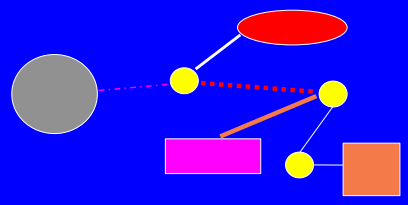
Internetworking, or IP and Networking Basics



Outline

- Origins of TCP/IP
- OSI Stack
- TCP/IP Architecture
- IP Addressing
- Large Network Issues
- Routers
- Routing Protocols

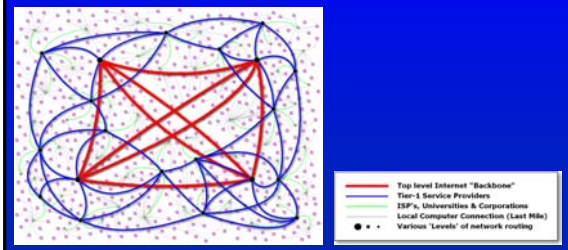
Origins of TCP/IP

- 1950's – 1960's – US Govt. requirement for "rugged" network
- RAND Corporation – Distributed Network Design
- 1968 – ARPA engineers propose Distributed network design for ARPANET (Defense Advanced Research Project Agency 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

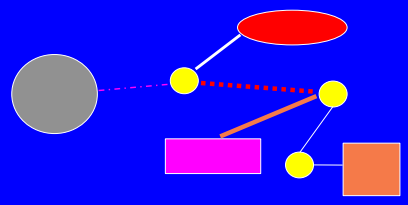
Simplified view of the Internet



What internetworks are

- 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

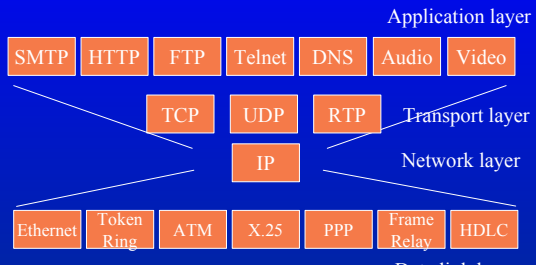
A small internetwork, or "internet"



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 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

Protocol Layers: The TCP/IP Hourglass Model



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

Functions of layers in the OSI 7-layer protocol stack

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

Layer 1

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



1101001000

Layer 2

- 2: Data Link layer
  - bundles bits into frames and moves frames between hosts on the same link
  - a frame has a definite start, end, size
    - » special delimiters to mark start and/or end
  - often also a definite source and destination link-layer address (e.g. ethernet MAC address)
  - some link layers detect corrupted frames
  - some link layers re-send corrupted frames (NOT ethernet)

Layer 3

- 3: Network layer (e.g. IP)
  - Single address space for the entire internetwork
  - adds an additional layer of addressing
    - » e.g. IP address is distinct from MAC address)
    - » so we need a way of mapping between different types of addresses
  - Unreliable (best effort)
    - » if packet gets lost, network layer doesn't care
    - » higher layers can resend lost packets

Layer 3

- 3: Network layer (e.g. IP)
  - 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

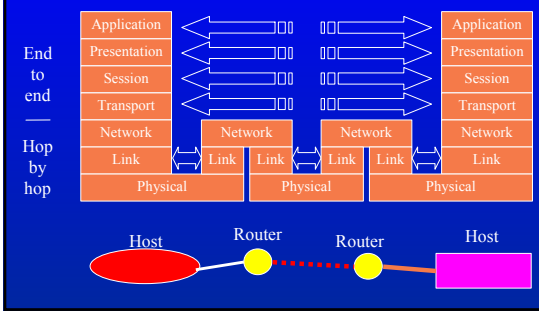
Layer 4

- 4: Transport layer (e.g. TCP)
  - end to end transport of segments
  - 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

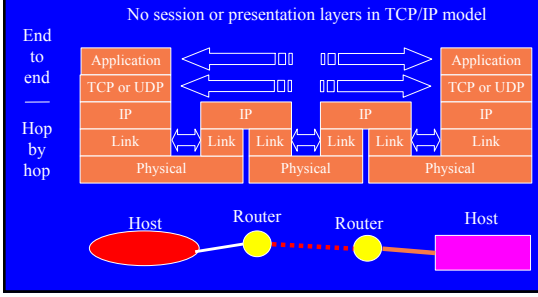
Layer 5, 6, 7

- 5: Session layer
  - not used in the TCP/IP network model
- 6: Presentation layer
  - not used in the TCP/IP network model
- 7: Application layer
  - Uses the underlying layers to carry out work
    - » e.g. SMTP (mail), HTTP (web), Telnet, FTP, DNS

Layer interaction: OSI 7-layer model



Layer interaction: TCP/IP Model

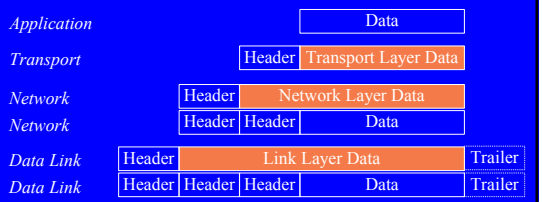


Layer interaction

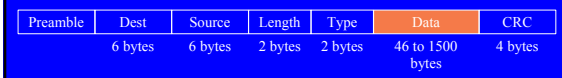
- Application protocol is end-to-end
- Transport protocol is end-to-end
  - encapsulation/decapsulation over network protocol on end systems
- Network protocol is throughout the internetwork
  - encapsulation/decapsulation over data link protocol at each hop
- Link and physical layers may be different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

Encapsulation

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

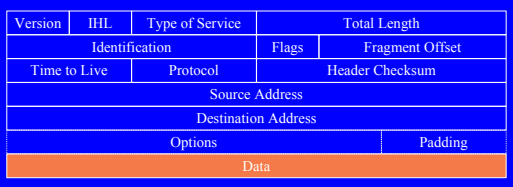


Layer 2 - Ethernet frame



- 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 = 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)

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

Basic Structure of an IP Address

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

133	27	162	125

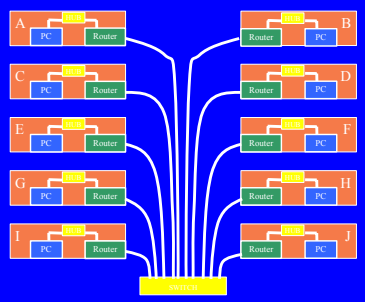
- Binary Representation:

10000101	00011011	10100010	01111101

- Hexadecimal Representation:

85	1B	A2	7D

Address Exercise



Address Exercise

- Construct an IP address for your router's connection to the backbone network.
- 81.199.108.x
- x = 1 for row A, 2 for row B, etc.
- Write it in decimal form as well as binary form.

Addressing in Internetworks

- 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



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

Network Masks

- Define which bits are used to describe the Network Part
- 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

Example Prefixes

- 137.158.128.0/**17** (netmask **255.255.128.0**)

1111 1111	**1111 1111**	**1**000 0000	**0000 0000**
1000 1001	1001 1110	1000 0000	0000 0000

- 198.134.0.0/**16** (netmask **255.255.0.0**)

1111 1111	**1111 1111**	**0000 0000**	**0000 0000**
1100 0110	1000 0110	0000 0000	0000 0000

- 205.37.193.128/**26** (netmask **255.255.255.192**)

1111 1111	**1111 1111**	**1111 1111**	**11**00 0000
1100 1101	0010 0101	1100 0001	10000000

Special Addresses

- All 0's in host part: Represents Network
  - e.g. 193.0.0.0/24
  - e.g. 138.37.128.0/17
- All 1's in host part: Broadcast
  - e.g. 137.156.255.255 (137.156.0.0/16)
  - e.g. 134.132.100.255 (134.132.100.0/24)
  - e.g. 190.0.127.255 (190.0.0.0/17)
- 127.0.0.0/8: Loopback address (127.0.0.1)
- 0.0.0.0: Various special purposes

More Address Exercises

- Assuming there are 11 routers on the classroom backbone network:
  - what is the minimum number of host bits needed to address each router with a unique IP address?
  - what is the corresponding prefix length?
  - what is the corresponding netmask (in decimal)?
  - how many hosts could be handled with that netmask?

Binary arithmetic tutorial

- In decimal (base 10), the number 403 means 4*10^2 + 0*10^1 + 3*10^0, or 4*100 + 0*10 + 3*1, or 400 + 0 + 3
- Similarly, in binary (base 2), the number 1011 means 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0, or 1*8 + 0*4 + 1*2 + 1*1, or 8 + 0 + 2 + 1, which is the same as the decimal number 11

Grouping of decimal numbers

- Suppose we have a lot of 4-digit decimal numbers, 0000 to 9999
- Want to make a group of 10^2 (100) numbers
- Could use 00xx (0000 to 0099), or 31xx (3100 to 3199), or 99xx (9900 to 9999), etc
- Should not use (0124 to 0223) or (3101 to 3200) etc, because they do not form groups in the same way

Grouping of binary numbers

- Suppose we have a lot of 4-bit binary numbers, 0000 to 1111
- Want to make a group of 2^2 (4) numbers
- Could use 00xx (0000 to 0011), or 01xx (0100 to 0111), or 10xx (1000 to 1011), or 11xx (1100 to 1111)
- Should not use (0101 to 1000) or (1001 to 1100) etc, because they do not form groups in the same way

Grouping of decimal numbers

- Given a lot of 4-digit numbers (0000 to 9999)
  - 10^4 = 10000 numbers altogether
- Can have 10^1 (10) groups of 10^3 (1000)
- Can have 10^2 (100) groups of 10^2 (100)
- Can have 10^3 (1000) groups of 10^1 (10)
- Can have 10^4 (10000) groups of 1
- Any large group can be divided into smaller groups, recursively

Grouping of binary numbers

- Given a lot of 4-bit binary numbers (0000 to 1111)
  - 2^4 = 16 numbers altogether
- Can have 2^1 (2) groups of 2^3 (8)
- Can have 2^2 (4) groups of 2^2 (4)
- Can have 2^3 (8) groups of 2^1 (2)
- Can have 2^4 (16) groups of 1
- Any large group can be divided into smaller groups, recursively

Grouping of binary numbers

- Given a lot of 32-bit numbers (0000...0000 to 1111...1111)
  - Can have 2^0 (1) groups of 2^32 numbers
  - Can have 2^8 (256) groups of 2^24 numbers
  - Can have 2^25 groups of 2^7 numbers
- Consider one group of 2^7 (128) numbers
  - » e.g. 11010001101000111011010010xxxxxxx
  - Can divide it into 2^1 (2) groups of 2^6 (64)
  - Can divide it into 2^3 (8) groups of 2^4 (16)
  - etc

More levels of address hierarchy

- Remember hierarchical division of IP address into network part and host part
- Similarly, we can group several networks into a larger block, or divide a large block into several smaller blocks
  - arbitrary number of levels of hierarchy
  - blocks don't all need to be the same size
- Old systems used more restrictive rules
  - New rules are "classless"
  - Old style used Class A, B, C networks

Old-style classes of IP addresses

- Different classes used to represent different sizes of network (small, medium, large)
- Class A networks (large):
  - 8 bits network, 24 bits host (/8, 255.0.0.0)
  - First byte in range 0-127
- Class B networks (medium):
  - 16 bits network, 16 bits host (/16 ,255.255.0.0)
  - First byte in range 128-191
- Class C networks (small):
  - 24 bits network, 8 bits host (/24, 255.255.255.0)
  - First byte in range 192-223

Old-style classes of IP addresses

- Just look at the address to tell what class it is.
  - Class A: 0.0.0.0 to 127.255.255.255
    - » binary 0xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  - Class B: 128.0.0.0 to 191.255.255.255
    - » binary 10xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  - Class C: 192.0.0.0 to 223.255.255.255
    - » binary 110xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  - Class D: (multicast) 224.0.0.0 to 239.255.255.255
    - » binary 1110xxxxxxxxxxxxxxxxxxxxxxxxxxxx
  - Class E: (reserved) 240.0.0.0 to 255.255.255.255

Implied netmasks of classful addresses

- A classful network has a “natural” or “implied” prefix length or netmask:
  - Class A: prefix length /8 (netmask 255.0.0.0)
  - Class B: prefix length /16 (netmask 255.255.0.0)
  - Class C: prefix length /24 (netmask 255.255.255.0)
- Old routing systems often used implied netmasks
- Modern routing systems always use explicit prefix lengths or netmasks

Traditional subnetting of classful networks

- Old routing systems allowed a classful network to be divided into subnets
  - All subnets (of the same classful net) had to be the same size and have the same netmask
  - Subnets could not be subdivided any further
- None of these restrictions apply in modern systems

Traditional supernetting

- Some traditional routing systems allowed supernets to be formed by combining adjacent classful nets.
  - e.g. combine two Class C networks (with consecutive numbers) into a supernet with netmask 255.255.254.0
- Modern systems use more general classless mechanisms.

Classless addressing

- Forget old Class A, Class B, Class C terminology and restrictions
- Internet routing and address management today is classless
- CIDR = Classless Inter-Domain Routing
  - routing does not assume that class A,B,C implies prefix length /8,/16,/24
- VLSM = Variable-Length Subnet Masks
  - routing does not assume that all subnets are the same size

Classless addressing example

- A large ISP gets a large block of addresses
  - e.g., a /16 prefix, or 65536 separate addresses
- Allocate smaller blocks to customers
  - e.g., a /22 prefix (1024 addresses) to one customer, and a /28 prefix (16 addresses) to another customer
- An organisation that gets a /22 prefix from their ISP divides it into smaller blocks
  - e.g. a /26 prefix (64 addresses) for one department, and a /27 prefix (32 addresses) for another department

Classless addressing exercise

- Consider the address block 133.27.162.0/23
- Allocate 8 separate /29 blocks, and one /28 block
- What are the IP addresses of each block?
  - in prefix length notation
  - netmasks in decimal
  - IP address ranges
- What is the largest block that is still available?
- What other blocks are still available?

An IP router

- A device with more than one link-layer interface
- Different IP addresses (from different subnets) on different interfaces
- Receives packets on one interface, and forwards them (usually out of another interface) to get them closer to their destination
- Maintains forwarding tables

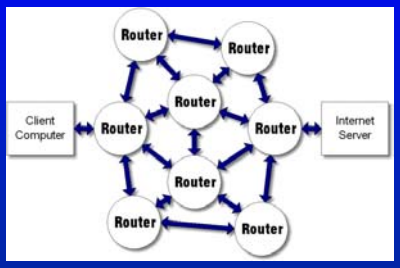
IP router - action for each packet

- Packet is received on one interface
- Check whether the destination address is the router itself
- Decrement TTL (time to live), and discard packet if it reaches zero
- Look up the destination IP address in the forwarding table
- Destination could be on a directly attached link, or through another router

Forwarding is hop by hop

- Each router tries to get the packet one hop closer to the destination
- Each router makes an independent decision, based on its own forwarding table
- Different routers have different forwarding tables
- Routers talk routing protocols to each other, to help update routing and forwarding tables

Hop by Hop Forwarding



Router Functions

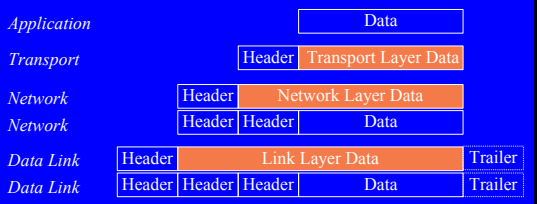
- Determine optimum routing paths through a network
  - » Lowest delay
  - » Highest reliability
- Transport packets through the network
  - » Examines destination address in packet
  - » Makes a decision on which port to forward the packet through
  - » Decision is based on the Routing Table
- Interconnected Routers exchange routing tables in order to maintain a clear picture of the network
- In a large network, the routing table updates can consume a lot of bandwidth
  - » a protocol for route updates is required

Forwarding table structure

- We don't list every IP number on the Internet - the table would be huge
- Instead, the forwarding table contains prefixes (network numbers)
  - "If the first /n bits matches this entry, send the datagram this way"
- If more than one prefix matches, the longest prefix wins (more specific route)
- 0.0.0.0/0 is "default route" - matches anything, but only if no other prefix matches

Encapsulation (reminder)

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



Classes of links

- Different strategies for encapsulation and delivery of IP packets over different classes of links
- Point to point (e.g. PPP)
- Broadcast (e.g. Ethernet)
- Non-broadcast multi-access (e.g. Frame Relay, ATM)

Point to point links

- Two hosts connected by a point-to-point link
  - data sent by one host is received by the other
- Sender takes IP datagram, encapsulates it in some way (PPP, SLIP, HDLC, ...), and sends it
- Receiver removes link layer encapsulation
- Check integrity, discard bad packets, process good packets

Broadcast links

- Many hosts connected to a broadcast medium
  - Data sent by one host can be received by all other hosts
  - example: radio, ethernet

Broadcast links

- Protect against interference from simultaneous transmissions interfering
- Address individual hosts
  - so hosts know what packets to process and which to ignore
  - link layer address is very different from network layer address
- Mapping between network and link address (e.g. ARP)

NBMA links (Non-broadcast multi-access)

- e.g. X.25, Frame Relay, SMDS
- Many hosts
- Each host has a different link layer address
- Each host can potentially send a packet to any other host
- Each packet is typically received by only one host
- Broadcast might be available in some cases

Ethernet Essentials

- Ethernet is a broadcast medium
- Structure of Ethernet frame:

Preamble	Dest	Source	Length	Type	Data	CRC

- Entire IP packet makes data part of Ethernet frame
- Delivery mechanism (CSMA/CD)
  - back off and try again when collision is detected

Ethernet/IP Address Resolution

- Internet Address
  - Unique worldwide (excepting private nets)
  - Independent of Physical Network
- Ethernet Address
  - Unique worldwide (excepting errors)
  - Ethernet Only
- Need to map from higher layer to lower *(i.e. IP to Ethernet, using ARP)*

Address Resolution Protocol

- Check ARP cache for matching IP address
- If not found, broadcast packet with IP address to every host on Ethernet
- "Owner" of the IP address responds
- Response cached in ARP table for future use
- Old cache entries removed by timeout