Networking details

ComS 252 — Iowa State University

Andrew Miner

How to configure network interfaces

- Easy way: run the GUI configuration tool
  - Works in Windows, Mac OS, and Linux
  - Dynamic IP address: super easy
    - Check "automatic" or "DHCP" and you're done
  - Static IP address: more work
    - You specify the IP address
    - You specify the Network or Subnet mask
    - You specify the Gateway or Router address
    - You specify the DNS address

How to configure network interfaces

- Easy way: run the GUI configuration tool
  - Works in Windows, Mac OS, and Linux
  - Dynamic IP address: super easy
    - Check “automatic” or “DHCP” and you’re done
  - Static IP address: more work
    - You specify the IP address
    - You specify the Network or Subnet mask
    - You specify the Gateway or Router address
    - You specify the DNS address
- No GUI in Linux? Try the TUI configuration tool
  - Just as easy as the GUI

How to configure network interfaces

- Easy way: run the GUI configuration tool
  - Works in Windows, Mac OS, and Linux
  - Dynamic IP address: super easy
    - Check "automatic" or "DHCP" and you're done
  - Static IP address: more work
    - You specify the IP address
    - You specify the Network or Subnet mask
    - You specify the Gateway or Router address
    - You specify the DNS address
- No GUI in Linux? Try the TUI configuration tool
  - Just as easy as the GUI
- Or use command-line tools . . .

Managing network connections

Utility: `nmcli`

- NetworkManager command line interface
- Can create, delete, display, edit network connections
  - Connections can be named
- Connections are associated with network devices
  - Need to know the device name
- Uses arguments as commands
  - Kind of like `systemctl`
  - No arguments? Shows detailed status of connections.

Network device names

Old way

- Ethernet interfaces are numbered arbitrarily
- `eth0` is the "first" ethernet interface
- `eth1` is the "second" ethernet interface...
- Issue: numbers may be assigned non-deterministically

"Consistent Network Device Naming" (since Fedora 15)

- Ethernet interfaces are named based on their physical location
- E.g.: `em3`, embedded interface in slot 3
- E.g.: `p3p1`, PCI slot 3 ethernet port 1

Using `nmcli`

- Commands start with an "object"
  - `connection`: for managing connections
  - `device`: for managing devices
  - `general`: general status
  - Others ...
- Any unique shortening of words may be used:

  `device`, `devic`, `devi`, `dev`, `de`, `d`
- Then specify what to do with the object
  - Default is usually `status`, to display status
- Can use "`help`" anywhere, for context sensitive help

Managing connections

What can come after `nmcli connection`

- `add`, `delete`: add or delete a connection
- `modify`: modify a connection
- `up`, `down`: bring up or shut down a connection

Connection settings

- `ipv4.method`
  - `manual` to manually specify (usually static IP)
  - `auto`, the default, for dynamic IP with DHCP
- `ipv4.address`: specify address and subnet
- `gw4`: specify gateway address

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
$ 
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
$ nmcli
```

Example for nmcli

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
$ nmcli
enp0s3:  connecting (getting IP configuration) to enp0s3
         "Intel 82540EM"
         ethernet (e1000), 08:00:27:AE:E4:B3, hw, mtu 15

lo:  unmanaged
         "lo"
         loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$ 
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
$ nmcli
enp0s3:  connecting (getting IP configuration) to enp0s3
         "Intel 82540EM"
         ethernet (e1000), 08:00:27:AE:E4:B3, hw, mtu 15

lo:  unmanaged
         "lo"
         loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$ nmcli dev
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
$ nmcli
enp0s3:  connecting (getting IP configuration) to enp0s3
         "Intel 82540EM"
         ethernet (e1000), 08:00:27:AE:E4:B3, hw, mtu 15

lo:  unmanaged
         "lo"
         loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$ nmcli dev
DEVICE  TYPE      STATE         CONNECTION
enp0s3  ethernet  disconnected  --
lo      loopback  unmanaged     --
$
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
$ nmcli
enp0s3:  connecting (getting IP configuration) to enp0s3
         "Intel 82540EM"
         ethernet (e1000), 08:00:27:AE:E4:B3, hw, mtu 15

lo:  unmanaged
         "lo"
         loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$ nmcli dev
DEVICE  TYPE      STATE         CONNECTION
enp0s3  ethernet  disconnected  --
lo      loopback  unmanaged     --
$ nmcli connecti
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
        "Intel 82540EM"
        ethernet (e1000), 08:00:27:AE:E4:B3, hw, mtu 15

lo:  unmanaged
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$ nmcli dev
DEVICE  TYPE      STATE         CONNECTION
enp0s3  ethernet  disconnected  --
lo      loopback  unmanaged     --
$ nmcli connecti
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  --
$
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
        "Intel 82540EM"
        ethernet (e1000), 08:00:27:AE:E4:B3, hw, mtu 15

lo:  unmanaged
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$ nmcli dev
DEVICE  TYPE      STATE         CONNECTION
enp0s3  ethernet  disconnected  --
lo      loopback  unmanaged     --
$ nmcli connecti
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  --
$ sudo nmcli c m enp0s3 ipv4.meth m ipv4.ad 10.3.3.3/24 gw4 10.3.3.1
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
        ethernet (e1000), 08:00:27:AE:E4:B3, hw, mtu 15

lo:  unmanaged
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$ nmcli dev
DEVICE  TYPE      STATE         CONNECTION
enp0s3  ethernet  disconnected  --
lo      loopback  unmanaged     --
$ nmcli connecti
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  --
$ sudo nmcli c m enp0s3 ipv4.meth m ipv4.ad 10.3.3.3/24 gw4 10.3.3.1
$
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
        ethernet (e1000), 08:00:27:AE:E4:B3, hw, mtu 15

lo:  unmanaged
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$ nmcli dev
DEVICE  TYPE      STATE         CONNECTION
enp0s3  ethernet  disconnected  --
lo      loopback  unmanaged     --
$ nmcli connecti
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  --
$ sudo nmcli c m enp0s3 ipv4.meth m ipv4.ad 10.3.3.3/24 gw4 10.3.3.1
$ nmcli conn
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
          "lo"
          loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$ nmcli dev
DEVICE  TYPE      STATE         CONNECTION
enp0s3  ethernet  disconnected  --
lo      loopback  unmanaged     --
$ nmcli connecti
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  --
$ sudo nmcli c m enp0s3 ipv4.meth m ipv4.ad 10.3.3.3/24 gw4 10.3.3.1
$ nmcli conn
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  enp0s3
$ 
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$ nmcli dev
DEVICE  TYPE      STATE         CONNECTION
enp0s3  ethernet  disconnected  --
lo      loopback  unmanaged     --
$ nmcli connecti
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  --
$ sudo nmcli c m enp0s3 ipv4.meth m ipv4.ad 10.3.3.3/24 gw4 10.3.3.1
$ nmcli conn
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  enp0s3
$ nmcli d
```

Example for nmcli

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
enp0s3  ethernet  disconnected  --
lo      loopback  unmanaged     --
$ nmcli connecti
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  --
$ sudo nmcli c m enp0s3 ipv4.meth m ipv4.ad 10.3.3.3/24 gw4 10.3.3.1
$ nmcli conn
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  enp0s3
$ nmcli d
DEVICE  TYPE      STATE      CONNECTION
enp0s3  ethernet  connected  enp0s3
lo      loopback  unmanaged  --
$
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
enp0s3  ethernet  disconnected  --
lo      loopback  unmanaged     --
$ nmcli connecti
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  --
$ sudo nmcli c m enp0s3 ipv4.meth m ipv4.ad 10.3.3.3/24 gw4 10.3.3.1
$ nmcli conn
NAME    UUID                                  TYPE      DEVICE
enp0s3  451df638-ef6d-3884-92d5-77a5b83837cf  ethernet  enp0s3
$ nmcli d
DEVICE  TYPE      STATE      CONNECTION
enp0s3  ethernet  connected  enp0s3
lo      loopback  unmanaged  --
$ nmcli
```

Example for `nmcli`

- Network device currently using DHCP
- Needs to be set statically
  - IP address and subnet is 10.3.3.3/24
  - Gateway address is 10.3.3.1

```
DEVICE   TYPE      STATE      CONNECTION
enp0s3   ethernet  connected  enp0s3
lo       loopback  unmanaged  --
$ nmcli
enp0s3:  connected to enp0s3
         "Intel 82540EM"
         ethernet (e1000), 08:00:27:AE:E4:B3, hw, mtu 15
         inet4 10.3.3.3/24
         route4 10.3.3.0/24

lo:  unmanaged
         "lo"
         loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
$
```

Other configuration (1)

`/etc/resolv.conf`

`search` `network names`:

- Try these suffixes to form a FQDN

`nameserver` `IP address`:

- Specify an IP address to use as a DNS server
- Can have several entries of this form

This file is usually constructed by the DHCP client

Other configuration (2)

/etc/hosts

- List of IP addresses and associated host names
- Linux looks here before asking DNS servers
- Useful to set up a few local host names
  - Alternative: set up your own DNS server
  - We will discuss how to do that, later

```
prompt$
```

Other configuration (2)

/etc/hosts

- List of IP addresses and associated host names
- Linux looks here before asking DNS servers
- Useful to set up a few local host names
  - Alternative: set up your own DNS server
  - We will discuss how to do that, later

```
prompt$ cat /etc/hosts
```

Other configuration (2)

/etc/hosts

- List of IP addresses and associated host names
- Linux looks here before asking DNS servers
- Useful to set up a few local host names
  - Alternative: set up your own DNS server
  - We will discuss how to do that, later

```
prompt$ cat /etc/hosts
127.0.0.1       localhost localhost.localdomain
192.168.1.1     router router.localdomain
192.168.1.100   server server.localdomain
prompt$
```

Linux Firewall

- Packet filtering is done in the kernel
- We can filter packets based on
  - Port number(s)
  - Protocol (e.g., UDP or TCP)
  - Sender and/or recipient IP address
  - Combinations of these

Linux Firewall

- Packet filtering is done in the kernel
- We can filter packets based on
  - Port number(s)
  - Protocol (e.g., UDP or TCP)
  - Sender and/or recipient IP address
  - Combinations of these
- Management of packet filtering is done in user space
  - Starting, stopping, changing filtering rules
- Different distributions use different systems
  - Similar to system initialization differences
    e.g., `sysvinit` vs. `systemd` vs. others
- This lecture: brief discussion of `firewalld`
  - Default for RHEL 7, Fedora 18 and later, CentOS 7
  - Available (e.g., as a package) for other distributions

`firewalld`

- User-space system for dynamic firewall management
- Can start or stop `firewalld` like any other service
  - Stopped: no filtering rules!
- Supports different *zones*
  - Typically each network connection belongs to a zone
  - Can have different rules in different zones
- Supports *services*
  - Can set up (or use default) filtering rules for services e.g., "HTTP: Allow port 80 TCP packets"
- Supports *run-time* or *permanent* rule changes
  - Run-time: just for now, can have timeout
  - Permanent: every time the system boots
- Different configuration front-ends
  - `firewall-config`: GUI configuration tool
  - `firewall-cmd`: command-line configuration tool

Firewall management with `firewall-cmd`

- Many arguments and switches (check `man` pages)
- Some useful "command" arguments:
  - `--add-service=foobar`
    Allow packets for service `foobar`
  - `--remove-service=foobar`
    Deny packets for service `foobar`
  - `--list-services`
    Show allowed services
- Some useful switches:
  - `--zone=zonename`
    For the specified zone (otherwise, uses default)
  - `--permanent`
    For the permanent rules (otherwise, run-time)
- We will discuss this in more detail, later

`firewall-cmd` example

```
prompt$
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
success
prompt$
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
success
prompt$ firewall-cmd --list-services
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
success
prompt$ firewall-cmd --list-services
http ssh
prompt$
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
success
prompt$ firewall-cmd --list-services
http ssh
prompt$ firewall-cmd --remove-service=http
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
success
prompt$ firewall-cmd --list-services
http ssh
prompt$ firewall-cmd --remove-service=http
success
prompt$
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
success
prompt$ firewall-cmd --list-services
http ssh
prompt$ firewall-cmd --remove-service=http
success
prompt$ firewall-cmd --list-services
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
success
prompt$ firewall-cmd --list-services
http ssh
prompt$ firewall-cmd --remove-service=http
success
prompt$ firewall-cmd --list-services
ssh
prompt$
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
success
prompt$ firewall-cmd --list-services
http ssh
prompt$ firewall-cmd --remove-service=http
success
prompt$ firewall-cmd --list-services
ssh
prompt$ firewall-cmd --list-services --permanent
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
success
prompt$ firewall-cmd --list-services
http ssh
prompt$ firewall-cmd --remove-service=http
success
prompt$ firewall-cmd --list-services
ssh
prompt$ firewall-cmd --list-services --permanent
mdns dhcpv6-client http
prompt$
```

firewall-cmd example

```
prompt$ firewall-cmd --add-service=ssh
success
prompt$ firewall-cmd --list-services
http ssh
prompt$ firewall-cmd --remove-service=http
success
prompt$ firewall-cmd --list-services
ssh
prompt$ firewall-cmd --list-services --permanent
mdns dhcpv6-client http
prompt$ firewall-cmd --add-service=ssh --permanent
```

firewall-cmd example

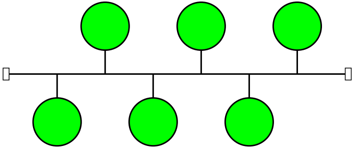
```
prompt$ firewall-cmd --add-service=ssh
success
prompt$ firewall-cmd --list-services
http ssh
prompt$ firewall-cmd --remove-service=http
success
prompt$ firewall-cmd --list-services
ssh
prompt$ firewall-cmd --list-services --permanent
mdns dhcpv6-client http
prompt$ firewall-cmd --add-service=ssh --permanent
success
prompt$
```

Topology

Choice of LAN topology:

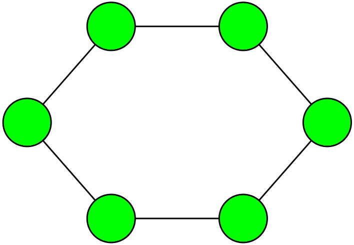
- May be based on available hardware
  - We will discuss some obsolete technologies
  - But they may still be in use
- May dictate the required protocols
- Will affect network performance
- Will affect ease of network maintenance
- We will discuss some simple LAN topologies
  - Other topologies are possible
  - Can combine simple toplogies to get complex ones

Bus Topology



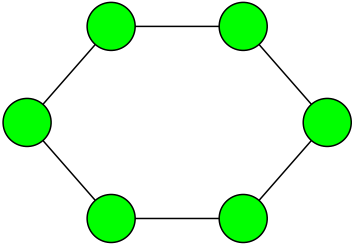
- All machines connected to a single bus cable
- The cable must have terminators on each end
  - These "absorb" the signal and prevent reflections
- When a machine sends a frame
  - All machines see the frame
  - Only the intended recipient keeps the frame
  - All other machines ignore the frame
- Broken cable — entire LAN goes down

Ring Topology



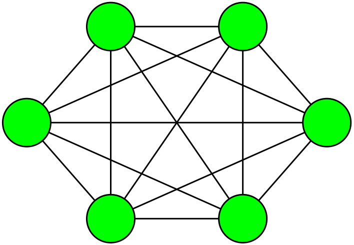
- 2 or more hosts are arranged in a circle
- Data travels in one direction around the ring
  - Use two rings for bi-directional communication
- Widely used with Fiberoptic Cable
- Better than Bus Topology for heavy load
- Token ring

Ring Topology

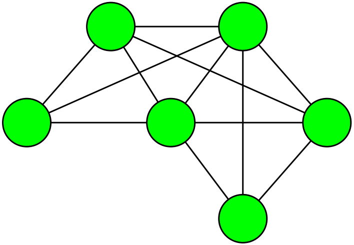


- 2 or more hosts are arranged in a circle
- Data travels in one direction around the ring
  - Use two rings for bi-directional communication
- Widely used with Fiberoptic Cable
- Better than Bus Topology for heavy load
- Token ring is NOT a ring topology

Mesh network



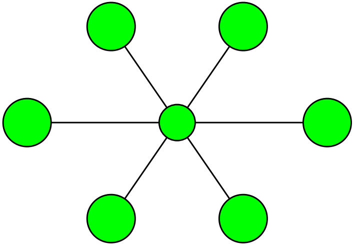
A fully–connected mesh



A partially–connected mesh

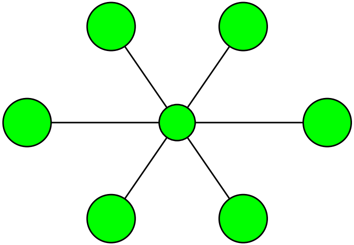
- Central nodes are connected to one or more other nodes
- Data must be “routed” to its destination
- Often used for wireless networks
  - Especially without centralized “base stations”
  - Links between machines are based on wireless signal strength

Star network



- Common LAN topology today
- Machines are connected to a central “hub”
  - May be a computer
  - May be dedicated hardware; e.g.:
    - Ethernet switch, wireless hub

Star network



- Common LAN topology today
- Machines are connected to a central "hub"
  - May be a computer
  - May be dedicated hardware; e.g.:
    - Ethernet switch, wireless hub, Token ring media access unit
- Any machine (except hub) may fail without affecting others

Token ring technology

- Can be used with star or bus topologies
- One or more frames continuously circle the LAN
  - Some are empty
  - Some contain messages
- To receive
  1. Examine destination field of every frame
  2. If a frame is for me, copy the message and clear the token bit
- To transmit
  1. Wait for an empty frame
  2. Set the token bit in the frame
  3. Insert message and destination into the frame
  4. Wait until the frame comes around again
  5. The token bit will be cleared if it was received
  6. Remove message from the frame

Ethernet — brief history

Invention

- Invented at Xerox PARC between 1973 and 1974
- Patent filed in 1975 lists inventors:

  Robert Metcalfe, David Boggs, Chuck Thacker, Butler Lampson
- 1979: Metcalfe leaves Xerox and forms 3Com

Technology

- Inspired by ALOHAnet
  - First wireless packet data network
  - Developed at University of Hawaii
- Originally used shared transmission medium
  - Coaxial cable to carry signals
- Name comes from old physics term "ether"

Ethernet — Shared medium technology

- Ethernet frame:
  - Frame header contains sender and recipient
    - As Media Access Control (MAC) addresses
  - Frame payload is between 42 and 1500 bytes
- Like token ring:
  - All machines see all transmitted frames
  - Intended recipient copies the frame
  - Everyone else ignores the frame
- Unlike token ring, machines may send frames at any time
- But, machines need to detect collisions
  - Happens when two machines try to send at the same time
  - Possible because speed of light is finite and cables have length
- CSMA/CD is the technical term for this
  - Carrier sense multiple access (with) Collision detection
- Senders "back off" when collision is detected
  - Wait for random time and resend

Ethernet — Cable evolution

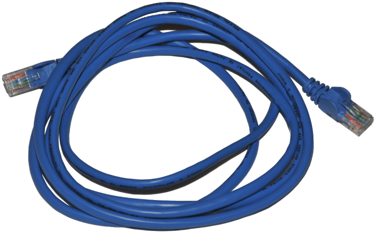
10BASE5 : Thick coaxial cable (a.k.a. "thicknet")
- Early 1980's
- 10 Mbit/s
- Baseband signalling
- 500 meter maximum cable length
- Need one long cable
- Cables needed terminators

10BASE2 : Thinner coaxial cable (a.k.a. "thinnet")
- Mid to late 1980's
- 185 ≈ 200 meter maximum cable length
- One cable but allowed BNC T–connectors
- Cables needed terminators

10BASE-T : Twisted–pair cables
- Used today

Ethernet — Cable evolution

Thicknet cable



Twisted pair cable

Base T Ethernet

How do we get "one" cable from several twisted pair cables?

Base T Ethernet

How do we get "one" cable from several twisted pair cables?

Ethernet hub

- Has several ports for Base T connections
  - Not to be confused with "session layer" ports
- Incoming frame on one port is copied onto all other ports

Base T Ethernet

How do we get "one" cable from several twisted pair cables?

Ethernet hub

- Has several ports for Base T connections
  - Not to be confused with "session layer" ports
- Incoming frame on one port is copied onto all other ports

Ethernet switch

- Has several ports for Base T connections
- Incoming frame on one port is copied only to recipient port
- This can happen between different port pairs simultaneously

Connecting LANs together

Why connect two LANs together? Why not one big LAN?

Connecting LANs together

Why connect two LANs together? Why not one big LAN?

- Geographical constraints
  - E.g., LANs are in different buildings
- Different network types
  - E.g., 10BASE2 ethernet and 100BASE-T ethernet
- Limits for each LAN
  - At most 100 nodes per 10BASE5 ethernet segment
  - At most 30 nodes per 10BASE2 ethernet segment
- Might have better performance
  - Fewer nodes per ethernet segment means fewer frame collisions
- May be easier to administer separately

Connecting LANs together

Why connect two LANs together? Why not one big LAN?

- Geographical constraints
  - E.g., LANs are in different buildings
- Different network types
  - E.g., 10BASE2 ethernet and 100BASE-T ethernet
- Limits for each LAN
  - At most 100 nodes per 10BASE5 ethernet segment
  - At most 30 nodes per 10BASE2 ethernet segment
- Might have better performance
  - Fewer nodes per ethernet segment means fewer frame collisions
- May be easier to administer separately

Ok, so how do we connect two LANs?

Connecting LANs together

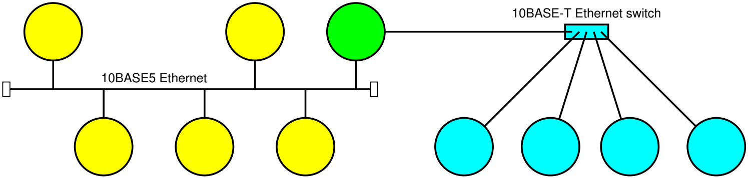
Why connect two LANs together? Why not one big LAN?

- Geographical constraints
  - E.g., LANs are in different buildings
- Different network types
  - E.g., 10BASE2 ethernet and 100BASE-T ethernet
- Limits for each LAN
  - At most 100 nodes per 10BASE5 ethernet segment
  - At most 30 nodes per 10BASE2 ethernet segment
- Might have better performance
  - Fewer nodes per ethernet segment means fewer frame collisions
- May be easier to administer separately

Ok, so how do we connect two LANs?

- Connect one machine to both LANs...

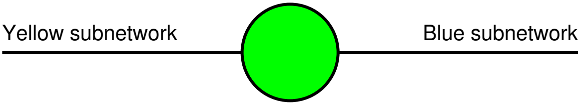
Network with 2 subnetworks



The green machine:

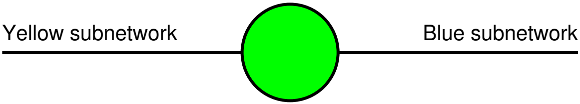
- Belongs to both LANs
- Has two network connections
  - Each will have its own name
- Is a known as a gateway
  - Because it is an entrance to another network

Gateway machines



- Can be routers
  - When they forward network packets to another subnetwork
  - May make decisions about where to forward network packets
    - Especially if the gateway is connected to 3 or more LANs
- Can be firewalls
  - E.g., certain packets from blue subnetwork are not forwarded to yellow subnetwork
  - Good way to restrict traffic for an entire subnetwork
- We will discuss this more, later

Gateway machines



- Can be routers
  - When they forward network packets to another subnetwork
  - May make decisions about where to forward network packets
    - Especially if the gateway is connected to 3 or more LANs
- Can be firewalls
  - E.g., certain packets from blue subnetwork are not forwarded to yellow subnetwork
  - Good way to restrict traffic for an entire subnetwork
- We will discuss this more, later
- Clever use of gateway machines is why the Internet works

ARPANET (1)

DARPA: Defense Advanced Research Projects Agency

- The "D" comes and goes over the years
- Started in 1958 in response to Sputnik launch
- Original mission: "prevent technological surprises"

Intergalactic Computer Network

- Idea of J. C. R. Licklider in early 1960's
- Bob Taylor took over in 1965
  - Had three separate terminals in his office
    - CTSS/MULTICS at MIT
    - Project Genie at UC Berkeley
    - System Development Corporation Q–32
  - Wanted to instead have one terminal . . .

Packet switching

- Developed by a few people in early 1960's
  - Paul Baran for the RAND Corporation
  - Donald Davies at the National Physical Laboratory (UK)
  - Leonard Kleinrock worked out mathematical models
- Before then: everything used circuit switching
  - Used in analog telephone networks
  - Establishes a dedicated channel from end to end
- Better fault tolerance than circuit switching
  - Computers were not as reliable as now
- More efficient use of bandwidth than circuit switching

ARPANET (2)

- 1968: Taylor completes plan for the ARPANET
  - Using packet switching
- 1969: First two machines connected
  - UCLA (Leonard Kleinrock's "Network Measurement Center")
  - Stanford Research Institute's Augmentation Research Center
  - First message sent was "`lo`"
    - Message was "`login`" but system crashed before "g"
  - NCP (Network Control Program) was the core protocol
- December 1969: initial 4–machine ARPANET is complete
  - UC Santa Barbara
  - University of Utah
- 1975: 57 machines; network is declared "operational"
  - Control handed over to Defense Communications Agency
  - Not a research project anymore!
- 1981: 213 machines
- February 1990: formally decommissioned

Meanwhile, also at DARPA. . .

- 1972: Robert Kahn was exploring other technologies
  - Satellite and ground–based radio packet networks
  - Different physical and data link layer protocols . . .
  - But those concepts did not exist yet
- Kahn realized the value of mixing technologies
- 1973: Robert Kahn and Vinton Cerf reformulated everything
  - A common "internetwork protocol" is used on top
  - Differences between network protocols are hidden
  - Now, machines are responsible for reliability, not the network
- 1974: Initial TCP/IP protocol drafted by Cerf and others
  - RFC 675 — Specification of Internet Transmission Control Program
- Early versions (0 — 3) of IP were used until 1979
- IPv4 released 1981, adopted, still used today
  - RFC 791 — Internet Protocol
- January 1, 1983: ARPANET stops supporting NCP

Aside — what's an RFC?

- Request For Comments
- Invented in 1969 by Steve Crocker
  - To record "unofficial notes" on ARPANET development
  - Actually encouraged comments from other researchers
- Now: RFCs are more "final"
  - The "official record" for Internet stuff
    - Specifications, protocols, procedures, etc.
- But: not all RFCs are standards
- Official source is `http://www.rfc-editor.org/rfc.html`
- Or use the IETF (Internet Engineering Task Force) site `http://www.ietf.org/rfc.html`
- Lecture slides link to IETF because those have hyperlinks

Network growth: 1985 — 1995

- Other agencies started developing networks
  - NASA (National Aeronautics and Space Agency) networks:
    - NSN (NASA Science Network)
    - Merged with SPAN (Space Physics Analysis Network)
    - Became NSI (NASA Science Internet)
  - NSF (National Science Foundation) networks:
    - CSNET (Computer Science Network)
    - NSFNET
  - DOE (Department of Energy) network:
    - ESNet (Energy Sciences Network)
- TCP/IP becomes standard for interconnecting these
  - Expression: you can run TCP/IP over two tin cans and a string
  - Or, over carrier pigeons
    - See RFC 1149 and note the date
    - People tried this: `http://www.blug.linux.no/rfc1149/`
- The rest of the world was paying attention; e.g.:
  - CERN starts supporting TCP/IP in 1989
  - NSFNET connected to Japan's JUNET in 1989

So, who invented the Internet?

So, who invented the Internet?

- Licklider, Taylor, and others at DARPA
- Baran, Davies, Kleinrock, and others for packet switching
- Kahn, Cerf, and others for TCP/IP
- For more: `http://www.internetsociety.org/internet/internet-51/history-internet`

So, who invented the Internet?

- Licklider, Taylor, and others at DARPA
- Baran, Davies, Kleinrock, and others for packet switching
- Kahn, Cerf, and others for TCP/IP
- For more: `http://www.internetsociety.org/internet/internet-51/history-internet`

What about Al Gore?

So, who invented the Internet?

- Licklider, Taylor, and others at DARPA
- Baran, Davies, Kleinrock, and others for packet switching
- Kahn, Cerf, and others for TCP/IP
- For more: `http://www.internetsociety.org/internet/internet-51/history-internet`

What about Al Gore?

High Performance Computing Act of 1991

- Created and introduced by Senator Albert Gore, Jr.
- Led to the National Information Infrastructure
  - I.e., several interconnected public and private networks
  - Called the “Information Superhighway”
- Built on prior infrastructure like ARPANET, NSFNET

So, who invented the Internet?

- Licklider, Taylor, and others at DARPA
- Baran, Davies, Kleinrock, and others for packet switching
- Kahn, Cerf, and others for TCP/IP
- For more: `http://www.internetsociety.org/internet/internet-51/history-internet`

What about Al Gore?

High Performance Computing Act of 1991

- Created and introduced by Senator Albert Gore, Jr.
- Led to the National Information Infrastructure
  - I.e., several interconnected public and private networks
  - Called the "Information Superhighway"
- Built on prior infrastructure like ARPANET, NSFNET

Here's what Kahn and Cerf say about Gore's role:

- `http://web.eecs.umich.edu/~fessler/misc/funny/gore,net.txt`

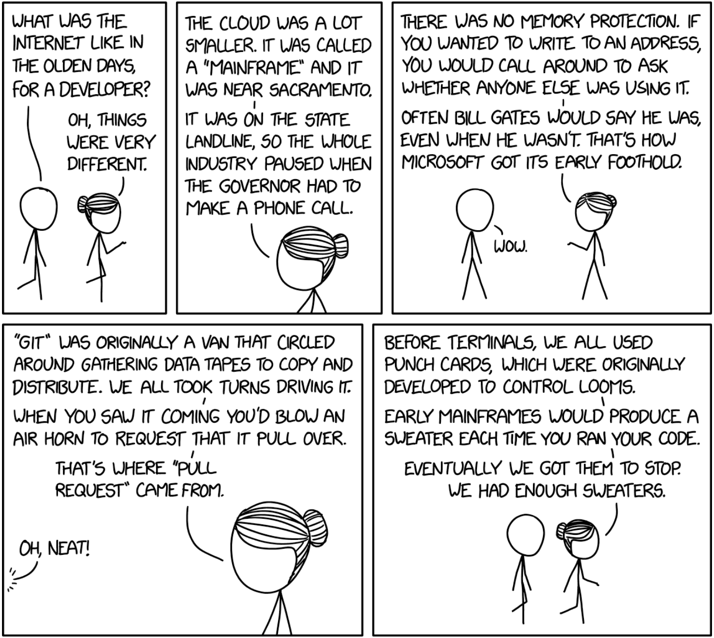
Summary of today's commands

`firewall-cmd` : Command-line firewall configuration

`firewall-config` : GUI firewall configuration

`nmcli` : Network configuration (command-line)

An appropriate xkcd comic: `http://www.xkcd.com/2324`



End of lecture