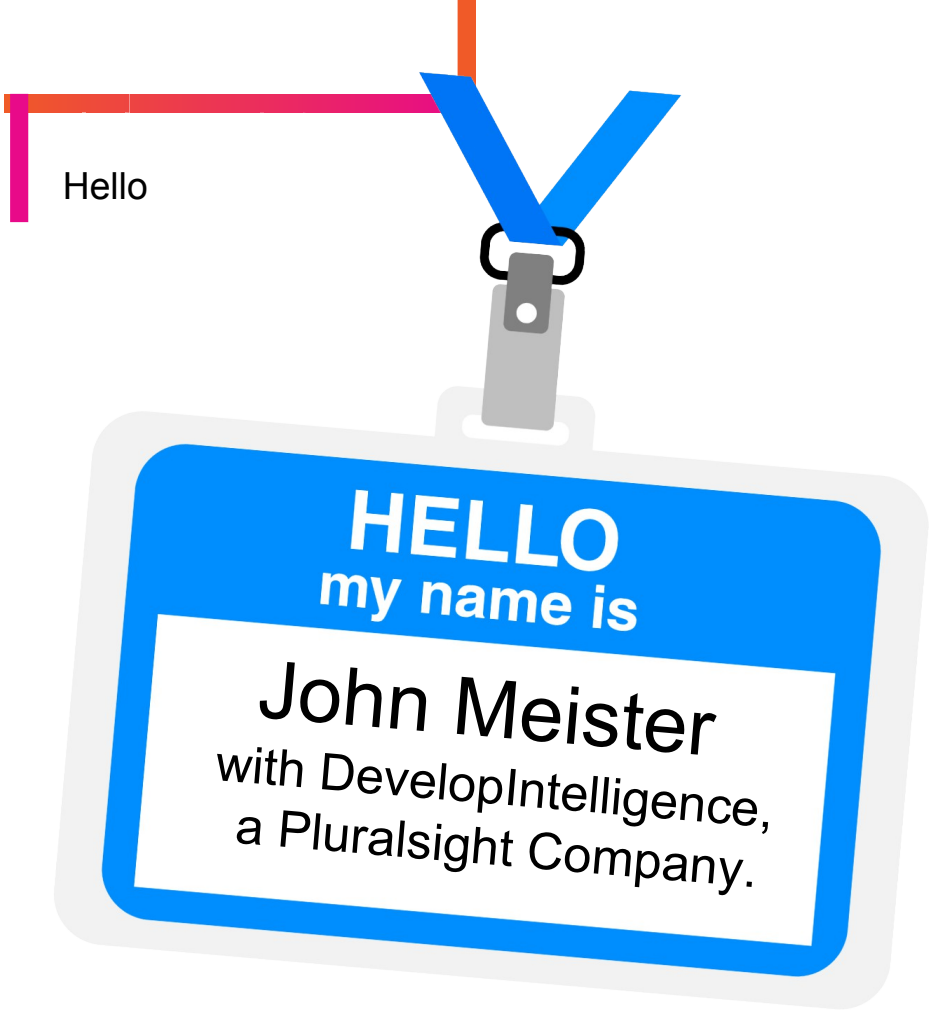
Welcome

Networking Fundamentals

Hello



- IT Systems Design and Integration Specialist 2005-2020
- Unix & Linux System Administrator 1994- present
- started teaching technical college courses in 1979
- US Army Signal School Instructor 1981-82
- Sr. Electronics Instructor ITT Tech 1982-83
- Adjunct Faculty City University of Seattle 2002-2008
- ebook technical author (Smashwords.com)
- O'Reilly author (Linux Sys Admin Video series)
- MSTM - Embry Riddle Aeronautical University
- enjoy photography (johnmeister.com) & kayaking

This course assumes you:

- Know how to use a computer and are familiar with the **command line.**
- Have **not taken a course on networking** and/or desire a better understanding of the basic concepts such as the OSI model, TCP/IP and network basic elements.

Please Note:

This course is designed to teach BASIC networking knowledge and skills.

Please message me in Chat so we can get you to the right resource.

Why study Networking Fundamentals?

- To understand the basic components of networking and the OSI Model
- To allow you to connect, troubleshoot and diagnose basic networks
- To understand overall network concepts to ensure security and function
- To learn the basic network commands used in Windows, Linux & Mac OSX
- To understand how traffic moves around a network

We teach over 400 technology topics

You experience our impact on a daily basis!

My pledge to you

I will...

- Make this interactive
- Ask you questions
- Ensure everyone can speak
- Discuss only commands available on Microsoft, Linux and Mac OSX by default

Objectives

At the end of this course you will be able to:

- From the command line, identify key network characteristics
- Identify OSI and TCP layers and their use
- Identify the key elements of basic networking
- Identify well known ports and related protocols
- Gather basic network information from the command line

How we're going to work together

- Make sure we're all on the same slide and topic
- Keep up with the "hands-on" exercises
- Make sure all recognized and expected results are at the command line
- Ask questions if needed
- WHEN AT THE COMMAND LINE please listen carefully to instructions or you could disconnect yourself! IF you do, you may need to reboot & log in again - YMMV!

Networking Fundamentals - OUTLINE

1) OVERVIEW: Introductions, History, Topologies & ACRONYMS
2) OSI Model - "*Programmers Do Not Throw Sausage Pizza Away*"
3) TCP Model
4) Protocols and Ports - Well Known Ports & Services
5) IP Addressing - Key elements &
   CIDR, Subnets, IPv6, IP Routing, DHCP, NAT
6) Domain Name System (DNS)

OVERVIEW

- Introductions & the Big Picture
- Basic Topologies
- Early Network Example
- Current Networking Example
- "Hands-on" command line view of YOUR network
- Acronyms (*e.g. YMMV - your mileage may vary!*)

INTRODUCTIONS

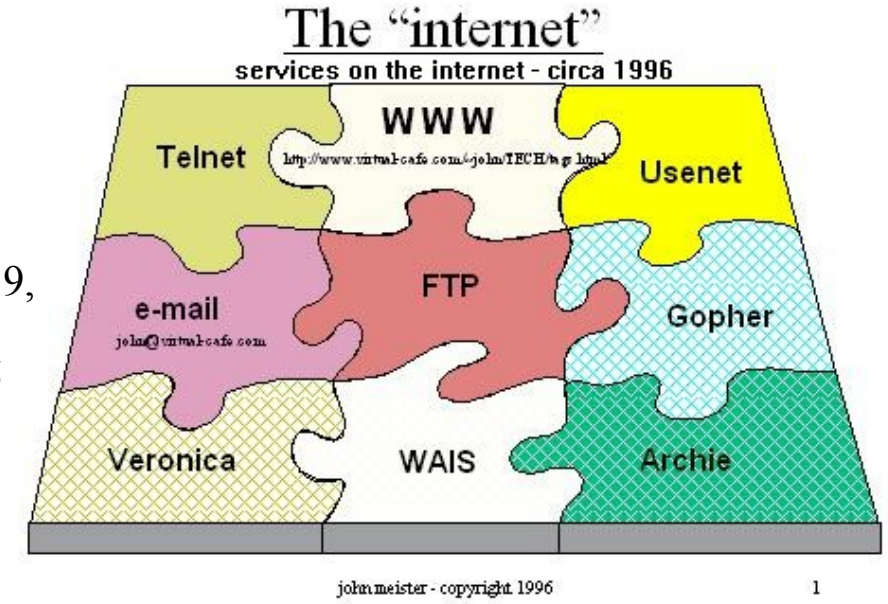
- Name - location - job - any interests you care to share
- work related NETWORK activity

  (*support / design / troubleshoot / sales / or use* ?)
- Operating System you are using today
- Familiar with the Command Line?

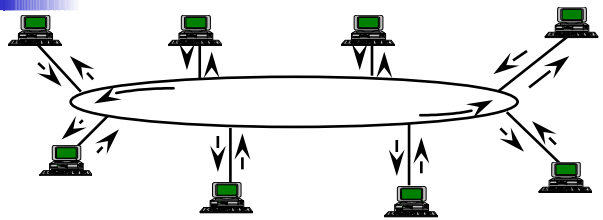
Network Overview - the Big Picture, circa 1996

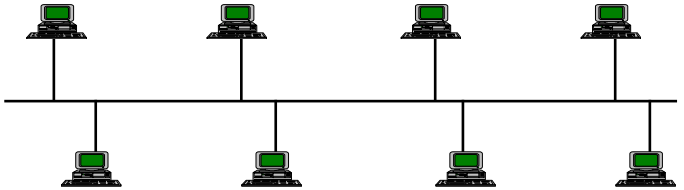
Services - and their well known PORTS:

Telnet - 23, **WWW - 80/443/8080,** Usenet - 119,
Mail 25, FTP - 20/21, Gopher -70,
Veronica - 2770, WAIS - 210, Archie - 1525

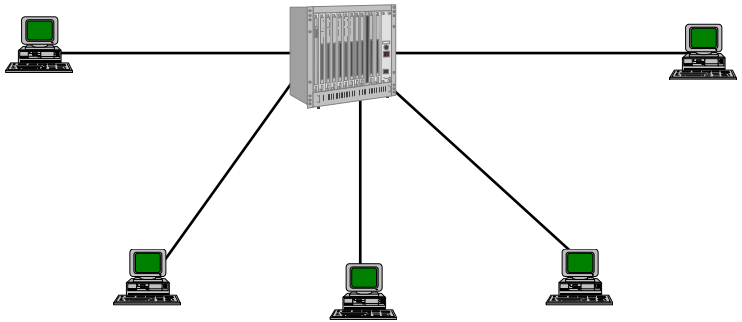


Ring, Bus and Star Topologies

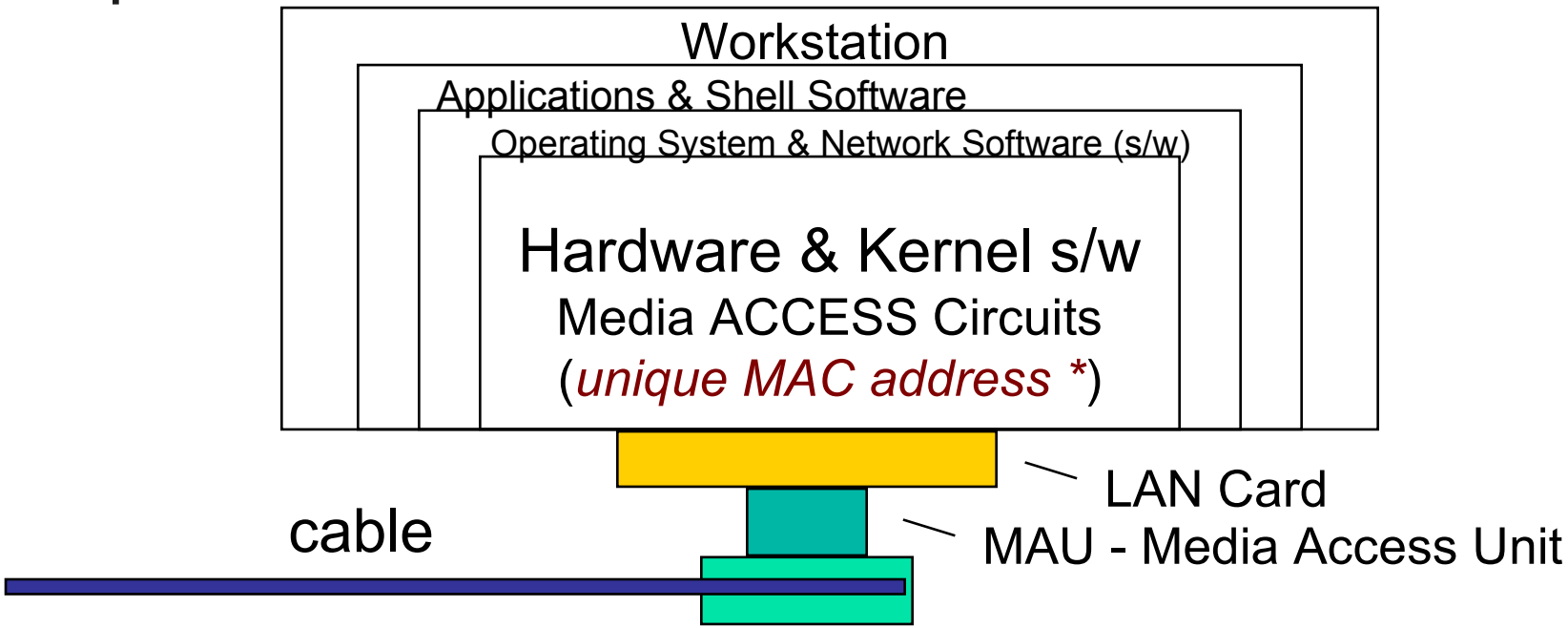




RING (e.g. Apollo Aegis 1980's) - (thick net) BUS using coax (1980-90's)
standard ethernet RJ45 (or wlan) to switch or hub



Basic LAN Components - Client

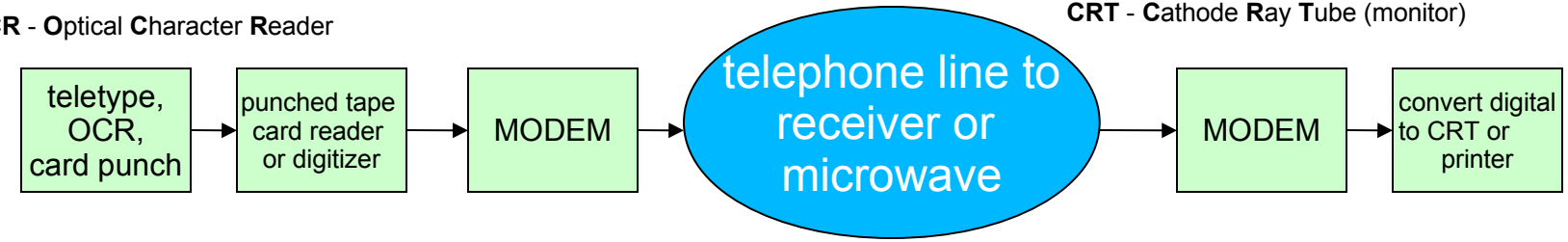


* - to lookup unique details or the NIC and vendor info, see: https://maclookup.app/search

Early Network Example 1970's

OCR - **O**ptical **C**haracter **R**eader

MODEM - **Mo**dulator - **Dem**odulator
CRT - **C**athode **R**ay **T**ube (monitor)

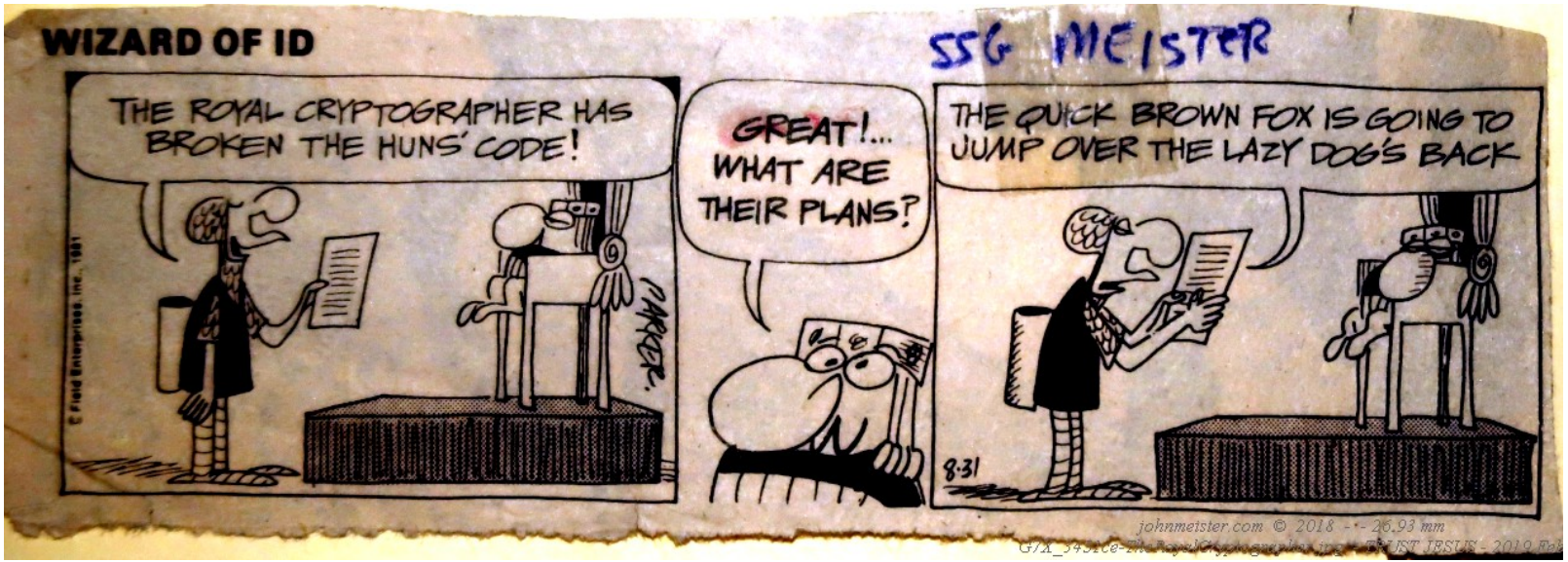


Early use of "networking" involved **digitizing information** and **modulating** it to be sent over wire and/or radio signals. It was hard wired and generally only point to point. Input to the system was by paper tape or **punched cards**, in the late 1970's OCR (*optical character recognition*) was used to convert typed (with a special font, *IBM selectric typewriters had a replaceable "wheel" that was used*) into the digital signals that may have been converted to paper tape or punched cards, or sent directly from the reader. The digital signals were converted to analog via a modem, then demodulated and digitized.

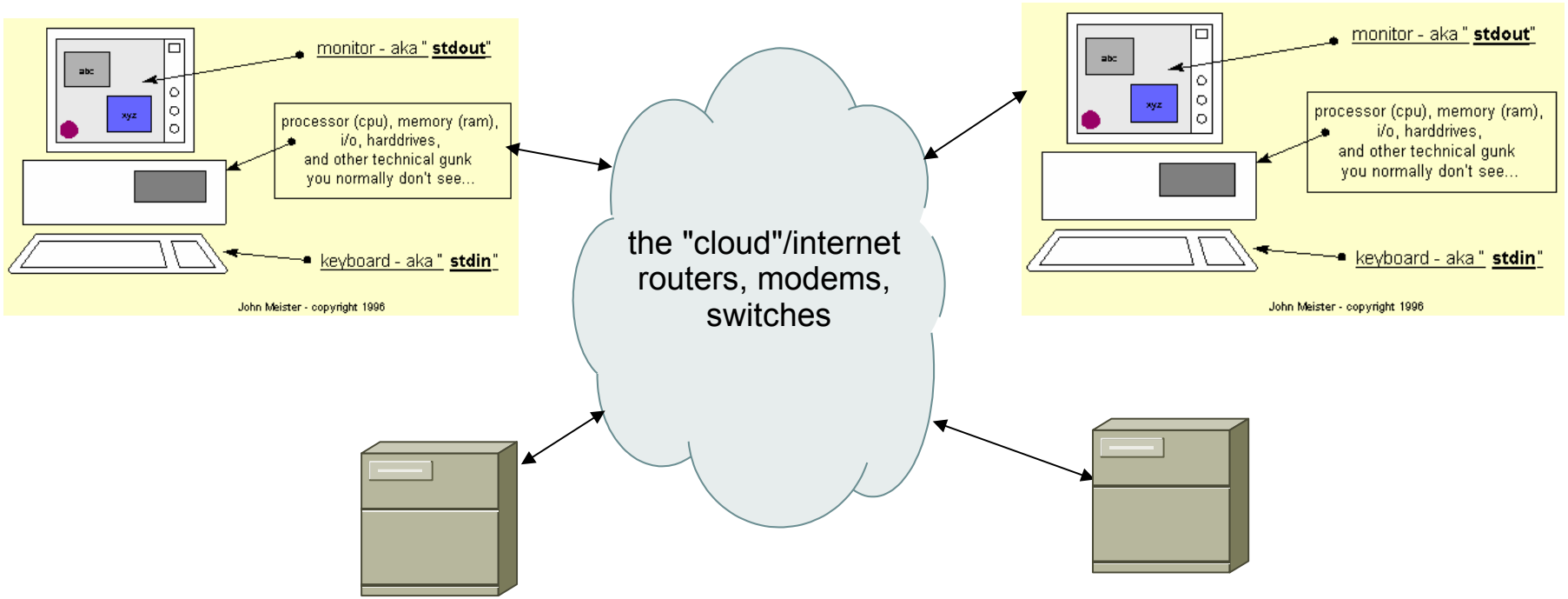
System testing before the "network"

NOTE: the "coded" message uses every letter of the alphabet - was used to test **teletypes** and communication systems

ACTUAL STRING USED: **THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG'S BACK**



Current Network Example



Acronyms

This is a **BNC connector**.

What do the letters B-N-C stand for?

*Can YOU answer **WITHOUT** using a search engine?*

BNC connector

https://www.amphenolrf.com/connectors/bnc-connectors.html

What is a BNC Connector?

The Bayonet **Neill Concelman** (BNC) connector was developed by **Paul Neill** of Bell Labs and Amphenol's own, **Carl Concelman**, and was named after both inventors and the innovative bayonet locking mechanism that the connector utilizes. The BNC connector was originally designed for military applications but is mostly used in the broadcast market today. (*circa 1950's*)

ACRONYMS

- BNC - Bayonet Neill Concelman
- OSI - Open Systems Interconnection
- TCP - Transmission Control Protocol
- IP - Internet Protocol
- DNS - Domain Name Service
- NAT - Network Address Translation
- CIDR - Classless Inter-Domain Routing
- DHCP - Dynamic Host Configuration Protocol
- DMZ - Demilitarized Zone
- HTTP - HyperText Transfer Protocol
- MAC - Media Access Control
- URL - Uniform Resource Locator

ACRONYMS

- POP - Post Office Protocol (MAIL)
- LAN - Local Area Network
- FTP - File Transfer Protocol
- MX - Mail Exchanger
- NIC - Network Interface Card
- NFS - Network File Service
- RDP - Remote Desktop Protocol
- SSH - Secure Shell (SCP = Secure Copy, SFTP - Secure FTP)
- RARP - Reverse Address Resolution Protocol
- RIP - Routing Internet Protocol
- NTP - Network Time Protocol
- SMTP - Simple Mail Transport Protocol

Computer Network Acronyms - Quiz

https://www.101computing.net/computer-network-acronyms-quiz/

complete the crossword at your leisure - *at lunch or after class.*

LAN WAN VPN WWW Wi-Fi WAP
NIC HTTP HTTPS FTP SMTP mbps
IP (address) MAC (address) URL DNS

a little history & humor...
before we hit the keys...

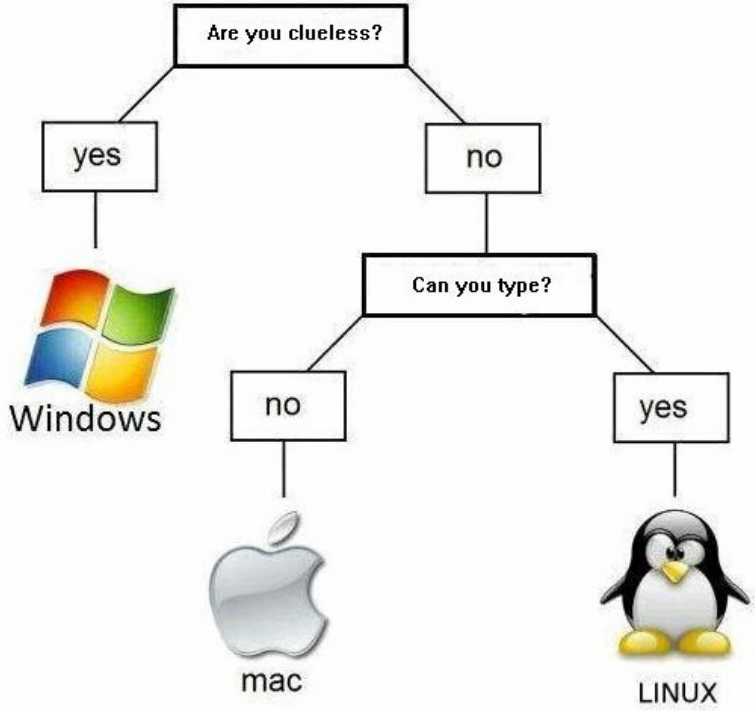
WYSIAYG

"What You See Is ALL You Get"

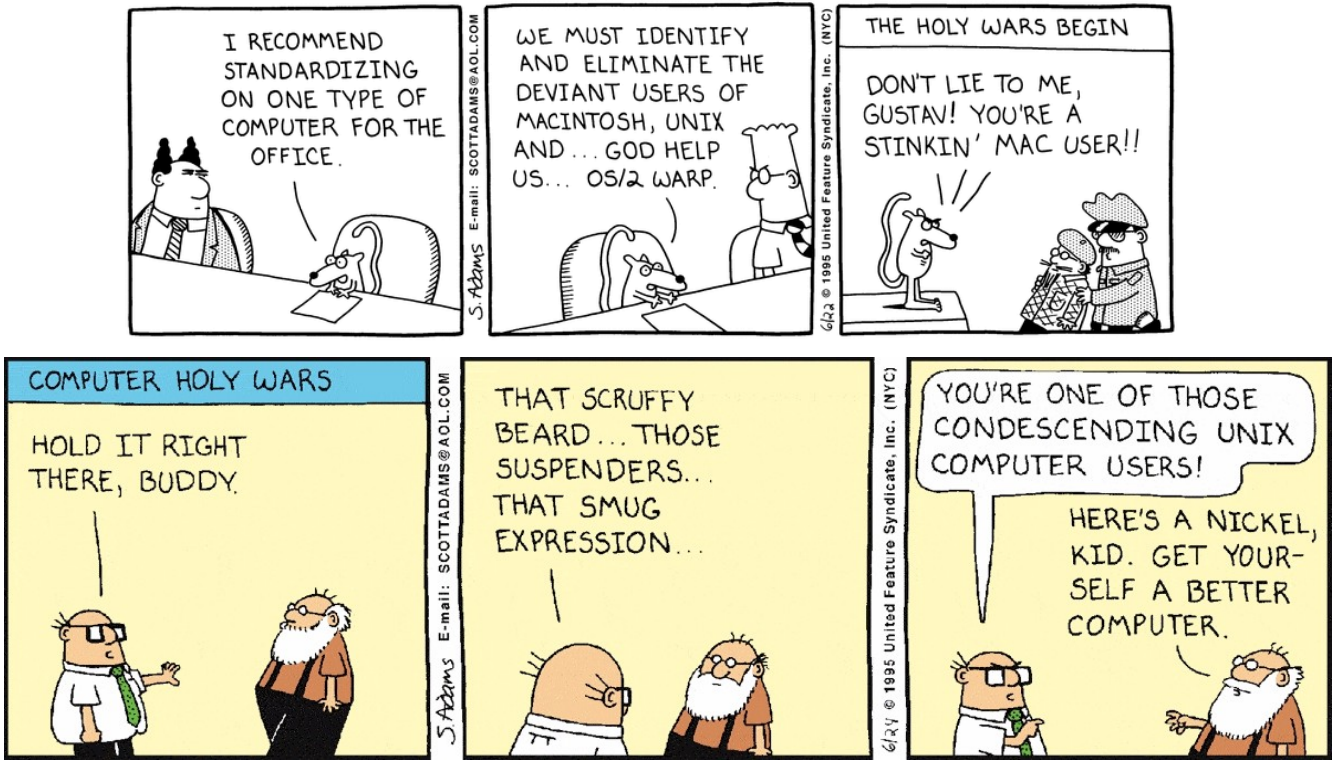
rian Kernighan, who also coined the term **UNIX, points out this means** real limitations with commands offered by GUI.

https://en.wikipedia.org/wiki/Brian_Kernighan
(GUI - graphical user interface)

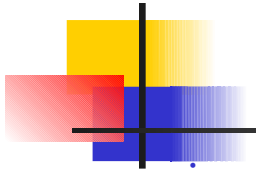
WYSIAYG with GUI - and with GUI often comes a performance penalty and increased security risks... in choosing a platform this decision tree might be helpful:



There are differences in the tools & commands offered... We'll cover **three** operating systems.



The "Hands-on" commands for networking



- In this class we will work with the following commands - please do NOT try them before we discuss them together. Many of these commands will alter your network configuration and may render you disconnected.
- There are DIFFERENCES between commands in Microsoft, Mac OSX and Linux.
- There are commands that are NOT installed by default and would require admin rights to install, and perhaps permission from your IT support group.
- The commands demonstrated and exercised in this class should NOT impact your configuration if typed as shown. The "Hands-on" commands are on all 3 systems.
- Some of the commands discussed have either been deprecated or updated with newer commands on the various systems, YMMV.
- Commands that are advanced or not installed by default will not be listed.
- Some commands may not work if your account or network has user restrictions.

The "Hands-on" commands for networking

Please do NOT try these before we discuss them

se the "**which**" command to see if the command is availble on your system, YMMV.

Microsoft Windows	Linux	Mac OSX
which ipconfig (which *not on W10?)*	*which ifconfig*	*which ifconfig*
ipconfig	ifconfig	ifconfig
netstat -r	netstat -r *or* route	netstat -r
hostname	hostname	hostname
nslookup cisco.com	nslookup cisco.com	nslookup cisco.com
ping -c 3 cisco.com	ping -c 3 cisco.com	ping -c 3 cisco.com
tracert cisco.com	traceroute cisco.com	traceroute cisco.com
netstat /?	netstat --help	netstat --help
arp ip address *(displays mac, but only those in arp table)*	arp ip address *(displays mac, but only those in arp table)*	arp ip address *(displays mac, but only those in arp table)*

"Hands-on" command line view of YOUR network details

1) get to a command line - *if you need help...* *ask.* (**WAIT** for all)

2) ... stand by...

The "Hands-on" commands for networking

ipconfig & ifconfig next...

Microsoft Windows	Linux	Mac OSX
which ipconfig *(which not on W10?)*	which ifconfig	which ifconfig
ipconfig	ifconfig	ifconfig
route	route	route
hostname	hostname	hostname
nslookup cisco.com	nslookup cisco.com	nslookup cisco.com
ping -c 3 cisco.com	ping -c 3 cisco.com	ping -c 3 cisco.com
tracert cisco.com	traceroute cisco.com	traceroute cisco.com
netstat /?	netstat --help	netstat --help
arp	arp	arp

"Hands-on" command line view of YOUR network details

- 1) *get to a command line - if you need help... ask. (WAIT for all)*
- 2) at the command line type **only** the following:
  - *For Windows type:* **ipconfig**
  - *For LINUX type:* **ifconfig**
  - *For Mac OSX type:* **ifconfig**
- 3) Let's discuss what is displayed from the command.

ipconfig - Microsoft Win10

C:\Users\luser>**ipconfig**

Windows IP Configuration

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : com

Link-local IPv6 Address . . . . . : fe80::4c3b:aecf:a4f6:16d6%6

IPv4 Address. . . . . : 10.0.2.15

Subnet Mask . . . . . : 255.255.255.0

Default Gateway . . . . . : 10.0.2.2 (*note: gateway=router*)

ifconfig - configure (view) network interface parameters

(Note: "if" is short for "interface")

NAME - ifconfig - configure a network interface

SYNOPSIS ifconfig [-v] [-a] [-s] [interface] ifconfig [-v] interface [aftype] options | address ...

DESCRIPTION

ifconfig is used to configure the kernel-resident network interfaces. (NOTE: 127.0.0.1)

It is used at boot time to set up interfaces as necessary.

If no arguments are given, **ifconfig displays the status of the active interfaces**.

If a single interface argument is given, it displays the status of the given interface only;

if a single -a argument is given, it displays the status of all interfaces, even those that are down.

Otherwise, it configures an interface.

https://www.oreilly.com/library/view/linux-network-administrators/1565924002/ch05s07.html

ifconfig - Linux Mint MATE

--> ifconfig

lan0: flags=4163<**UP,BROADCAST,RUNNING,MULTICAST**> mtu 1500
inet 192.168.1.31 **netmask** 255.255.255.0 **broadcast** 192.168.1.255
i**net6** fe80::9ae7:f4ff:fee9:ef39 **prefixlen** 64 scopeid 0x20<link>
ether 98:e7:f4:e9:ef:59 txqueuelen 1000 (Ethernet)
RX packets 2526776 bytes 2509693875 (2.5 GB)
RX errors 0 dropped 26219 overruns 0 frame 0
TX packets 1535939 bytes 1063212830 (1.0 GB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
device interrupt 16 memory 0xe1300000-e1320000

lo: flags=73<**UP,LOOPBACK,RUNNING**> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 83 bytes 6868 (6.8 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 83 bytes 6868 (6.8 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ifconfig - Mac OSX Darwin Kernel Version 19.6.0

lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
inet 127.0.0.1 netmask 0xff000000
inet6 ::1 prefixlen 128
inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=400<CHANNEL_IO>
ether e0:ac:cb:91:f1:7c
inet6 fe80::187b:5d93:599:ee4d%en0 prefixlen 64 secured scopeid 0x4
inet 192.168.1.125 **netmask** 0xffffff00 **broadcast** 192.168.1.255
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active

en1:flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
options=460<TSO4,TSO6,CHANNEL_IO>
ether 82:18:0f:77:77:80
media: autoselect <full-duplex>
status: inactive
bridge0:flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=63<RXCSUM,TXCSUM,TSO4,TSO6>
ether 82:18:0f:77:77:80
Configuration: id 0:0:0:0:0:0 priority 0 hellotime 0 fwddelay 0
maxage 0 holdcnt 0 proto stp maxaddr 100 timeout 1200
root id 0:0:0:0:0:0 priority 0 ifcost 0 port 0
ipfilter disabled flags 0x0
member: en1 flags=3<LEARNING,DISCOVER>
ifmaxaddr 0 port 5 priority 0 path cost 0
nd6 options=201<PERFORMNUD,DAD>
media: <unknown type>
status: inactive
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
options=400<CHANNEL_IO>
ether 02:ac:cb:91:f1:7c
media: autoselect
status: inactive
awdl0:flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1484
options=400<CHANNEL_IO>
ether 86:ce:9c:b0:26:03
inet6 fe80::84ce:9cff:feb0:2603%awdl0 prefixlen 64 scopeid 0x8
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active

llw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=400<CHANNEL_IO>
ether 86:ce:9c:b0:26:03
inet6 fe80::84ce:9cff:feb0:2603%llw0 prefixlen 64 scopeid 0x9
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active
utun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
inet6 fe80::81f2:8414:8c19:e4e8%utun0 prefixlen 64 scopeid 0xa
nd6 options=201<PERFORMNUD,DAD>
utun1: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 2000
inet6 fe80::850c:67f4:28c5:fd74%utun1 prefixlen 64 scopeid 0xb
nd6 options=201<PERFORMNUD,DAD>
utun2: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
inet6 fe80::4fe8:1122:2f53:b6a1%utun2 prefixlen 64 scopeid 0xc
nd6 options=201<PERFORMNUD,DAD>
utun3: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
inet6 fe80::50d5:c5ae:69fd:a0c1%utun3 prefixlen 64 scopeid 0xd
nd6 options=201<PERFORMNUD,DAD>

"Hands-on" command line view of YOUR network details

1) *get to a command line - if you need help... ask. (WAIT for all)*

2) at the command line type **only** the following:

   For Windows type: **ipconfig /all**

   For LINUX type: **ifconfig -a**

   For Mac OSX type: **ifconfig -a**

3) Let's discuss WHAT YOU SEE (you'll need to scroll the screen)

"Hands-on" command line view of YOUR network details

1) *get to a command line - if you need help... ask. (WAIT for all)*

2) at the command line type **only** the following:

For Windows type: ***ipconfig /all***

type: **ipconfig /help** *or* **ipconfig /?**

For LINUX type: ***ifconfig -a***

type: **ifconfig --help** *and* **man ifconfig**

For Mac OSX type: ***ifconfig -a***

type: **ifconfig --help** *and* **man ifconfig**

3) Let's **discuss** WHAT YOU SEE on each operating system.

Key Network Elements - ifconfig
IP address, Netmask, Broadcast, Route & MAC

ifconfig/ipconfig provide **key network elements** *(except route)*:

inet 192.168.1.31 **netmask** 255.255.255.0 **broadcast** 192.168.1.255
inet6 fe80::9ae7:f4ff:fee9:ef79 **prefixlen** 64 **scopeid** 0x20<link>

1. "inet" - the IPv4 or IPv6 **ADDRESS**
2. the **netmask**, or prefixlen
3. the **broadcast,** or scopeid
4. the **MAC** (media access control) information (*UNIQUE*):

ether 98:e7:f4:e9:ef:79 txqueuelen 1000 (Ethernet)

TO LOOK UP A MAC: https://maclookup.app/search

The "Hands-on" commands for networking

Please do NOT try these before we discuss them

Next command: netstat -r

Microsoft Windows	Linux	Mac OSX
which ipconfig (which *not on W10?)*	*which ifconfig*	*which ifconfig*
ipconfig	ifconfig	ifconfig
netstat -r	netstat -r *or* route	netstat -r
hostname	hostname	hostname
nslookup cisco.com	nslookup cisco.com	nslookup cisco.com
ping -c 3 cisco.com	ping -c 3 cisco.com	ping -c 3 cisco.com
tracert cisco.com	traceroute cisco.com	traceroute cisco.com
netstat /?	netstat --help	netstat --help
arp ip address *(displays mac, but only those in arp table)*	arp ip address *(displays mac, but only those in arp table)*	arp ip address *(displays mac, but only those in arp table)*

"Hands-on" netstat -r

- the netstat -r command lists the routing table: (*Linux & Mac OSX first, then MS Windows below*)

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 lan0
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 lan0
_gateway        0.0.0.0         255.255.255.255 UH        0 0          0 lan0
```

C:\WINDOWS>netstat -r *(results shown edited for brevity)*

```
Route Table / Interface List
0x1 ........................... MS TCP Loopback interface
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          127.0.0.0          255.0.0.0        127.0.0.1     127.0.0.1       1
Persistent Routes:   None
```

"Hands-on" command line - REFINE RESULTS & REVIEW network details

- at the command line type **only** the following:
  - *For Windows type:* **ipconfig /all | grep inet** *(ymmv)*
  - *For LINUX type:* **ifconfig -a | grep inet**
  - *For Mac OSX type:* **ifconfig -a | grep inet**
  - *For all, type:* netstat -r
  - ESSENTIAL ITEM: THE LOOPBACK ADDRESS

    IPv4: 127.0.0.1

    IPv6: inet6 ::1

Networking Fundamentals - OUTLINE

1) OVERVIEW: Introductions, History, Topologies & ACRONYMS
2) **OSI Model -**
   "Programmers Do Not Throw Sausage Pizza Away"
3) TCP Model
4) Protocols and Ports - Well Known Ports & Services
5) IP Addressing - Key elements and
   CIDR, Subnets, IPv6, IP Routing, DHCP, NAT
6) Domain Name System (DNS)

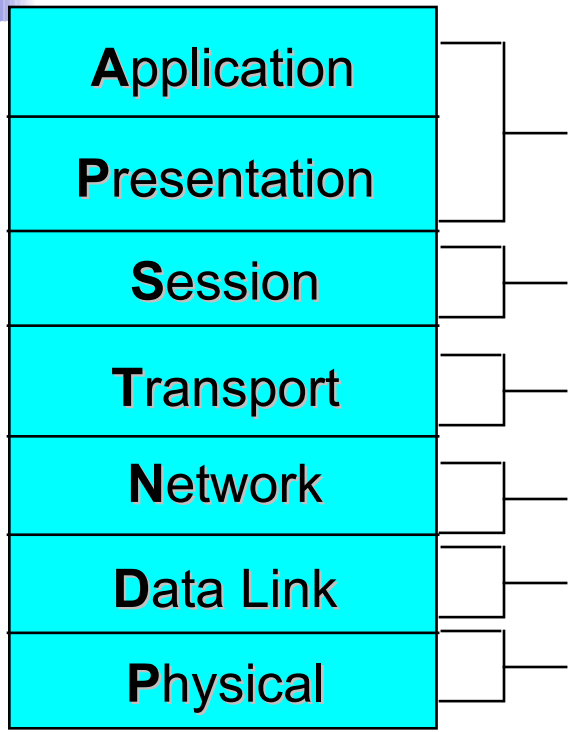
The Open Systems Interconnection (OSI) model

- **Open Systems Interconnection** - The model partitions the flow of data in a communication system into **seven abstraction layers**, from the physical implementation of transmitting bits across a communications medium to the highest-level representation of data of a distributed application. Each intermediate layer serves a class of functionality to the layer above it and is served by the layer below it. Classes of functionality are realized in software by standardized communication protocols.
- The OSI model was developed starting in the late 1970s to support the emergence of the diverse computer networking methods that were competing for application in the large national networking efforts in the world. In the 1980s, the model became a working product of the Open Systems Interconnection group at the **International Organization for Standardization (ISO).**

- *retrieved from:* https://en.wikipedia.org/wiki/OSI_mode on 21 January 2022

Internet Protocols - OSI Model

"Programmers Do Not Throw Sausage Pizza Away"



e.g. ssh runs at the Application layer on port 22

Network Filing System (NFS)

SNMP, FTP, SMTP, TELNET

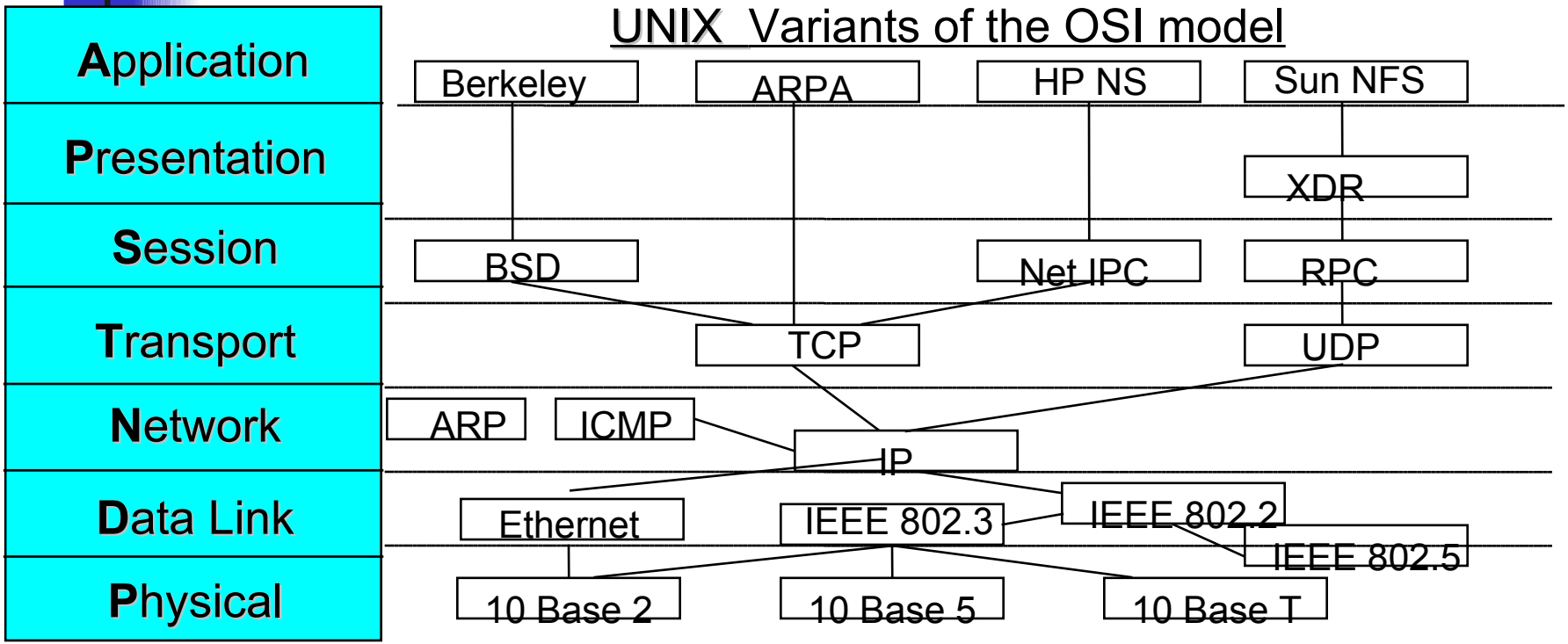
TCP

IP

Lan drivers
Media Access Control
Physical: NIC, Transceiver, Cable & Connectors

Components

Internet Protocols - UNIX Variants



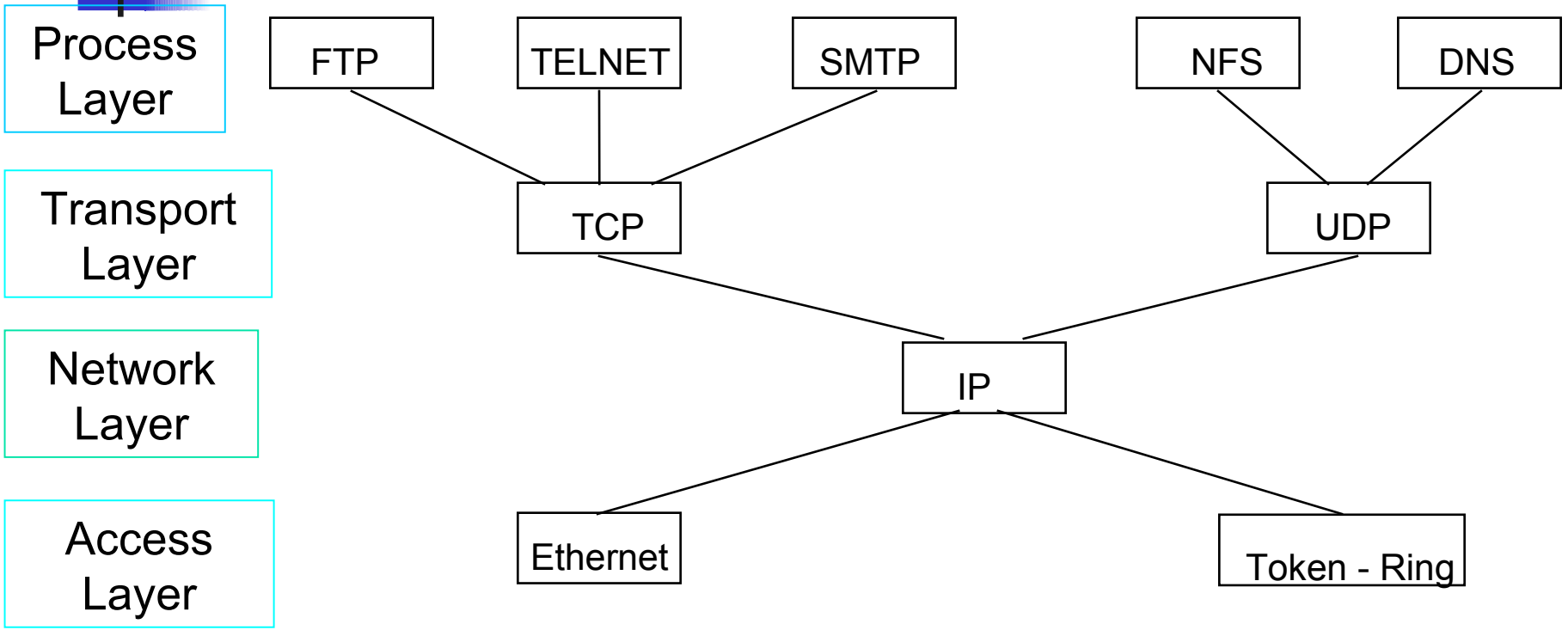
Networking Fundamentals - OUTLINE

1) OVERVIEW: Introductions, History, Topologies & ACRONYMS
2) OSI Model - "Programmers Do Not Throw Sausage Pizza Away"
3) **TCP Model**
4) Protocols and Ports - Well Known Ports & Services
5) IP Addressing - Key elements and
   CIDR, Subnets, IPv6, IP Routing, DHCP, NAT
6) Domain Name System (DNS)

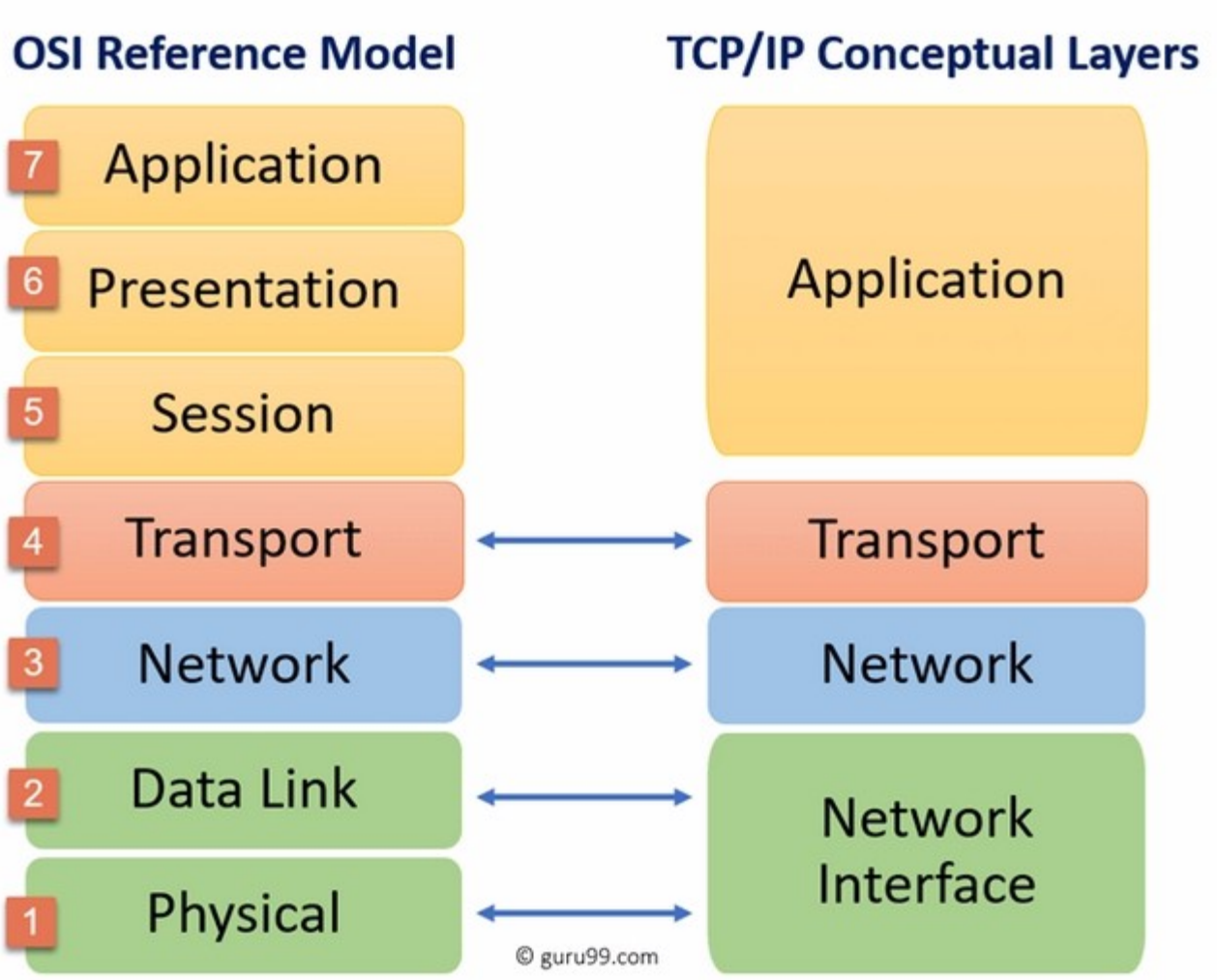
TCP - Transmission Control Protocol

- Application Layer - Includes the OSI application, presentation and most of the session layers.
- Transport Layer - Remaining parts of the OSI session layer plus the transport layer
- Network Layer - Subset of the OSI network layer
- Link Layer - OSI data link layer and sometimes the physical layers, as well as some protocols of the OSI's network layer.

Structure of TCP/IP Components



Compare Models for: OSI and TCP



Networking Fundamentals - OUTLINE

1) OVERVIEW: Introductions, History, Topologies & ACRONYMS
2) OSI Model - "Programmers Do Not Throw Sausage Pizza Away"
3) TCP Model
4) **Protocols and Ports - Well Known Ports & Services**
5) IP Addressing - Key elements and
   CIDR, Subnets, IPv6, IP Routing, DHCP, NAT
6) Domain Name System (DNS)

Ports used for TCP and UDP

--> more /etc/services

Network services, Internet style

#

Note that it is presently the policy of IANA to assign a single well-known

port number for both TCP and UDP; hence, officially ports have two entries

even if the protocol doesn't support UDP operations.

#

Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

Well Known Ports (/etc/services Mac & Linux (and MS))

https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

```
netstat     15/tcp
ftp-data    20/tcp
ftp         21/tcp
ssh         22/tcp          # SSH Remote Login Protocol
telnet      23/tcp
smtp        25/tcp          mail
time        37/tcp/udp      timserver   time
whois       43/tcp          nicname
domain      53/tcp/udp          # Domain Name Server
tftp        69/udp
gopher      70/tcp          # Internet Gopher
finger      79/tcp
http        80/tcp      www     # WorldWideWeb HTTP
pop3        110/tcp     pop-3       # POP version 3
sunrpc      111/tcp/upd     portmapper  # RPC 4.0 portmapper
nntp    119/tcp  readnews untp   # USENET News Transfer Protocol
```

```
ntp         123/udp         # Network Time Protocol
imap2       143/tcp     imap        # Interim Mail Access P 2 and 4
snmp        161/tcp/udp         # Simple Net Mgmt Protocol
xdmcp       177/udp         # X Display Manager Control Protocol
bgp         179/tcp         # Border Gateway Protocol
ldap        389/tcp/udp         # Lightweight Directory Access Protocol
https       443/tcp         # http protocol over TLS/SSL
# UNIX specific services
biff        512/udp     comsat
login       513/tcp
who         513/udp     whod
syslog      514/udp
printer     515/tcp     spooler     # line printer spooler
route       520/udp     router routed   # RIP
uucp        540/tcp     uucpd       # uucp daemon
rsync       873/tcp
```

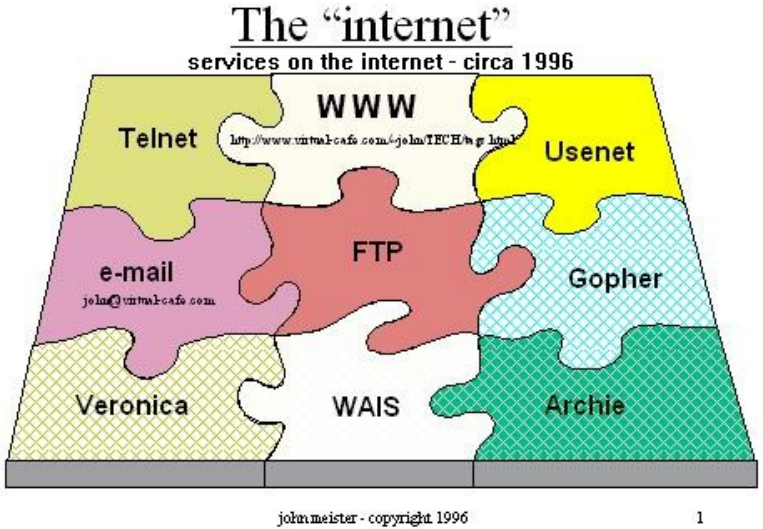
Selected Ports from /etc/services

- netstat 15/tcp
- ftp-data 20/tcp
- ftp 21/tcp
- ssh 22/tcp # SSH Remote Login Protocol
- telnet 23/tcp
- smtp 25/tcp mail
- time 37/tcp timserver
- time 37/udp timserver
- whois 43/tcp nicname
- domain 53/tcp # Domain Name Server
- domain 53/udp
- tftp 69/udp
- gopher 70/tcp # Internet Gopher
- http 80/tcp www # WorldWideWeb

TCP - handshakes - reliable
UDP - does not handshake
MSBlaster used open ports on W2K most were udp
https://en.wikipedia.org/wiki/Blaster_%28computer_worm%29

Well Known Ports

- telnet - 23
- www - 80/443/8080
- usenet - 119
- mail 25
- ftp - 20/21
- gopher -70
- veronica - 2770
- wais - 210
- archie - 1525



The "Hands-on" commands for networking

Please do NOT try these before we discuss them

Next command: netstat -r

Microsoft Windows	Linux	Mac OSX
which ipconfig (which *not on W10?)*	*which ifconfig*	*which ifconfig*
ipconfig	ifconfig	ifconfig
netstat -r	netstat -r *or* route	netstat -r
hostname	hostname	hostname
nslookup cisco.com	nslookup cisco.com	nslookup cisco.com
ping -c 3 cisco.com	ping -c 3 cisco.com	ping -c 3 cisco.com
tracert cisco.com	traceroute cisco.com	traceroute cisco.com
netstat /?	netstat --help	netstat --help
arp ip address *(displays mac, but only those in arp table)*	arp ip address *(displays mac, but only those in arp table)*	arp ip address *(displays mac, but only those in arp table)*

"Hands-on" netstat -r

the netstat -r command lists the routing table:

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 lan0
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 lan0
_gateway        0.0.0.0         255.255.255.255 UH        0 0          0 lan0

C:\WINDOWS>netstat -r
Route Table / Interface List
0x1 ........................... MS TCP Loopback interface
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
Persistent Routes:   None
```

Additional Network Terms & Acronyms

- **SNMP** - Simple Network Management Protocol, a standard Internet protocol used to monitor network devices
- **ICMP** - Internet Control Message Protocol - An Internet protocol that handles error and control messages.
- **MIB** - Management Information Base, a set of variables stored as a database. Contains information about the element to be managed.
- **Ping** - Packet InterNet Groper , name used for a Internet program used to test reachability of destinations by sending an ICMP echo request.
- **UDP** - User Datagram Protocol - Asyncronous, unreliable and connectionless.

Networking Fundamentals - OUTLINE

1) OVERVIEW: Introductions, History, Topologies & ACRONYMS
2) OSI Model - "Programmers Do Not Throw Sausage Pizza Away"
3) TCP Model
4) Protocols and Ports - Well Known Ports & Services
5) **IP Addressing - Key elements and CIDR, Subnets, IPv6, IP Routing, DHCP, NAT**
6) Domain Name System (DNS)

IP Addresses - Warriors of the NET

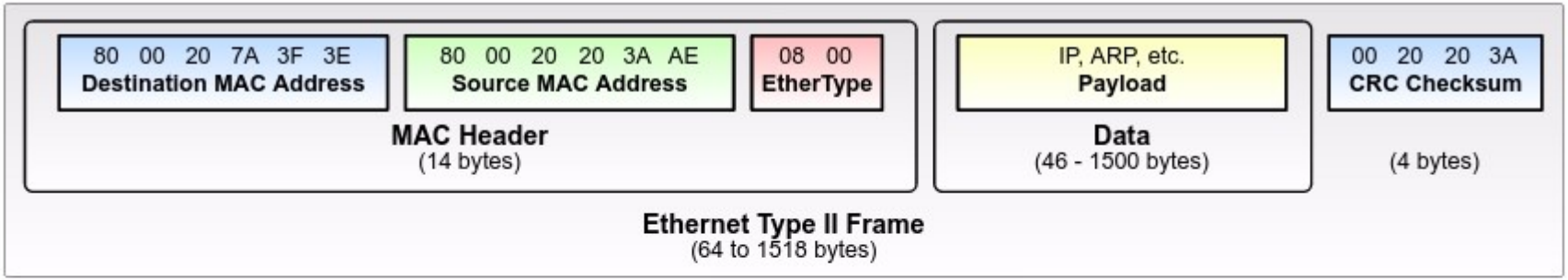
During Lunch - this feature "film" will be "playing"... *or view it "after school" - a useful presentation of the concepts.*

https://www.youtube.com/watch?v=EOYe71RWMvk

This was shown as part of an *Advanced Microsoft Networking* class I took in 2002.

Warriors of the Net - An original production from the creative development team at Ericsson in Europe way back in 1999. *This is a very novel and simple, yet easy to understand visual explanation of how the Internet, using the IP protocol, actually works in non technical terms.*

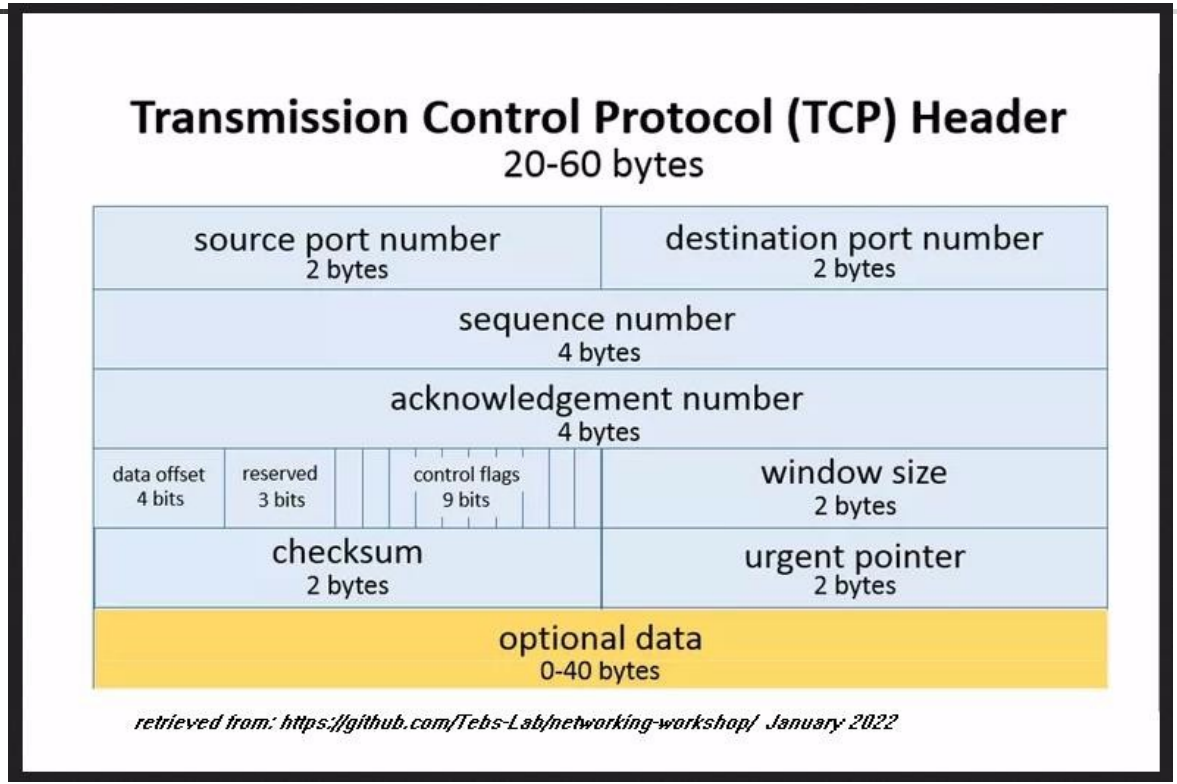
Parts of an IP Address showing the MAC



retrieved from: https://github.com/Tebs-Lab/networking-workshop/ January 2022

- The MAC addresses represent the two physical devices (our computers ethernet card, and the models ethernet card).
- Ethertype indicates which type of ethernet frame this is, Type II is the most common but there are others.
- The data is ALLLLLLLLL the data from the HTTP, TCP, and IP layers.
- The CRC checksum is a mechanism to check that data has not been corrupted in transit.

Additional details found in a packet



Classful IP Addressing

Class	First Few Bits	First Byte	Prefix Length	Intent
A	0	1-126*	8	Very large networks
B	10	128-191	16	Large networks
C	110	192-223	24	Small networks
D	1110	224-239	NA	IP multicast
E	1111	240-255	NA	Experimental

*Addresses starting with 127 are reserved for IP traffic local to a host.
127.0.0.1 is the loopback for the system kernel.

Private Addresses - Non-routable

- Non-Routable IP Addresses
- RFC1918, an Internet standards document, describes several non-routable IP address blocks. These blocks of IP addresses have been set aside for use by individuals and companies for use in private networks. Properly configured routers on the Internet will not route these IP addresses.
- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

Subnet Mask

- 32 bits long
- Specifies which part of an IP address is the network/subnet field and which part is the host field
  - The network/subnet portion of the mask is all 1s in binary.
  - The host portion of the mask is all 0s in binary.
  - Convert the binary expression back to dotted-decimal notation for entering into configurations.
- Alternative
  - Use slash notation (for example /24)
  - Specifies the number of 1s

Subnet Mask Example

- 11111111 11111111 11111111 00000000
- 255 255 255 0
- What is this in slash notation? /24 (24 1's)
- What is this in dotted-decimal notation? 255.255.255.0
- 1's, 2, 4, 8, 16, 32, 64, 128 - powers of 2
  (2 raised to a power, each position increases by a factor of 2 for its value)
- add up 1+2+4+8+16+32+64+128 = ?
- BETTER: 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 (reading left to right)

Class A Networks

- **N . H . H . H**
  - Class **A** - 126 networks - 16,777,214 hosts
  - 1st "N" is the Network ID
  - 3 remaining "H" are the Host ID's
  - First Octet: '1-126' represents Class **A** address
  - Subnet is 255.0.0.0

Class B Networks

- **N . N . H . H**
  - Class **B** - 16,384 networks - 65,534 hosts
  - 1st two "N's" are the Network ID
  - 2 remaining "H's" are the Host ID's
  - First Octet: '128-191' represents Class **B** address
  - Subnet is 255.255.0.0

Class C Networks

- **N . N . N . H**
  - Class **C** - 2,097,152 networks - 254 hosts
  - 1st three "N's" are the Network ID
  - 1 remaining "H" is the Host ID's
  - First Octet: '192-223' represents Class **C** address
  - Subnet is 255.255.255.0

- **CLASS A, B, or C**

  A – 255.0.0.0 - 126 networks 16,777,214 hosts

  B – 255.255.0.0 – 16,384 networks 65,534 hosts

  C – 255.255.255.0 – 2,097,152 networks 254 hosts

  D - Multicasting

  E - Experimental

- 127.0.0.0 - LoopBack Address
- lowest number reserved for network
- highest number (255) reserved for Broadcast

Division of the Classful Address Space

Class	**Prefix Length**	**Number of Addresses per Network**
A	8	$2^{24}-2$ = **16,777,214**
B	16	$2^{16}-2$ = 65,534
C	24	$2^{8}-2$ = **254**

Classful IP is Wasteful

- Class A uses 50% of address space
- Class B uses 25% of address space
- Class C uses 12.5% of address space
- Class D and E use 12.5% of address space

Classless Addressing

- Prefix/host boundary can be anywhere
- Less wasteful
- Supports route summarization
  - Also known as
    - Aggregation
    - Supernetting
    - Classless routing
    - Classless inter-domain routing (CIDR)
    - Prefix routing

- # Classless Inter-Domain Routing
- CIDR translates all IP address and subnet masks to binary notation. CIDR divides an IP address into a set of 32 values, in place of the four values used in the classful system.
- CIDR does not define a default subnet mask based on the IP address. Each host is configured with a custom subnet mask.
- e.g. 192.168.201.24/24
  255.255.255.0/24
  converts to binary notation

IP version 6

- An **Ipv6 address uses 128 bits as opposed to 32 bits in IPv4**.
- Because an hexadecimal number uses 4 bits this means that an IPv6 address consists of **32 hexadecimal numbers**.
- These numbers are **grouped in 4's giving 8 groups** or blocks. The groups are written with a : (colon) as a separator.

How to Simplify Shorten and Compress IPv6 Addresses

- IPv6 addresses are 128 bit binary numbers (hexadecimal format)

  2001:0db8:0000:000b:0000:0000:0000:001A

- To shorten and compress Omit leading zeros.
- After removing the leading zeros, the IPv6 Address quoted above is:

  2001:db8:0:b:0:0:0:1A

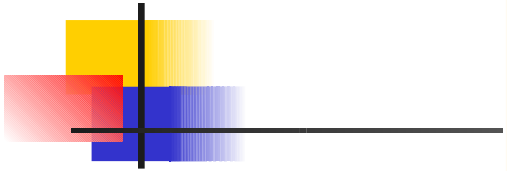
- We can shorten further, but only **once** and **one way**... i.e.
  2001:db8:0:b:**0:0:0:**1A, there is series of **three** consecutive fields of hexadecimal zeros, in **bold**,
  replace with double colon: **2001:db8:0:b:0:0:0:1A** - to further simplify
  and shorten the IPv6 the Address to: **2001:db8:0:b::1A**

DHCP - Dynamic Host Configuration Protocol

- **DHCP uses 2 UDP ports, a connectionless service model:**
  UDP port number 67 is the destination port of a server, and
  UDP port number 68 is used by the client.
- DHCP operations consists of **four phases**:
  1) server discovery,
  2) IP lease offer,
  3) IP lease request, and
  4) IP lease acknowledgement.
- Abbreviated as DORA for Discovery, Offer, Request, and Acknowledgement.
- for additional information see the RFC, wiki & related references:
  http://www.faqs.org/rfcs/rfc2131.html
  https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

TCP ports and services

- ports and services examples
- details on setting up ssh, using port 22
- details on configuring a reverse tunnel in ssh to show how NAT - Network Address Translation works in a real scenario.



cd ~/.ssh *(mkdir if neeed)*
ssh-keygen *(hit enter until done)*
cp id_rsa.pub id_rsa.*USER.HOST*

never share id_rsa - that is your PRIVATE KEY. RENAME id_rsa.pub, e.g. id_rsa-NAME.pub

scp (secure copy) the public key copy with your username and client name to the server you need to remote to. LOGIN ID and Password.

Once on the server, cd .ssh copy existing authorized_keys with a DATE suffix, and then APPEND (>>) the newly added id_rsa.pub.user.client to the authorized_keys file. logout, and then ssh to that host.

TO SETUP new CLIENTS SSH

QUICK **SETUP ONE WAY SSH**... from client to host only.

---

start on the authorized **client**:

---

mkdir .ssh
cd .ssh
ssh-keygen
cp id_rsa.pub id_rsa.clientname
scp id_rsa.pub.clientname user@host:/home/user/.ssh/ ---------- *(login and password required)*
--------*you just copied a file to the other system...*
ssh host **(login and password required)**
---------- *at this point you are on the other system...*
cd .ssh
cat id_rsa.pub.clientname >> authorized_keys
exit
--------*you're back on your system...*
ssh host *(login and password **NOT** required)*
---------- *back on the other system... w/o passwd!*

QUICK SETUP **ONE WAY SSH**... from client to host.

---

start on the authorized client:

```
cd
mkdir .ssh
cd .ssh
ssh-keygen
cp id_rsa.pub id_rsa.clientname
scp id_rsa.pub.clientname user@host:/home/user/.ssh/
ssh host
cd .ssh
cat id_rsa.pub.clientname >> authorized_keys
exit
ssh host
```

Reverse Tunneling

with SSH, an example showing use of ports and NAT.

REVERSE SSH TUNNEL - summarized

1) from behind NAT - on **RODENT**: ssh -R 19999:localhost:22 john@**CAT.com**
2) from external box - on **CAT.com**: ssh localhost -p 19999

OPTIONAL:

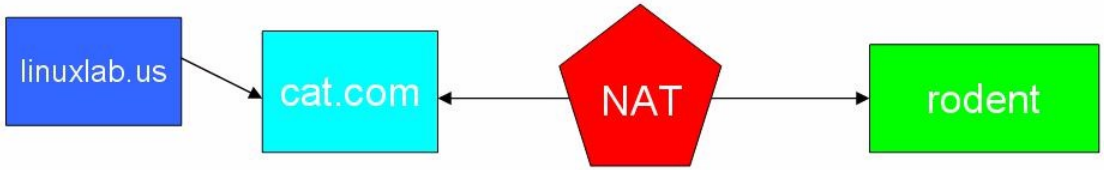
3) from another external box: ssh to **CAT.com**; then ssh localhost -p 19999

---

rodent:	ssh -R 19999:localhost:22 cat.com	# establish reverse tunnel using port 19999 on localhost using incoming port 22 to cat.com
rodent:	top	# need to keep connection alive - will timeout
cat.com:	ssh localhost -p 19999	# one on external system, use ssh to localhost port 19999
linuxlab.us:	ssh cat.com; ssh localhost -p 19999	# from another external server, ssh to cat.com

---

cat.com (*124.23.26.201*) -> | **Network-Address-Translator** | -> rodent (*10.10.10.45*)



ssh -R 12345:localhost:22 catuser@124.23.26.201

* the port can be any unused port: check /etc/system, netstat commands.

ssh localhost -p 12345

BillyBob's server -> cat (124.23.26.201) -> |NAT| -> rodent (10.10.10.45)

1 From BillyBob's server linuxlab.us:
ssh catuser@124.23.26.201
2 After the login to cat:
ssh localhost -p 12345

Networking Fundamentals - OUTLINE

1) OVERVIEW: Introductions, History, Topologies & ACRONYMS
2) OSI Model - "*Programmers Do Not Throw Sausage Pizza Away*"
3) TCP Model
4) Protocols and Ports - Well Known Ports & Services
5) IP Addressing - Key elements and
   CIDR, Subnets, IPv6, IP Routing, DHCP, NAT
6) **Domain Name System (DNS)**

Domain Name System (DNS)

- Maps names to IP addresses
- Supports hierarchical naming
  - example: frodo.rivendell.middle-earth.com
- A DNS server has a database of resource records (RRs) that maps names to addresses in the server's "zone of authority" (ZOA)
- Client queries server
  - Uses **UDP port 53** for name queries and replies
  - Uses **TCP port 53** for zone transfers

The "Hands-on" commands for networking

Please do NOT try these before we discuss them

Next commands: hostname, nslookup and ping

Microsoft Windows	Linux	Mac OSX
which ipconfig (which *not on W10?)*	*which ifconfig*	*which ifconfig*
ipconfig	ifconfig	ifconfig
netstat -r	netstat -r *or* route	netstat -r
hostname	hostname	hostname
nslookup cisco.com	nslookup cisco.com	nslookup cisco.com
ping -c 3 cisco.com	ping -c 3 cisco.com	ping -c 3 cisco.com
tracert cisco.com	traceroute cisco.com	traceroute cisco.com
netstat /?	netstat --help	netstat --help
arp ip address *(displays mac, but only those in arp table)*	arp ip address *(displays mac, but only those in arp table)*	arp ip address *(displays mac, but only those in arp table)*

HANDS-ON (from the command line) hostname

All three OSes will display the system name, however, Microsoft is limited & Mac OSX follows BSD

WHAT DOES YOUR SYSTEM REPORT?

hostname - will display the system name (*may be used to change the hostname in Linux; may not be persistent*)

Program options:

-a, --alias alias names
-A, --all-fqdns all long host names (FQDNs)
-b, --boot set default hostname if none available
-d, --domain DNS domain name
-f, --fqdn, --long long host name (FQDN)
-F, --file read host name or NIS domain name from given file
-i, --ip-address addresses for the host name
-I, --all-ip-addresses all addresses for the host
-s, --short short host name
-y, --yp, --nis NIS/YP domain name

HANDS-ON (from the command line) nslookup

TYPE:

nslookup cisco.com

Server: 75.75.75.75

Address: 75.75.75.75#53

Non-authoritative answer:

Name: cisco.com

Address: 72.163.4.185

Name: cisco.com

Address: 2001:420:1101:1::185

Specify DNS server in Command:

TYPE:

nslookup cisco.com 1.1.1.1

Server: 1.1.1.1

Address: 1.1.1.1#53

Non-authoritative answer:

Name: cisco.com

Address: 72.163.4.185

Name: cisco.com

Address: 2001:420:1101:1::185

DNS Details

- Client/server model
- Client is configured with the IP address of a DNS server
  - Manually or DHCP can provide the address
- DNS *resolver software* on the client machine sends a query to the DNS server. Client may ask for *recursive lookup.*
- Having multiple DNS servers in the /etc/resolv.conf file allows for a failover if there is a name resolution error. In addition, most services may be reached by typing in the actual IP address, if known.

RFC Guideline for Assigning Host Names

- **According to RFC 952:** 1. A "name" (Net, Host, Gateway, or Domain name) is a text string up to 24 characters drawn from the alphabet (A-Z), digits (0-9), minus sign (-), and period (.). Note that periods are only allowed when they serve to delimit components of "domain style names". (See RFC-921, "Domain Name System Implementation Schedule", for background). No blank or space characters are permitted as part of a name. No distinction is made between upper and lower case. The first character must be an alpha character (*was relaxed in an update to allow a digit*). The last character must not be a minus sign or period.
- Names should be: Short, Meaningful, Unambiguous, Distinct, and Case insensitive
- see: https://datatracker.ietf.org/doc/html/rfc952
- Avoid names with special characters (see next slide for list)

SPECIAL CHARACTERS

& ; | * ? ' " ` [ ] ( ) $ < > { } ^ # / \ % ! ~ - (spaces have significance)

Be careful with these characters. If you have unexpected results with a command or a script it may be related to one of these.

&	- used to place jobs in the background, *ex: mozilla & - also used by sed to pass variables*
;	- separates commands, *ex: ls ; cd / ; ls*
|	- used to "pipe" processes, *ex: cat filename | col -b > newfile*
*	- wildcard, one or more characters
?	- wildcard, ONE character only
' " `	- various uses within shells and scripts (single quote, quote, and grave)
[ ] ( )	- used to pass variables and define ranges
$	- represents a process or a variable, *ex: echo $SHELL*
< >	- redirection
{ }	- used to pass parameters, see find examples later in course
^	- up caret - often used to define keys, *ex: stty erase ^H (sets backspace)*
#	- pound, used at the front of a comment, ignored by shell and scripts
/	- path or escape sequences
%	- used by sed as a parameter, see later examples of use in vi and sed
!	- bang, used to escape a shell or vi, also combined with pound to instruct "command interpreter", *ex: #!/bin/sh*
~	- tilde, used to indicate home directory on many systems, other uses
-	- dash, used to provide switch information to commands

DNS Recursion

- A DNS server may offer *recursion,* which allows the server to ask **other** servers
  - Each server is configured with the IP address of one or more root DNS servers.
- When a DNS server receives a response from another server, it replies to the resolver client software. The server also caches the information for future requests.
  - The network administrator of the authoritative DNS server for a name defines the length of time that a non-authoritative server may cache information.

HANDS-ON (from the command line):
nslookup and ping (best to use a count of ~3: ping -c 3)

TYPE:
nslookup cisco.com
Server: 75.75.75.75
Address: 75.75.75.75#53

Non-authoritative answer:
Name: cisco.com
Address: 72.163.4.185
Name: cisco.com
Address: 2001:420:1101:1::185

TYPE:
ping -c 3 cisco.com
PING 72.163.4.185 (72.163.4.185) 56(84) bytes of data.
64 bytes from 72.163.4.185: icmp_seq=1 ttl=239 time=51.2 ms
64 bytes from 72.163.4.185: icmp_seq=2 ttl=239 time=51.6 ms
64 bytes from 72.163.4.185: icmp_seq=3 ttl=239 time=52.1 ms

--- 72.163.4.185 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 51.190/51.604/52.071/0.361 ms

Note the details for each result, the port information from nslookup as well as the ipv6 address. From ping, the IP address and the times, and any packet losses. Limiting counts is better for traffic. Some servers will block pings to reduce issues with possible DOS (denial of service) attacks.

The "Hands-on" commands for networking

Next commands: traceroute and arp

Microsoft Windows	Linux	Mac OSX
which ipconfig (which *not on W10?)*	*which ifconfig*	*which ifconfig*
ipconfig	ifconfig	ifconfig
netstat -r	netstat -r *or* route	netstat -r
hostname	hostname	hostname
nslookup cisco.com	nslookup cisco.com	nslookup cisco.com
ping -c 3 cisco.com	ping -c 3 cisco.com	ping -c 3 cisco.com
tracert cisco.com	traceroute cisco.com	traceroute cisco.com
netstat /?	netstat --help	netstat --help
arp ip address *(displays mac, but only those in arp table)*	arp ip address *(displays mac, but only those in arp table)*	arp ip address *(displays mac, but only those in arp table)*

HANDS-ON – traceroute *(tracepath)*

TYPE: **traceroute cisco.com**

```
traceroute to cisco.com (72.163.4.185), 64 hops max
 1   192.168.1.1  0.357ms  0.255ms  0.166ms
 2   96.120.101.121  9.715ms  10.350ms  9.689ms
 3   24.153.80.205  9.826ms  9.424ms  7.651ms
 4   69.139.160.185  10.875ms  9.688ms  9.774ms
 5   24.124.128.249  10.992ms  11.808ms  8.963ms
 6   24.124.128.122  10.605ms  10.188ms  10.099ms
 7   4.68.37.129  9.949ms  12.976ms  18.816ms
 8   4.69.208.229  49.048ms  57.776ms  51.968ms
 9   4.59.34.66  50.115ms  50.341ms  50.163ms
10   128.107.2.5  50.816ms  50.793ms  50.471ms
11   72.163.0.102  50.361ms  50.031ms  50.795ms
12   72.163.0.186  51.267ms  50.680ms  51.010ms
13   72.163.2.202  50.259ms  72.163.3.2  51.102ms  51.049ms
63   *  *  *
64   *  *  *
```

"Hands-on" - arp
The arp table & command

- ARP (Address Resolution Protocol) bridges Layer 2 and Layer 3 of the OSI model. This function allows for the discovery of MAC (media access control) addresses for its known IP addresses.
- FOR WINDOWS: arp -a
- For Mac OSX: arp <hostname>
- For Linux: arp

```
Address          HWtype   HWaddress           Flags Mask            Iface
mainbox          ether    e0:ad:cb:91:f2:8c   C                     lan0
_gateway         ether    04:c1:51:2f:b5:6d   C                     lan0
genesis          ether    d3:be:d9:9f:4c:62   C                     lan0
revelation       ether    c8:d4:ff:6a:63:24   C                     lan0
```

MS, Linux & Mac OSX files for Networking

Microsoft (XP--Win10) related files: **C:\WINDOWS\system32\drivers\etc**
hosts, networks, protocol, services

/etc/hosts

```
# The following lines are desirable
      for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1 localhost
192.168.1.99 penguin
```

/etc/nsswitch.conf

```
hosts:   files dns mdns4_minimal [NOTFOUND=return] myhostname
      networks:     files
      services:     db files
      ethers:       db files
```

/etc/resolv.conf

```
search com net gov edu
      nameserver 75.75.75.75
      nameserver 8.8.8.8
```

/etc/services

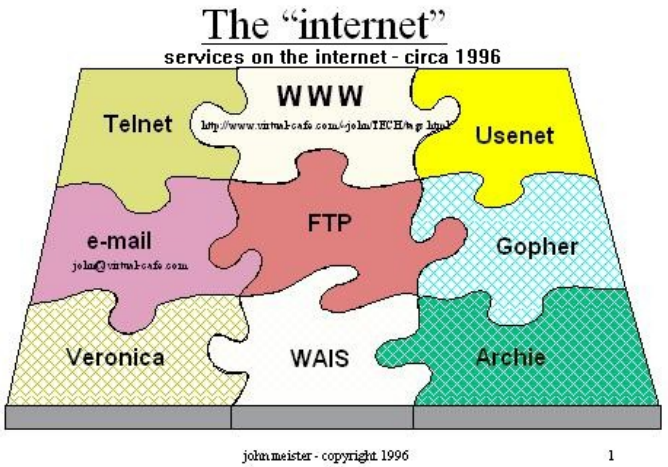
port numbers and tcp/udp as shown previously

Networking Fundamentals - OUTLINE

1) OVERVIEW: Introductions, History, Topologies & ACRONYMS
2) OSI Model - "Programmers Do Not Throw Sausage Pizza Away"
3) TCP Model
4) Protocols and Ports - Well Known Ports & Services
5) IP Addressing - Key elements and
   CIDR, Subnets, IPv6, IP Routing, DHCP, NAT
6) Domain Name System (DNS)
7) **REVIEW AND SUMMARY**

REVIEW **Network** Commands - discussed and exercised

ifconfig -a (or ipconfig /all)
ifconfig -a | grep inet (*MS users, try ipconfig - ymmv!*)
netstat --help ; netstat -r
hostname
nslookup cisco.com
ping -c 3 cisco.com
tracert cisco.com ; **traceroute** cisco.com
arp



KEY files (YMMV - mostly Linux):
/etc/hosts - host names and IP addresses - generally local systems only
/etc/resolv.conf - lists nameservers, e.g. 1.1.1.1, 8.8.8.8, 75.75.75.75 - used for DNS lookup
/etc/nsswitch.conf - identifies which is looked at first, /etc/host FILES, or DNS or ...
/etc/hostname - the local hostname, also found in /etc/hosts
http://johnmeister.com/linux/SysAdmin/NETWORK/networking-analysis-and-tools.html

Thank you!

If you have additional questions,
please reach out to me at: meistertech@gmail.com