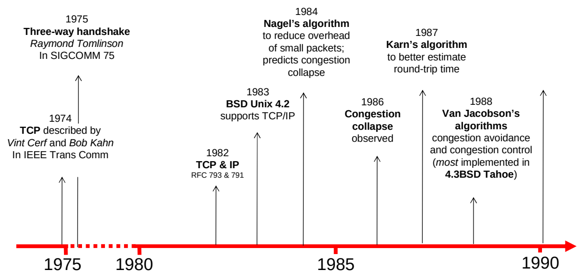
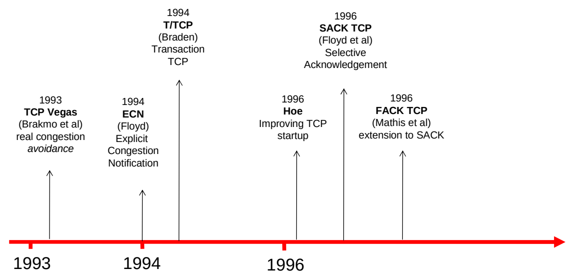
TCP Evolution



1

TCP Through the 90s



2

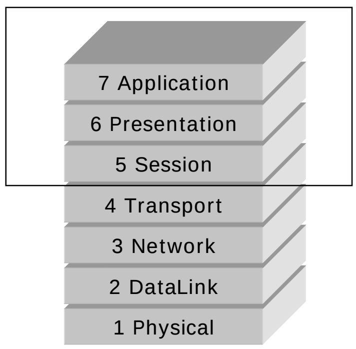
TCP / UDP Comparison

- TCP:
  - Connection-oriented protocol
  - Full-duplex
  - Messages received in order, no loss or duplication
  - Error, flow and congestion control
  - $\Rightarrow$ *Reliable but with overhead*

- UDP:
  - Messages called ``datagrams''
  - Messages may be lost or duplicated
  - Messages may be received out of order
  - $\Rightarrow$ *Unreliable but potentially faster*

3

Where Are We?



4

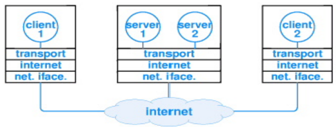
Separation of Duties

- Network
  - – Transfer bits/bytes
  - – Operates at the application's request
- Applications
  - – What data to transmit
  - – When to transmit data
  - – Where to transmit data to
  - – Meaning of bits/bytes

5

Client-Server Paradigm

- Network applications use a form of communication known as the client-server paradigm
- A server application waits passively for contact, while a client application initiates communication actively



6

Characteristics

- Client
  - – Actively initiates contact with server(s)
  - – Invoked directly by a user
  - – Uses different source port for each connection
- Server
  - – Passively awaits connections from clients
  - – Accepts contact from arbitrary clients, but offers a single service
  - – Uses the same source port for all clients

7

Use of Ports

- Each service given unique port number, *P*
- Server
  - – Informs operating system it is using port *P*
  - – Waits for requests to arrive
- Client
  - – Forms request
  - – Sends request to port *P* on server computer

8

Interacting with Protocol Software

- Client or server uses transport protocols
- Protocol software resides in OS (kernel)
- Applications outside the OS (userland)
- Mechanism to bridge the two
  - *Application Programming Interface (API)*

9

Application Programming Interface

- Part of the operating system
- Permits applications to use protocols
- Defines
  - Operations allowed
  - Arguments for each operation

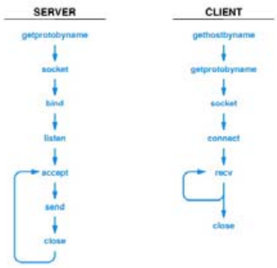
10

Example API: Sockets

- Originally designed
  - On and for BSD UNIX
  - To use with TCP/IP protocols
- Now
  - Industry standard
  - Available on many operating systems and languages
    - E.g. Winsock
    - Java, Python, etc. sockets

11

Sample Socket Procedure



12

Automated Protocol Configuration

- In a layered protocol stack, like TCP/IP, layers are configured from lowest to highest
- Higher-layer protocols use lower layer protocols to obtain configuration information
- Broadcasting main mechanism

- Must be supported by the networking technology, e.g. Ethernet

- Applications that use broadcasting to obtain configuration information must be prepared to receive multiple responses

  * One approach: Accept first, ignore others

13

Bootstrapping: Sequence of Protocols Used

- Step 1: Broadcast *RARP request* to obtain IP address
- Step 2: Wait for *RARP response*; if none arrives within T1 seconds go to Step 1
- Step 3: Broadcast *ICMP address mask request*
- Step 4: Wait for *ICMP address mask response*; if none arrives within T2 seconds go to Step 3
- Step 5: Use *ICMP gateway discovery* to find IP address of default router and add default route to the routing table

14

RARP Limitations

- Provides a mechanism for a host to determine its IP address when it is only aware of its MAC address
- RARP can determine and configure a diskless workstation with an IP address
- No method for determining and delivering other configuration data
- RARP server can service only a single subnet because of its complete reliance on Ethernet broadcasts

15

Bootstrap Protocol (BOOTP)

- BOOTP allows hosts to be dynamically configured to use the TCP/IP protocol suite
- BOOTP delivers configuration information automatically instead of each TCP/IP host on a network being manually configured
- BOOTP is based on UDP so it uses IP for transport and is routeable
- RFC 951

16

BOOTP Operation

- BOOTP is a client/server process where the BOOTP client, during the boot phase, requests configuration information from a BOOTP server
- After receiving a request from a BOOTP client, the server looks up the clients MAC address in its BOOTP configuration database and sends a reply containing IP configuration information
- Client receives the reply and configures its TCP/IP stack
- BOOTP client will also load a boot file if the BOOTP server supplies a path using the fully qualified filename.
- The magic cookie, a mechanism for a BOOTP server to supply vendor-specific operating system options to a BOOTP client

  - Options include DNS servers, WINS or NetBIOS name server, time servers, etc.

17

BOOTP Messages

- BOOTREQUEST

  - Includes information that allows the BOOTP server to determine what configuration data it must supply to the requesting host
- BOOTREPLY

  - Includes the information that the client requested from the server
- Have the same packet structure

18

BOOTREQUEST (1/2)

- Contains following information:

1. Source's MAC address
2. Destination's MAC address
3. Destination's IP address
4. Source's IP address
5. Destination server hostname
6. Boot filename
7. Vendor-specific data (e.g. boot image via TFTP)

19

BOOTREQUEST (2/2)

- Appears to use IP that has not been configured
- IP uses the all-1's broadcast address as a destination address, and all-0's as a source address
- If a computer uses the all-0's address to send a request a BOOTP server either uses broadcast to respond or uses the MAC address of the incoming frame to respond via a unicast
- Server must avoid ARP requests since the client does not know its IP address and cannot answer them

20

BOOTP Limitations

- BOOTP database is a static text file
  - File has to be maintained by hand as changes are made to the network
- Unable to dynamically allocate and distribute IP addresses
  - Dynamic IP addressing critical for many application scenarios

Dynamic Host Configuration Protocol

- DHCP: Dynamic Host Configuration Protocol (RFC 1531)
- Superset of BOOTP, provides the same service with more options
- Does not require an administrator to add an entry for each computer to the database that a server uses
- *Plug-and-play networking*

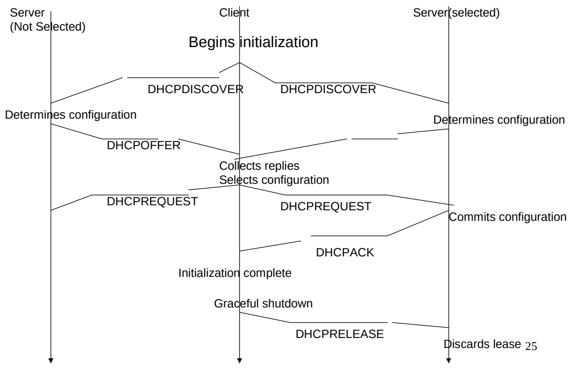
DHCP Operation (1/2)

- IP addresses bound to workstations dynamically
  - Workstation broadcasts DHCPDISCOVER message on boot
  - Several DHCP servers may respond with DHCPOFFER messages containing:
    - * IP address, subnet mask
    - * Default router IP address
    - * Renewal time

DHCP Operation (2/2)

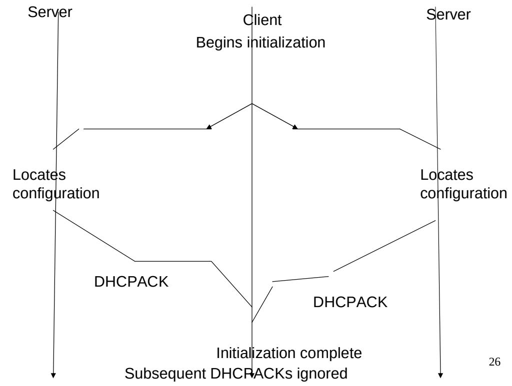
- Workstation responds to one offer with DHCPREQUEST
  - * Request may include items like: DNS servers, time servers, boot files, etc.
- DHCP server now binds IP address and replies with DHCPACK message with requested options

DHCP New Address Allocation



25

DHCP Previously Allocated Address



26

DHCP Server Configuration

- System administrator assigns multiple ranges of IP addresses to each DHCP server and server manages distribution to clients
- Client must renew IP address at regular intervals indicated by *Renewal time*

27

DHCP Message Format (1/2)

op (1)	htype (1)	hlen (1)	hops (1)
xid (4)			
secs (2)		flags (2)	
ciaddr (4)			
yiaddr (4)			
siaddr (4)			
giaddr (4)			
chaddr (16)			
sname (64)			
file (128)			
options (312)			

28

DHCP Message Format (2/2)

Fields	Bytes	Descriptions
op	1	op code/message type
htype	1	Hardware address type
hlen	1	hardware address length
hops	1	Client sets to zero, optionally used by relay agents when booting via relay agent
xid	4	Transaction ID
secs	2	Filled in by client, seconds elapsed since client began address acquisition or renewal process.
flags	2	Flags
ciaddr	4	Client IP address
yiaddr	4	``your'' (client) IP address
siaddr	4	IP address of next server to use in bootstrap; returned in DHCPOFFER, DHCPACK by server.
giaddr	4	Relay agent IP address
chaddr	16	Client hardware address
sname	64	Optional server host name, null terminated string.
file	128	Boot file name
options	var	Optional parameters field.

29

IPv6 Autoconfiguration

- IPv6 has a stateless autoconfiguration capability
- Stateless autoconfiguration does not require manual host configuration, minimal router configuration, and no additional servers
- A host is able generate its own addresses using a combination of locally available information and information advertised by routers

30

DHCPv6 (DHCP for IPv6)

- DHCPv6 complements IPv6 by providing a stateful autoconfiguration option to facilitate the automatic configuration of DHCP clients (RFC 3315)
- Hosts obtain interface addresses and/or configuration information and parameters from a server
- The server maintains a database of the addresses that have been assigned to hosts
- DHCPv6 uses a combination of unicast and multicast messages instead of broadcast messages for the majority of its conversation

31