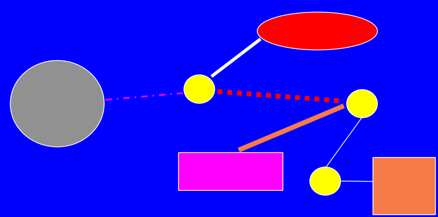
IP and Networking Basics



Outline

- Origins of TCP/IP
- OSI Stack & TCP/IP Architecture
- Client Server Architecture
- IP Addressing & Numbering Rules
- IP Forwarding and default route
- Network Troubleshooting Tools

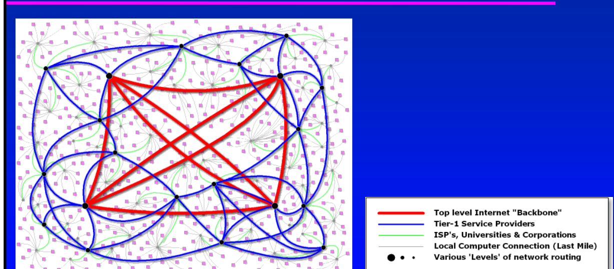
Origins of TCP/IP

- 1950's – 1960's – US Govt. requirement for "rugged" network that would continue to work in case of a nuclear attack
- RAND Corporation (America's leading think thank) & DoD formed ARPA (Advanced Research Project Agency)
- 1968 – ARPA engineers proposed Distributed network design for ARPANET Network

Distributed Network Design

- Pre-ARPANET networks
  - "connection oriented"
  - Management & control was centralized
- "New" Network – ARPANET
  - Connectionless
  - Decentralised
- Modern Internet has evolved from the ARPANET

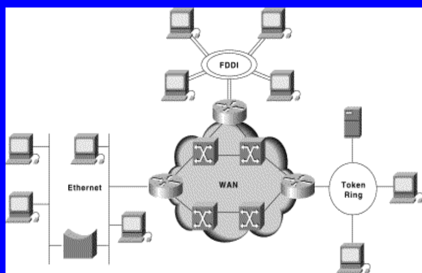
Simplified view of the Internet



Internetworks

- Start with lots of little networks
- Many different types
  - Ethernet, dedicated leased lines, dialup, ATM, Frame Relay, FDDI
- Each type has its own idea of addressing and protocols
- Want to connect them all together and provide a unified view of the whole lot (i.e. act as a single large network)

A small internetwork or "Internet"



The unifying effect of the network layer

- Define a protocol that works in the same way with any underlying network
- Call it the network layer (IP)
- IP routers operate at the network layer
- There are defined ways of using:
  - » IP over Ethernet
  - » IP over ATM
  - » IP over FDDI
  - » IP over serial lines (PPP)
  - » IP over almost anything

OSI Stack & TCP/IP Architecture

What is TCP/IP?

- In simple terms is a language that enables communication between computers
- A set of rules (protocol) that defines how two computers address each other and send data to each other
- Is a suite of protocols named after the two most important protocols TCP and IP but includes other protocols such as UDP, RTP, etc

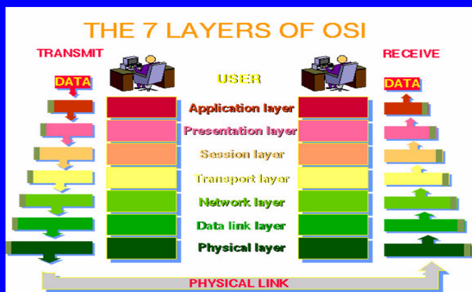
Open Systems & TCP/IP

- TCP/IP formed from standardized communications procedures that were platform independent and open
- Open systems
  - – open architecture - readily available to all
- What is open system networking?
  - – network based on well known and standardized protocols
  - – standards readily available
  - – networking open systems using a network protocol

OSI - Layered Model Concept

- Divide-and-conquer approach
- Dividing requirements into groups, e.g transporting of data, packaging of messages, end user applications
- Each group can be referred to as a **layer**
  - – Upper layers are logically closer to the user and deal with more abstract data, relying on lower layer protocols to translate data into forms that can eventually be physically transmitted.
- Open Systems Interconnection Reference Model (OSI-RM) adopted as a standard for networking

OSI Model



OSI Model

	Layer	
7	Application	APPLICATION • Upper Layers • Application oriented • Independent of layers below
6	Presentation	
5	Session	
4	Transport	TRANSPORT • Lower Layers • Transmission of data • No differentiation of upper layers
3	Network	
2	Data Link	
1	Physical	

Layers 7, 6, 5

- 7: Application layer
  - Provides different services to the applications
  - Uses the underlying layers to carry out work
    - » e.g. SMTP (mail), HTTP (web), Telnet, FTP, DNS
- 6: Presentation layer
  - Converts data from applications into common format and vice versa
- 5: Session layer
  - organizes and synchronizes the exchange of data between application processes

Layer 4

- 4: Transport layer
  - Provides end to end transportation of segments
  - E.g. TCP
    - » encapsulates TCP segments in network layer packets
    - » adds reliability by detecting and retransmitting lost packets
    - » uses acknowledgements and sequence numbers to keep track of successful, out-of-order, and lost packets
    - » timers help differentiate between loss and delay
  - UDP is much simpler: no reliability features

Layer 3

- 3: Network layer
  - Routes the information in the network
  - E.g. IP is a network layer implementation which defines addresses in such a way that route selection can be determined.
    - » Single address space for the entire internetwork
    - » adds an additional layer of addressing, e.g. IP address, which is different from MAC address.

Layer 3

- 3: Network layer (e.g. IP)
  - Unreliable (best effort)
    - » if packet gets lost, network layer doesn't care for higher layers can resend lost packets
  - Forwards packets hop by hop
    - » encapsulates network layer packet inside data link layer frame
    - » different framing on different underlying network types
    - » receive from one link, forward to another link
    - » There can be many hops from source to destination

Layer 3

- 3: Network layer (e.g. IP)
  - Makes routing decisions
    - » how can the packet be sent closer to its destination?
    - » forwarding and routing tables embody “knowledge” of network topology
    - » routers can talk to each other to exchange information about network topology

Layer 2

- 2: Data Link layer
  - Provides reliable transit of data across a physical network link
  - bundles bits into frames and moves frames between hosts on the same link
  - a frame has a definite start, end, size
  - often also a definite source and destination link-layer address (e.g. Ethernet MAC address)
  - some link layers detect corrupted frames while other layers re-send corrupted frames (NOT Ethernet)

Layer 1

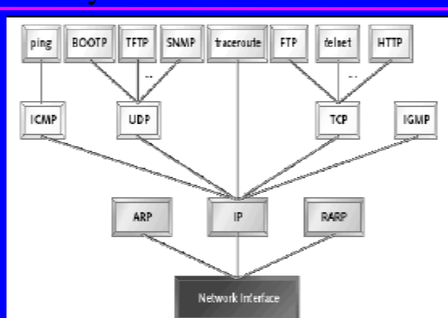
- 1: Physical layer
  - moves bits using voltage, light, radio, etc.
  - no concept of bytes or frames
  - bits are defined by voltage levels, or similar physical properties

1101001000

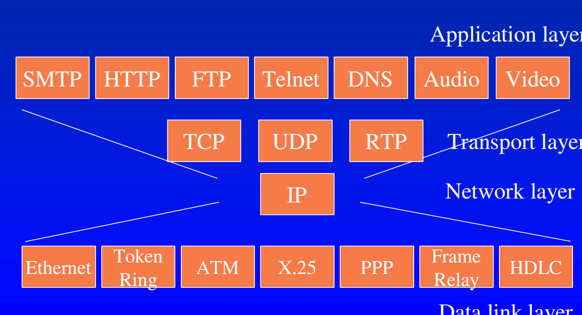
OSI and TCP/IP

	OSI	TCP/IP	
7	Application	Application	
6	Presentation		*Mail, Web, etc.*
5	Session		
4	Transport	Transport	*TCP/UDP – end to end reliability*
3	Network	Network	*IP - Forwarding (best-effort)*
2	Data Link	Data Link & Physical	*Framing, delivery*
1	Physical		*Raw signal*

TCP/IP Layer Model



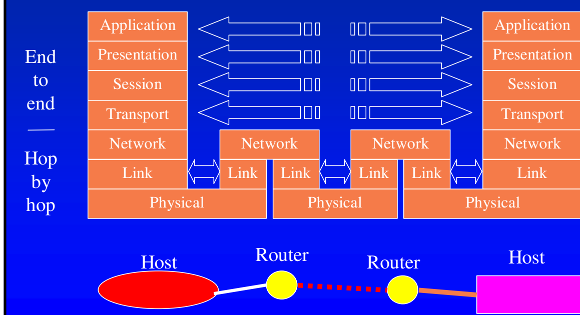
Protocol Layers: The TCP/IP Hourglass Model



Layer Interaction

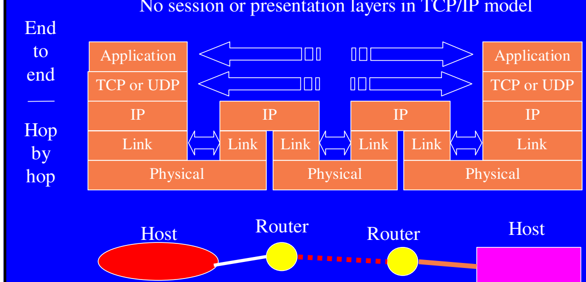
- Application, Presentation and Session protocols are end-to-end
- Transport protocol is end-to-end
  - encapsulation/decapsulation over network protocol on end systems
- Network protocol is throughout the internetwork
  - encapsulation/decapsulation over data link protocol at each hop
- Link and physical layers may be different on each hop

Layer Interaction: OSI 7-Layer Model



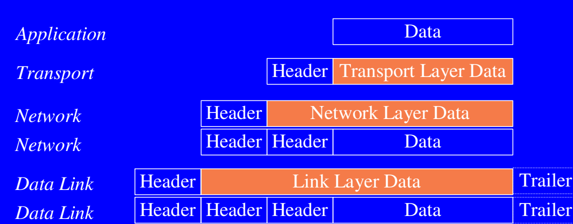
Layer Interaction: TCP/IP Model

No session or presentation layers in TCP/IP model



Encapsulation & Decapsulation

- Lower layers add headers (and sometimes trailers) to data from higher layers



Frame, Datagram, Segment, Packet

- Different names for packets at different layers
  - Ethernet (link layer) frame
  - IP (network layer) datagram
  - TCP (transport layer) segment
- Terminology is not strictly followed
  - we often just use the term "packet" at any layer

Layer 2 - Ethernet frame

Preamble	Dest	Source	Length	Type	Data	CRC
	6 bytes	6 bytes	2 bytes	2 bytes	46 to 1500 bytes	4 bytes

- Destination and source are 48-bit MAC addresses
- Type 0x0800 means that the data portion of the Ethernet frame contains an IP datagram. Type 0x0806 for ARP.

Layer 3 - IP datagram

Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options				Padding
Data				

- Version = 4
- If no options, IHL = 5
- Source and Destination are 32-bit IP addresses
- Protocol = 6 means data portion contains a TCP segment. Protocol = 17 means UDP.

Layer 4 - TCP segment

Source Port								Destination Port
Sequence Number								
Acknowledgement Number								
Data Offset	Reserved	URG	ACK	EOL	RST	SYN	FIN	Window
Checksum								Urgent Pointer
Options								Padding
Data								

Source and Destination are 16-bit TCP port numbers (IP addresses are implied by the IP header)

If no options, Data Offset = 5 (which means 20 octets)

Client Server Architecture

- simple example layer 7 protocol: HTTP
- Client makes requests, Server serves requests – e.g HTTP for transferring “websites”. This is the easiest way to provide services on demand and provides a means of sharing resources more effectively.
- Example: Mimicking the browser with telnet (client) talking to a web server (server)
  - telnet www.google.com 80
  - GET / HTTP/1.1
  - Host: www.google.com

IP Addressing

Purpose of an IP address

- Unique Identification of
  - Source
    Sometimes used for security or policy-based filtering of data
  - Destination
    So the networks know where to send the data
- Network Independent Format
  - IP over anything

Purpose of an IP Address

- Identifies a machine’s connection to a network
- Physically moving a machine from one network to another requires changing the IP address
- TCP/IP uses unique 32-bit addresses

Basic Structure of an IP Address

- 32 bit number (4 octet number):
  (e.g. 133.27.162.125)
- Decimal Representation:

133	27	162	125

- Binary Representation:

10000101	00011011	10100010	01111101

- Hexadecimal Representation:

85	1B	A2	7D

IP Address Allocation

- Private IP address ranges:
  - 10/8 (10.0.0.0 – 10.255.255.255)
  - 192.168/16 (192.168.0.0 – 192.168.255.255)
  - 172.16/12 (172.16.0.0 – 172.31.255.255)
- Public IP address space
  - Assigned by an appropriate authority such as RIPE, ARIN, AFRINIC, etc. or Local Internet Registries (LIRs)
  - Public Address space for the Africa Region available from AfriNIC
- Choose a small block from whatever range you have, and subnet your networks (to avoid problems with broadcasts)

Addressing in Internetworks

- The problem we have
  - More than one physical network
  - Different Locations
  - Larger number of computers
- Need structure in IP addresses
  - network part identifies which network in the internetwork (e.g. the Internet)
  - host part identifies host on that network

Address Structure Revisited

- Hierarchical Division in IP Address:
  - Network Part (Prefix)
    - » describes which physical network
  - Host Part (Host Address)
    - » describes which host on that network

205 . 154 . 8	1
11001101 10011010 00001000	00000001
Network	Host

  - Boundary can be anywhere
    - » very often NOT at a multiple of 8 bits

Network Masks

- Network Masks help define which bits are used to describe the Network Part and which for hosts
- Different Representations:
  - decimal dot notation: 255.255.224.0
  - binary: 11111111 11111111 11100000 00000000
  - hexadecimal: 0xFFFFE000
  - number of network bits: /19
- Binary AND of 32 bit IP address with 32 bit netmask yields network part of address

Classless Addressing

- IP address with the subnet mask defines the range of addresses in the block
  - E.g 10.1.1.32/28 (subnet mask 255.255.255.240) defines the range 10.1.1.32 to 10.1.1.47
  - 10.1.1.32 is the network address
  - 10.1.1.47 is the broadcast address
  - 10.1.1.33 ->46 assignable addresses

Forwarding

- Computers can only send packets directly to other computers on their subnet
- If the destination computer is not on the same subnet, packets are sent via a “gateway”
- defaultrouter option in /etc/rc.conf sets the default gateway for this system.
- IP forwarding on a FreeBSD box
  - turned on with the gateway_enable option in /etc/rc.conf otherwise the box will not forward packets from one interface to another.

How DNS fits

- Computers use IP Addresses but Humans find names easier to remember
- DNS provides a mapping of IP Addresses to names and vice versa
- Computers may be moved between networks, in which case their IP address will change BUT their names can remain the same

Network Troubleshooting Tools

- ping
- traceroute
- tcpdump