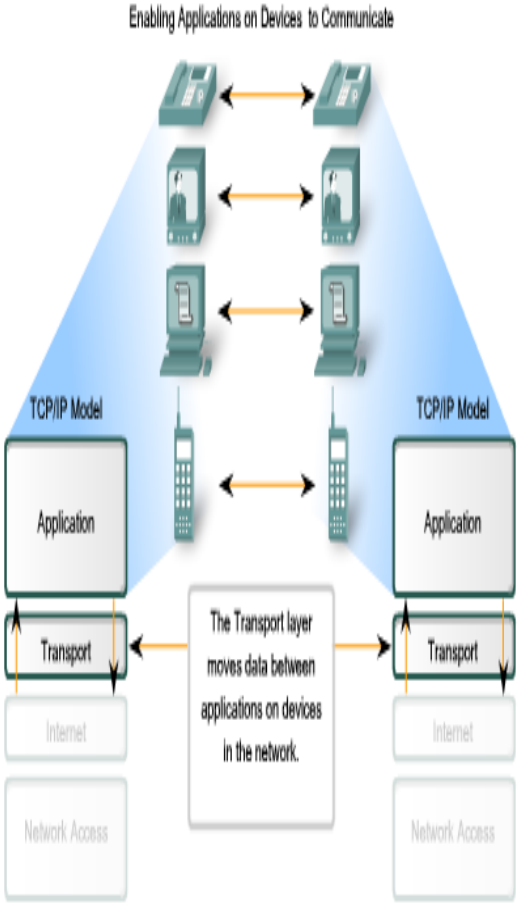
Transport Layer – TCP & UDP
Week 7

Module : Computer Networks
Lecturers : Lucy White lbwhite@wit.ie
Office : 324

Purpose of the Transport Layer

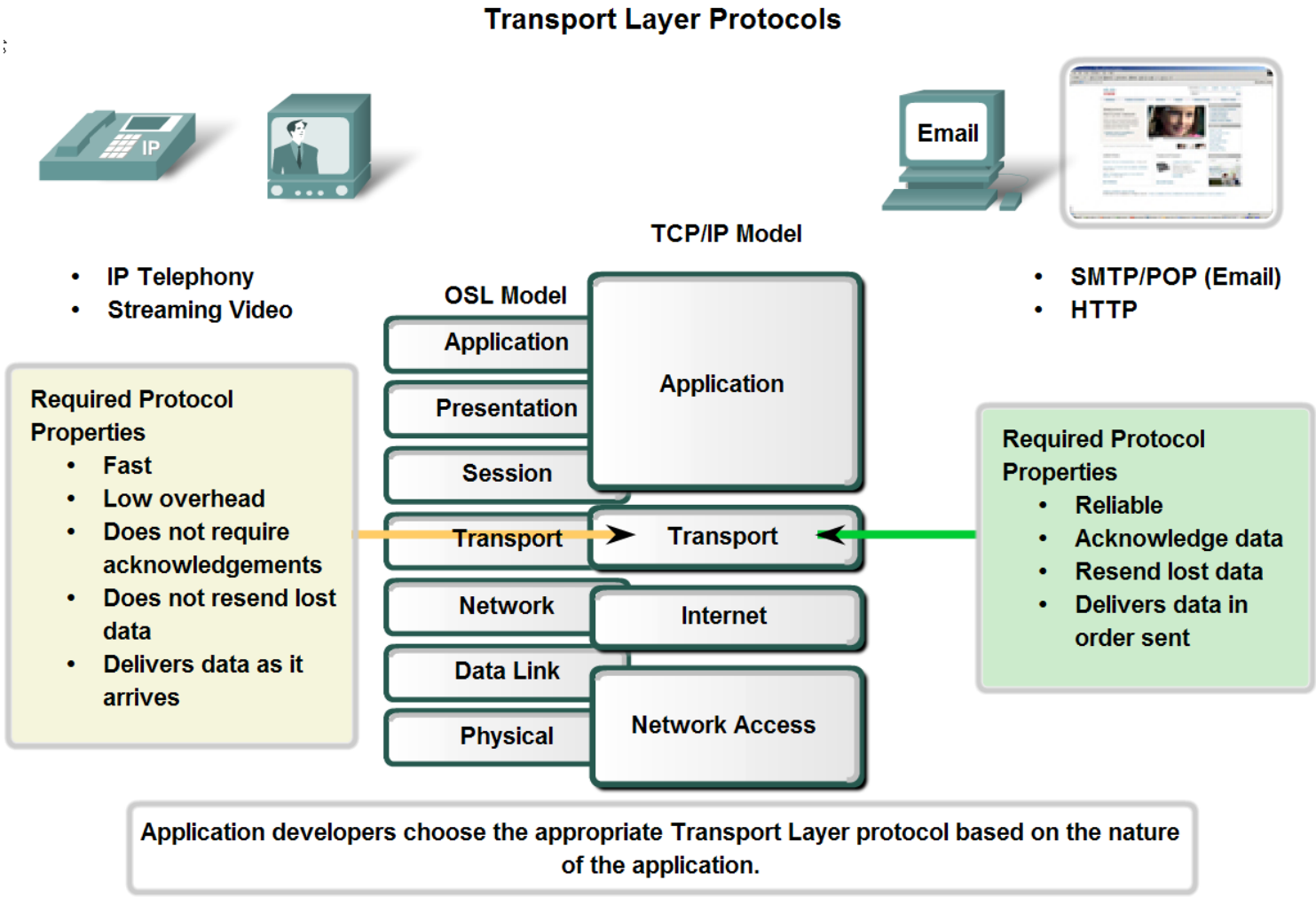
- The Transport layer provides for the segmentation of data necessary to reassemble these pieces into the various communication streams.
- Its primary responsibilities to accomplish this are:
  - Tracking the individual communication between applications on the source and destination hosts
  - Segmenting data and managing each piece
  - Reassembling the segments into application data
  - Identifying the different applications



In order to pass data streams to the proper applications, the Transport layer must identify the target application. Transport layer assigns an application an identifier called a port number. Each software process that needs to access the network is assigned a port number unique in that host.

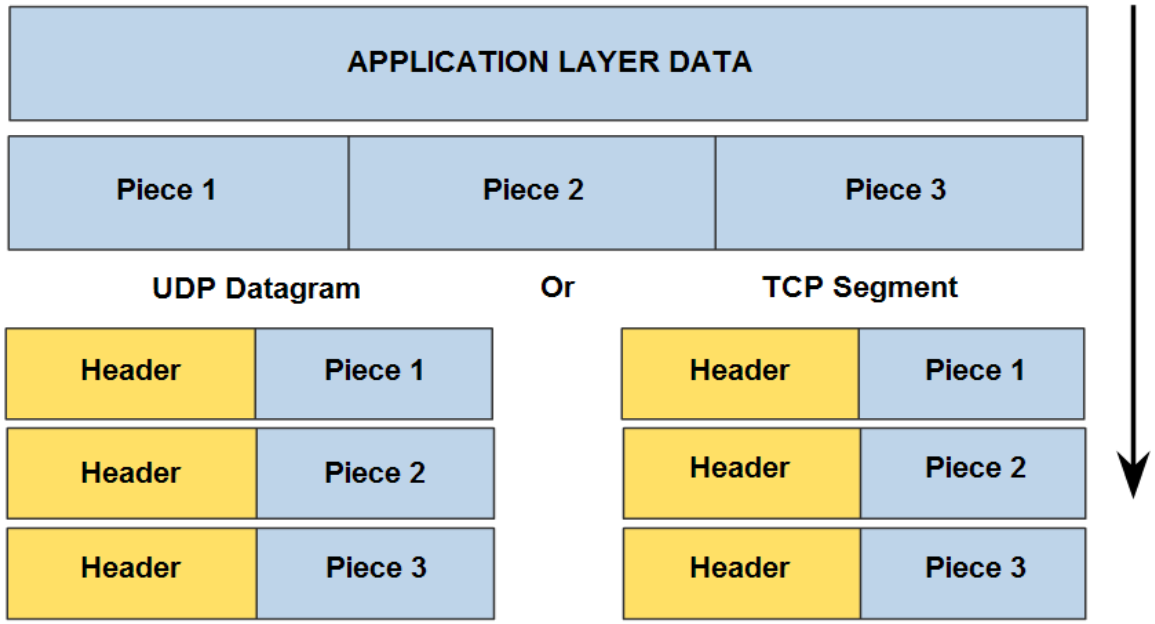
Transport Layer Role and Services

- Supporting Reliable Communication



Transport Layer Functions

The Transport layer divides the data into pieces and adds a header for delivery over the network.



UDP Header provides for:

- Source and destination (ports)

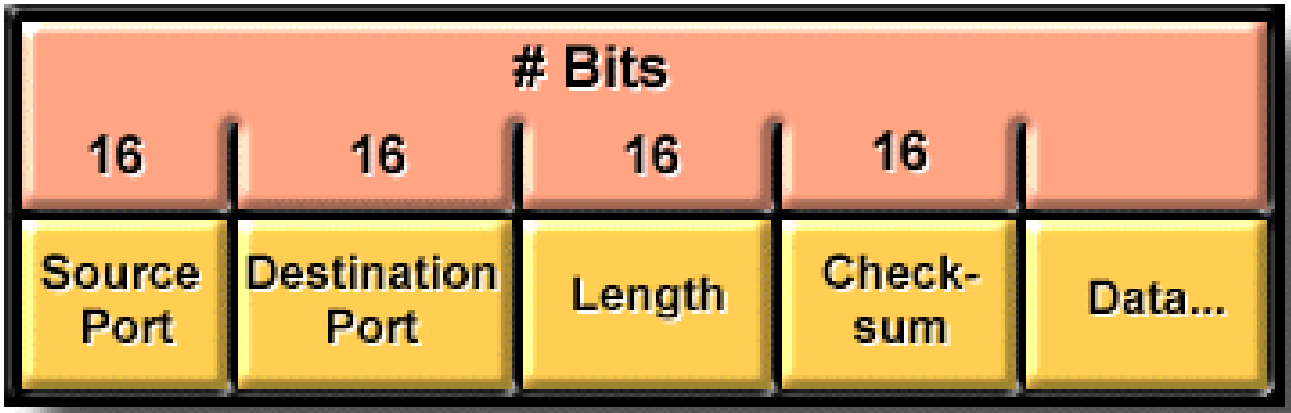
TCP Header provides for:

- Source & destination (ports)
- Sequencing for same order delivery
- Acknowledgement of received segments
- Flow control and congestion management

Transport Layer

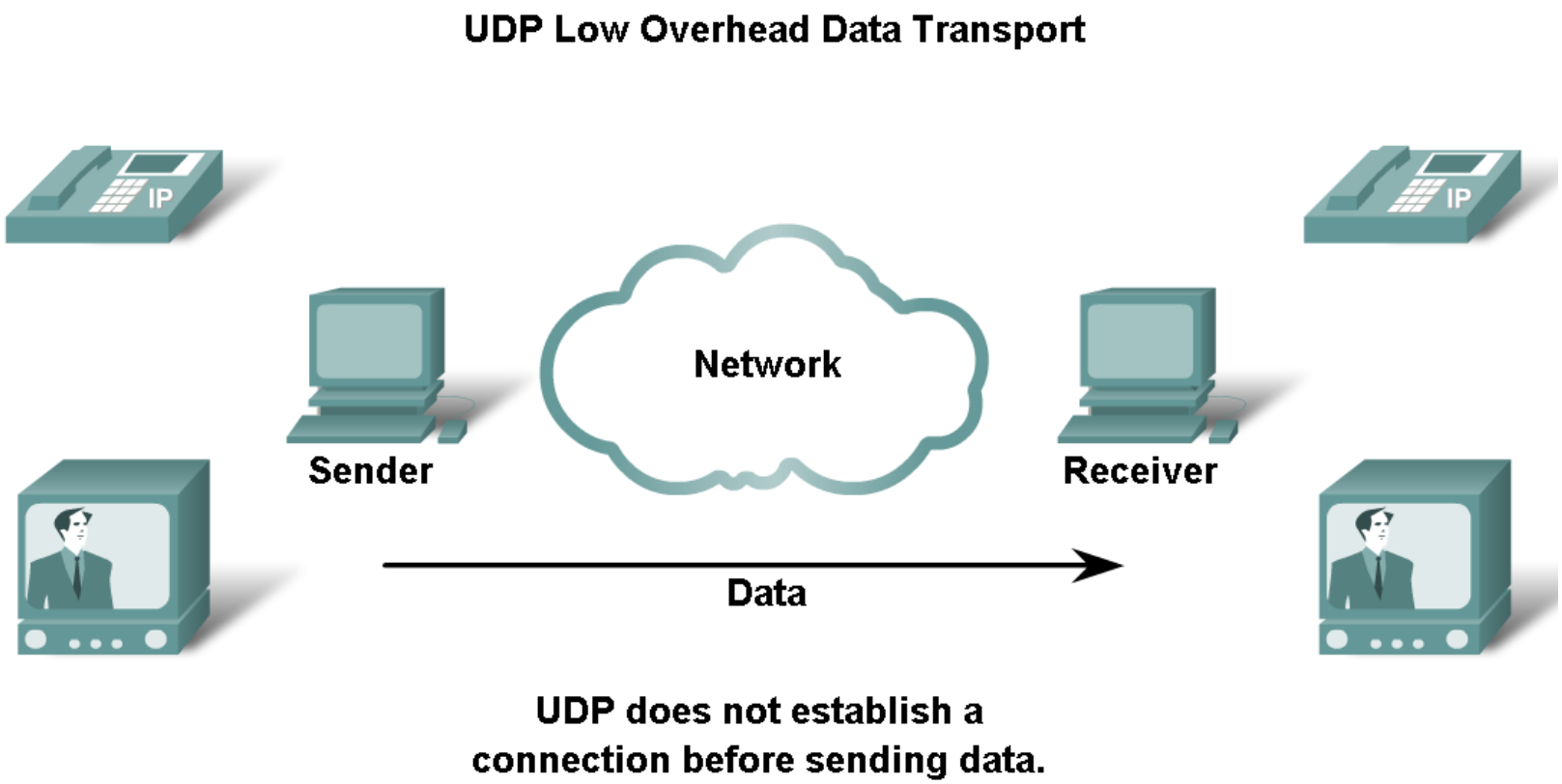
- *TCP* -- a connection-oriented, reliable protocol; provides flow control by providing sliding windows, and reliability by providing sequence numbers and acknowledgments. TCP re-sends anything that is not received and supplies a virtual circuit between end-user applications. The advantage of TCP is that it provides guaranteed delivery of the segments.
- *UDP* -- connectionless and unreliable; although responsible for transmitting messages, no software checking for segment delivery is provided at this layer. The advantage that UDP provides is speed. Since UDP provides no acknowledgments, less traffic is sent across the network, making the transfer faster.

The UDP Segment Format



- No sequence or acknowledgement fields

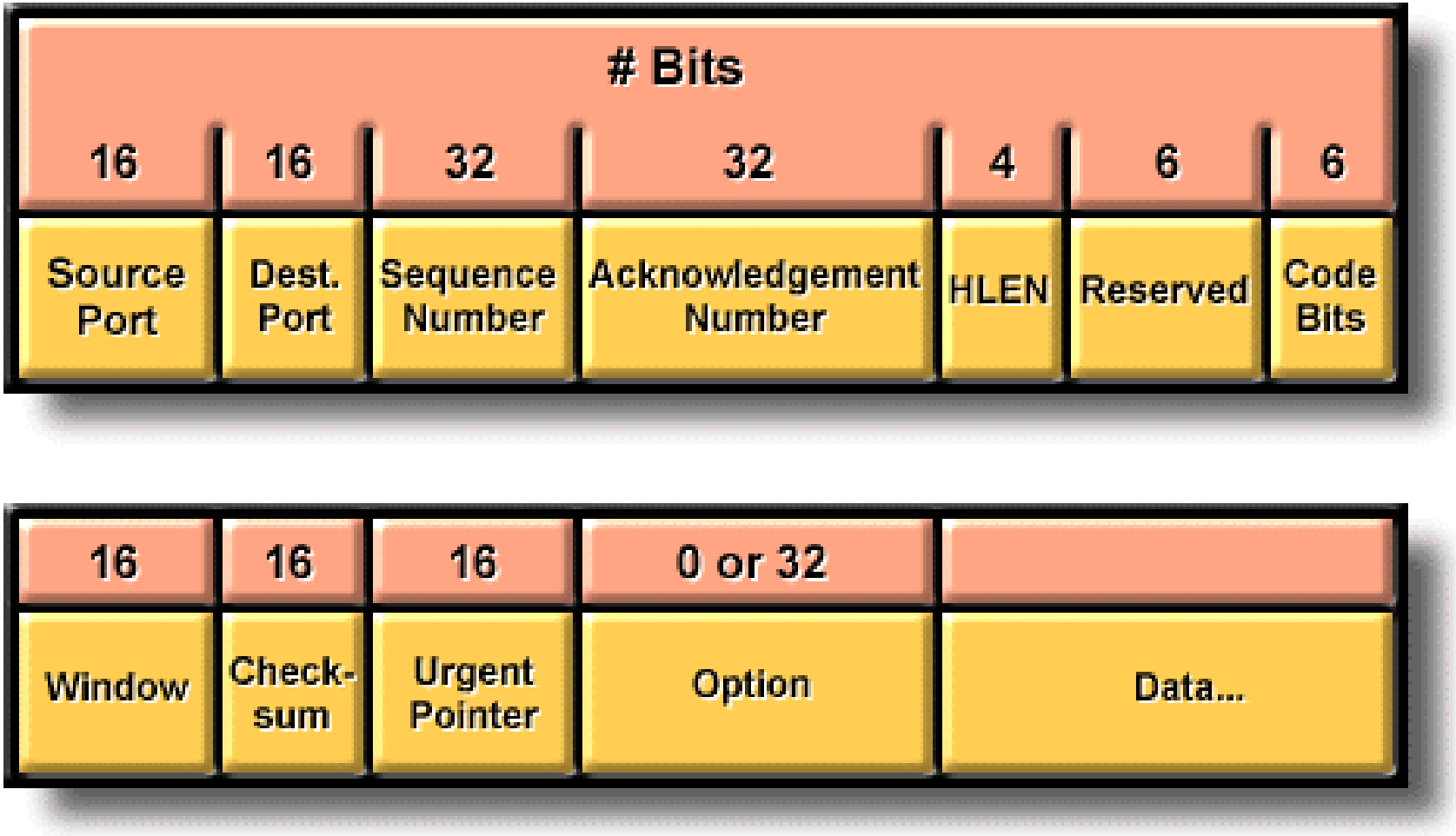
UDP Protocol



UDP – Low Overhead vs. Reliability

- UDP is a simple protocol that provides the basic Transport layer functions.
  - –It is not connection-oriented
  - –It does not provide retransmission, sequencing, and flow control.
- This does not mean that applications that use UDP are always unreliable.
  - –It simply means that these functions are not provided by the Transport layer protocol and must be implemented elsewhere if required.
- key Application layer protocols that use UDP include:
  - –Domain Name System (DNS)
  - –Simple Network Management Protocol (SNMP)
  - –Dynamic Host Configuration Protocol (DHCP)
  - –Routing Information Protocol (RIP)
  - –Trivial File Transfer Protocol (TFTP)
  - –Online games
- If these applications used TCP, they may experience large delays while TCP detects data loss and retransmits data.
  - –These delays would be more detrimental to the application than small data losses.

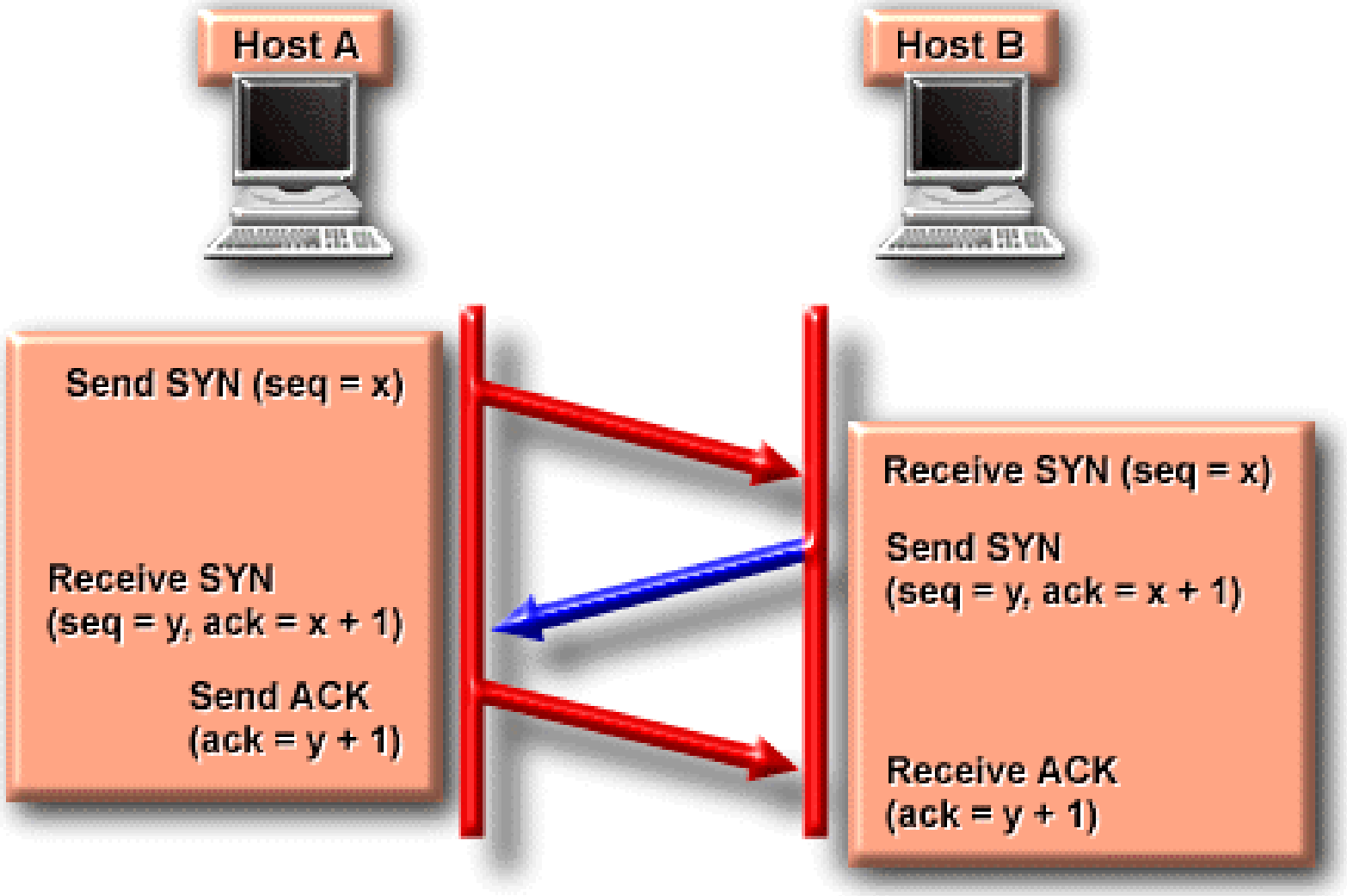
The TCP Segment Format



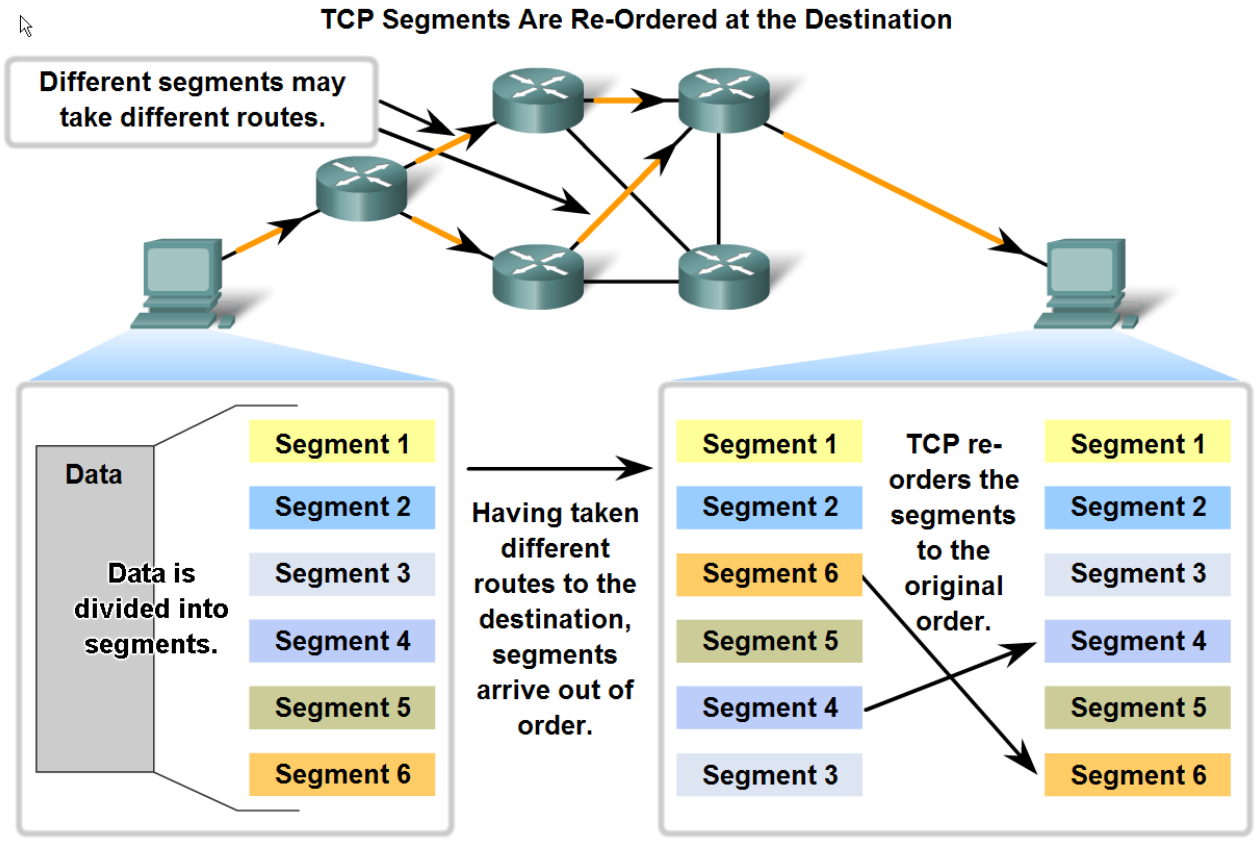
Transmission Control Protocol (TCP)

- Flow Control provided by sliding window
- Reliability provided by Sequence numbers and Acknowledgement numbers
- 3-way Handshake used to establish TCP connection and initialise fields

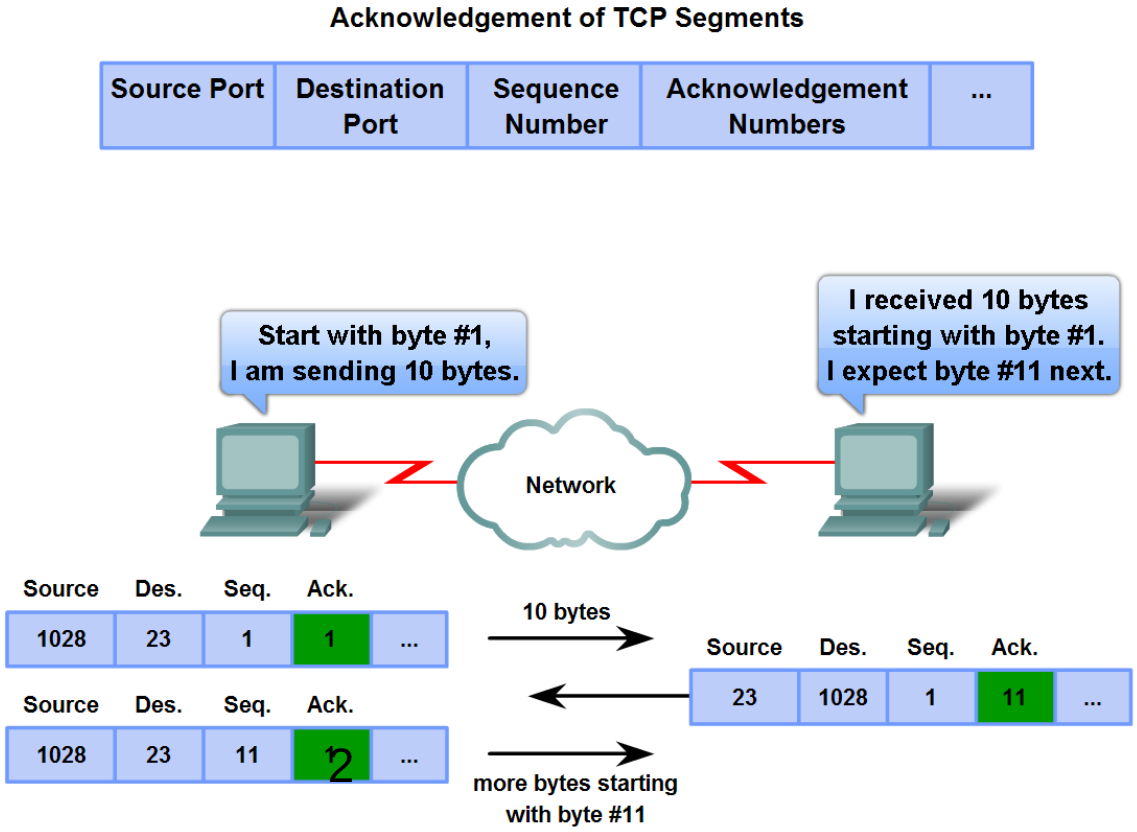
The TCP Three-Way Handshake/Open Connection



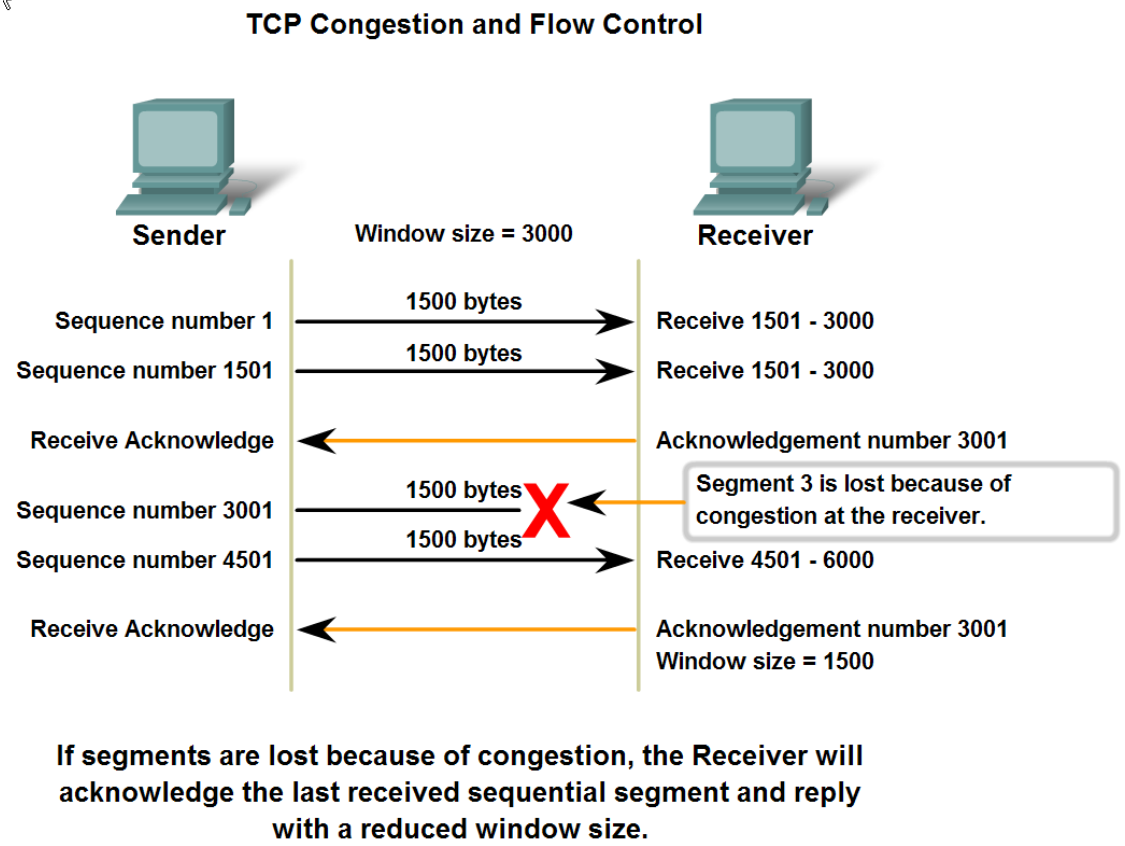
- TCP sequence numbers are used to reconstruct the data stream with segments placed in the correct order



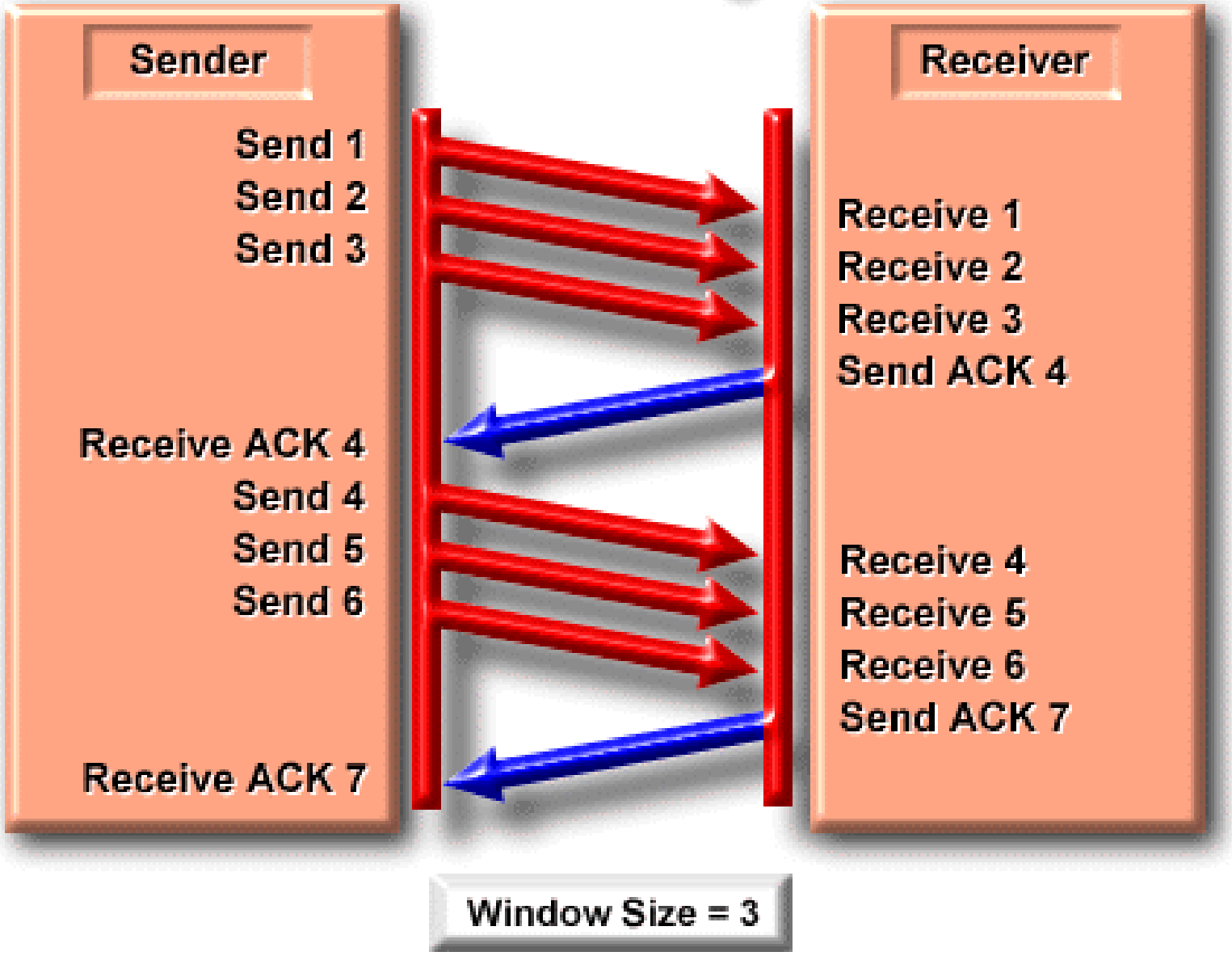
- Steps used by the TCP protocol in which sequence numbers and acknowledgement numbers are used to manage exchanges in a conversation



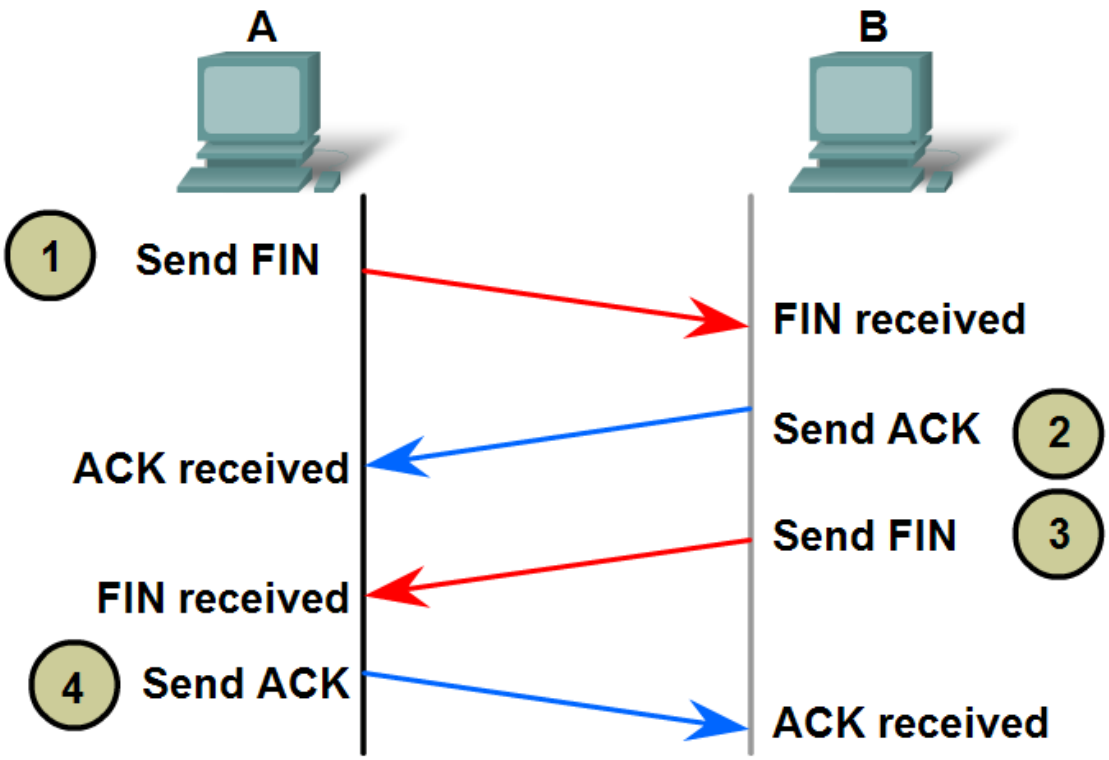
- Mechanisms in TCP that manage the interrelationship between window size, data loss and congestion during a session



The TCP Sliding Window



TCP Connection Establishment and Termination



Port Numbers

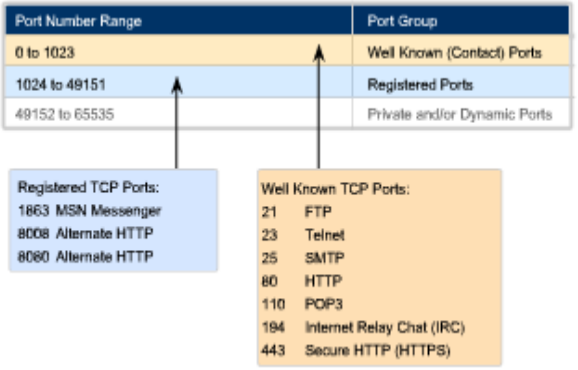
- TCP and UDP are both transport protocols above the IP layer, which are interfaces between IP and upper-layer processes. TCP and UDP protocol port numbers are designed to distinguish multiple applications running on a single device from one another.
- Since many network applications may be running on the same machine, computers need something to make sure the correct software application on the destination computer gets the data packets from the source machine, and some way to make sure replies get routed to the correct application on the source computer. This is accomplished through the use of the TCP or UDP "port numbers". In the TCP and UDP header, there are "Source Port" and "Destination Port" fields which are used to indicate the message sending process and receiving process identities defined.

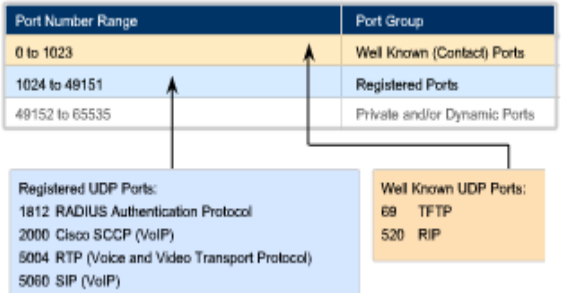
Port Numbers

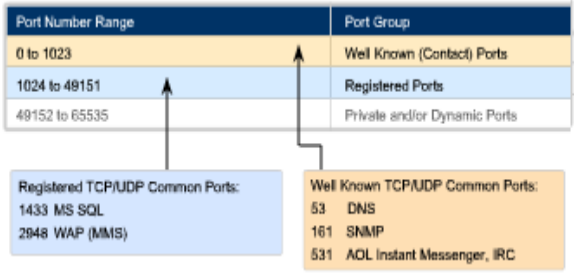
- Application software developers have agreed to use the well-known port numbers that are defined in RFC 1700.
- Port numbers have the following assigned ranges:
- Numbers below 1023 are for public applications – well known port numbers.
- Numbers above 1023 are unregulated.

The IANA assigns port numbers

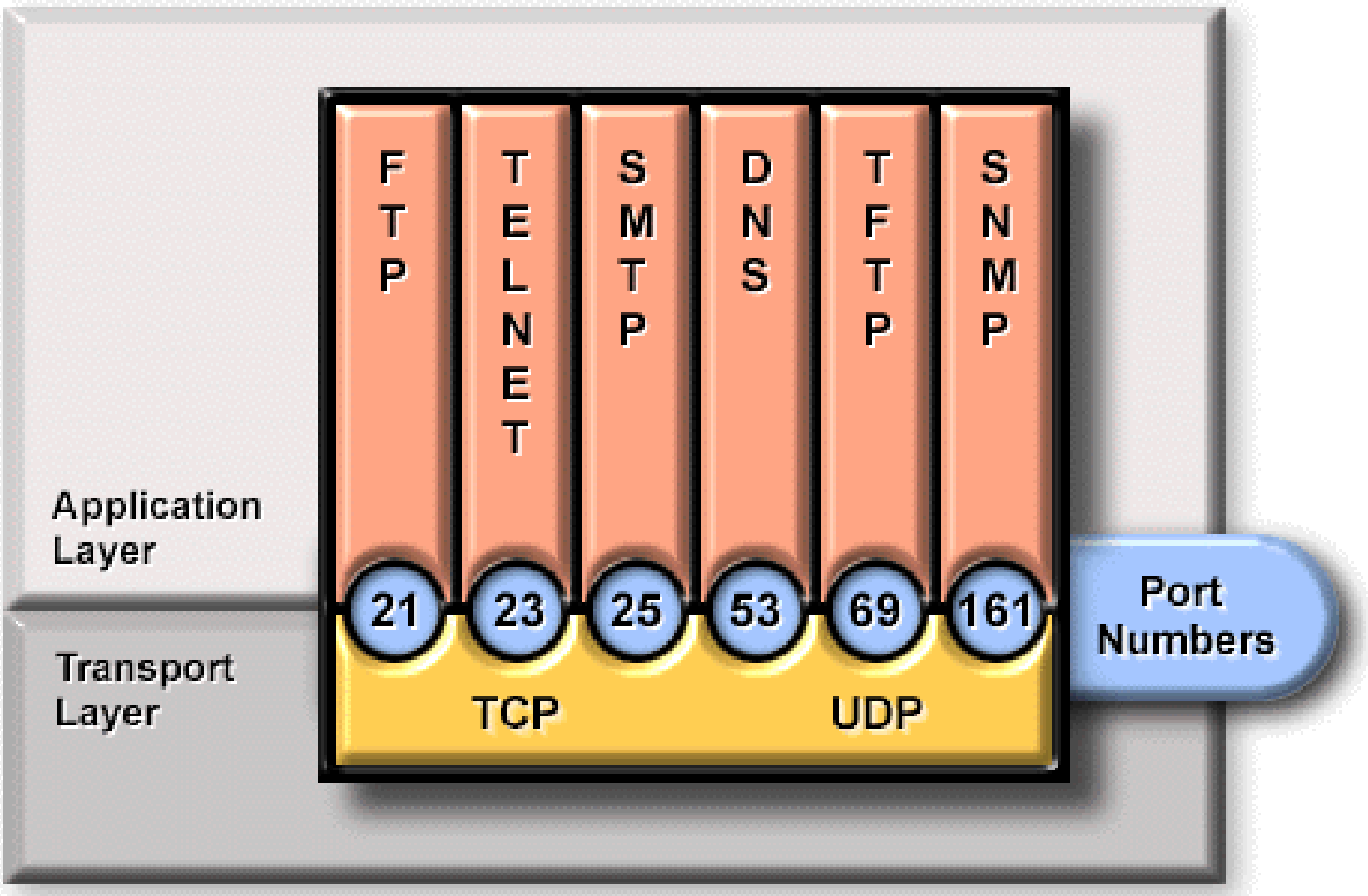
- Well Known Ports (Numbers 0 to 1023) - These numbers are reserved for services and applications.
  - –HTTP (web server) POP3/SMTP (e-mail server) and Telnet.
- Registered Ports (Numbers 1024 to 49151) - These port numbers are assigned to user processes or applications.
  - –These processes are primarily individual applications that a user has chosen to install.
  - –When not used for a server resource, these ports may also be used dynamically selected by a client as its source port.
- Dynamic or Private Ports (Numbers 49152 to 65535) - Also known as Ephemeral Ports, these are usually assigned dynamically to client applications when initiating a connection.
  - –It is not very common for a client to connect to a service using a Dynamic or Private Port.
- Using both TCP and UDP
  - –Some applications may use both TCP and UDP.
    - For example, the low overhead of UDP enables DNS to serve many client requests very quickly.
    - Sometimes, however, sending the requested information may require the reliability of TCP.



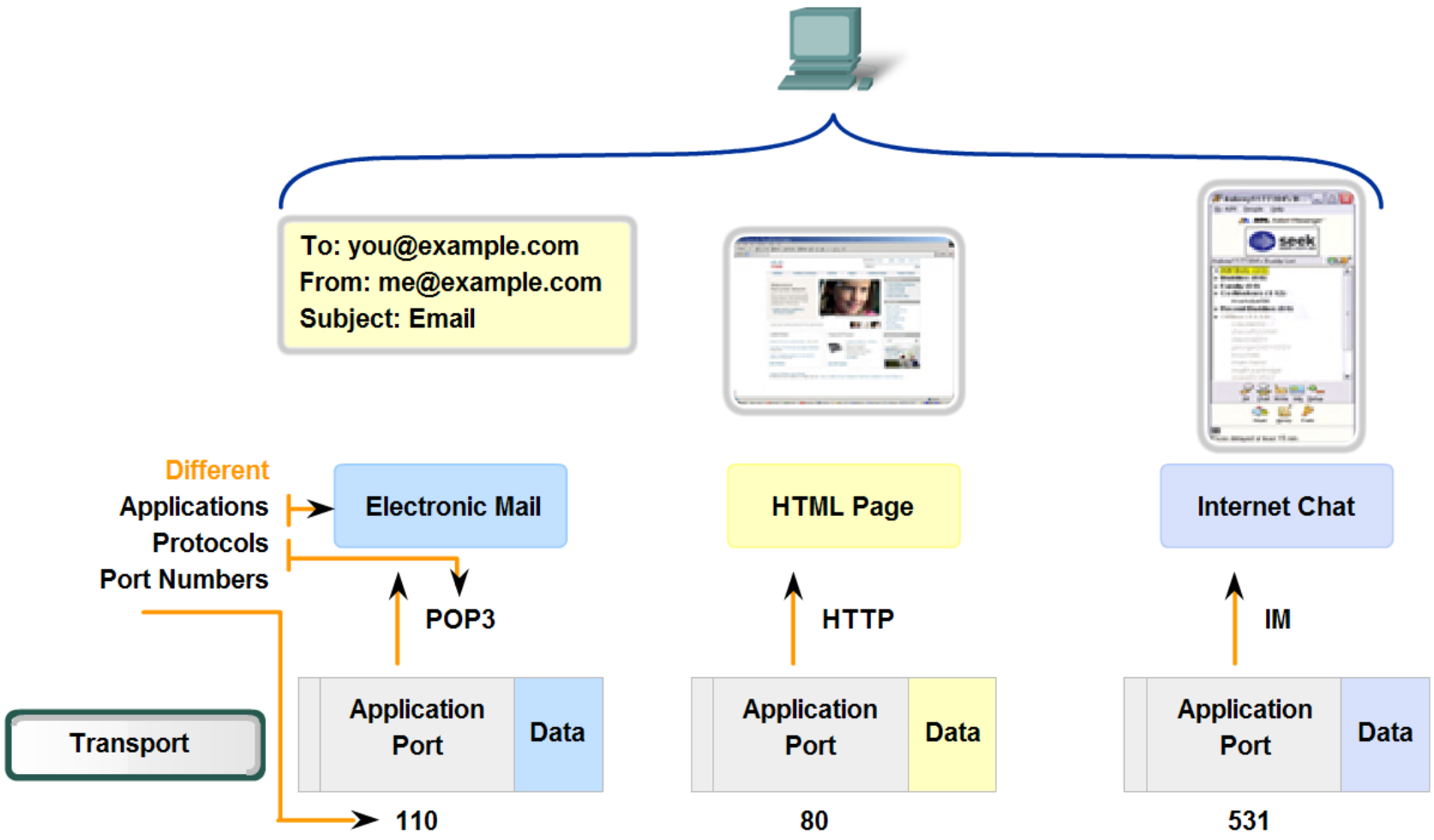




Port Numbers

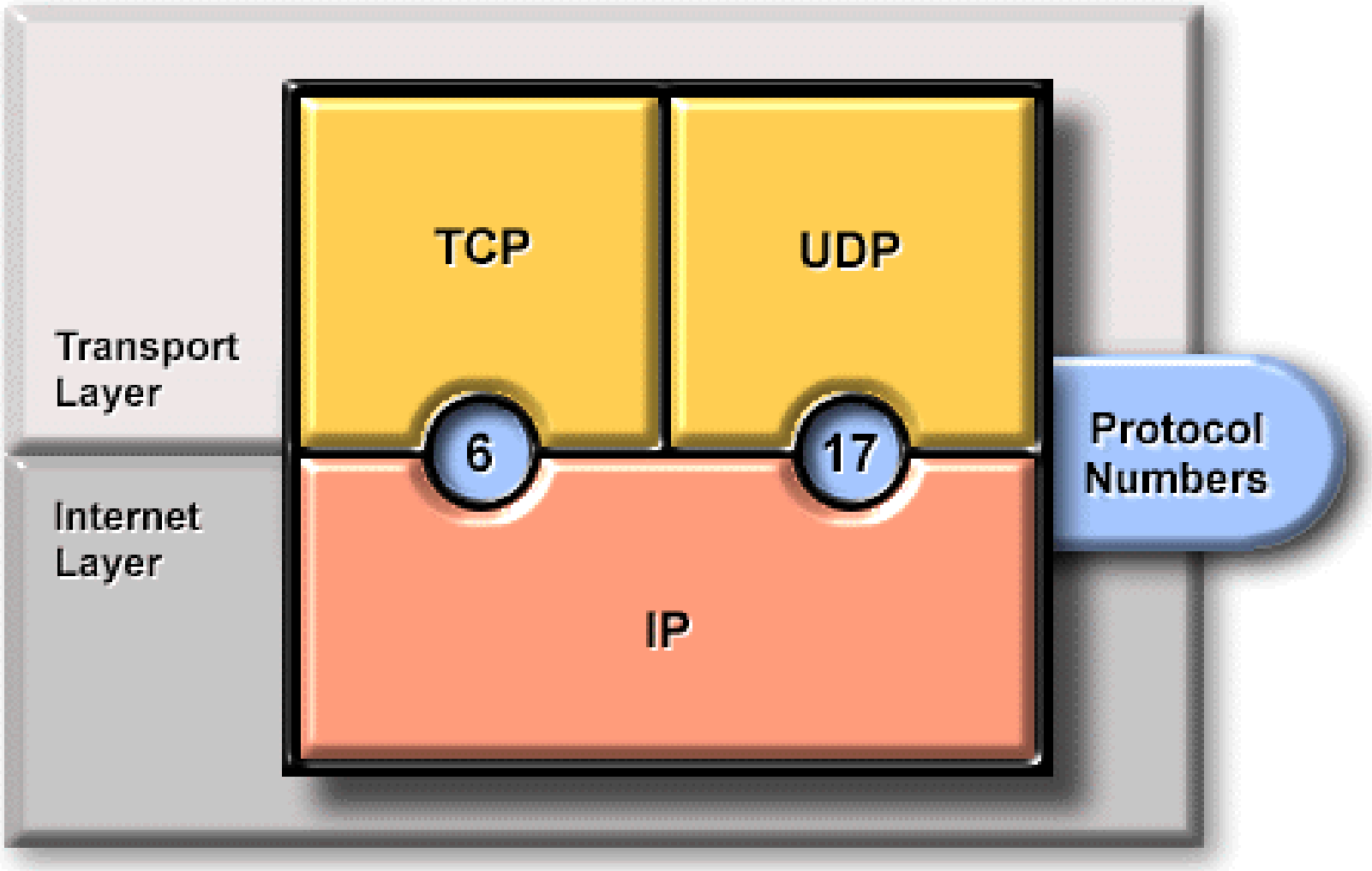


Port Addressing



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

The Protocol Field



- Determines destination upper-layer protocol