 billion address total.
(Reality - much smaller!)

Conclusion: Problem - out of space.

Solutions

1. IPv6
2. NAT
3. Subnetting

None is a perfect cure but all have been used to offset issues.

① IPv6

- Invented in 1998
- Allows for 128-bit addresses (versus 32-bit) 2^{32} vs. 2^{128}
- Transition has been slower than expected: as of Nov. 2012, reported to be ~1% of total traffic.

IPv6 details:

- Packet headers are twice as long.
- However, processing is actually simpler + faster at routers.
- Privacy extensions exist to "hide" identity: OS generates random host identifier

(2) Network Address Translation

A router stands between a private network & outside world.

Every internal IP address maps to a single IP/port which is all outside sees.

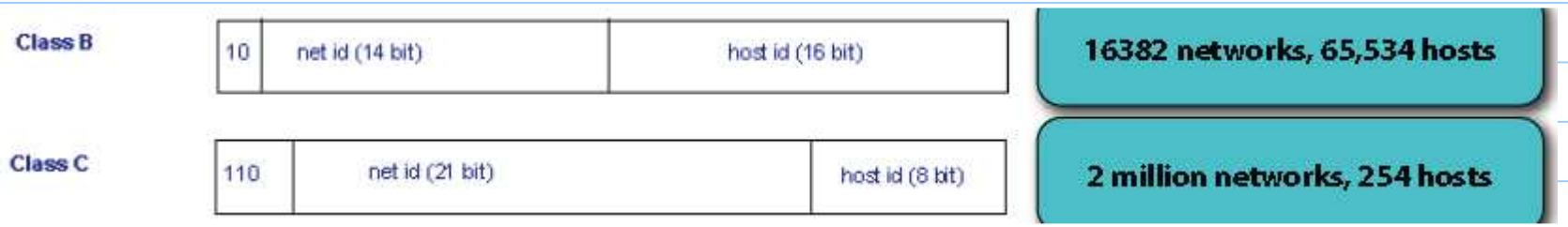
(Combines well with firewall functionality.)

Pros of NAT: - effectively a firewall
- only need 1 IP

Cons: - single point of failure
- significant overhead

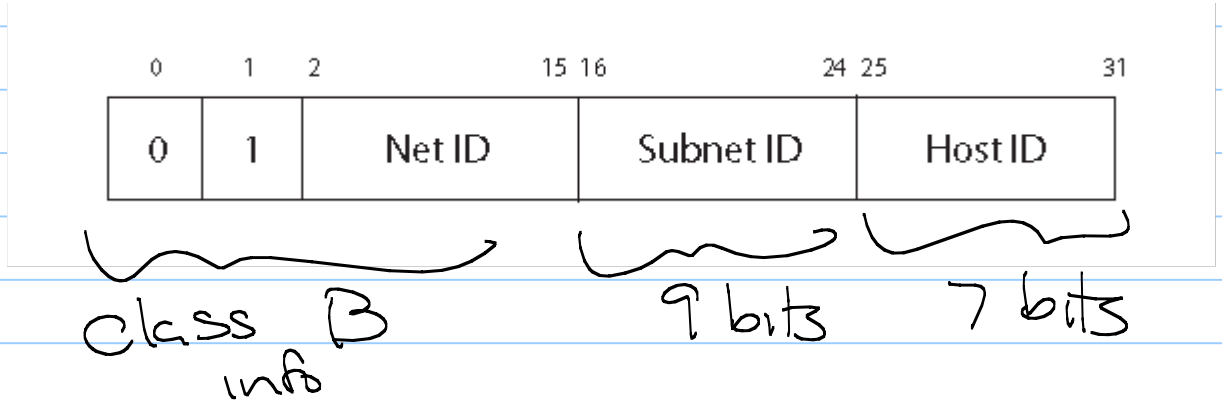
③ Subnetting

There is a jump between class B and C sizes.



Many larger networks actually subdivide them furth.

Example:



subnets: 2^9

hosts in each: 2^7

Subnets cont:

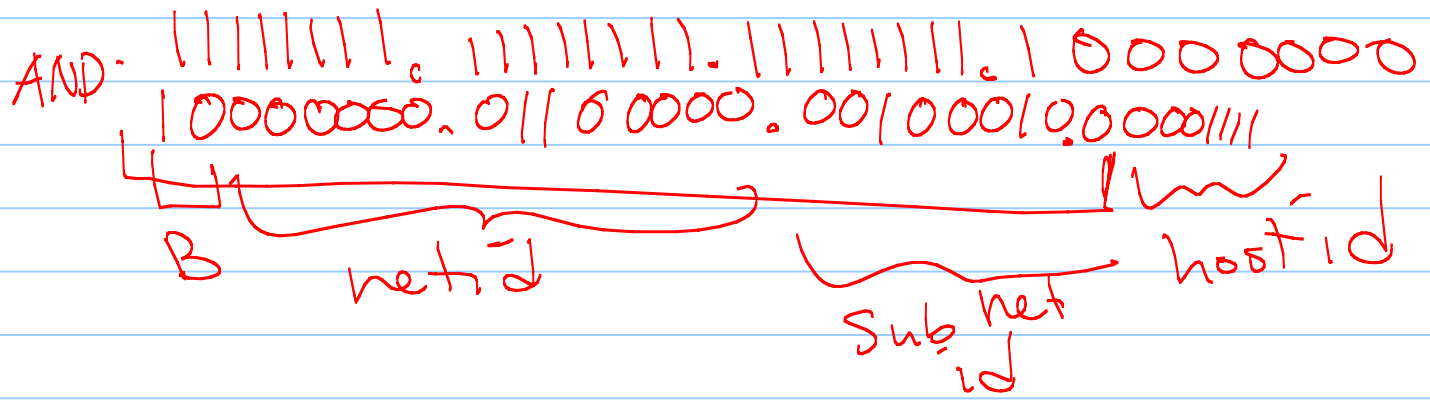
Every computer gets a subnet mask,
eg 255.255.255.128

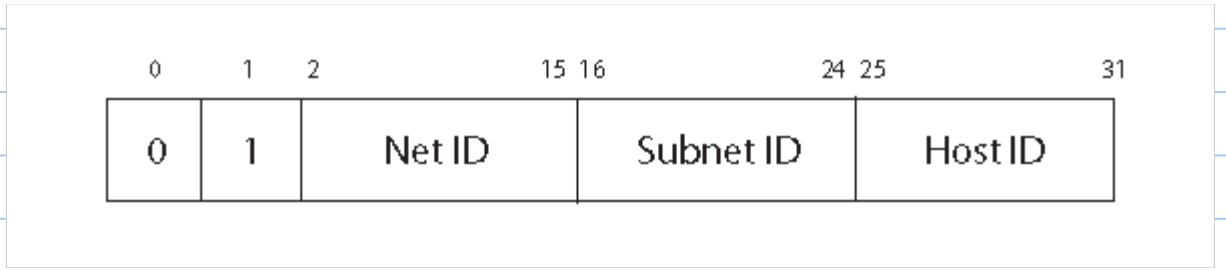
11111111.11111111.11111111.10000000

As well as an IP: 128.96.34.15

10000000.01100000.00100010.00001111

Take the bitwise AND to get the subnet versus host id:





Local Area Networks

A LAN is a "small interconnection infrastructure that typically uses a shared transmission medium".

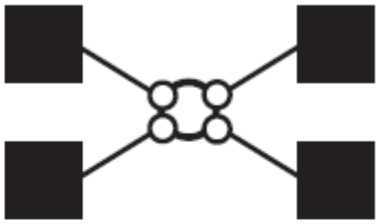
From Computer and Communication Networks by N. Mir

Note: A single LAN may actually be huge.

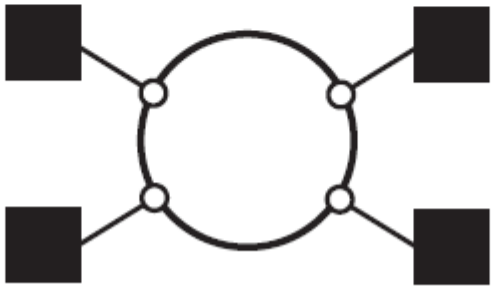
"Local" is relative, but generally these all connect to same router or switch.

LAN Topologies

Bus Configuration

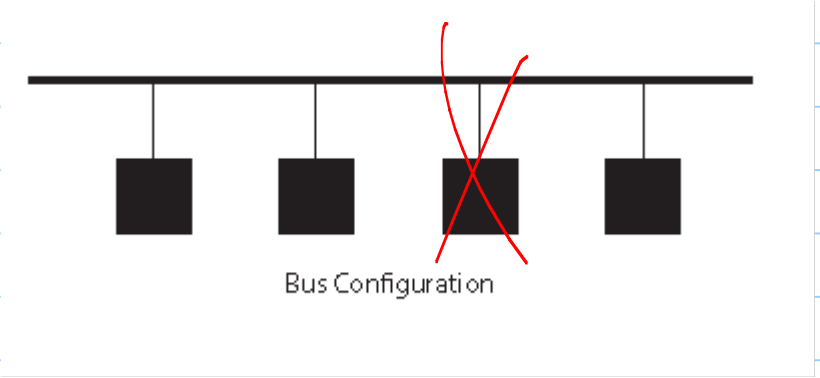


Star Configuration



Ring Configuration

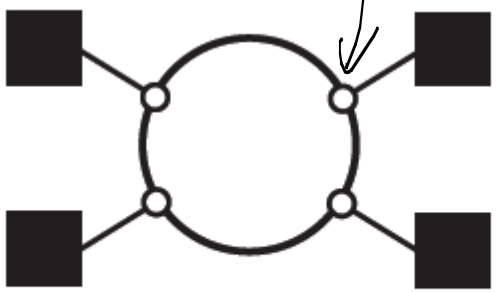
Bus Configuration



Transmissions are propagated on the "bus" in both directions.

All users (locally) will receive all packets.

Ring Configuration



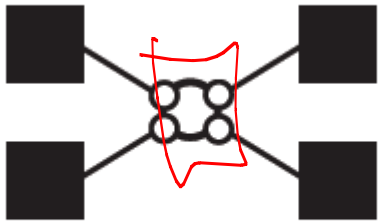
Ring Configuration

Sender gives packet to repeater.

Repeater forwards until reaches destination.

If reaches original source, not forwarded any further.

Star configuration



Star Configuration

Center of "star" is a multi-port hub or switch.

Frames are sent to center, which either broadcasts or sends to targeted destination.

Note:

None of these have much security!

All are vulnerable to attack and to eavesdropping.

Adding Security

To address this, we'll dive down a level to the Link layer.

(More next time!)