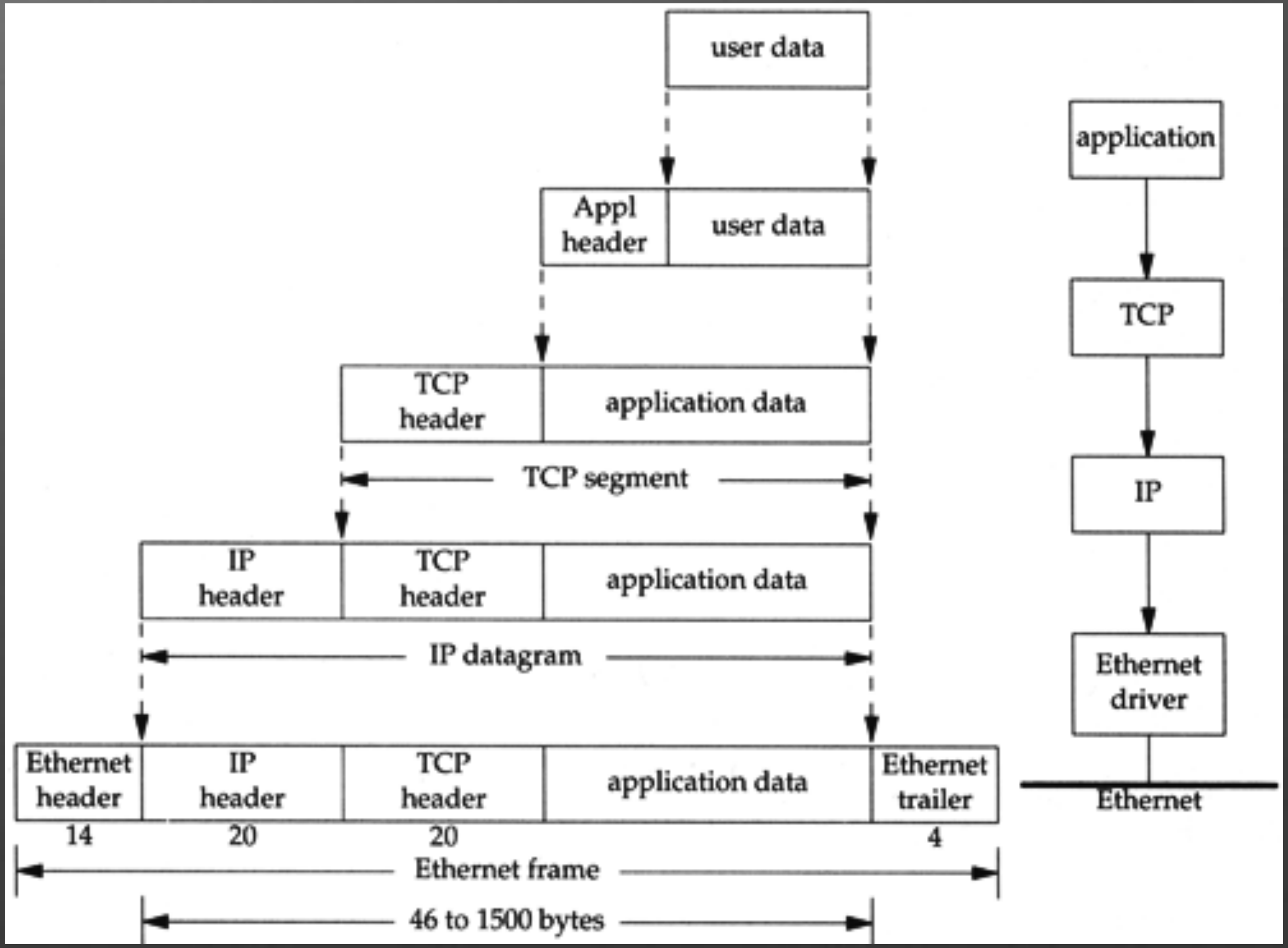
Linux Network Configuration

Linux System Administration
COMP2018 Summer 2017

TCP/IP

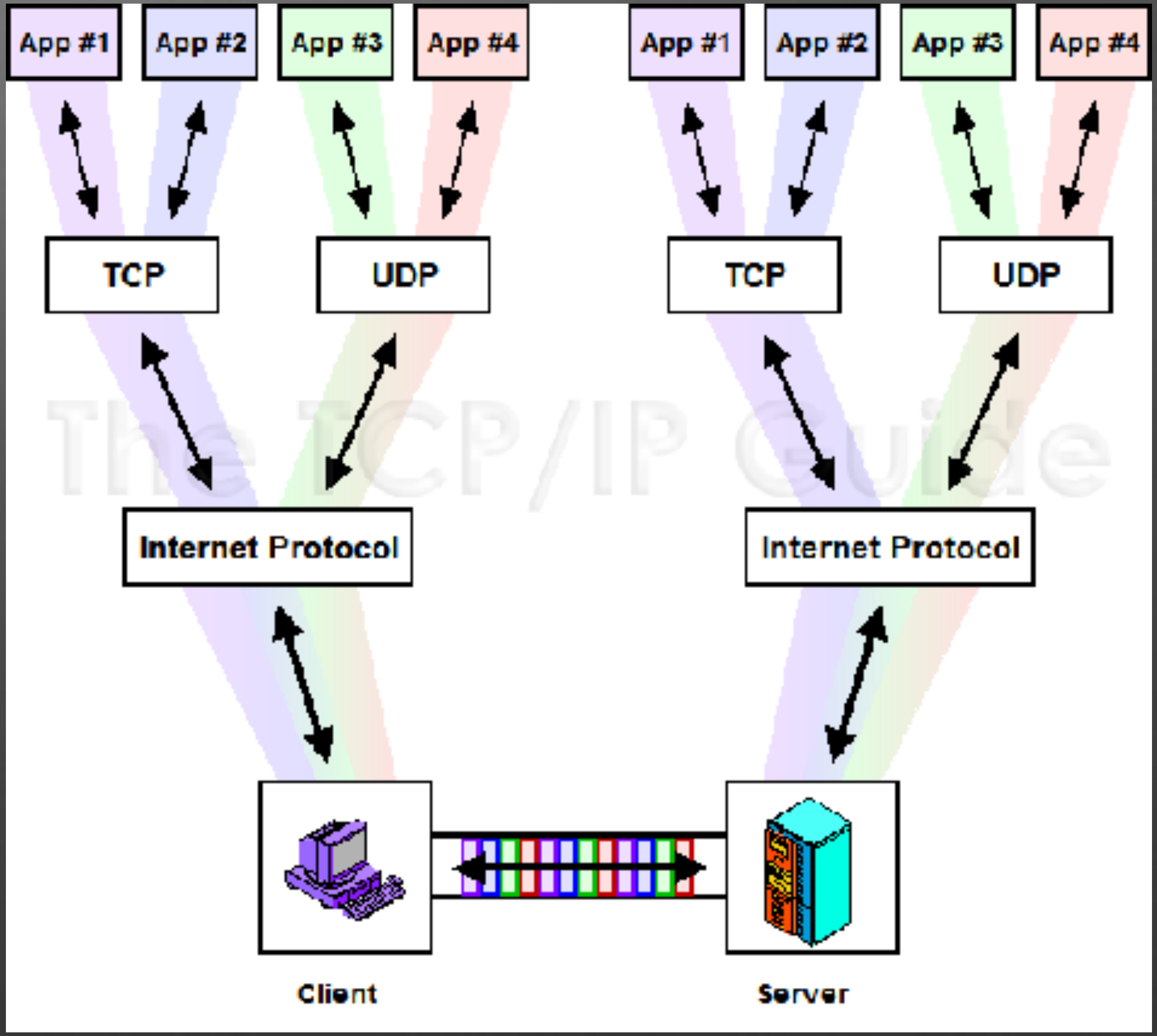
- TCP/IP is a suite of protocols and data formats that enable communications

- It makes use of data encapsulation to allow separation of functionality into layers, abstracting the details of each layer from the others so that the most suitable technologies can be used for each part of the communications task



Addressing

- An IP address uniquely identifies a communications end point host machine
- You need to know your destination's IP address if you want to communicate with them
- TCP/IP uses a combination of IP address (V4 or V6), protocol (TCP, UDP, etc.), and port (0-65535) to fully identify a communications end point (sometimes called a socket, used by a process)



Network Connections

- Processes establish network connections
- The kernel keeps tables of current connections
- The netstat command lets us view these tables
- netstat -t shows TCP connections
- netstat -u shows UDP connections
- netstat -l shows processes with an open connection end point listening for connections
- netstat -p shows which processes have which connections
- These options can be combined, used with -n to show numbers instead of names for addresses and ports, and there are many more options for netstat

Host Names vs. Addresses

- Addresses for TCP/IP are either 4 or 16 bytes (IPV4 vs. IPV6)

```
192.168.1.1
127.0.0.1
2607:f8b0:400b:80a::200e
```

- We use names instead of addresses for convenience and flexibility

```
router
localhost
google.com
```

- Our host's primary name is kept in /etc/hostname and is used to set the name of the machine at boot

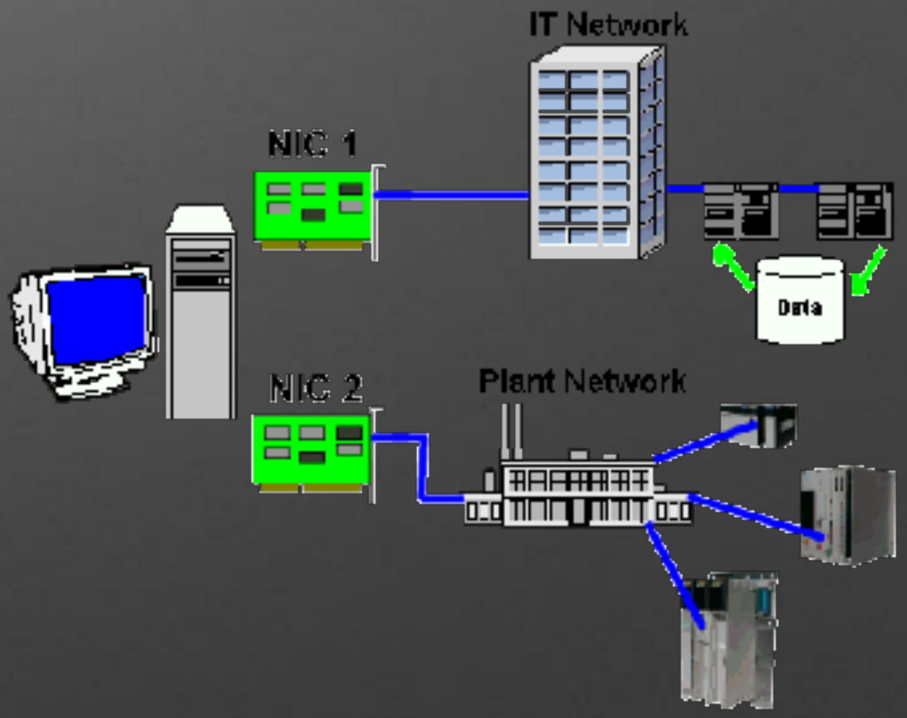
```
myhostname
```

- Name to address conversion (known as resolving names) is automated using /etc/hosts

```
192.168.1.1                 router
2607:f8b0:400b:80a::200e  google.com
127.0.0.1                   localhost
```

Interfaces

- Interfaces provide a way to connect a computer to a network for the purpose of communication between processes which may be on separate computers
- Interfaces may be physical (e.g. ethernet or wireless) or virtual (bridge, tunnel, virtual network, loopback)



Physical Interfaces

- Ethernet is the most common at speeds of 100Mb, 1Gb, 10Gb and is an implementation of the IEEE 802.3 communications standard
- Wi-fi is the next most common in many different speeds and implements various IEEE 802.11 standards (a/b/g/n/ac/ad)
- Other physical connection technologies include fibre, bluetooth, coaxial cable, DSL, dialup, etc.
- Physical interface names are composed of a short name for a driver and an instance number, e.g. eth0, ens33, eno1, wlan0

Interface Identification

- lshw is a command that shows the hardware which Linux has recognized on our system
- It can be limited to the network hardware devices, making it easier to find the device names to use for networking commands

```
# lshw -class network
  *-network
       description: Ethernet interface
       product: 82579LM Gigabit Network Connection
       vendor: Intel Corporation
      logical name: eno1
      size: 1Gbit/s
       capacity: 1Gbit/s
      capabilities: pm msi bus_master cap_list ethernet physical tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
       configuration: autonegotiation=on broadcast=yes driver=e1000e driverversion=3.2.6-k duplex=full firmware=0.13-4 latency=0 link=yes multicast=yes port=twisted pair speed=1Gbit/s
       resources: irq:26 memory:f7c00000-f7c1ffff memory:f7c39000-f7c39fff ioport:f080(size=32)
  *-network DISABLED
       description: Ethernet interface
      logical name: virbr0-nic
      size: 10Mbit/s
       capabilities: ethernet physical
       configuration: autonegotiation=off broadcast=yes driver=tun driverversion=1.6 duplex=full link=no multicast=yes port=twisted pair speed=10Mbit/s
```

Working With Interfaces

- Network interfaces have a physical configuration and a logical configuration
- Physical configuration is usually automatic (e.g. network speed, flow control, MAC address, etc.)
- Logical configuration is the assignment of one or more IP addresses to the interface with a netmask so that packets to/from the connection can be handled properly
- Addresses are dynamically assigned to interfaces using the ifconfig or ip commands
- Both of these are used regularly to view network interface logical configurations, changing configurations is normally done by modifying configuration files and applying those changes

Interface Control

- ifup and ifdown are used to turn interfaces on and off based on the configuration file(s), and normally only happens at system boot and system shutdown
- /etc/network/interfaces contains the default configurations for non-GUI managed interfaces
- GUIs change the rules and make up their own rules
- ifup -a runs at boot

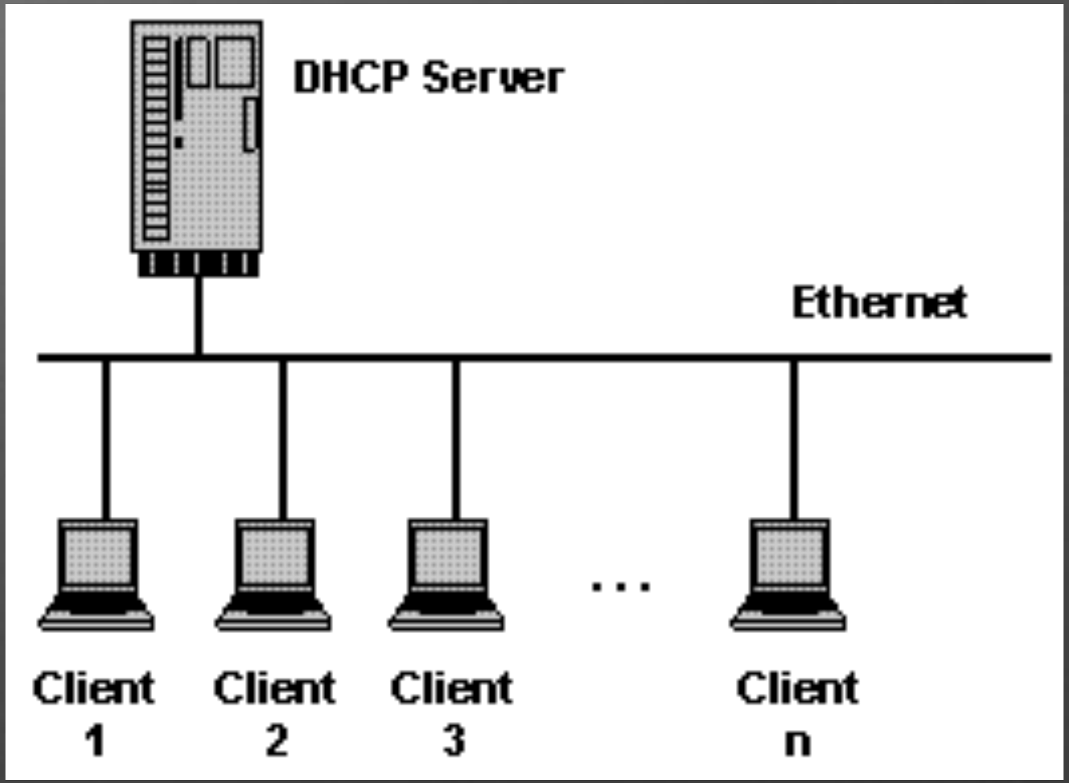
/etc/network/interfaces

- Two main lines control each interface
- auto *interfacename*
- iface *interfacename* inet **static|dhcp|manual|loopback**
- Additional lines are used for static, minimum is address and netmask
- Servers are normally statically configured, desktops are typically dynamically configured

```
auto lo eth0 eth1 eth1:1
iface lo inet loopback

iface eth0 inet dhcp

iface eth1 inet static
    address 192.168.1.2/24
    gateway 192.168.1.1

iface eth1:1 inet dhcp
```

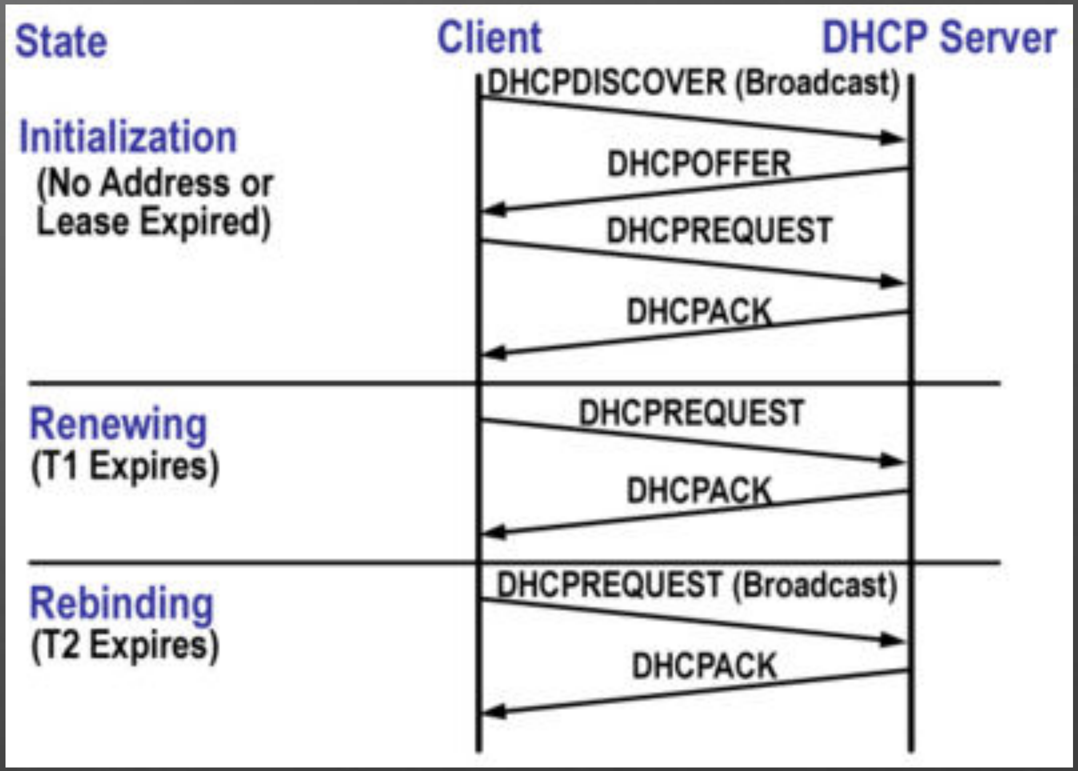
DHCP Concepts

- DHCP, the Dynamic Host Configuration Protocol, is used to provide automated interface configuration to hosts on a network
- A dhcp client retrieves information from a dhcp server
- Clients can choose what to use from the server without telling the server what they used



DHCP Client

- Default configuration invoked by using the dhcp keyword on the iface line in /etc/network/interfaces
- Retrieves and by default uses IP address and netmask, DNS server(s), default gateway
- Expects server to be directly connected on LAN, DHCP uses broadcast
- Check /var/log/syslog when things aren't working right



DHCP Server

- Server provides configuration in an advisory role
- Can supply many things, usually just ip address and netmask from a pre-defined range, default gateway, and DNS server(s)
- Can provide local data, or perform proxy service for a remote dhcp server
- Configurations are provided to clients on a lease basis, they expire if not renewed by the client on a regular basis

Route Table

- The route table is a table in memory used by the kernel whenever it is asked to send data using TCP/IP
- It provides a list of destinations and identifies where to send data so that it can reach those destinations
- Traffic for directly connected hosts is sent to the interface for that network
- Traffic for remote destinations is sent to a router that knows how to reach the destination network
- A default route is usually set up so that data packets not destined for local networks can be sent to remote hosts such as those found on the internet

```
# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.3.1     0.0.0.0         UG    0      0        0 br0
10.23.134.0     0.0.0.0         255.255.255.0   U     0      0        0 lxdbr0
192.168.3.0     0.0.0.0         255.255.255.0   U     0      0        0 br0
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
```

Working With the Route Table

- netstat -r shows the route table, similar to route
- route is used to add routes (route add), delete routes (route delete), and show the route table
- traceroute sends packets to a destination with increasing TTL values, causing them to fail at the first router, then the next and so on
- The diagnostic packets coming back from each router allows traceroute to display a list of the routers that handled the packets on the way
- The -n option can be used on these commands to show addresses instead of hostnames

File Sharing

- Network-based file sharing involves mounting a filesystem similarly to how disk-based filesystems are mounted
- The difference is that the name of the device mounted is not a disk device or file, but a network share address
- Different network sharing protocols have different formats for specifying the server and shared filesystem name, NFS is the UNIX/Linux native file sharing protocol
- Persistence for network filesystem mounts is achieved by putting them into /etc/fstab
- NFS source filesystems are specified as server:/path/to/mount
- CIFS source filesystems (Microsoft) are specified as //server/sharename and you need a username option for the typical Windows user-based security model, and you will need the cifs-utils package installed to use the mount command with cifs