Session 2B

TCP/UDP Ports and TCP Header

Mouli Sankaran

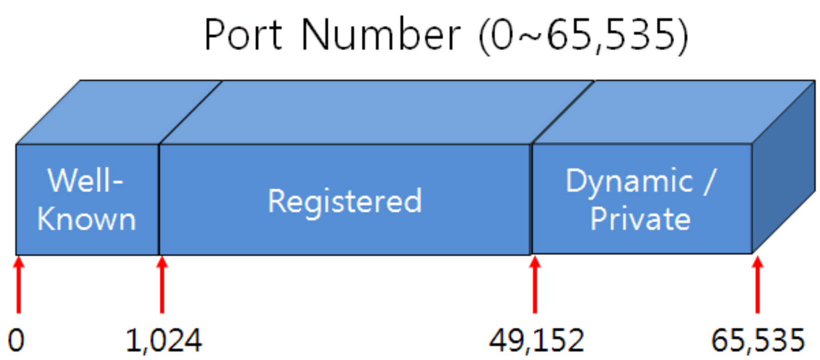
Session 2B: Focus

- TCP/UDP Ports
  - Port Allocations
  - TCP-UDP flow of segments
  - Well-Known Ports
- TCP Segment Header Structure
  - Sequence Number
  - ACK bit and Acknowledgement Number
  - TCP Checksum
    - Pseudo-Header

Course page where the course materials will be posted as the course progresses:

TCP/UDP- Ports

TCP/UDP Port Allocations – Well-Known Ports



How many **bits** are reserved for **port numbers** on the **TCP header**? **ANS: 16 bits**

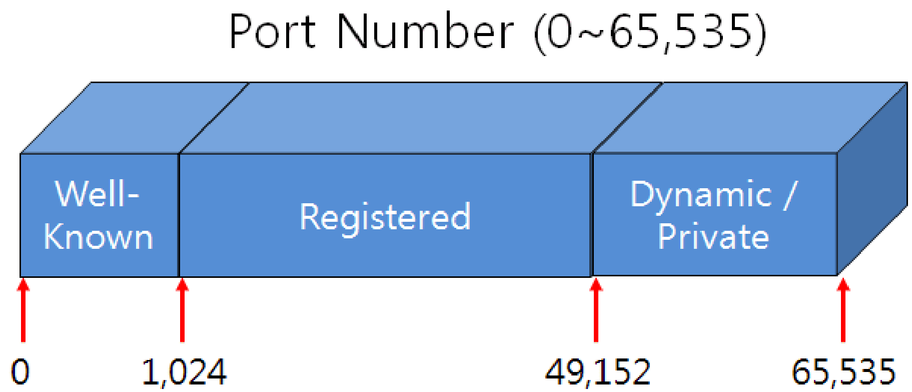
Well-Known ports These ports are reserved for widely used protocols and services, ensuring consistent communication across different systems and networks.

The **port** numbers direct packets to the appropriate applications running in the servers.

- The **port numbers** in the range from **0 to 1023** are the **Well-Known ports**.
- They are used by system processes that provide widely used types of network services (FTP, HTTP, DNS, SMTP, DHCP
  - On Unix-like operating systems, a process must execute with **super-user privileges** to be able to bind a network socket to an IP address using one of the well-known ports
- The **dynamic port numbers** (also known as the private port numbers) from **49,152 to 65,535** are the port numbers that are available for **temporary use** by any application for communicating with any other application
- These private ports cannot be registered with **IANA.**

IANA: Internet Assigned Numbers Authority

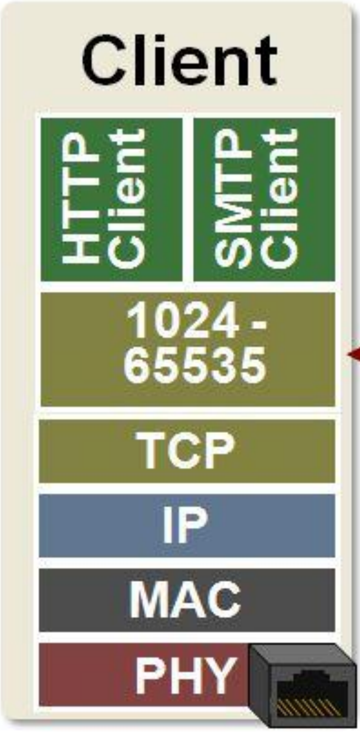
TCP/UDP Port Allocations: Registered Ports



- The range of port numbers from **1024 to 49151** are the **registered ports**.
- They are assigned by **IANA** for specific service, based on specific network applications offered by any organization.
- Commonly used by proprietary applications, third-party software, and specialized services that are not covered by Well-Known Ports
  - **Example**: Microsoft SQL Server (1433), NFS (2049), MySQL (3306), SIP (5060), etc.
- Administrative privileges are not required, user applications can use it
- On most systems, registered ports can be used by ordinary users as well

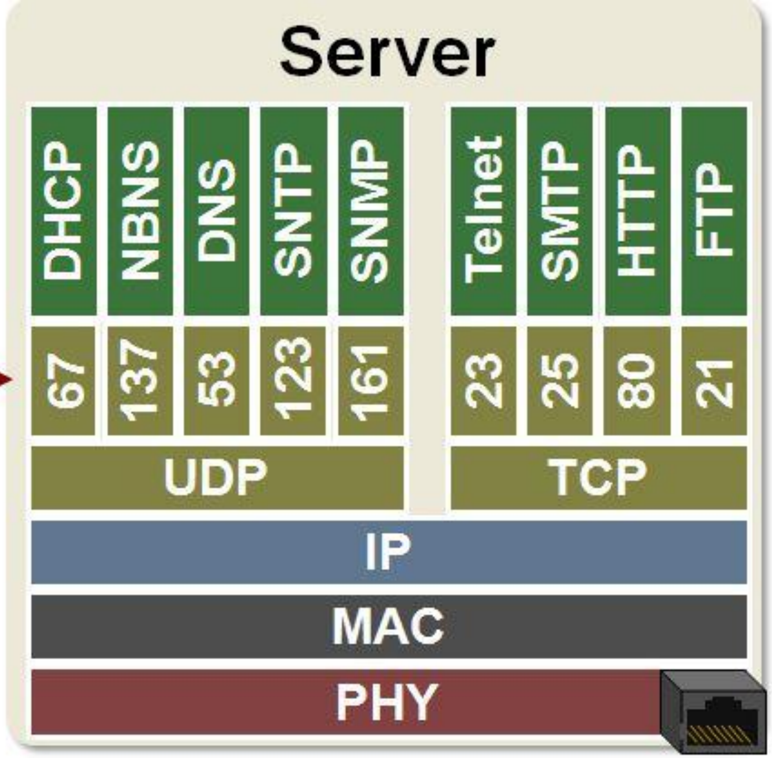
NFS: Network File System (file sharing), SIP: Session Initiation Protocol (for VoIP applications)

Ports used by Servers and Clients



NBNS: NetBIOS Name Service

SNTP: Simple Network Time Protocol



- Well defined Server ports for specific purposes/applications
- Clients connect to the specific server ports based on the application
  - Web browsers (clients) connect to HTTP server ports (80) on the Web server
  - Port number 8080 is also used for web services, as an alternative port for HTTP traffic, often used for proxy servers, for development, and testing

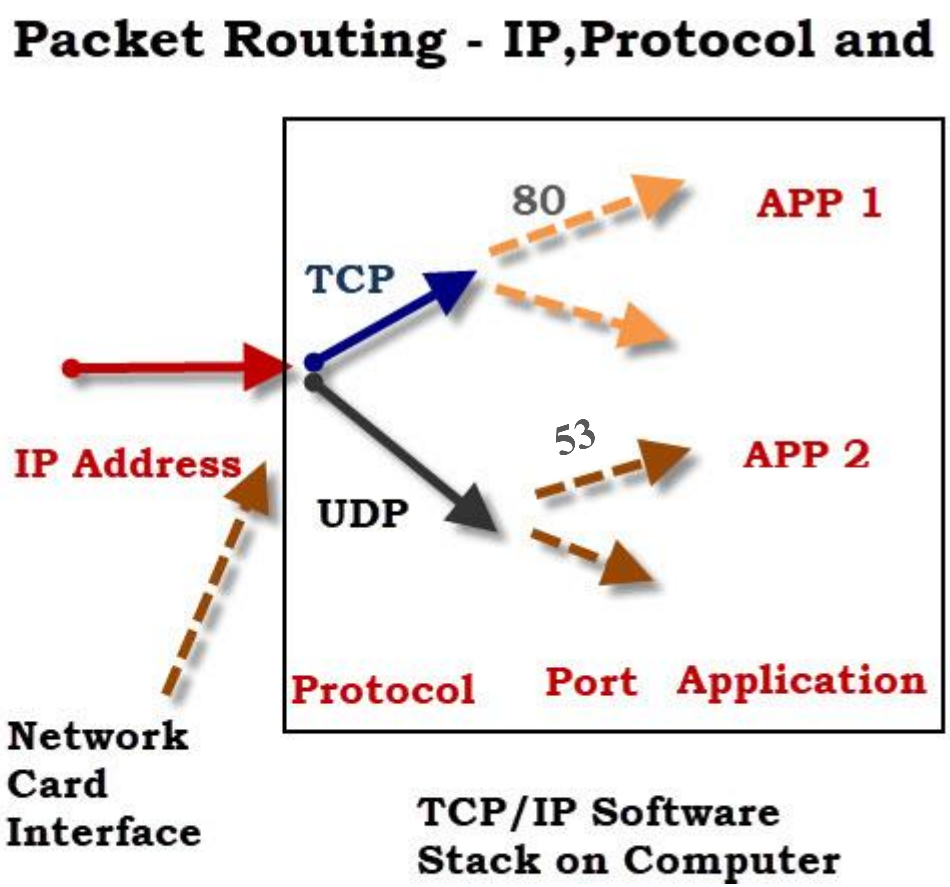
Well-known: TCP/UDP Ports

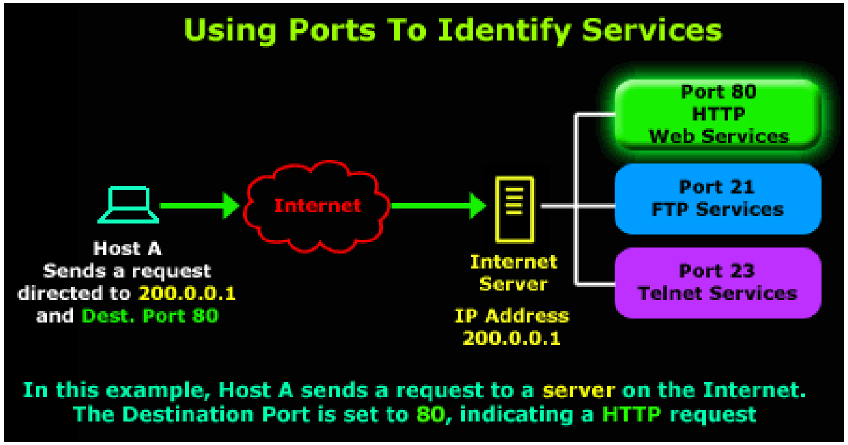
7	Echo	554	RTSP	2745	Bagle.H	6891-6901	Windows Live
19	Chargen	546-547	DHCPv6	2967	Symantec AV	6970	Quicktime
20-21	FTP	560	rmonitor	3050	Interbase DB	7212	GhostSurf
22	SSH/SCP	563	NNTP over SSL	3074	XBOX Live	7648-7649	CU-SeeMe
23	Telnet	587	SMTP	3124	HTTP Proxy	8000	Internet Radio
25	SMTP	591	FileMaker	3127	MyDoom	8080	HTTP Proxy
42	WINS Replication	593	Microsoft DCOM	3128	HTTP Proxy	8086-8087	Kaspersky AV
43	WHOIS	631	Internet Printing	3222	GLBP	8118	Privoxy
49	TACACS	636	LDAP over SSL	3260	iSCSI Target	8200	VMware Server
53	DNS	639	MSDP (PIM)	3306	MySQL	8500	Adobe ColdFusion
67-68	DHCP/BOOTP	646	LDP (MPLS)	3389	Terminal Server	8767	TeamSpeak
69	TFTP	691	MS Exchange	3689	iTunes	8866	Bagle.B
70	Gopher	860	iSCSI	3690	Subversion	9100	HP JetDirect
79	Finger	873	rsync	3724	World of Warcraft	9101-9103	Bacula
80	HTTP	902	VMware Server	3784-3785	Ventrilo	9119	MXit
88	Kerberos	989-990	FTP over SSL	4333	mSQL	9800	WebDAV
102	MS Exchange	993	IMAP4 over SSL	4444	Blaster	9898	Dabber
110	POP3	995	POP3 over SSL	4664	Google Desktop	9988	Rbot/Spybot
113	Ident	1025	Microsoft RPC	4672	eMule	9999	Urchin
119	NNTP (Usenet)	1026-1029	Windows Messenger	4899	Radmin	10000	Webmin

Note: TCP and UDP ports are independent of each other. They need not be mutually exclusive. Which means that the same port number can provide different services on TCP and UDP.

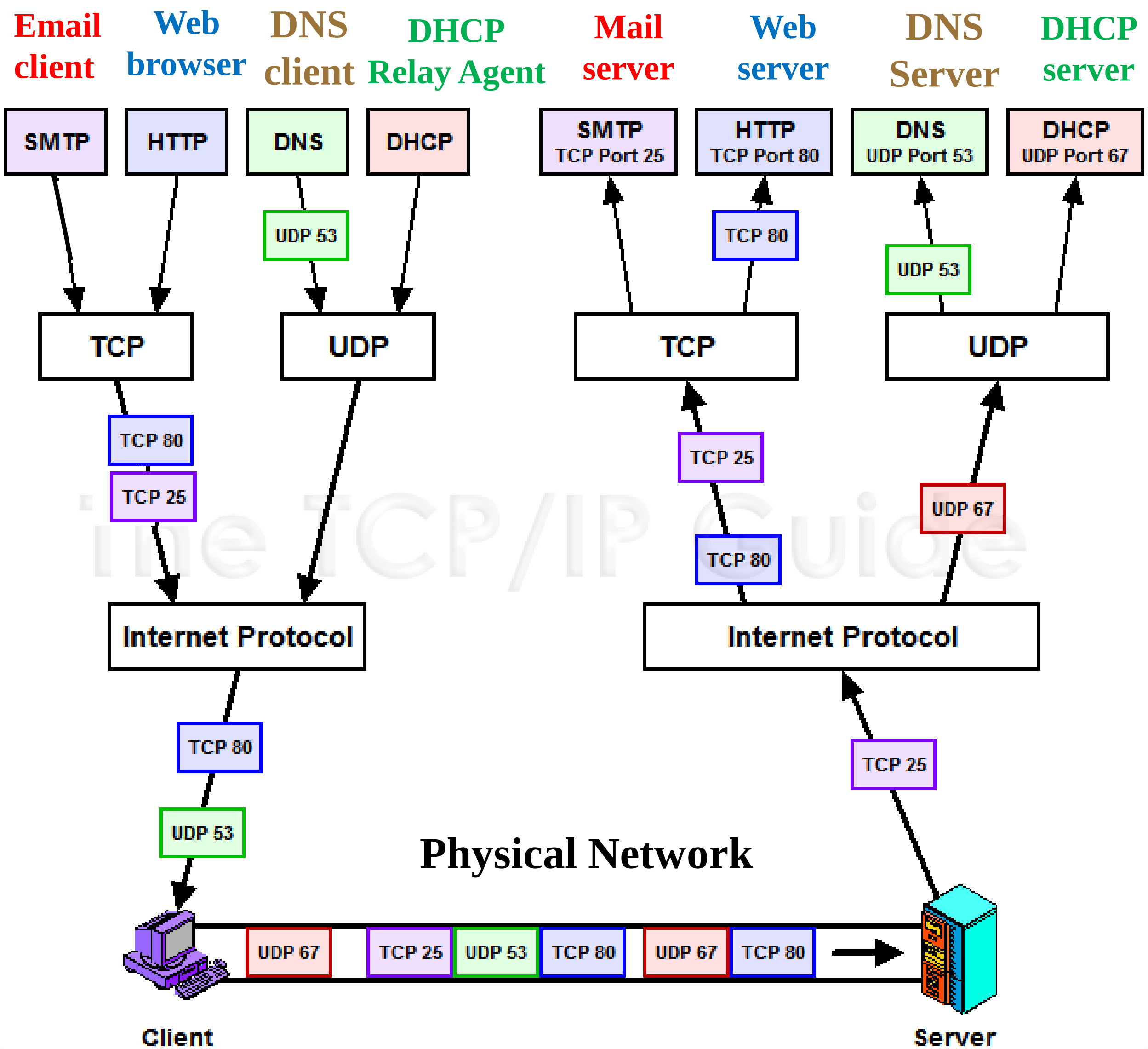
Ref: TCP/UDP well-known ports

Services mapped to Ports on a Host





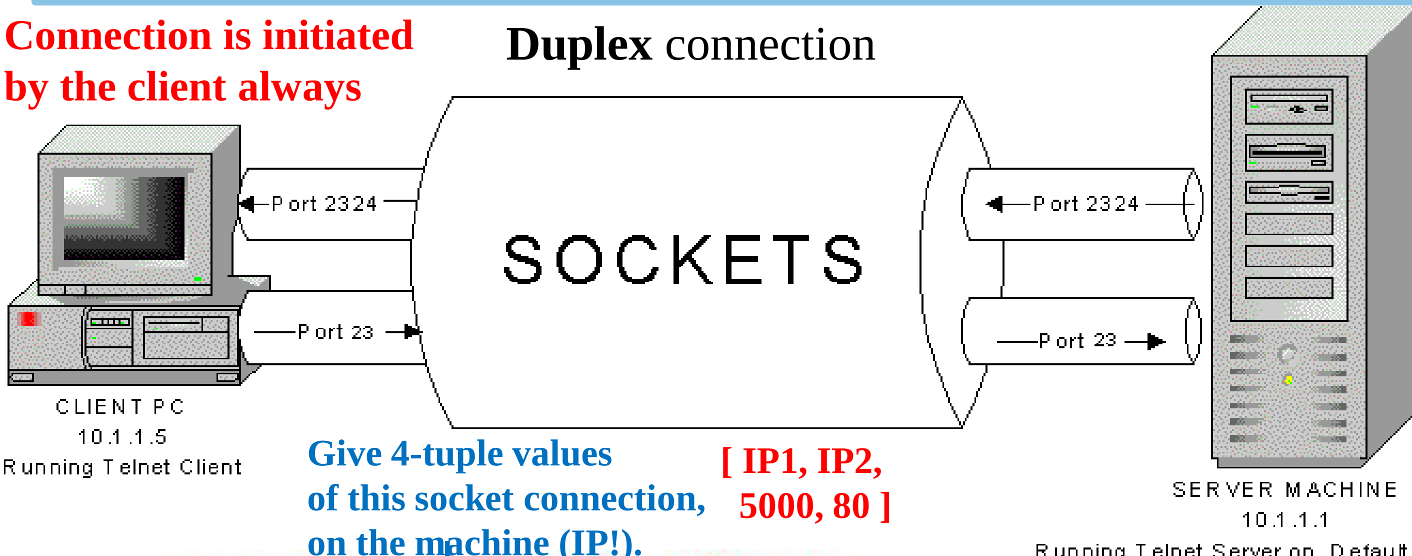
Flow of TCP and UDP Segments (Client-server model)



TCP/IP Sockets

Connection is initiated by the client always

Duplex connection

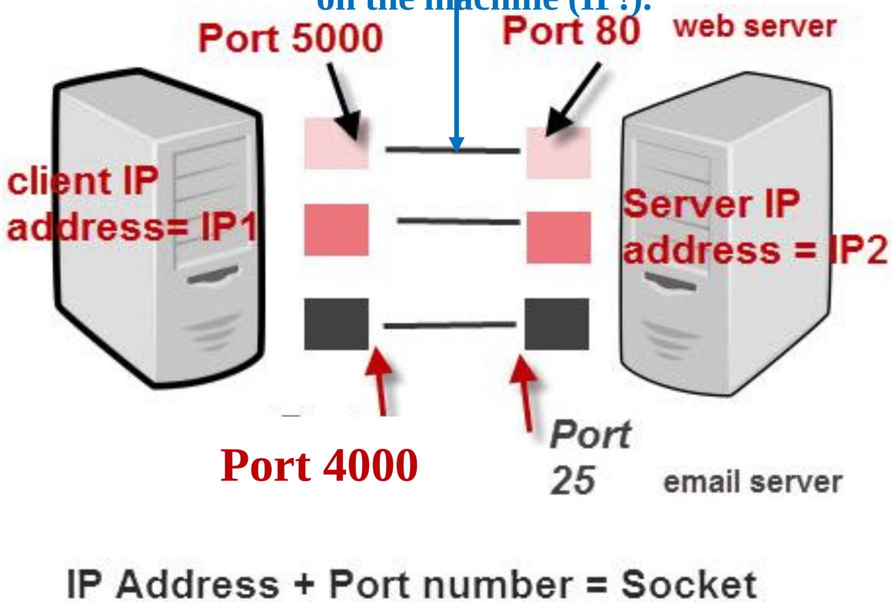


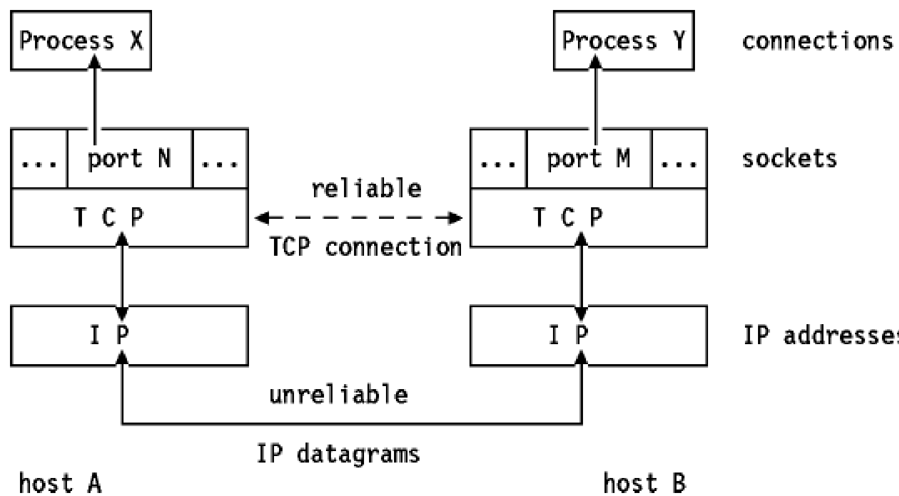
Give 4-tuple values of this socket connection, on the machine (IP!).

[IP1, IP2, 5000, 80]

Telnet: Remote login

Telnet is a protocol used on the Internet or LAN to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection

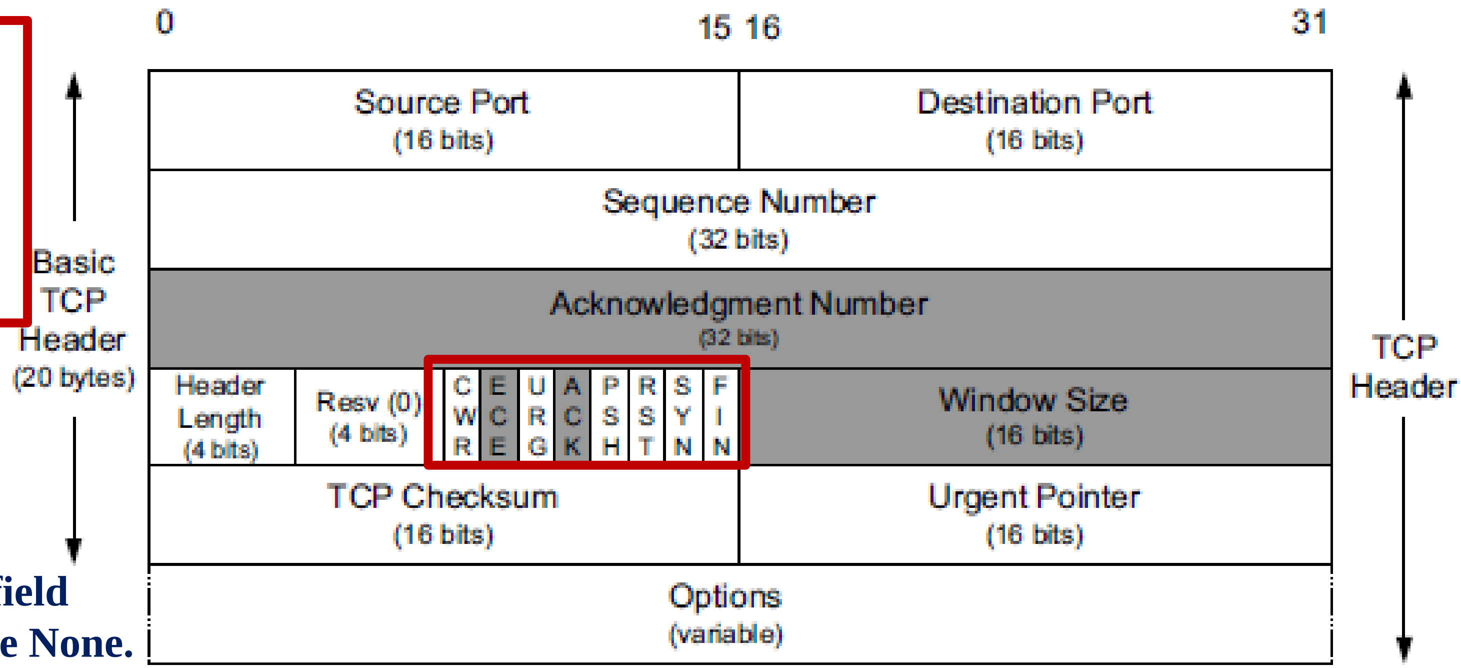




TCP Segment Structure

TCP Segment Structure (Header)

These are bit fields, which will be explained soon

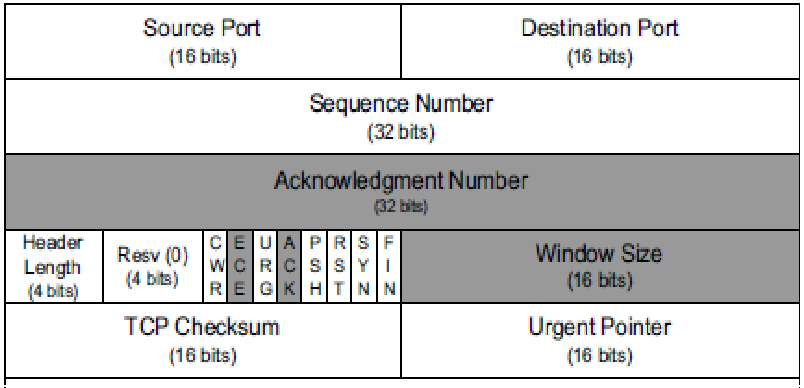


Note: Options field is assumed to be None.

- The **shaded fields** (Acknowledgment Number, Window Size, plus ECE and ACK bits) **refer to** the **data flowing** in the **opposite direction relative** to the **sender of this segment**.
- Remember, TCP is a duplex connection, between two end points. The segments from a sender carry some information about the data the sender has received from the other end, in the shaded fields. We will study shortly

TCP Header Fields: Sequence Number

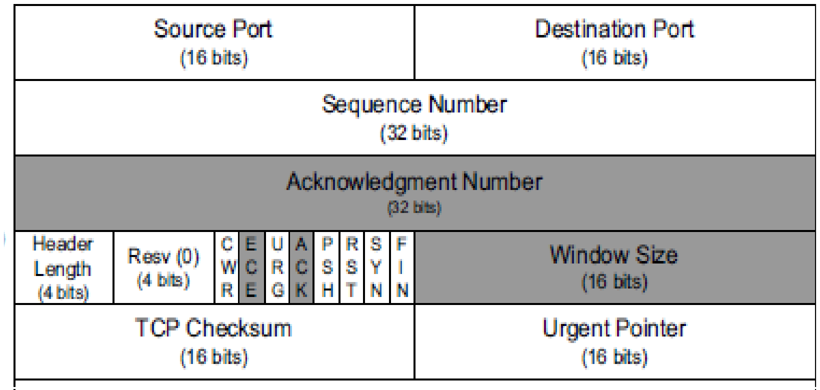
- Recall, TCP is used to exchange a byte-stream of data. The data could be a file, email, HTML page, etc.
- TCP needs to make sure that every byte sent by the sender reaches the other end reliably.



- The **Sequence Number** field identifies the **byte** in the **stream of data** from the **sending TCP** to the **receiving TCP.** It **refers** to the **first byte of data** being carried by the segment, of which this header is part of.
- If we consider the stream of bytes flowing in one direction between two applications, TCP numbers each byte with a sequence number.
- This **sequence number** is a **32-bit unsigned** number that wraps back around to 0 after reaching $(2^{32}) - 1$.
- When a host initiates a TCP session, the **Initial Sequence Number (ISN)** is always chosen at **random**. As the data gets exchanged, it wraps around.

TCP Header Fields: Acknowledgement Number

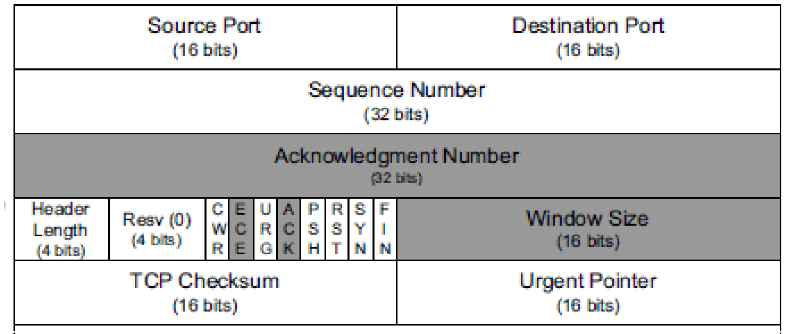
- As you are aware, the TCP connection is reliable, the receiver sends acknowledgment to the sender when the data is received correctly by it
- Since the connection is duplex, normally acknowledgment is sent along with the data flowing in the other direction



- Because **every byte** exchanged is **numbered**, the Acknowledgment Number field (also called the **ACK Number** or **ACK field** for short) contains the **next sequence number** that the **sender** of the **acknowledgment** **expects** to **receive**.
- This is therefore the sequence number of the last successfully received byte of data plus 1.
- This field is valid only if the ACK bit field (described later in this section) is ON (set to one), which it usually is ON for all but initial and closing segments.

TCP Header Fields: Acknowledgement Number

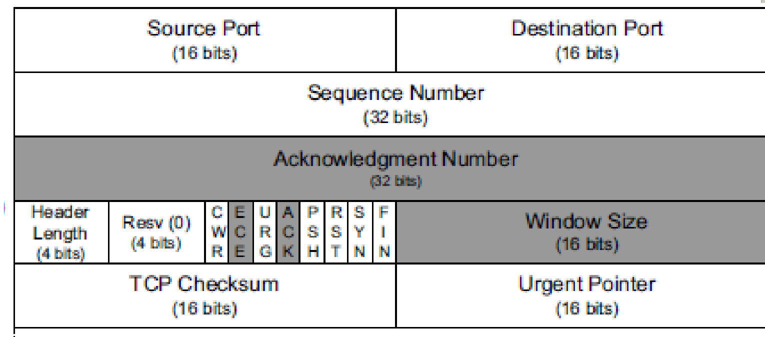
- Sending an ACK costs nothing more than sending any other TCP segment because the 32-bit ACK Number field is always part of the header, as is the ACK bit field.
- ACK and Sequence numbers represent the data flowing in two different directions



- As the **ISN** (Initial Sequence Numbers) are chosen by the senders **at random** while establishing the connection, the ACK and Sequence numbers will be totally different though they are part of a Segment header field
  - Because the first sequence number of the data flowing in the reverse direction would have been chosen by the receiver at random
- Suppose, a client machine receives a TCP segment, with a sequence number 100, and a data size of 50 bytes. **What would be the values of ACK no. and ACK bit when the client sends a TCP segment to the other end?**
- When this client machine has some data to be sent to other end of the connection, it would make the ACK number as **150** and set ACK bit, in the TCP segment that is carrying its own data to the machine on the other end

TCP Header Fields: TCP Checksum

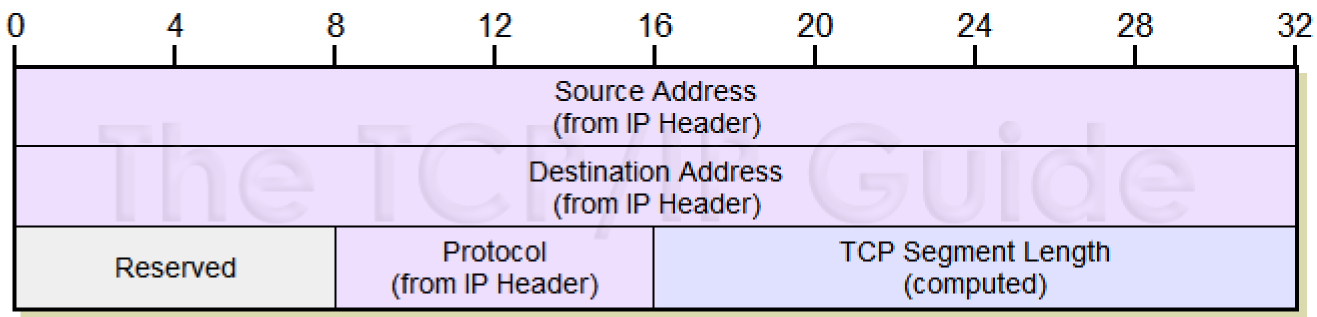
- The TCP/IP checksum is used to detect corruption of data over a TCP or IPv4 connection.
- If a bit is flipped, a byte mangled, or some other badness happens to a packet, then it is highly likely that the receiver of that broken packet will notice the problem due to a checksum mismatch.



- This provides end-to-end assurance that the data stream is correct.
- The TCP protocol includes an extra checksum that protects the packet "payload" as well as the header. This is in addition to the header-checksum of IP.
- The algorithm for the TCP and IPv4 checksums is identical. The data is processed a word (16 bits, two bytes) at a time.
- The TCP Checksum field covers the TCP header and data and some fields in the IP header, which is called a pseudo-header.

Pseudo-Header from IP Header Used for TCP Checksum

TCP “**Pseudo Header**” for TCP’s Checksum Calculation



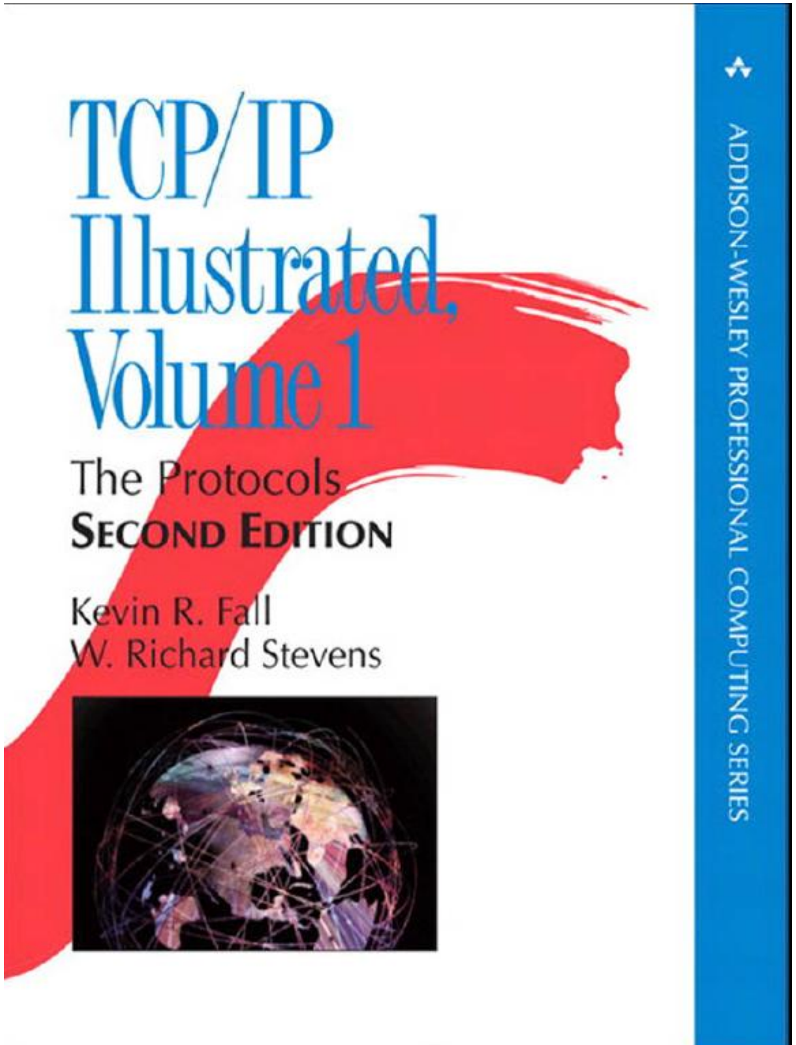
- Instead of computing the checksum over only the actual TCP header and data fields of the TCP segment, the above 12-byte from the IP header, on which this TCP segment is going to be part of, is also included in the checksum calculation
- Addition of “pseudo header” is done to make sure that at the receiving end, the host can be assured that the received TCP segment is indeed from the original sender that is addressed to it
  - This takes care of detecting “man-in the middle attack” if the TCP segment is tampered by someone enroute or corrupted IP packets being accepted by TCP

Session 2B: Summary

- TCP/UDP Ports
  - Port Allocations
  - TCP-UDP flow of segments
  - Well-Known Ports
- TCP Segment Header Structure
  - Sequence Number
  - ACK bit and Acknowledgement Number
  - TCP Checksum
    - Pseudo-Header

References

Ref 1



Ref 2

[TCP Congestion Control: A Systems Approach]

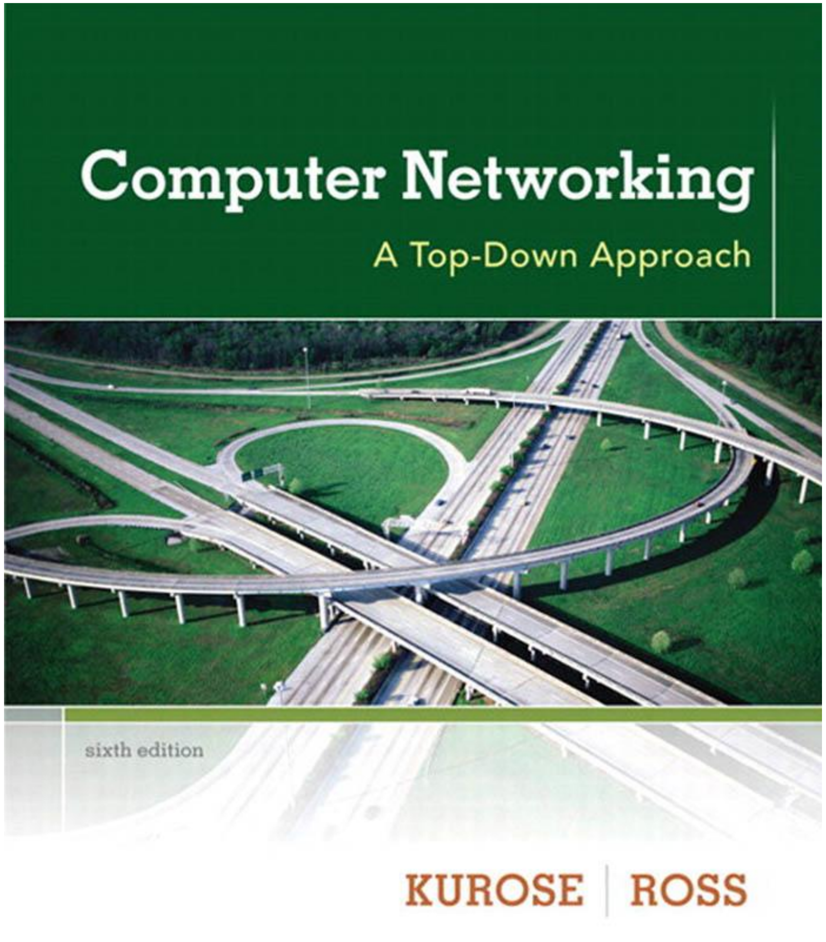
HTTPS://SYSTEMSAPPROACH.ORG

TCP Congestion Control: A Systems Approach

Peterson, Brakmo, and Davie

References

Ref 3



Ref 4