CpSc 360: Distributed and Network Programming

TCP/IP Protocols

James Wang

Textbook chapter 2.

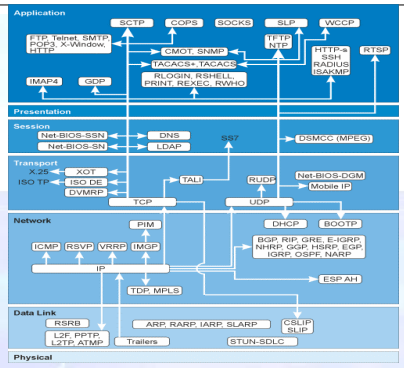
Slides adapted from:
http://www.garykessler.net/library/tcpip.html#IP
http://www.protocols.com/pbook/tcpip1.htm

History of TCP/IP

- The Defense Advance Research Projects Agency (DARPA) originally developed Transmission Control Protocol/Internet Protocol (TCP/IP) to interconnect various defense department computer networks.
- The Internet, an international Wide Area Network, uses TCP/IP to connect government and educational institutions across the world.
- TCP/IP is also in widespread use on commercial and private networks.

2

TCP/IP Suite



3

Ethernet - A Real Data-Link Layer

- It will be useful to discuss a real data-link layer.
- Ethernet (really IEEE 802.3) is widely used.
- Supported by a variety of physical layer implementations.
- Multi-access (shared medium).
- Every Ethernet interface has a unique 48 bit address (a.k.a. *hardware address*).
- Example: `C0:B3:44:17:21:17`
- The broadcast address is all 1's.
- Addresses are assigned to vendors by a central authority.

4

CSMA/CD

- ***Carrier Sense Multiple Access with Collision Detection***
- ***Carrier Sense***: can tell when another host is transmitting
- ***Multiple Access***: many hosts on 1 wire
- ***Collision Detection***: can tell when another host transmits at the same time.

5

An Ethernet Frame

Preamble	Destination Address	Source Address	Len	DATA	CRC
8 bytes	6	6	2	0-1500	4

- The preamble is a sequence of alternating 1s and 0s used for synchronization.
- CRC is Cyclic Redundancy Check
- Each interface looks at every frame and inspects the destination address. If the address does not match the hardware address of the interface (or the broadcast address), the frame is discarded.
- Some interfaces can also be programmed to recognize multicast addresses.

6

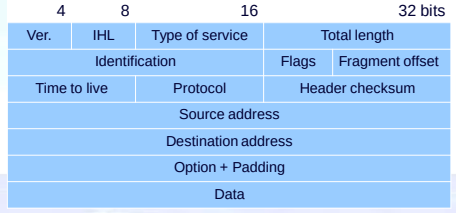
IP

- **The Internet Protocol (IP), defined by IETF RFC791, is the routing layer datagram service of the TCP/IP suite.**
- **All other protocols within the TCP/IP suite, except ARP and RARP, use IP to route frames from host to host.**
- **The IP frame header contains routing information and control information associated with datagram delivery.**
- **IP is the network layer**
  - packet delivery service (host-to-host).
  - translation between different data-link protocols.
- **IP provides connectionless, unreliable delivery of *IP datagrams.***
  - *Connectionless*: each datagram is independent of all others.
  - *Unreliable:* there is no guarantee that datagrams are delivered correctly or even delivered at all.

RFC 791: http://www.cis.ohio-state.edu/htbin/rfc/rfc791.html

7

The IP header structure



8

IP Fields

- **Version:** Version field indicates the format of the Internet header.
- **IHL:** Internet header length is the length of the Internet header in 32-bit words. Points to the beginning of the data. The minimum value for a correct header is 5.
- **Type of service**
  Indicates the quality of service desired. Networks may offer service precedence, meaning that they accept traffic only above a certain precedence at times of high load. There is a three-way trade-off between low delay, high reliability and high throughput.

9

IP Fields (Conti.)

- **Total length**
  - Length of the datagram measured in bytes, including the Internet header and data.
  - This field allows the length of a datagram to be up to 65,535 bytes, although such long datagrams are impractical for most hosts and networks.
  - All hosts must be prepared to accept datagrams of up to 576 bytes, regardless of whether they arrive whole or in fragments.
  - It is recommended that hosts send datagrams larger than 576 bytes only if the destination is prepared to accept the larger datagrams.
- **Identification:** Identifying value assigned by the sender to aid in assembling the fragments of a datagram.
- **Flags:** 3 bits Control flags.

10

IP Fields (Conti.)

- **Fragment offset:** 13 bits. Indicates where this fragment belongs in the datagram. The fragment offset is measured in units of 8 bytes (64 bits). The first fragment has offset zero.
- **Time to live:** Indicates the maximum time the datagram is allowed to remain in the Internet system. If this field contains the value zero, the datagram must be destroyed. This field is modified in Internet header processing. The time is measured in units of seconds. However, since every module that processes a datagram must decrease the TTL by at least one (even if it processes the datagram in less than 1 second), the TTL must be thought of only as an upper limit on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded and to bound the maximum datagram lifetime.
- **Protocol:** Indicates the next level protocol used in the data portion of the Internet datagram.

11

IP Fields (Conti.)

- **Header checksum:** A checksum on the header only. Since some header fields change, e.g., Time To Live, this is recomputed and verified at each point that the Internet header is processed.
- **Source address / destination address**
  32 bits each. A distinction is made between names, addresses and routes. A *name* indicates an object to be sought. An *address* indicates the location of the object. A *route* indicates how to arrive at the object. The Internet protocol deals primarily with addresses. It is the task of higher level protocols (such as host-to-host or application) to make the mapping from names to addresses. The Internet module maps Internet addresses to local net addresses. It is the task of lower level procedures (such as local net or gateways) to make the mapping from local net addresses to routes.
- **Options:** Options may or may not appear in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments, the security option may be required in all datagrams.
- **Data**: IP data or higher layer protocol header.

12

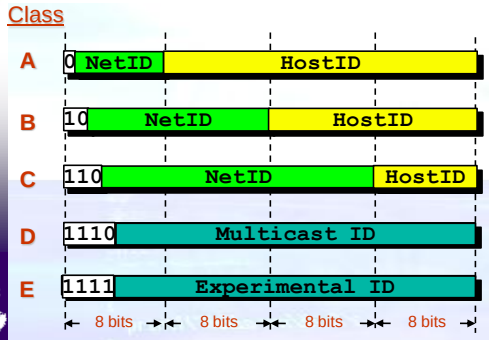
IPv4 Addresses

- IP addresses are not the same as the underlying data-link (MAC) addresses.
- IP is a network layer - it must be capable of providing communication between hosts on different kinds of networks (different data-link implementations).
- The address must include information about what *network* the receiving host is on. This is what makes routing feasible.
- IP addresses are *logical* addresses (not physical)
- 32 bits.
- Includes a network ID and a host ID.
- Every host must have a unique IP address.
- IP addresses are assigned by a central authority (*American Registry for Internet Numbers* for North America).
- IP Addresses are usually shown in *dotted decimal* notation:

  1.2.3.4 00000001 00000010 00000011 00000100

13

Formats of IP Addresses



14

Network and Host IDs

- A Network ID is assigned to an organization by a global authority.
- Host IDs are assigned locally by a system administrator.
- Both the Network ID and the Host ID are used for routing.
- Class A:
  - 128 possible network IDs
  - over 4 million host IDs per network ID
- Class B:
  - 16K possible network IDs
  - 64K host IDs per network ID
- Class C:
  - over 2 million possible network IDs
  - about 256 host IDs per network ID

15

Special Network Addresses

- **Addresses beginning with 01111111, or 127 decimal, are reserved for loopback and for internal testing on a local machine.**
  - You can test this: you should always be able to ping 127.0.0.1, which points to yourself.
- **There are three IP network addresses reserved for private networks. The addresses are 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16.**
  - They can be used by anyone setting up internal IP networks, such as a lab or home LAN behind a NAT or proxy server or a router.
  - It is always safe to use these because routers on the Internet will never forward packets coming from these addresses.

16

Host and Network Addresses

- **A single network interface is assigned a single IP address called the *host* address.**
- **A host may have multiple interfaces, and therefore multiple *host* addresses.**
- **Hosts that share a network all have the same IP *network* address (the network ID).**

17

IP Broadcast and Network Addresses

- **An IP broadcast addresses has a host ID of all 1s.**
- **IP broadcasting is not necessarily a true broadcast, it relies on the underlying hardware technology.**
- **An IP address that has a host ID of all 0s is called a *network address* and refers to an entire network.**

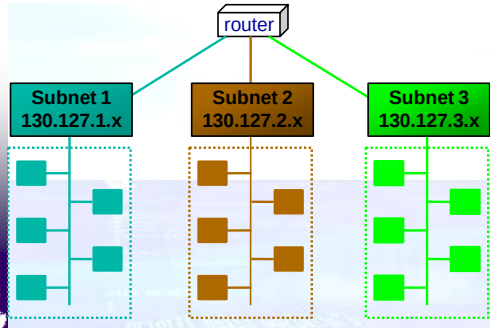
18

Subnet Addresses

- An organization can subdivide it's host address space into groups called subnets.
- The subnet ID is generally used to group hosts based on the physical network topology.

19

Subnetting



20

Subnetting

- Subnets can simplify routing.
- IP subnet broadcasts have a hostID of all 1s.
- It is possible to have a single wire network with multiple subnets.

21

Mapping Addresses

- IP Addresses are not recognized by hardware.
- If we know the IP address of a host, how do we find out the hardware address ?
- The process of finding the hardware address of a host given the IP address is called ***Address Resolution***
- The process of finding out the IP address of a host given a hardware address is called ***Reverse Address Resolution***
- Reverse address resolution is needed by diskless workstations when booting (which used to be quite common).

22

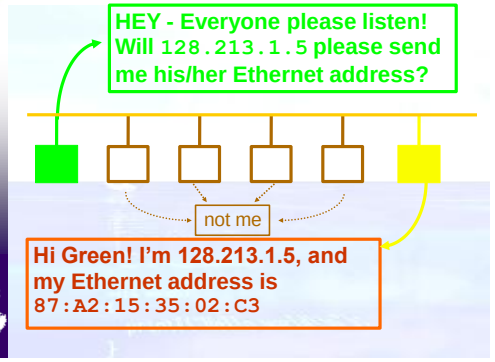
ARP/RARP

- TCP/IP uses the Address Resolution Protocol (ARP) and the Reverse Address Resolution Protocol (RARP) to initialize the use of Internet addressing on an Ethernet or other network that uses its own media access control (MAC).
- The *Address Resolution Protocol* is used by a sending host when it knows the IP address of the destination but needs the Ethernet (or whatever) address.
- ARP is a broadcast protocol - every host on the network receives the request.
- Each host checks the request against it's IP address - the right one responds.
- ARP does not need to be done every time an IP datagram is sent - hosts *remember* the hardware addresses of each other.
- Part of the ARP protocol specifies that the receiving host should also remember the IP and hardware addresses of the sending host.

RFC826 http://www.cis.ohio-state.edu/htbin/rfc/rfc826.html
RFC1293 http://www.cis.ohio-state.edu/htbin/rfc/rfc1293.html
RFC2390 http://www.cis.ohio-state.edu/htbin/rfc/rfc2390.html
RFC1390 http://www.cis.ohio-state.edu/htbin/rfc/rfc1390.html

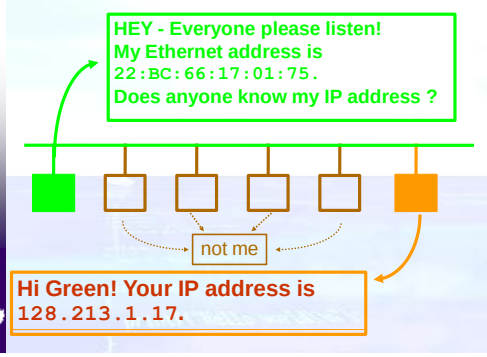
23

ARP conversation



24

RARP conversation



25

The ARP/RARP header structure

16	32 bits
Hardware Type	Protocol Type
HLen (8) / Plen (8)	Operation
Sender Hardware Address	
Sender Protocol Address	
Target Hardware Address	
Target Protocol Address	

- **Operation:**

1. ARP request.
2. ARP response.
3. RARP request.
4. RARP response.
5. Dynamic RARP request.
6. Dynamic RARP reply.
7. Dynamic RARP error.
8. InARP request.
9. InARP reply.

26

Fragmentation, Flow Control & Error Detection

- Each fragment (packet) has the same structure as the IP datagram.
- IP specifies that datagram reassembly is done only at the destination (not on a hop-by-hop basis).
- If any of the fragments are lost - the entire datagram is discarded (and an ICMP message is sent to the sender).
- If packets arrive too fast - the receiver discards excessive packets and sends an ICMP message to the sender (SOURCE QUENCH).
- If an error is found (header checksum problem) the packet is discarded and an ICMP message is sent to the sender.

27

ICMP

- IETF RFC792 defines the Internet Control Message Protocol (ICMP).
- ICMP messages generally contain information about routing difficulties with IP datagrams or simple exchanges such as time-stamp or echo transactions.
- ICMP is a protocol used for exchanging control messages.
- ICMP uses IP to deliver messages.
- ICMP messages are usually generated and processed by the IP software, not the user process.

RFC792 http://www.cis.ohio-state.edu/htbin/rfc/rfc792.html
RFC950 http://www.cis.ohio-state.edu/htbin/rfc/rfc950.html

28

UDP

- The User Datagram Protocol (UDP), defined by IETF RFC768, provides a simple, but unreliable message service for transaction-oriented services.
- Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among hosts.
- UDP is a transport protocol
  - communication between processes
- UDP uses IP to deliver datagrams to the right host.
- UDP uses *ports* to provide communication services to individual processes.

RFC768 http://www.cis.ohio-state.edu/htbin/rfc/rfc768.html

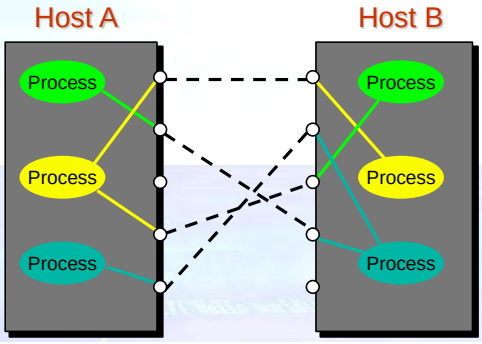
29

Ports

- TCP/IP uses an abstract destination point called a protocol port.
- Ports are identified by a positive integer.
- Operating systems provide some mechanism that processes use to specify a port.

30

Ports



31

The UDP header structure

16	32 bits
Source port	Destination port
Length	Checksum
Data	

- **Source port:** Source port is an optional field. When used, it indicates the port of the sending process and may be assumed to be the port to which a reply should be addressed in the absence of any other information. If not used, a value of zero is inserted.
- **Destination port:** Destination port has a meaning within the context of a particular Internet destination address.
- **Length:** The length in octets of this user datagram, including this header and the data. The minimum value of the length is eight.
- **Checksum:** The 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.
- **Data:** UDP data field.

32

TCP

- **IETF RFC793 defines the Transmission Control Protocol (TCP).**
- **TCP provides a reliable stream delivery and virtual connection service to applications through the use of sequenced acknowledgment with retransmission of packets when necessary.**
- **TCP is an alternative transport layer protocol supported by TCP/IP.**

RFC793 http://www.cis.ohio-state.edu/htbin/rfc/rfc793.html
RFC1146 http://www.cis.ohio-state.edu/htbin/rfc/rfc1146.html
RFC1072 http://www.cis.ohio-state.edu/htbin/rfc/rfc1072.html
RFC 1323 http://www.cis.ohio-state.edu/htbin/rfc/rfc1323.html
RFC1693 http://www.cis.ohio-state.edu/htbin/rfc/rfc1693.html

33

TCP Features

- **Connection-Oriented**
  - *Connection oriented* means that a virtual connection is established before any user data is transferred.
  - If the connection cannot be established - the user program is notified (finds out).
  - If the connection is ever interrupted - the user program(s) is finds out there is a problem.
- **Reliable**
  - *Reliable* means that every transmission of data is acknowledged by the receiver.
  - If the sender does not receive acknowledgement within a specified amount of time, the sender retransmits the data.

34

TCP Features (Conti.)

- **Data Stream**
  - *Stream* means that the connection is treated as a stream of bytes.
  - The user application does not need to package data in individual datagrams (as with UDP).
- **Buffering**
  - TCP is responsible for buffering data and determining when it is time to send a datagram.
  - It is possible for an application to tell TCP to send the data it has buffered without waiting for a buffer to fill up.
- **Full Duplex**
  - TCP provides transfer in both directions (over a single virtual connection).
  - To the application program these appear as 2 unrelated data streams, although TCP can piggyback control and data communication by providing control information (such as an ACK) along with user data.

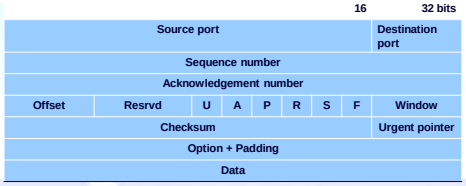
35

Addressing in TCP/IP

- **Each TCP/IP address includes:**
  - Internet Address
  - Protocol (UDP or TCP)
  - Port Number
- **TCP Ports**
  - Interprocess communication via TCP is achieved with the use of ports (just like UDP).
  - UDP ports have no relation to TCP ports (different name spaces).

36

The TCP header structure



37

TCP Fields

- **Source port:** Source port number.
- **Destination port:** Destination port number.
- **Sequence number:** The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present, the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.
- **Acknowledgment number:** If the ACK control bit is set, this field contains the value of the next sequence number which the sender of the segment is expecting to receive. Once a connection is established, this value is always sent.
- **Data offset:** 4 bits. The number of 32-bit words in the TCP header, which indicates where the data begins. The TCP header (even one including options) has a length which is an integral number of 32 bits.

38

TCP Fields (conti.)

- **Reserved:** 6 bits. Reserved for future use. Must be zero.
- **Control bits:** 6 bits. The control bits may be (from right to left):
  - U (URG)Urgent pointer field significant.
  - A (ACK)Acknowledgment field significant.
  - P (PSH)Push function.
  - R (RST) Reset the connection.
  - S (SYN)Synchronize sequence numbers.
  - F (FIN)No more data from sender.
- **Window:** 16 bits. The number of data octets which the sender of this segment is willing to accept, beginning with the octet indicated in the acknowledgment field.

39

TCP Fields (conti.)

- **Checksum**
  16 bits. The checksum field is the 16 bit one's complement of the one's complement sum of all 16-bit words in the header and text. If a segment contains an odd number of header and text octets to be checksummed, the last octet is padded on the right with zeros to form a 16-bit word for checksum purposes. The pad is not transmitted as part of the segment. While computing the checksum, the checksum field itself is replaced with zeros.
- **Urgent Pointer**
  16 bits. This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field can only be interpreted in segments for which the URG control bit has been set.
- **Options**
  Options may be transmitted at the end of the TCP header and always have a length which is a multiple of 8 bits. All options are included in the checksum. An option may begin on any octet boundary.
- **Data:** TCP data or higher layer protocol.

40

TCP vs. UDP

Q: Which protocol is better ?

A: It depends on the application.

- **TCP provides a connection-oriented, reliable, byte stream service (lots of overhead).**
- **UDP offers minimal datagram delivery service (as little overhead as possible).**

41

TCP/IP Summary

- **IP: network layer protocol**
  - unreliable datagram delivery between hosts.
- **UDP: transport layer protocol**
  - unreliable datagram delivery between processes.
- **TCP: transport layer protocol**
  - reliable, byte-stream delivery between processes.

42