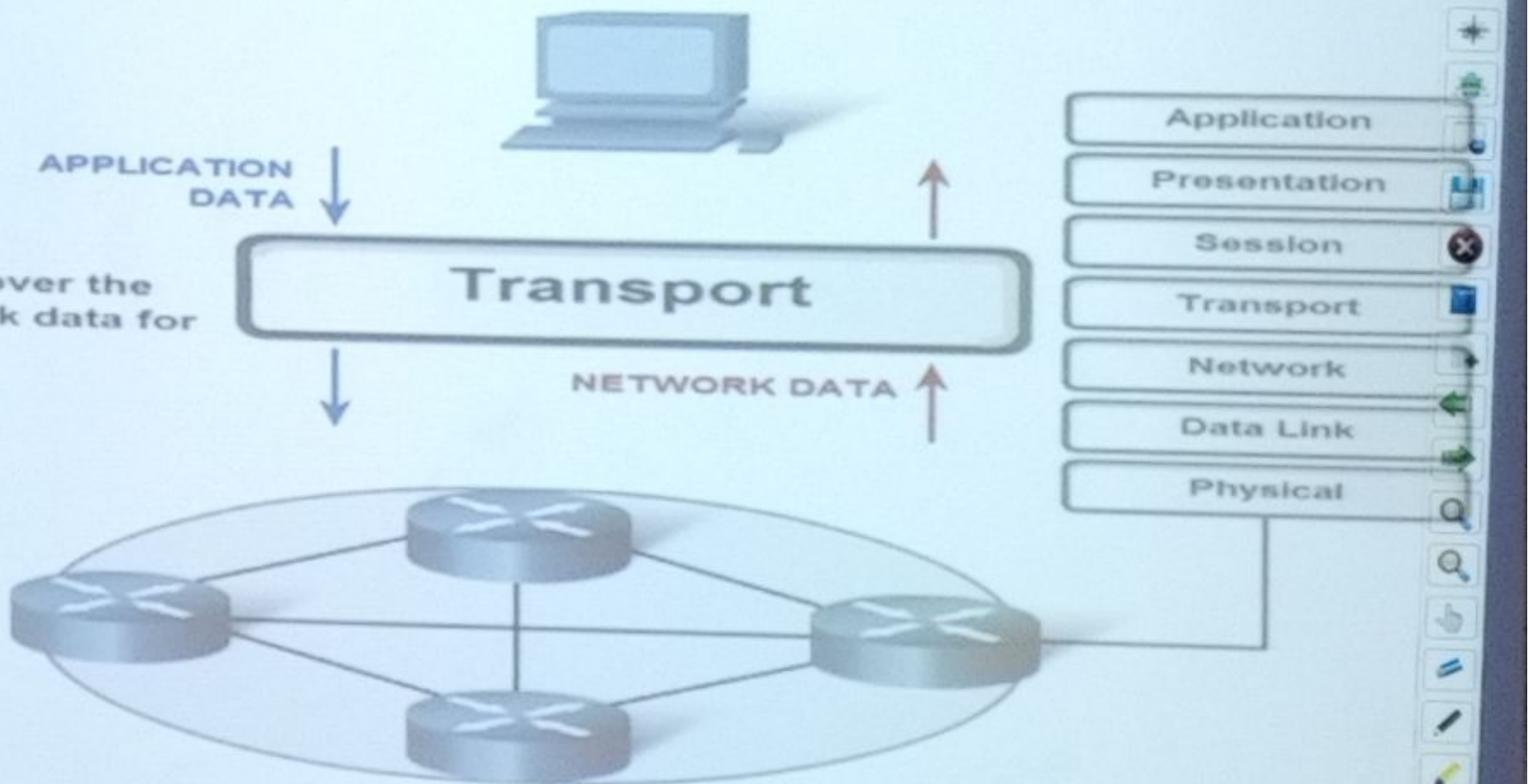
Transport Layer Protocols (End-to-End Communication)

Presentation By:
Dr. Rakesh Rathi
Asst. Professor
Department of Computer Science and Engineering
Government Engineering College Ajmer

The OSI Transport Layer

The Transport layer prepares application data for transport over the network and processes network data for use by applications.



Transport Layer Role and Services

Transport layer offers end-to-end connection between two processes on remote hosts.

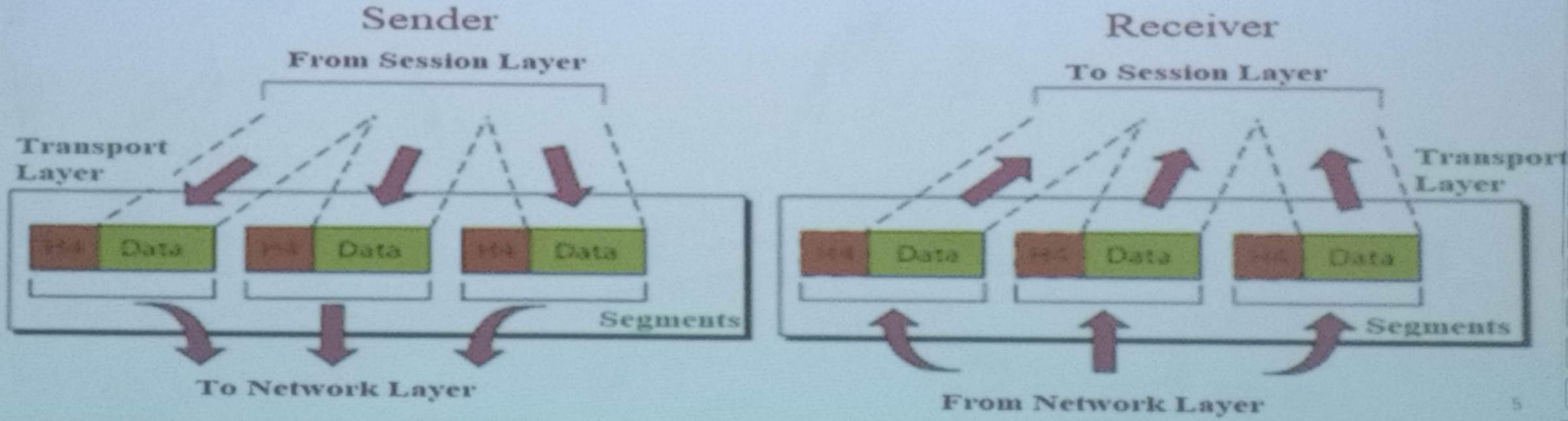
Transport layer takes data from upper layer (i.e. Application layer) and then breaks it into smaller size of segments, numbers each segment, and hands over to lower layer (Network Layer) for delivery.

- Service Point Addressing(Process-Process delivery)
- Segmentation and Reassembly
- Connection Control
- Flow Control(QoS) – MUX & Demux
- Error Control – Error Checking and Recovery
- Congestion Control

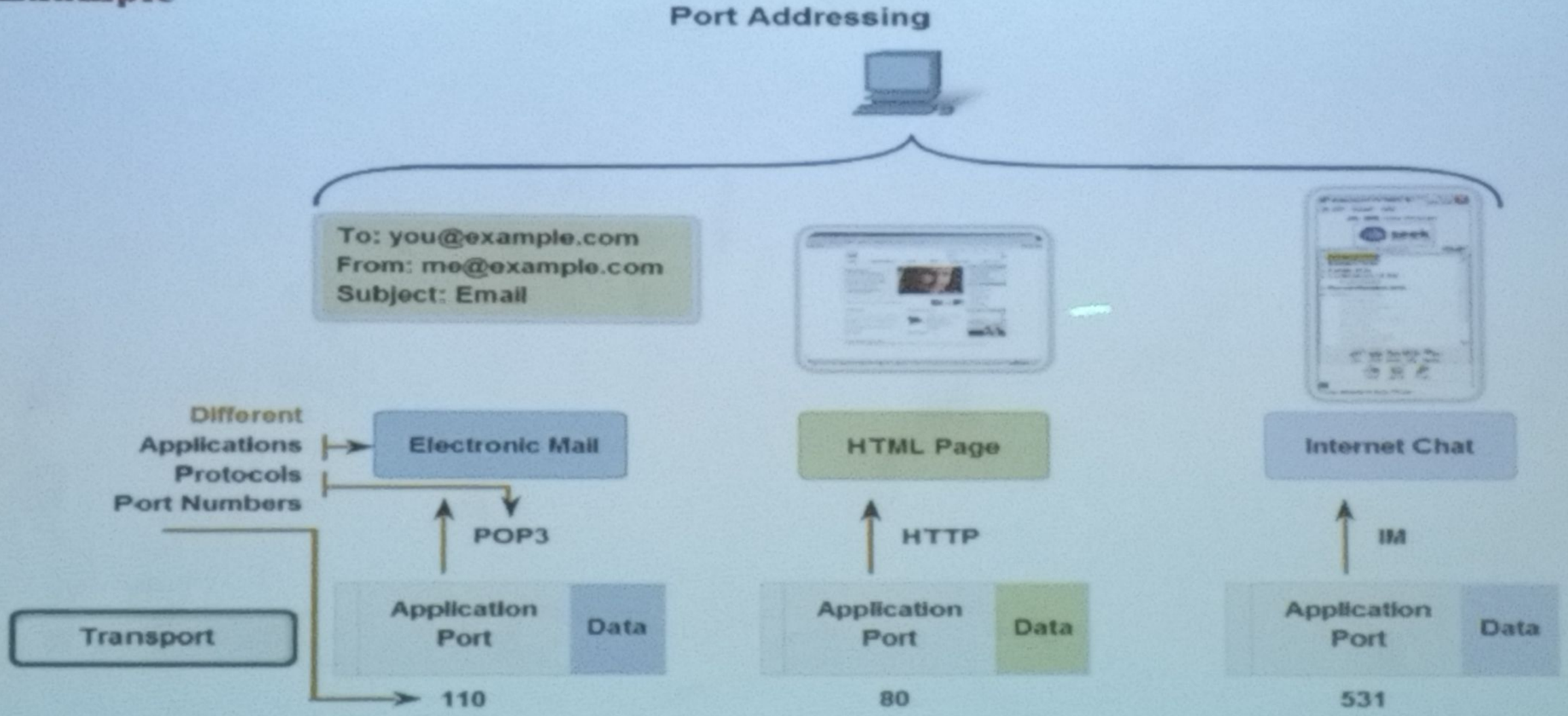
End-to-End Communication

A process on one host identifies its peer host on remote network by means of TSAPs (Transport Service Access Point) also known as Port numbers. TSAPs are very well defined and a process which is trying to communicate with its peer knows this in advance.

The transport layer is responsible for process-to-process delivery of the entire message.



Example

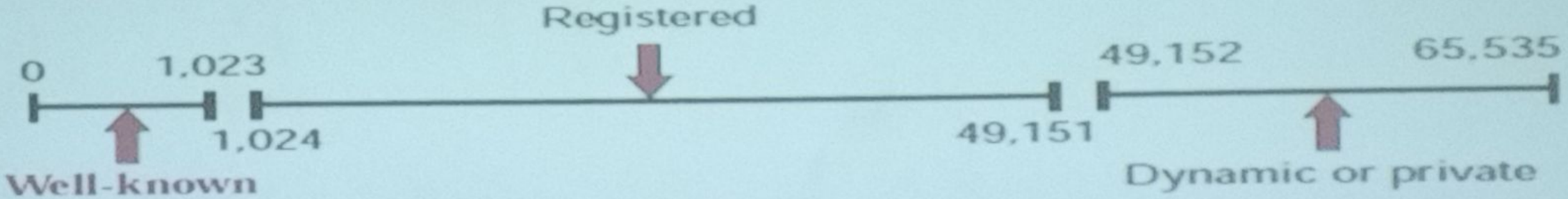


Data for different applications is directed to the correct application because each application has a unique port number.

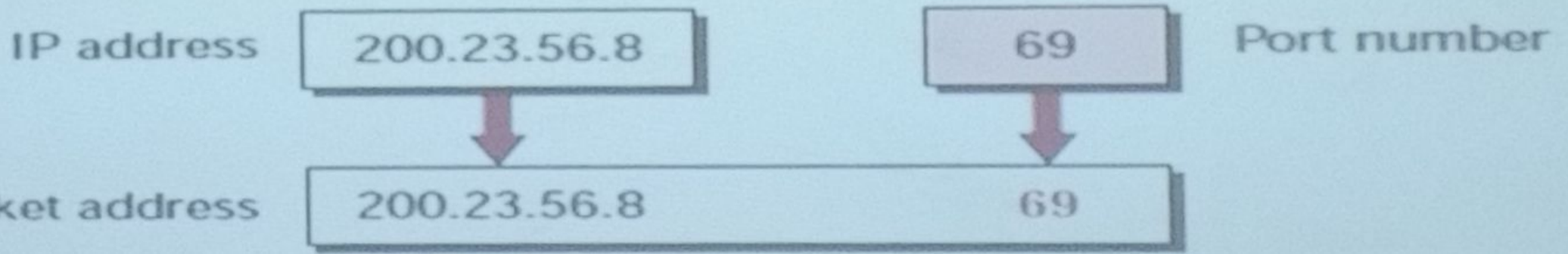
PORT NUMBER	TRANSPORT PROTOCOL	SERVICE NAME	RFC
20, 21	TCP	File Transfer Protocol (FTP)	RFC 959
22	TCP and UDP	Secure Shell (SSH)	RFC 4250-4256
23	TCP	Telnet	RFC 854
25	TCP	Simple Mail Transfer Protocol (SMTP)	RFC 5321
53	TCP and UDP	Domain Name Server (DNS)	RFC 1034-1035
67, 68	UDP	Dynamic Host Configuration Protocol (DHCP)	RFC 2131
69	UDP	Trivial File Transfer Protocol (TFTP)	RFC 1350
80	TCP	HyperText Transfer Protocol (HTTP)	RFC 2616
110	TCP	Post Office Protocol (POP3)	RFC 1939
119	TCP	Network News Transport Protocol (NNTP)	RFC 8977
123	UDP	Network Time Protocol (NTP)	RFC 5905
135-139	TCP and UDP	NetBIOS	RFC 1001-1002
143	TCP and UDP	Internet Message Access Protocol (IMAP4)	RFC 3501
161, 162	TCP and UDP	Simple Network Management Protocol (SNMP)	RFC 1901-1908, 3411-
179	TCP	Border Gateway Protocol (BGP)	RFC 4271
389	TCP and UDP	Lightweight Directory Access Protocol	RFC 4510
443	TCP and UDP	HTTP with Secure Sockets Layer (SSL)	RFC 2818
500	UDP	Internet Security Association and Key Management Protocol (ISAKMP) / Internet Key Exchange (IKE)	RFC 2408 - 2409
636	TCP and UDP	Lightweight Directory Access Protocol over TLS/SSL (LDAPS	RFC 4513
989/990	TCP	FTP over TLS/SSL	RFC 4217

Addressing: TCP communication between two remote hosts is done by means of p
numbers which is Transport Service Access Points (TSAPs). Ports numbers can ran
from 0 – 65535 (2^{16})which are divided as:

- System Ports (0 – 1023)
- User Ports (1024 – 49151)
- Private/Dynamic Ports (49152 – 65535)



Socket address: Combination of an IP address and a Port Number.



The Two Main Transport Layer Protocols are:

1. **Transmission Control Protocol (TCP)**
   It provides reliable communication between two hosts.
2. **User Datagram Protocol (UDP)**
   It provides unreliable communication between two hosts.

Applications Requirements Vary

- Because different applications have different requirements, there are multiple Transport layer protocols.

Transport Layer Role and Services

Supporting Reliable Communication

Transport Layer Protocols

- IP Telephony
- Streaming Video

Required Protocol Properties

- Fast
- Low overhead
- Does not require acknowledgements
- Does not resend lost data
- Delivers data as it arrives

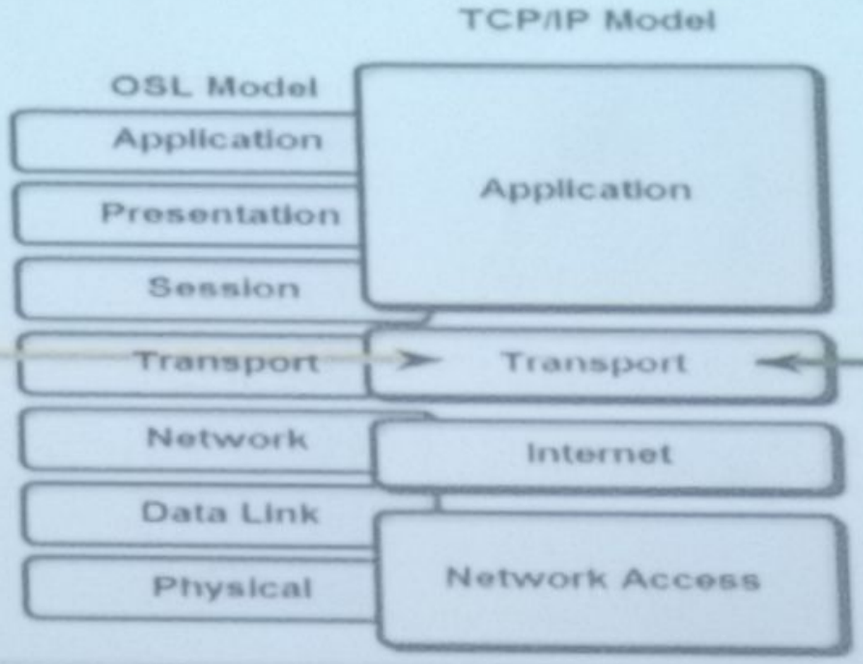
UDP

- SMTP/POP (Email)
- HTTP

Required Protocol Properties

- Reliable
- Acknowledge data
- Resend lost data
- Delivers data in order sent

T



Application developers choose the appropriate Transport Layer protocol based on the nature of the application.

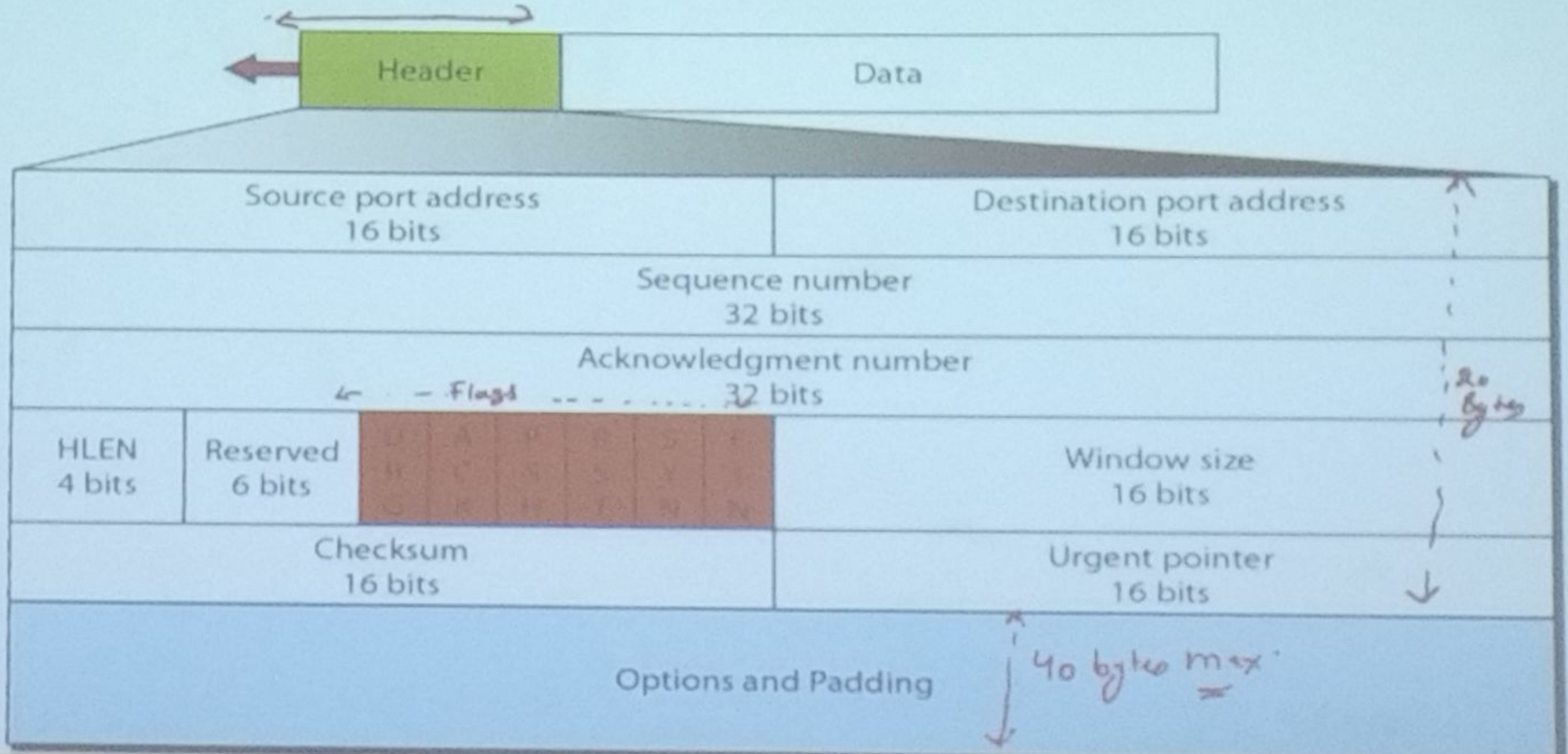
Transmission Control Protocol TCP

Features

- TCP is reliable protocol. That is, the receiver always sends either positive or negative acknowledgement about the data packet to the sender, so that the sender always has bright clue about whether the data packet is reached the destination or it needs to resend it.
- TCP ensures that the data reaches intended destination in the same order it was sent.
- TCP is connection oriented. TCP requires that connection between two remote points be established before sending actual data.
- TCP provides error-checking and recovery mechanism.
- TCP provides end-to-end communication.
- TCP provides flow control and quality of service.
- TCP provides full duplex server, i.e. it can perform roles of both receiver and sender.

Figure : *TCP segment format*

The length of TCP header is minimum 20 bytes and maximum 60 bytes.



Source Port (16-bits): It identifies source port of the application process on the sending device
Destination Port (16-bits): It identifies destination port of the application process on th
receiving device.
Sequence Number (32-bits): Sequence number of data bytes of a segment in a session.
Acknowledgement Number (32-bits): When ACK flag is set, this number contains the n
sequence number of the data byte expected and works as acknowledgement of the previous
received.
Header Length (4-bits): This field shows the size of TCP header including options and padd
(20-60 bytes). Value of header length vary between 0101 to 1111. It is work on scale of 4.
Reserved (6-bits): Reserved for future use and all are set zero by default.
Flags (1-bit each):

- **URG:** It indicates that Urgent Pointer field has significant data and should be processed.
- **ACK:** It indicates that Acknowledgement field has significance. If ACK is cleared to
  indicates that packet does not contain any acknowledgement.
- **PSH:** When set, it is a request to the receiving station to PUSH data as soon as it come
  the receiving application without buffering it.

- **RST:** Reset flag has the following features:
  - ✓ It is used to refuse an incoming connection.
  - ✓ It is used to reject a segment.
  - ✓ It is used to restart a connection.
- **SYN:** This flag is used to set up a connection between hosts.
- **FIN:** This flag is used to release a connection and no more data is exchanged thereafter. Because packets with SYN and FIN flags have sequence numbers, they are processed in correct order.

Windows Size: This field is used for flow control between two stations and indicates amount of buffer (in bytes) the receiver has allocated for a segment, i.e. how much data is receiver expecting.

Checksum: This field contains the checksum of Header, Data, and Pseudo Headers.

Urgent Pointer: It points to the urgent data byte if URG flag is set to 1.

Options: It facilitates additional options which are not covered by the regular header. Opt field is always described in 32-bit words. If this field contains data less than 32-bit, paddin used to cover the remaining bits to reach 32-bit boundary. These options include Maxi segment size (MSS), Window scale, Selective acknowledgment (SACK), Timestamp measure the round-trip time (RTT) of every packet that is acknowledged).