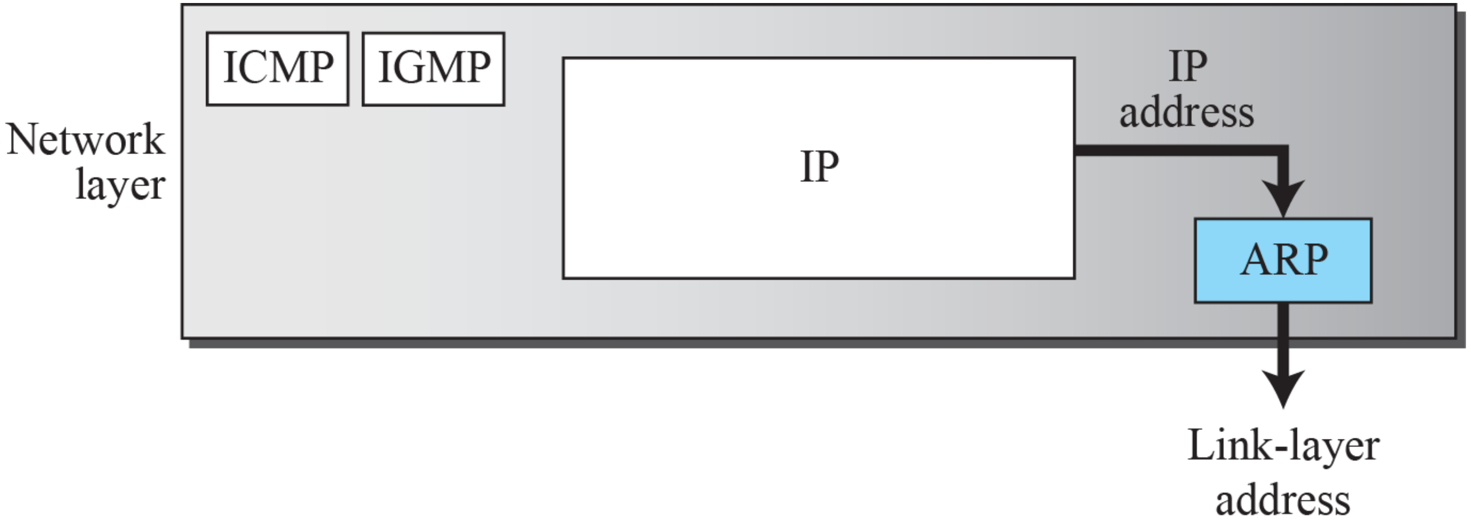
Protocols

Topics of Discussion

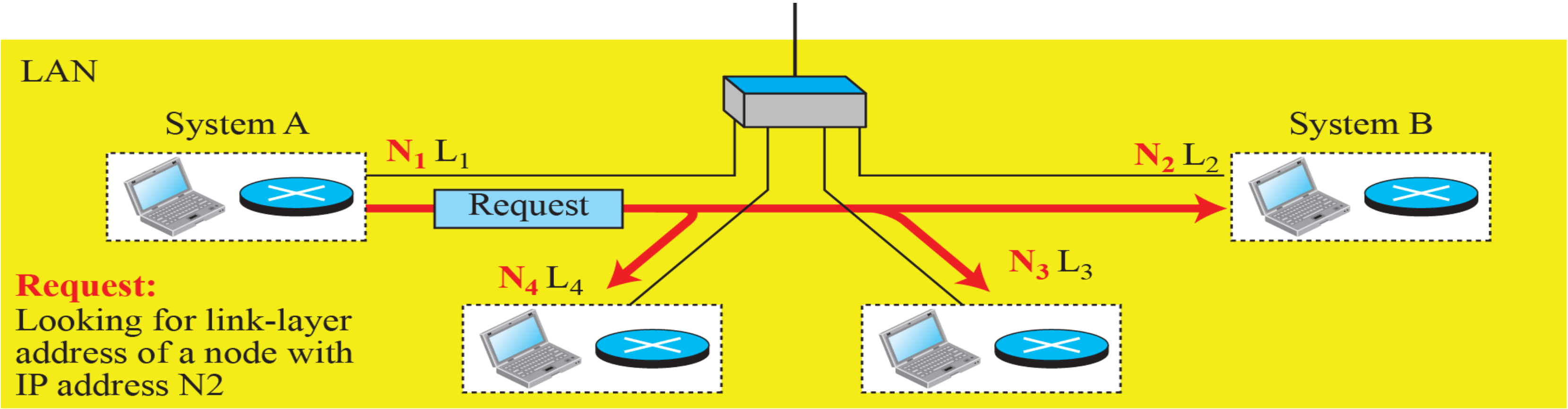
- *ARP Header*
- *IP Header*
- *Transport layer protocols*
  - *User Datagram Protocol*
    - *UDP Header Format*
  - *Transmission Control Protocol*
    - *TCP Header Format*
  - *Stream Control Transmission Protocol*
    - *Services*
    - *Features*
- *Port Numbers*

Address Resolution Protocol (ARP)

- Anytime a node has an IP datagram to send to another node in a link, it has the IP address of the receiving node.
- However, the IP address of the next node is not helpful in moving a frame through a link; we need the link-layer address of the next node.
- This is the time when the Address Resolution Protocol (ARP) becomes helpful.
- Position of ARP in TCP/IP protocol suite is shown below.

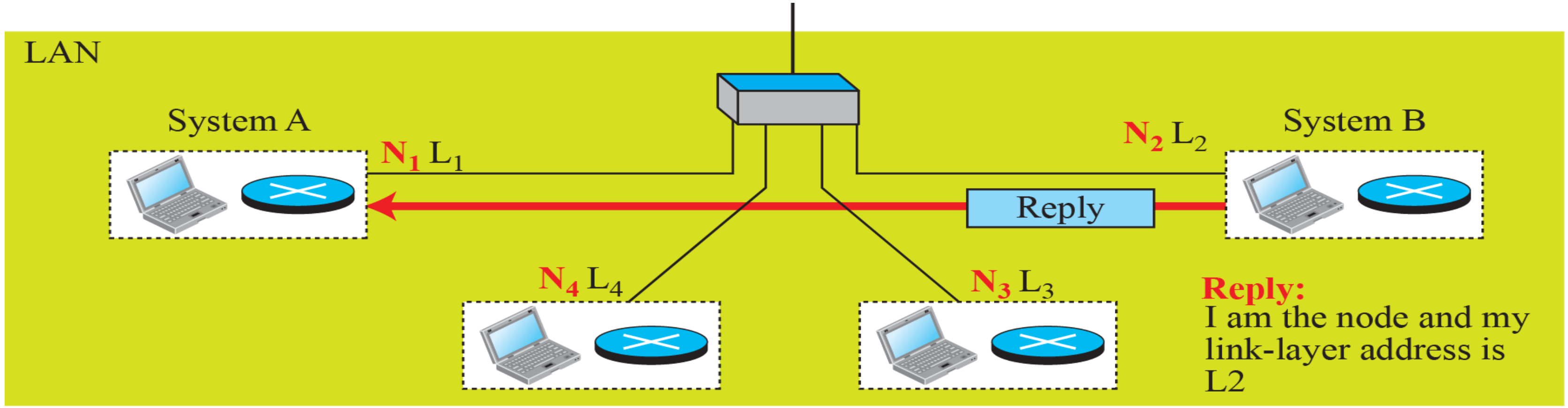


ARP Operation



Request:
Looking for link-layer address of a node with IP address N2

a. ARP request is broadcast



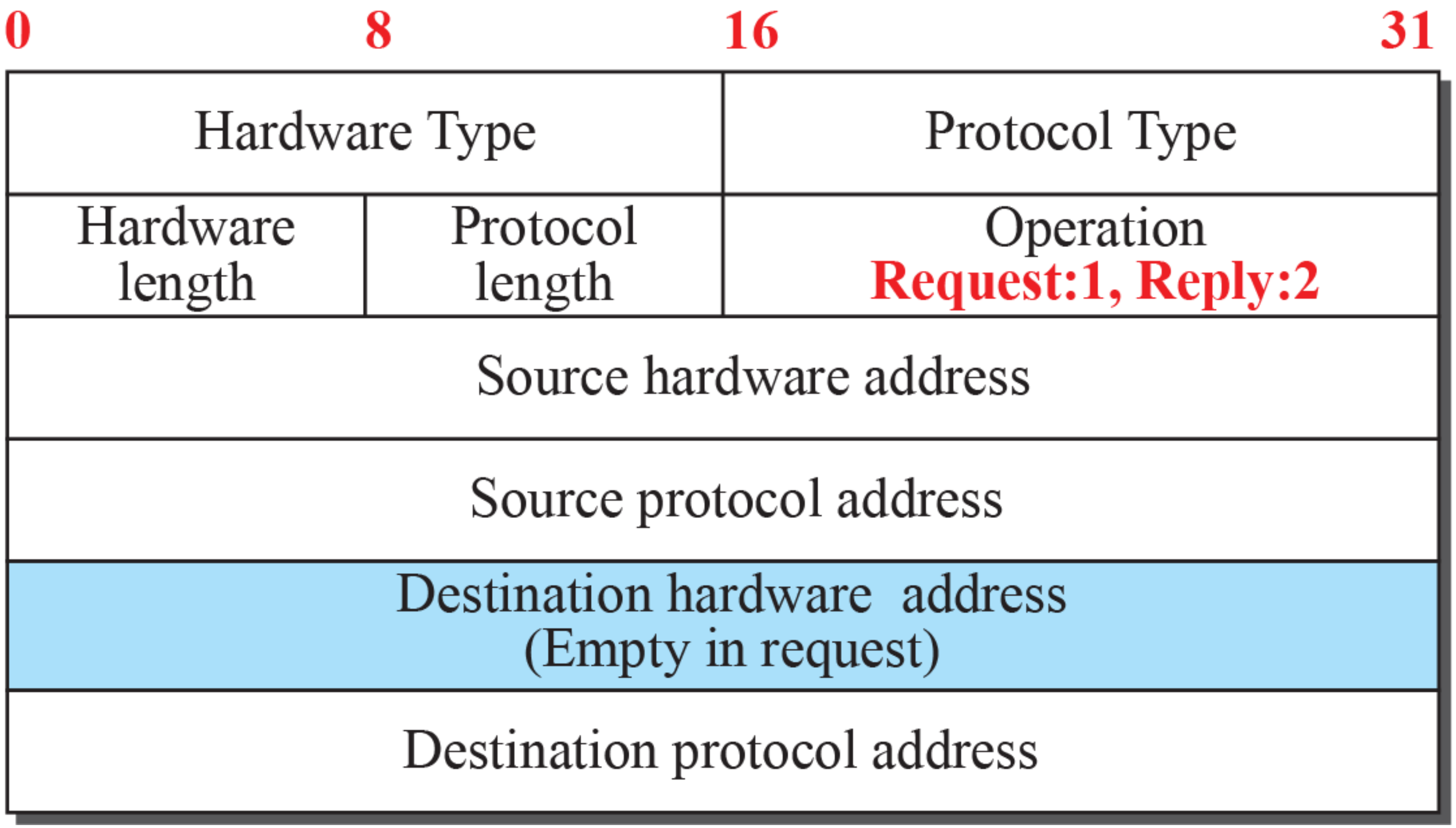
Reply:
I am the node and my link-layer address is L2

b. ARP reply is unicast

ARP Packet Format

Hardware: LAN or WAN protocol

Protocol: Network-layer protocol



0 8 16 31

Hardware Type

Protocol Type

Hardware length

Protocol length

Operation

Request:1, Reply:2

Source hardware address

Source protocol address

Destination hardware address

(Empty in request)

Destination protocol address

IPv4 protocol

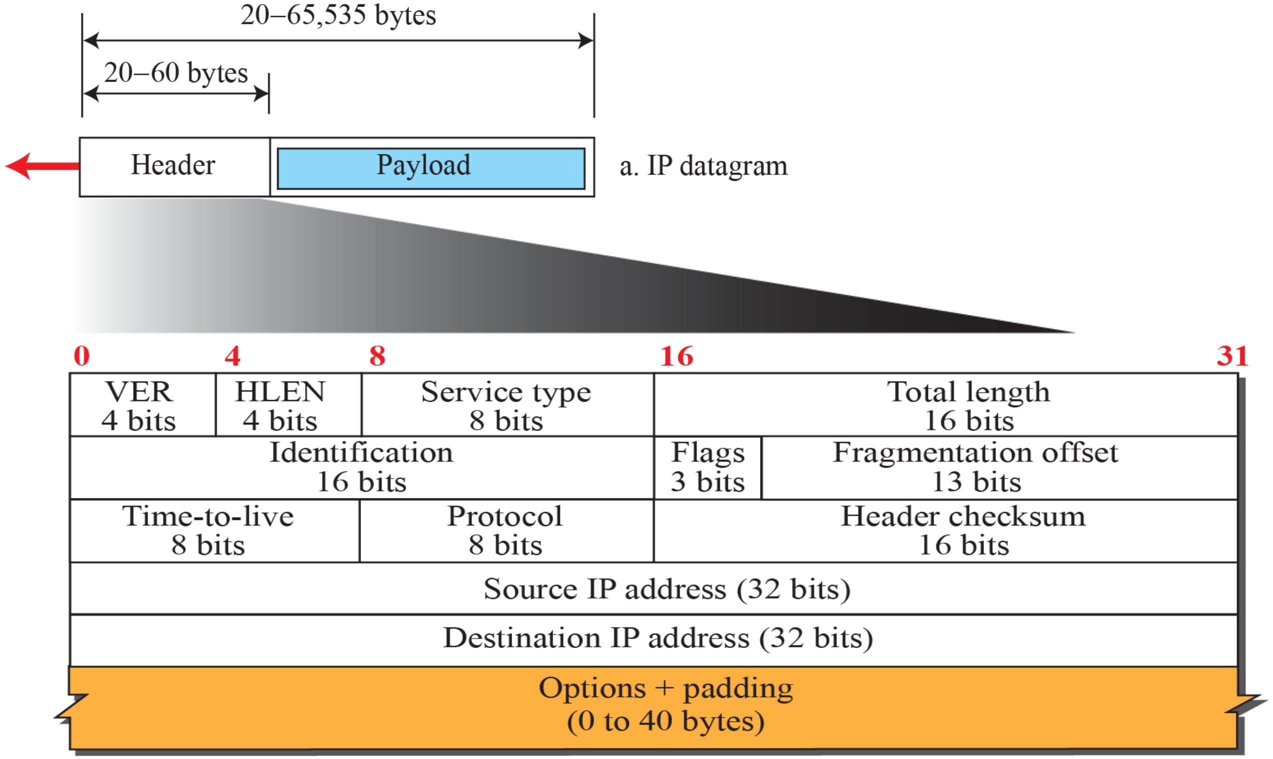
Legend

VER: version number
HLEN: header length
byte: 8 bits

- Packets used by the IP are called datagrams. Figure below shows the IPv4 datagram format.
- A datagram is a variable-length packet consisting of two parts: **header and payload (data).**
- The header is 20 to 60 bytes in length and contains information essential to routing and delivery.
- It is customary in TCP/IP to show the header in 4-byte sections.

ICMP: 01	UDP: 17
IGMP: 02	OSPF: 89
TCP: 06	

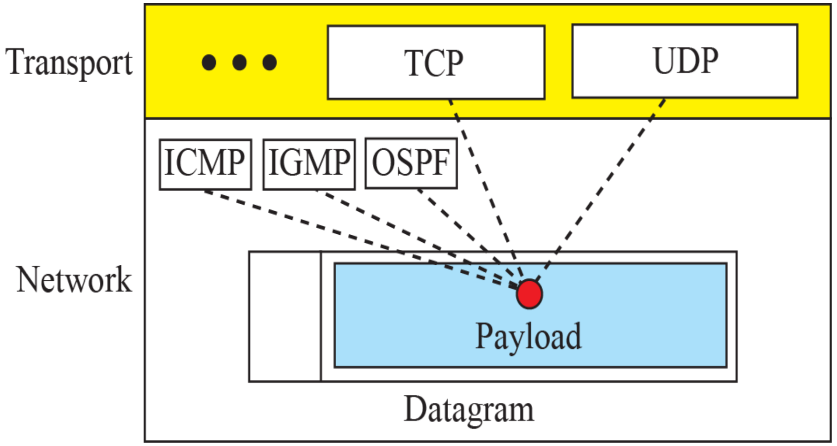
Some protocol values



a. IP datagram

0	4	8	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits	Total length 16 bits	
Identification 16 bits			Flags 3 bits	Fragmentation offset 13 bits
Time-to-live 8 bits		Protocol 8 bits	Header checksum 16 bits	
Source IP address (32 bits)				
Destination IP address (32 bits)				
Options + padding (0 to 40 bytes)				

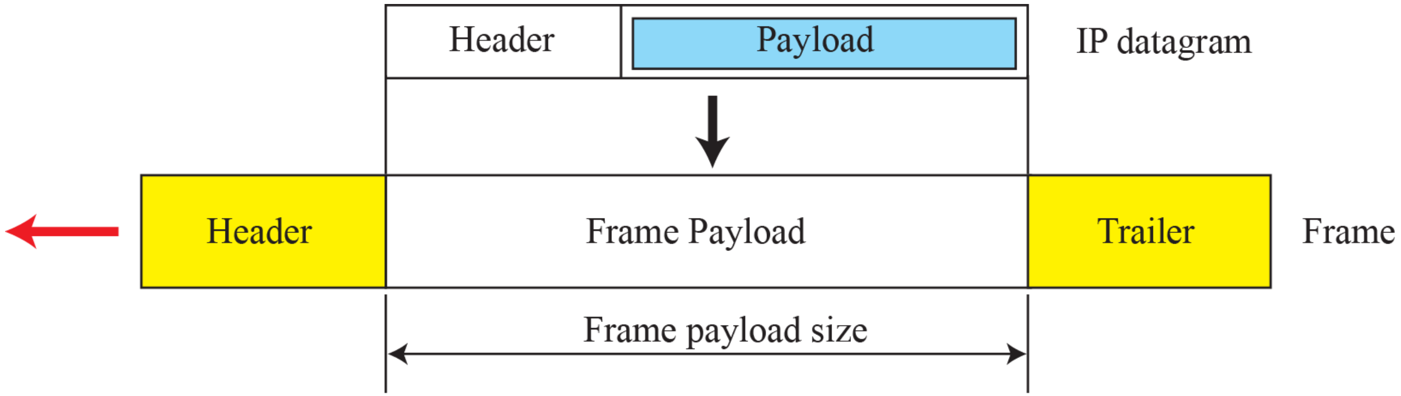
b. Header format



a. Multiplexing

Fragmentation Offset

- A datagram can travel through different networks.
- Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame.
- The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled.
- The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.
- For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.



MTU: Maximum size of frame payload

Options in the frame format

- The header of the IPv4 datagram is made of two parts: **a fixed part and a variable part.**
- The fixed part is 20 bytes long and was discussed in the previous section.
- The variable part comprises the options that can be a maximum of 40 bytes (in multiples of 4-bytes) to preserve the boundary of the header.

Figure 20.14 *Taxonomy of options in IPv4*

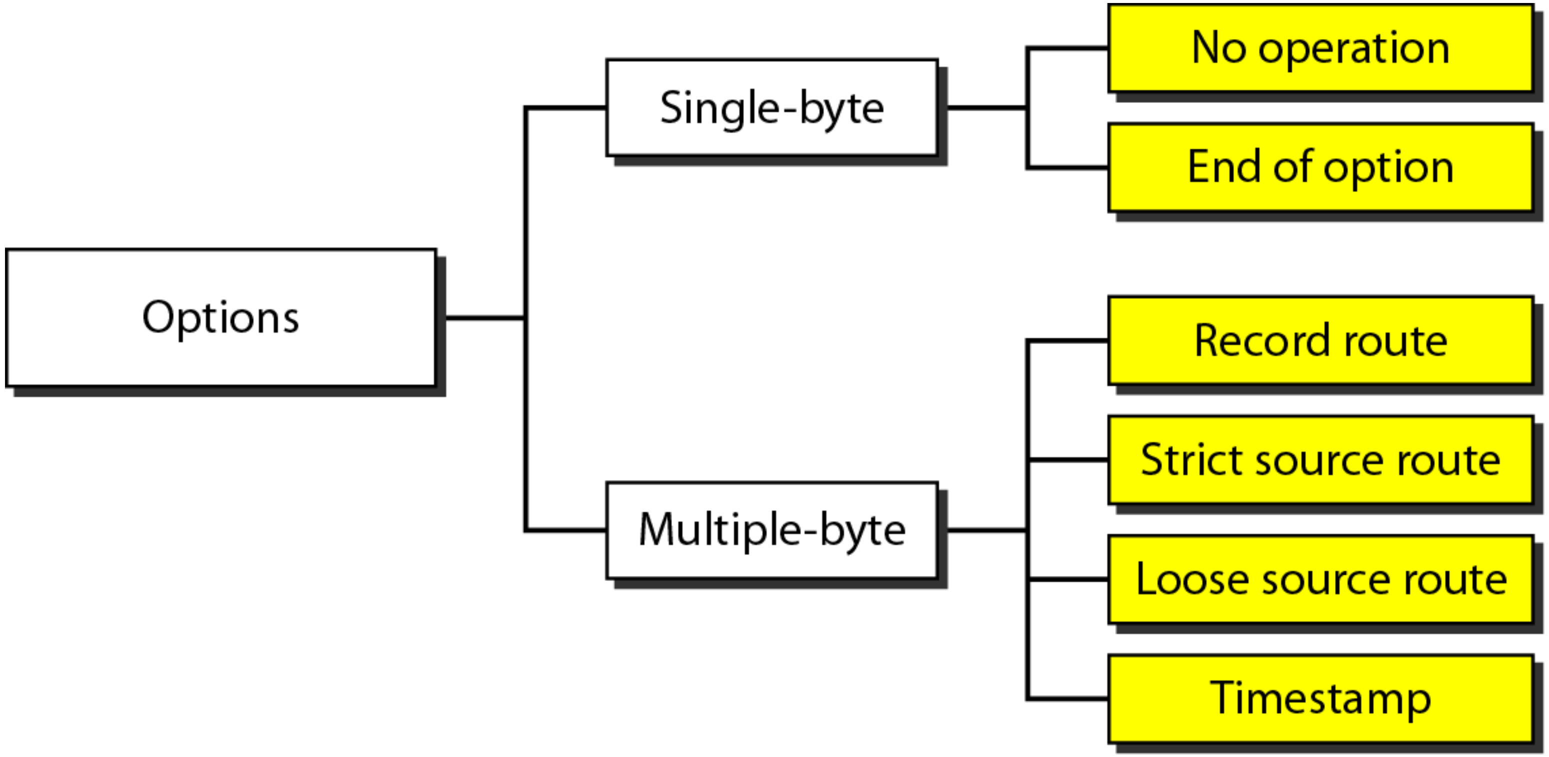
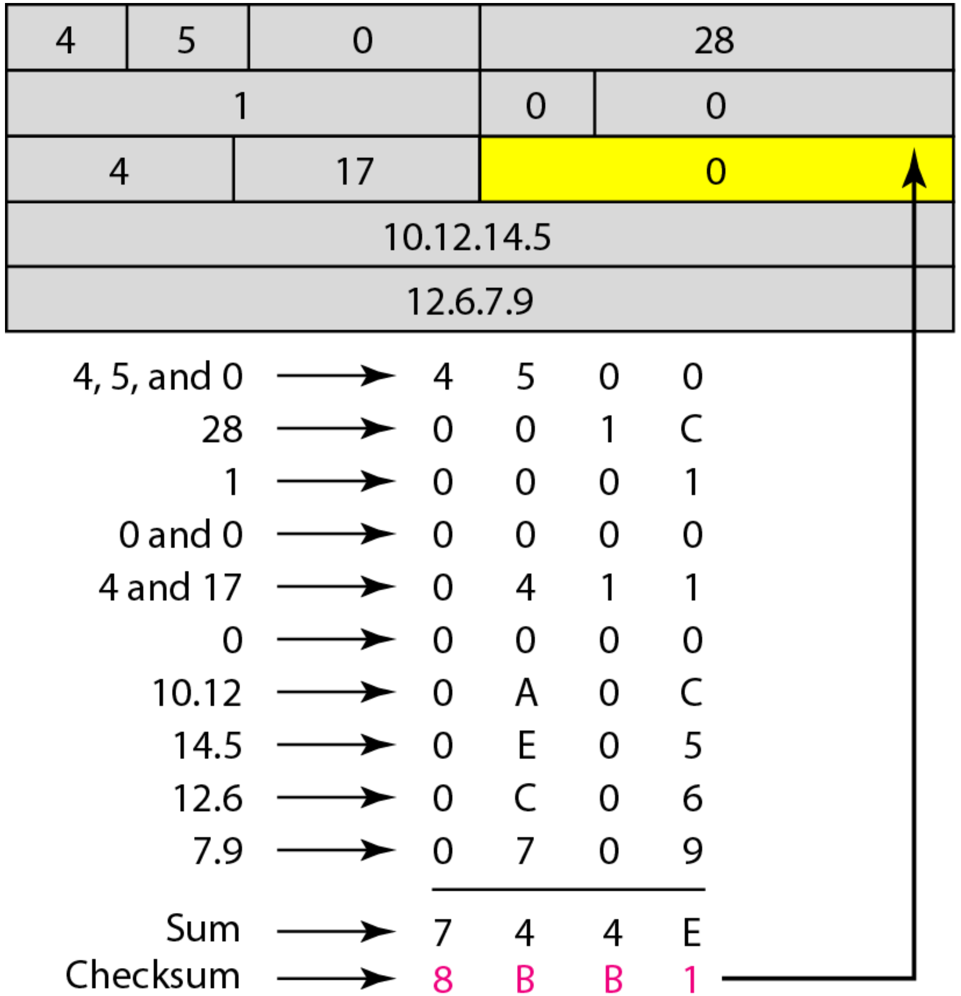


Figure 20.13 *Example of checksum calculation in IPv4*



4	5	0	28
1		0	0
4	17		0
10.12.14.5			
12.6.7.9			

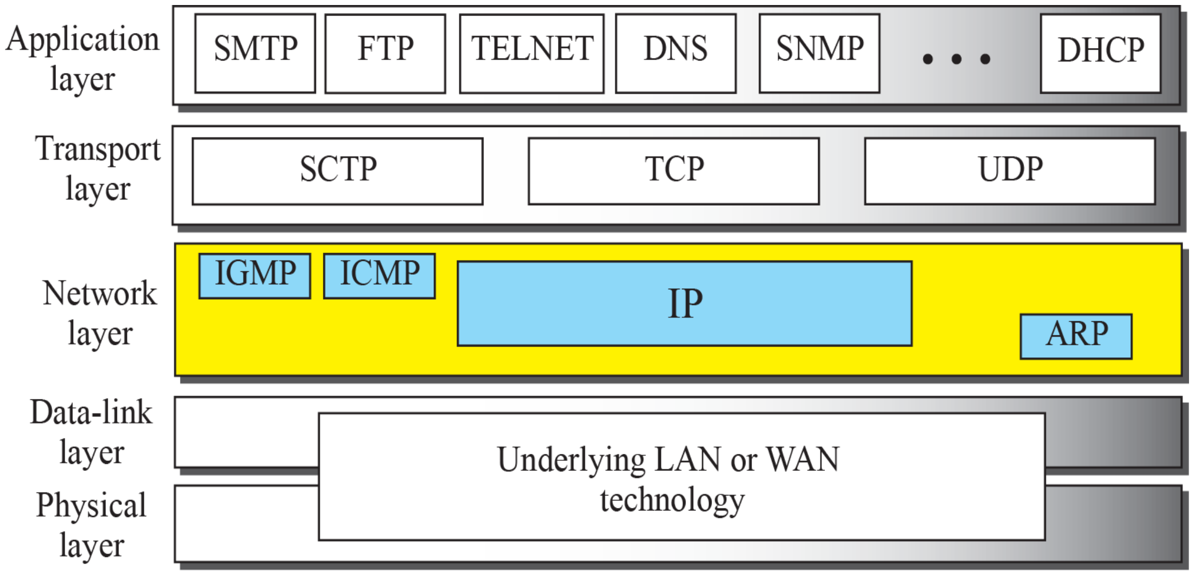
4, 5, and 0	→	4	5	0	0
28	→	0	0	1	C
1	→	0	0	0	1
0 and 0	→	0	0	0	0
4 and 17	→	0	4	1	1
0	→	0	0	0	0
10.12	→	0	A	0	C
14.5	→	0	E	0	5
12.6	→	0	C	0	6
7.9	→	0	7	0	9
Sum	→	7	4	4	E
Checksum	→	8	B	B	1

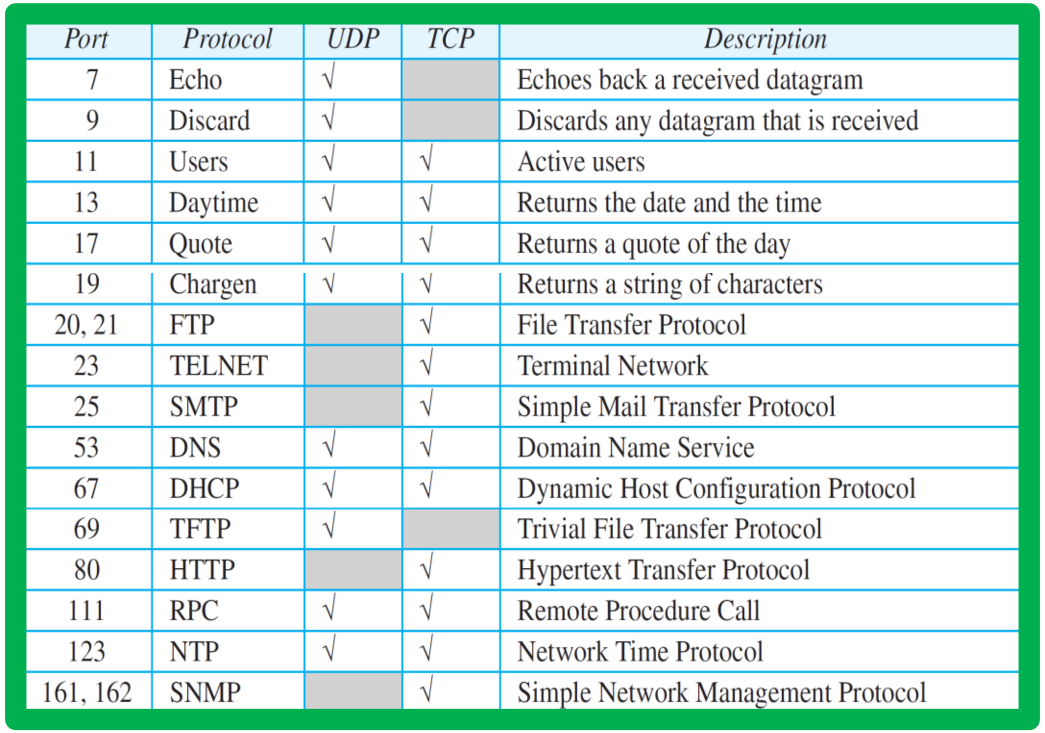
It Provide Services to the Application Layer and it receives services from the Network Layer

Transport Layer Protocols

It acts as a licence between the Sever and The Client Programs

- A transport-layer protocol usually has several responsibilities.
- One is to create a **process-to-process** communication; these protocols use **port numbers** to accomplish this.
- Port numbers provide **end-to-end addresses** at the transport layer and allow multiplexing and demultiplexing at this layer, just as IP addresses do at the network layer.

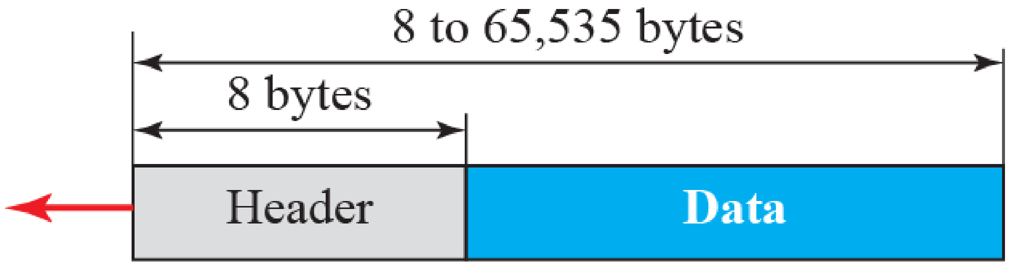




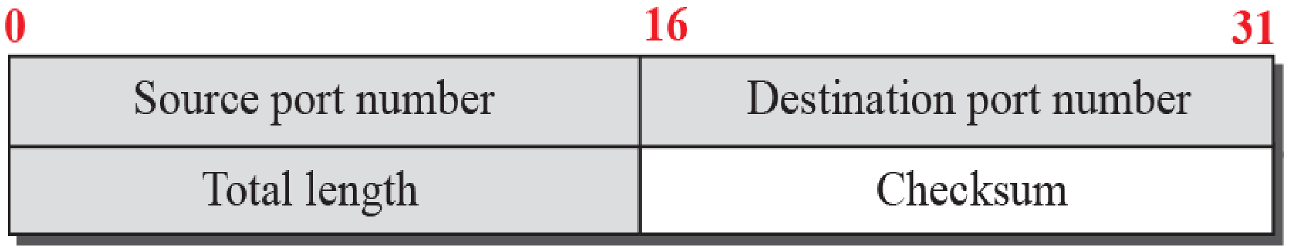
Port	*Protocol*	*UDP*	*TCP*	*Description*
7	Echo	√		Echoes back a received datagram
9	Discard	√		Discards any datagram that is received
11	Users	√	√	Active users
13	Daytime	√	√	Returns the date and the time
17	Quote	√	√	Returns a quote of the day
19	Chargen	√	√	Returns a string of characters
20, 21	FTP		√	File Transfer Protocol
23	TELNET		√	Terminal Network
25	SMTP		√	Simple Mail Transfer Protocol
53	DNS	√	√	Domain Name Service
67	DHCP	√	√	Dynamic Host Configuration Protocol
69	TFTP	√		Trivial File Transfer Protocol
80	HTTP		√	Hypertext Transfer Protocol
111	RPC	√	√	Remote Procedure Call
123	NTP	√	√	Network Time Protocol
161, 162	SNMP		√	Simple Network Management Protocol

UDP protocol

- The User Datagram Protocol (UDP) is a **connectionless, unreliable transport protocol**.
- UDP is a very simple protocol using a minimum of overhead.
- UDP packets, called user datagrams, have a fixed-size header of 8 bytes made of four fields, each of 2 bytes (16 bits).



a. UDP user datagram



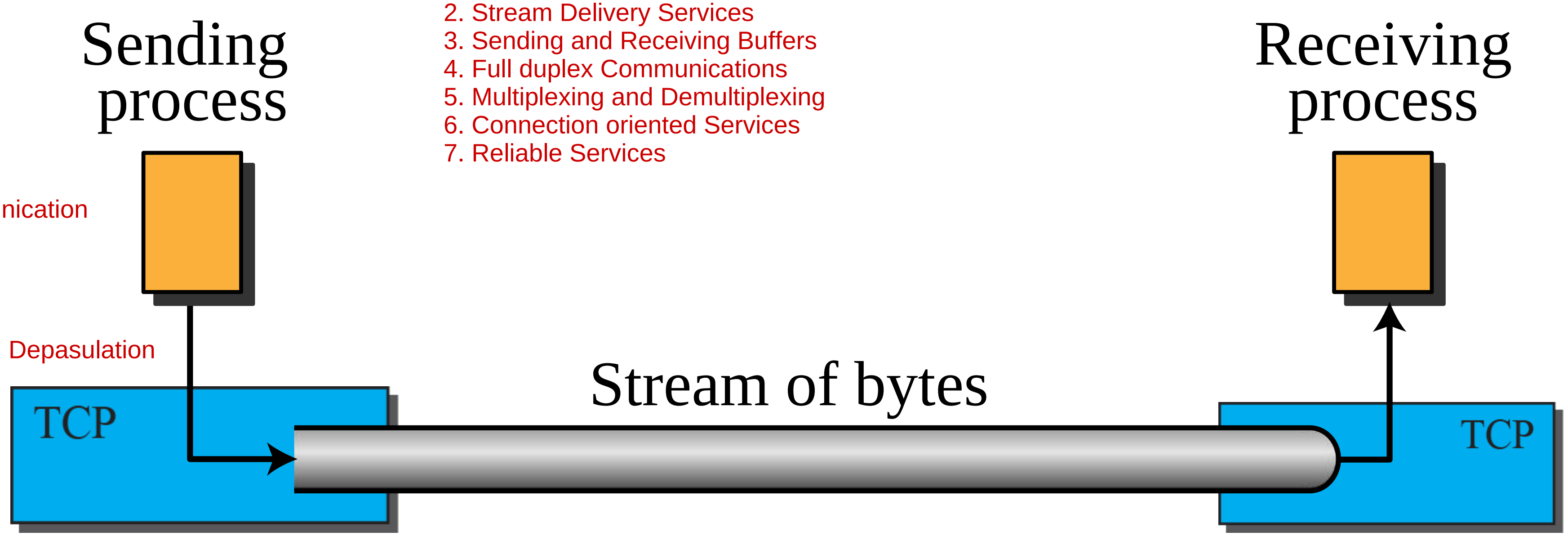
b. Header format

- The first two fields define the source and destination port numbers.
- The third field defines the total length of the user datagram, (header plus data).
- The 16 bits can define a total length of 0 to 65,535 bytes.
- However, the total length needs to be less because a UDP user datagram is stored in an IP datagram with the total length of 65,535 bytes.
- The last field can carry the optional checksum (explained later).

TCP Protocol

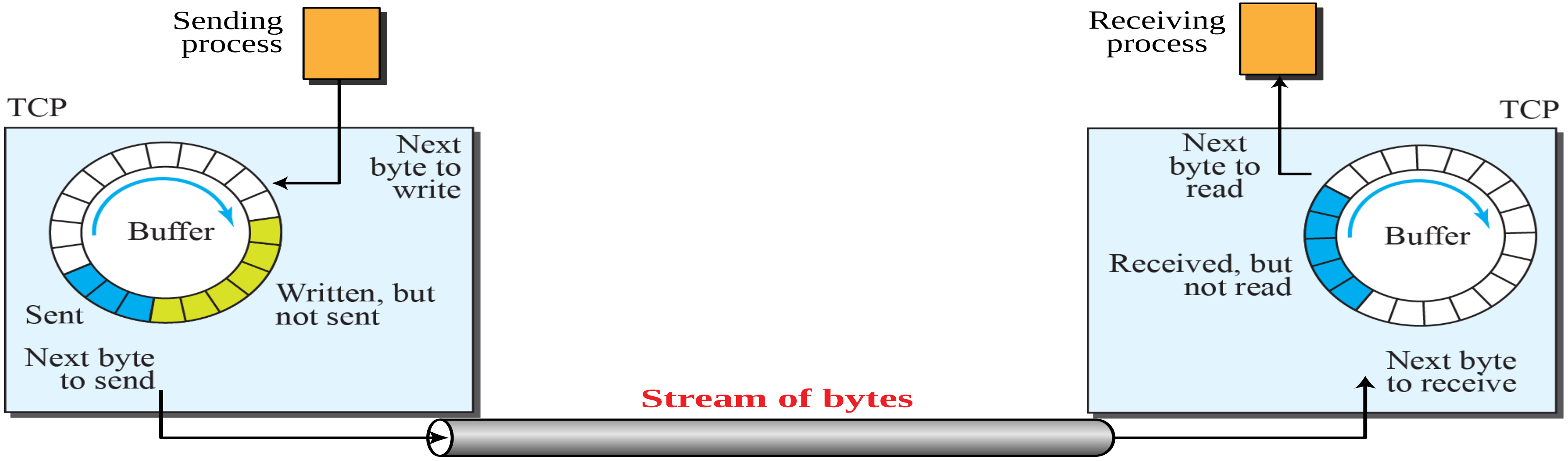
- Transmission Control Protocol (TCP) is a connection-oriented, reliable protocol.
- TCP explicitly defines connection establishment, data transfer, and connection teardown phases to provide a connection-oriented service.
- TCP uses a combination of GBN(Go-Back-N) and SR(Selective Repeat) protocols to provide reliability.

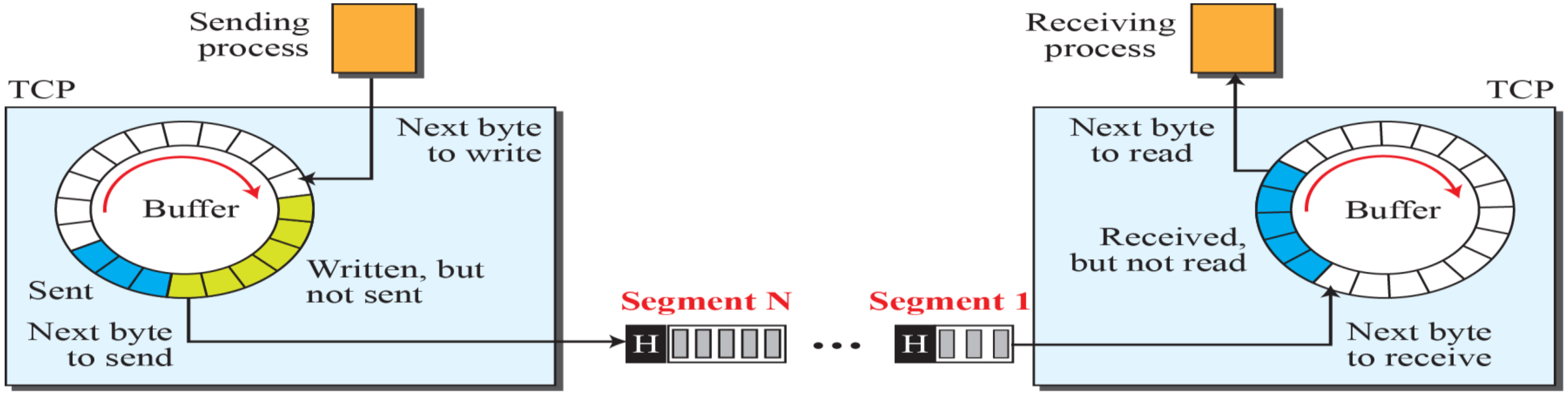
Services Provided By TCP:
1. Process to Process Communication (also provided by UDP)
2. Stream Delivery Services
3. Sending and Receiving Buffers
4. Full duplex Communications
5. Multiplexing and Demultiplexing
6. Connection oriented Services
7. Reliable Services



Services Provided By UDP:
1. Process to Process Communication
2. Connection less Service
3. No Flow Control
4. No Error control
5. Check sum is optional
6. Provides Encapsulation and Depasulation

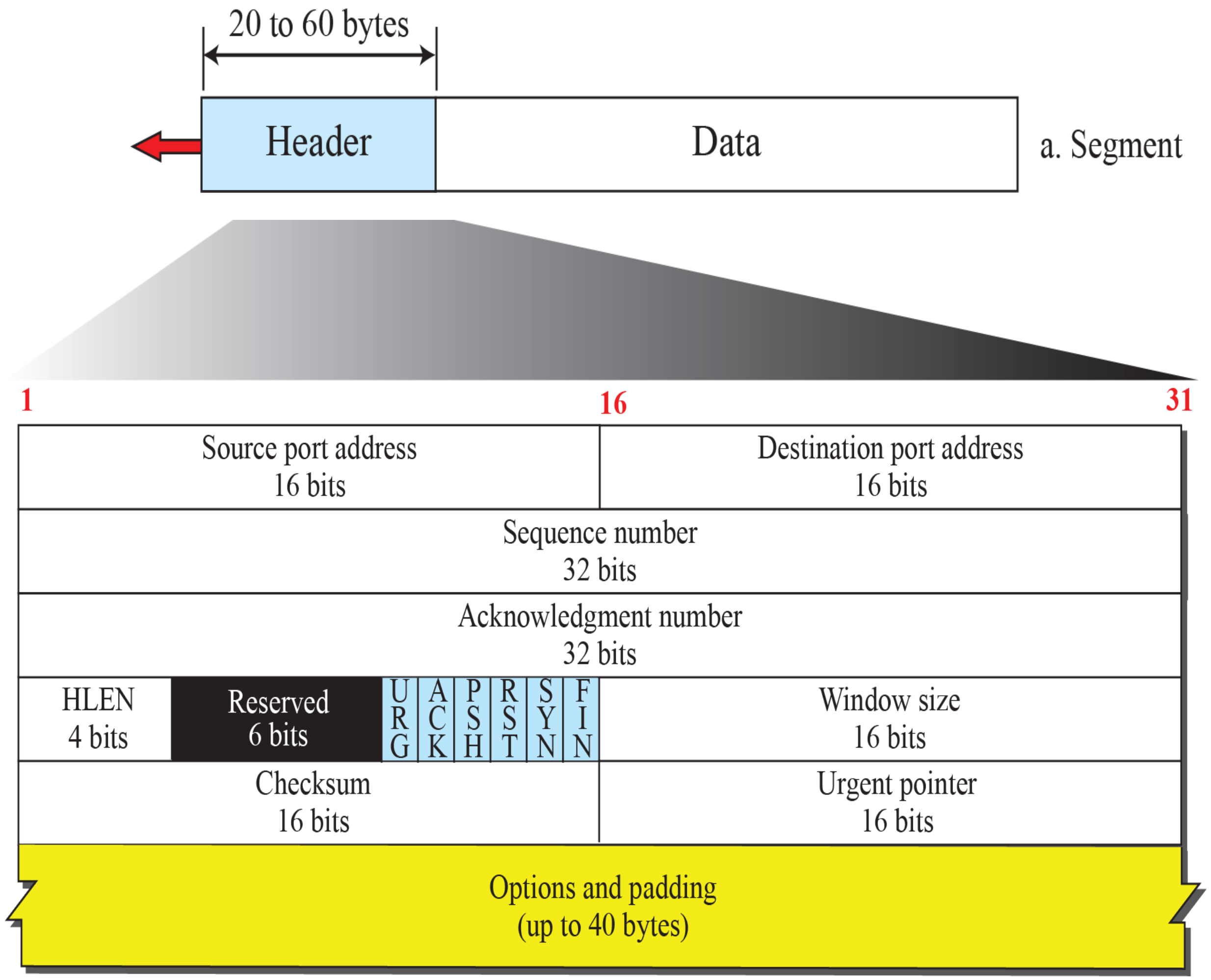
Sending and Receiving Buffers





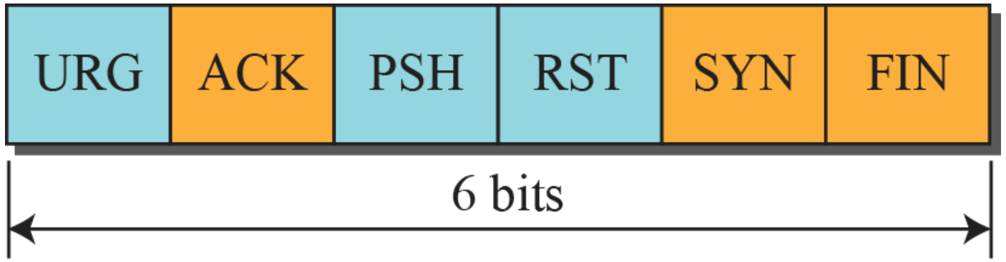
TCP Segment

TCP Segment Format



a. Segment

b. Header



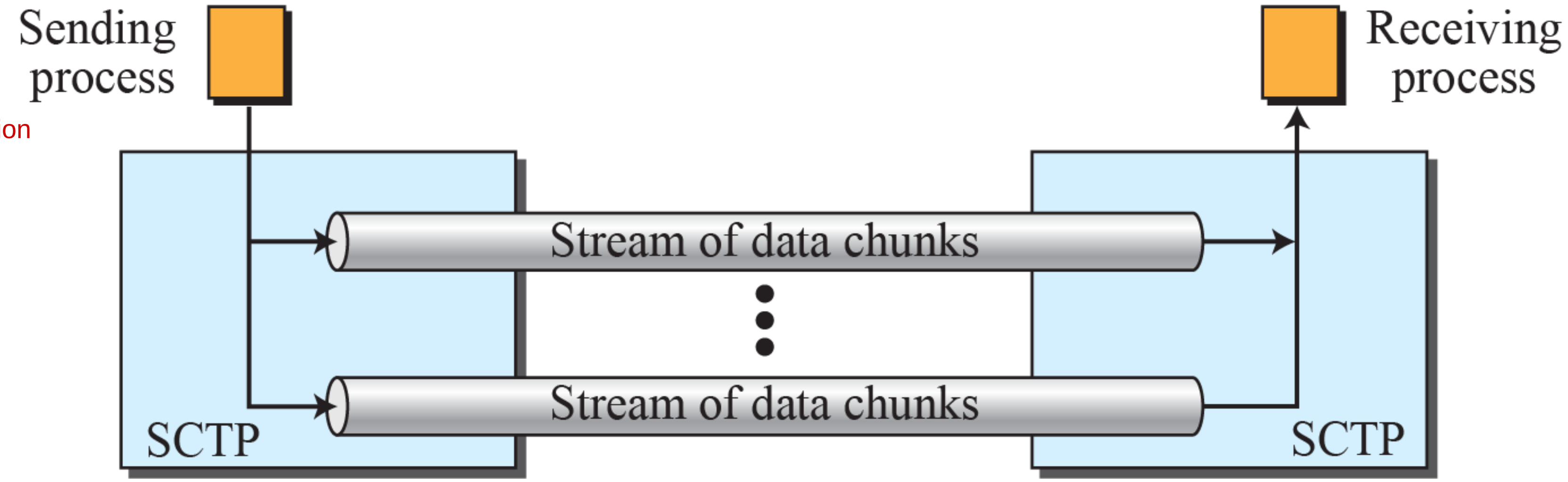
URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection

SCTP Protocol

Stream Control Transmission Protocol (SCTP) is a new transport-layer protocol designed to combine some features of UDP and TCP in an effort to create a protocol for multimedia communication.

Services Provided By SCTP:

1. Process to Process Communication
2. Multiple Screens
3. Multihoming
4. Full Duplex Communication
5. Connection Oriented Services
6. Reliable Services



Multiple-stream concept

Figure 39 : Multihoming concept

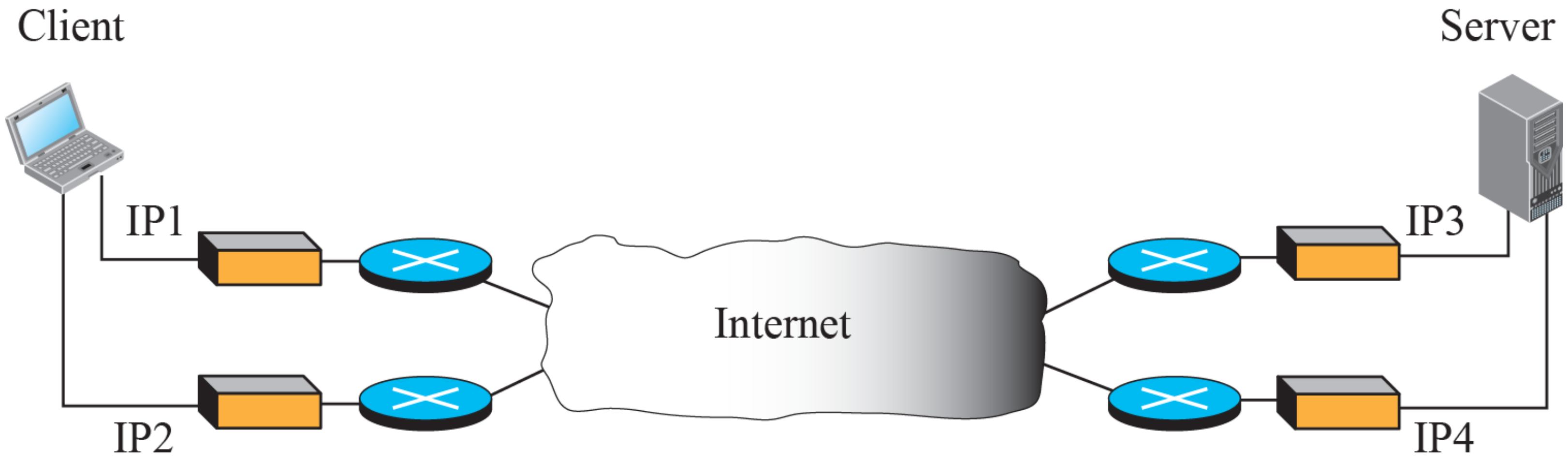
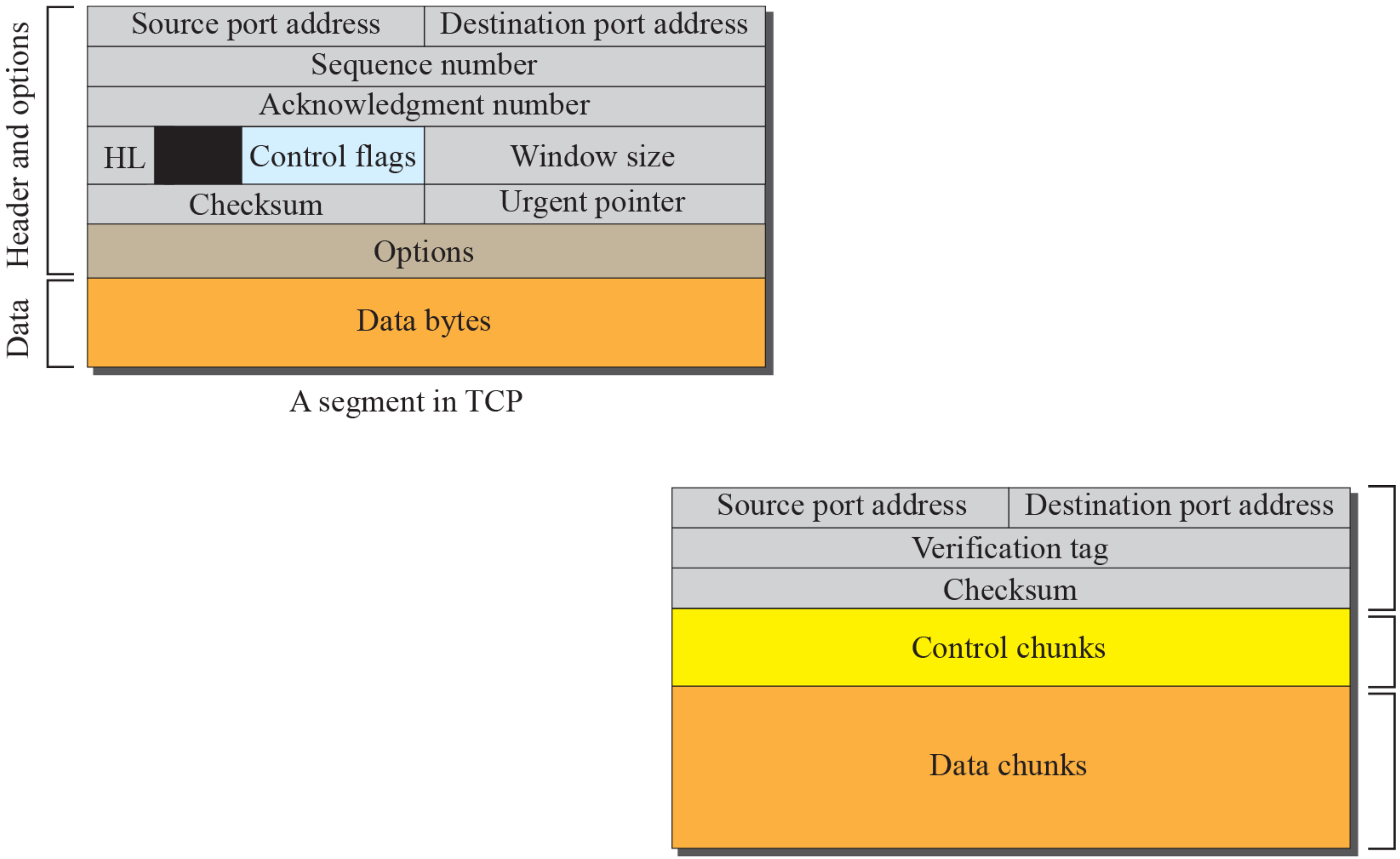


Figure 40 : Comparison between a TCP segment and an SCTP packet



Header and options

Data

Source port address | Destination port address

Sequence number

Acknowledgment number

HL | Control flags | Window size

Checksum | Urgent pointer

Options

Data bytes

A segment in TCP

Source port address | Destination port address

Verification tag

Checksum

Control chunks

Data chunks

A packet in SCTP

SCTP Features

The following shows the general features of SCTP:

Transmission Sequence Number (TSN)

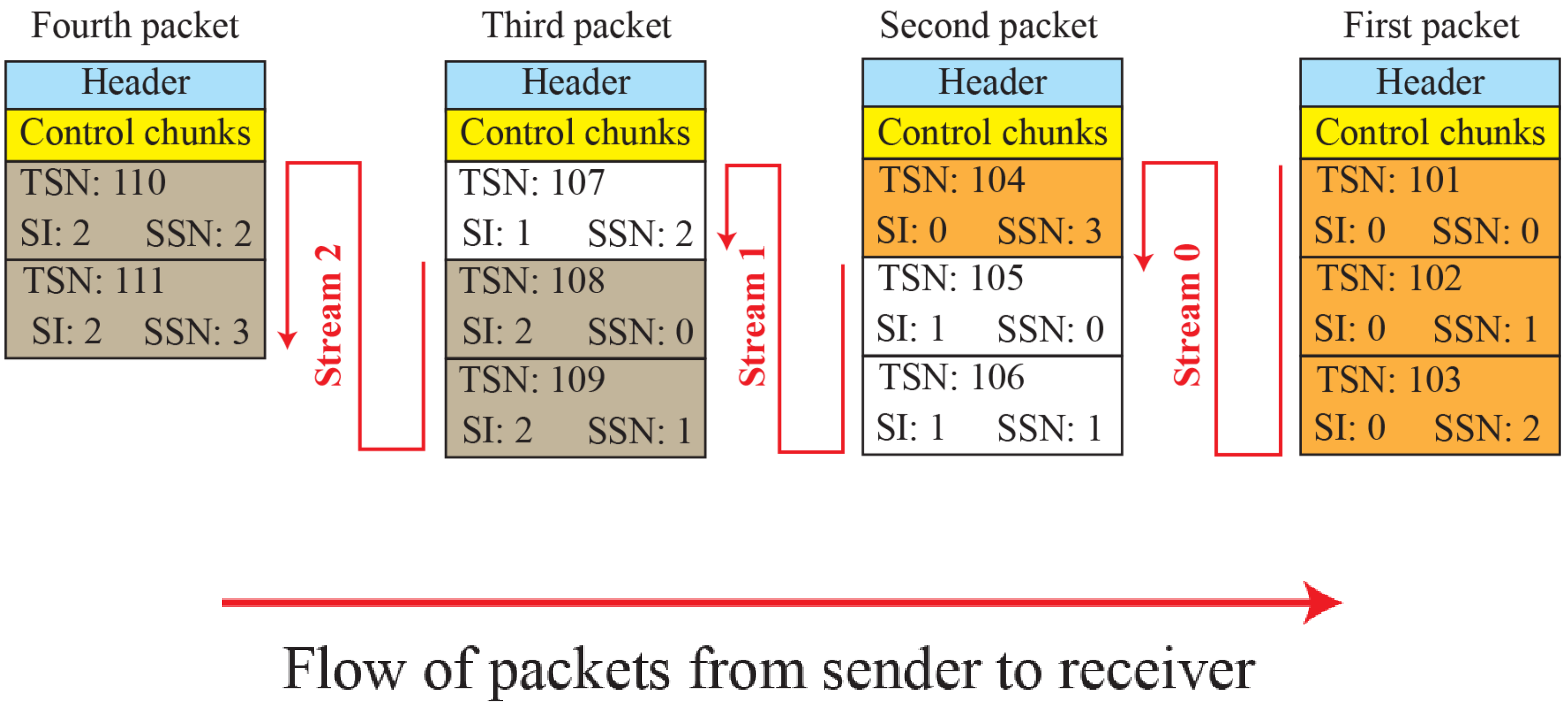
Stream Identifier (SI)

Stream Sequence Number (SSN)

Packets

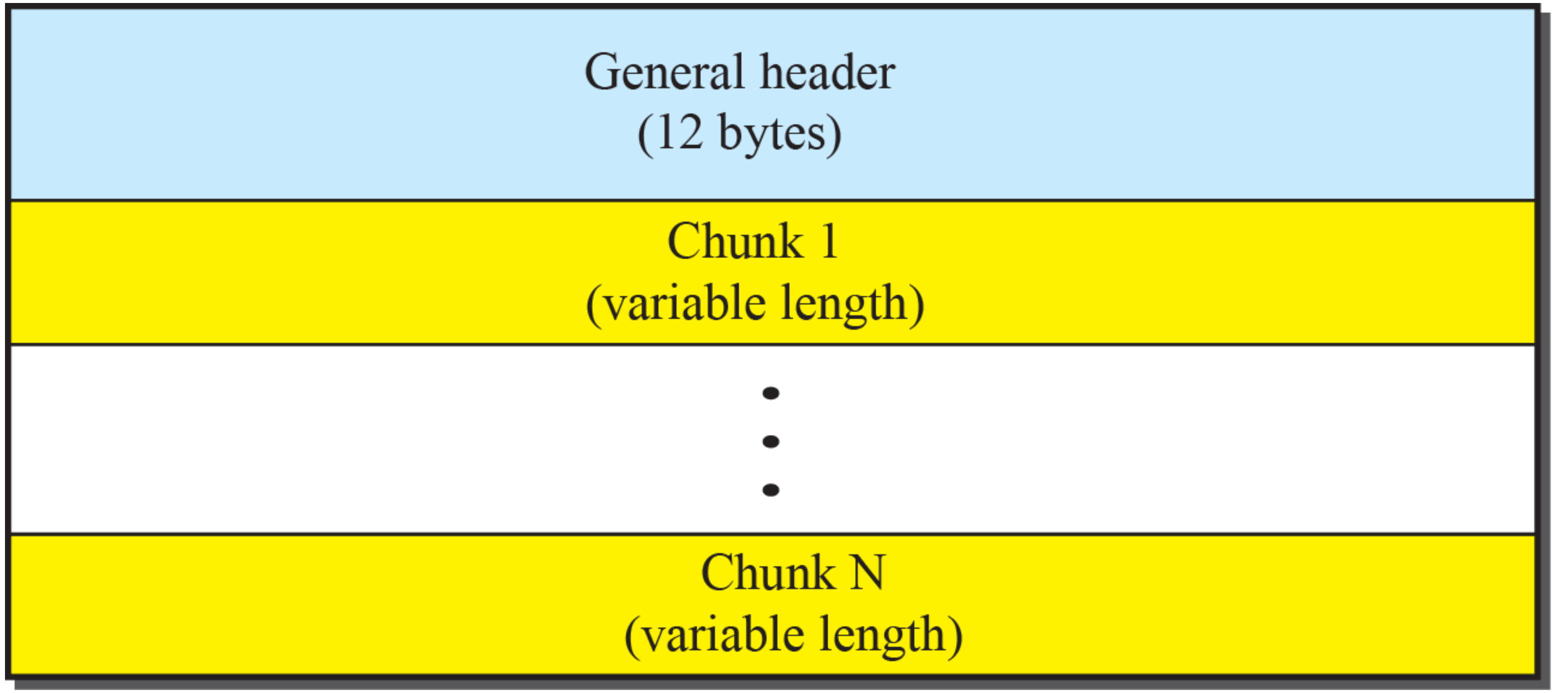
Acknowledgement Number

Figure 41 : *Packets, data chunks, and streams*



Packet Format

- *An SCTP packet has a mandatory general header and a set of blocks called chunks.*
- *There are two types of chunks: control chunks and data chunks.*
- *A control chunk controls and maintains the association; a data chunk carries user data.*
- *In a packet, the control chunks come before the data chunks.*
- *Figure 42 shows the general format of an SCTP packet.*



General header
(12 bytes)

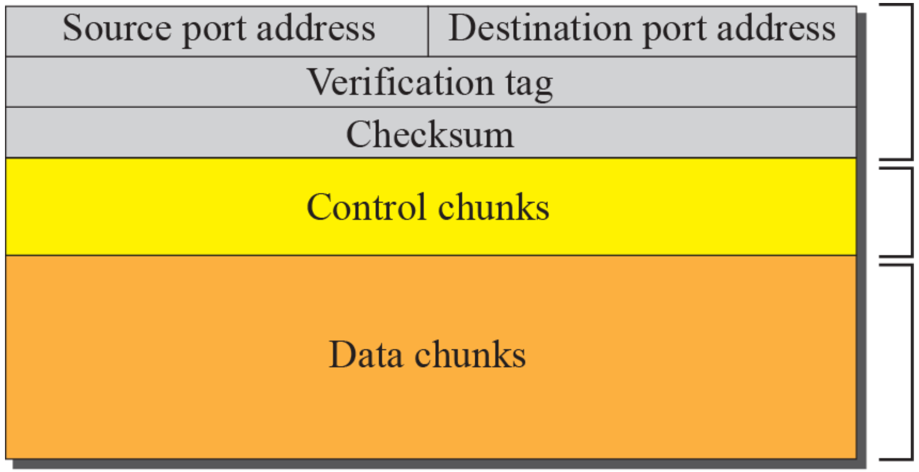
Chunk 1
(variable length)

⋮

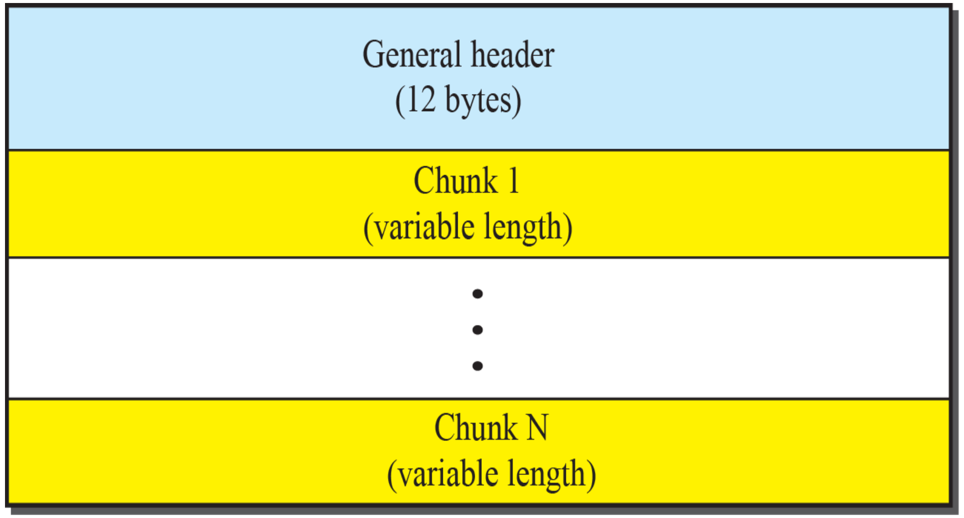
Chunk N
(variable length)

Figure 42 : SCTP packet format

SCTP Frame format and Chunks

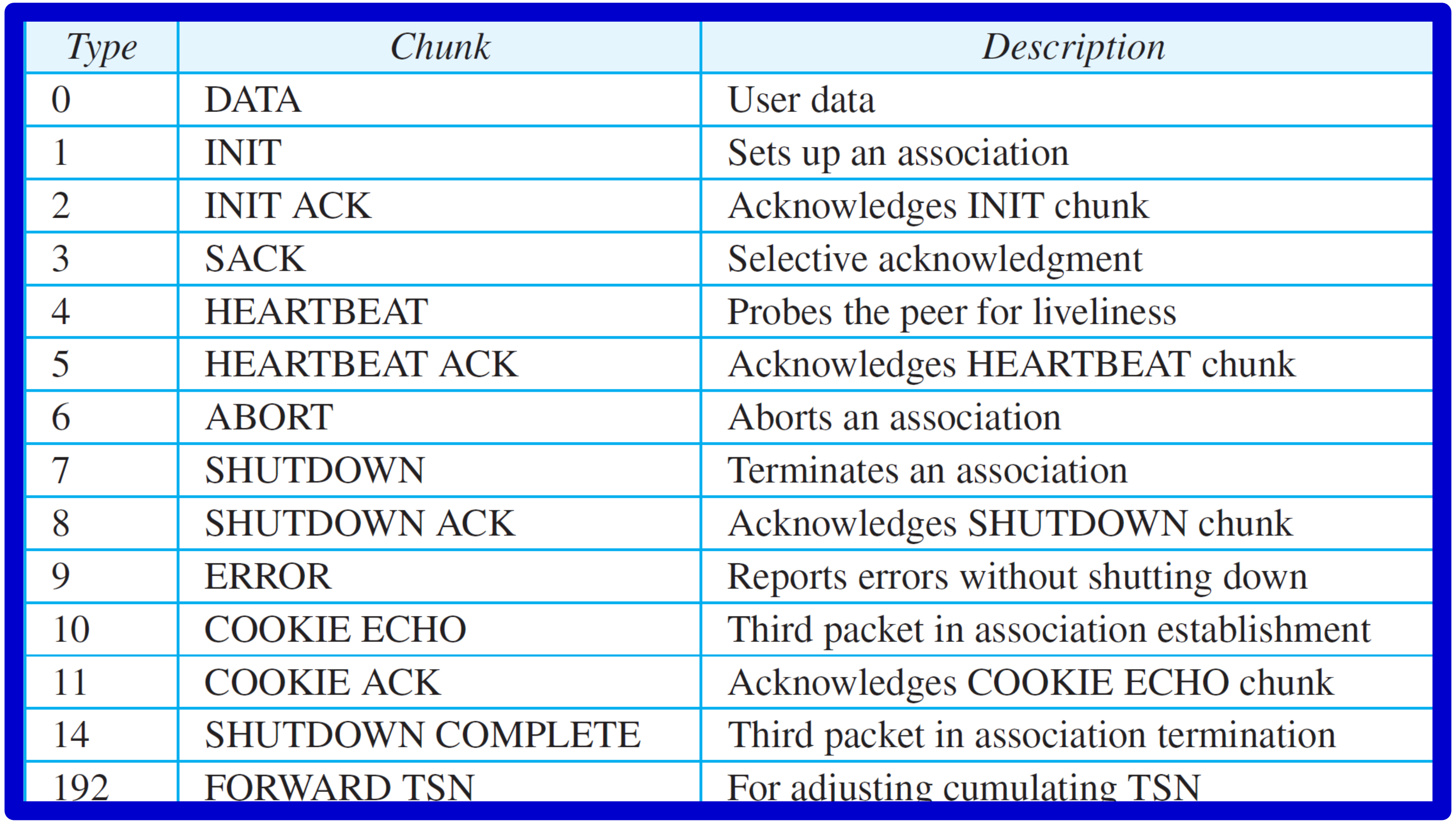


A packet in SCTP



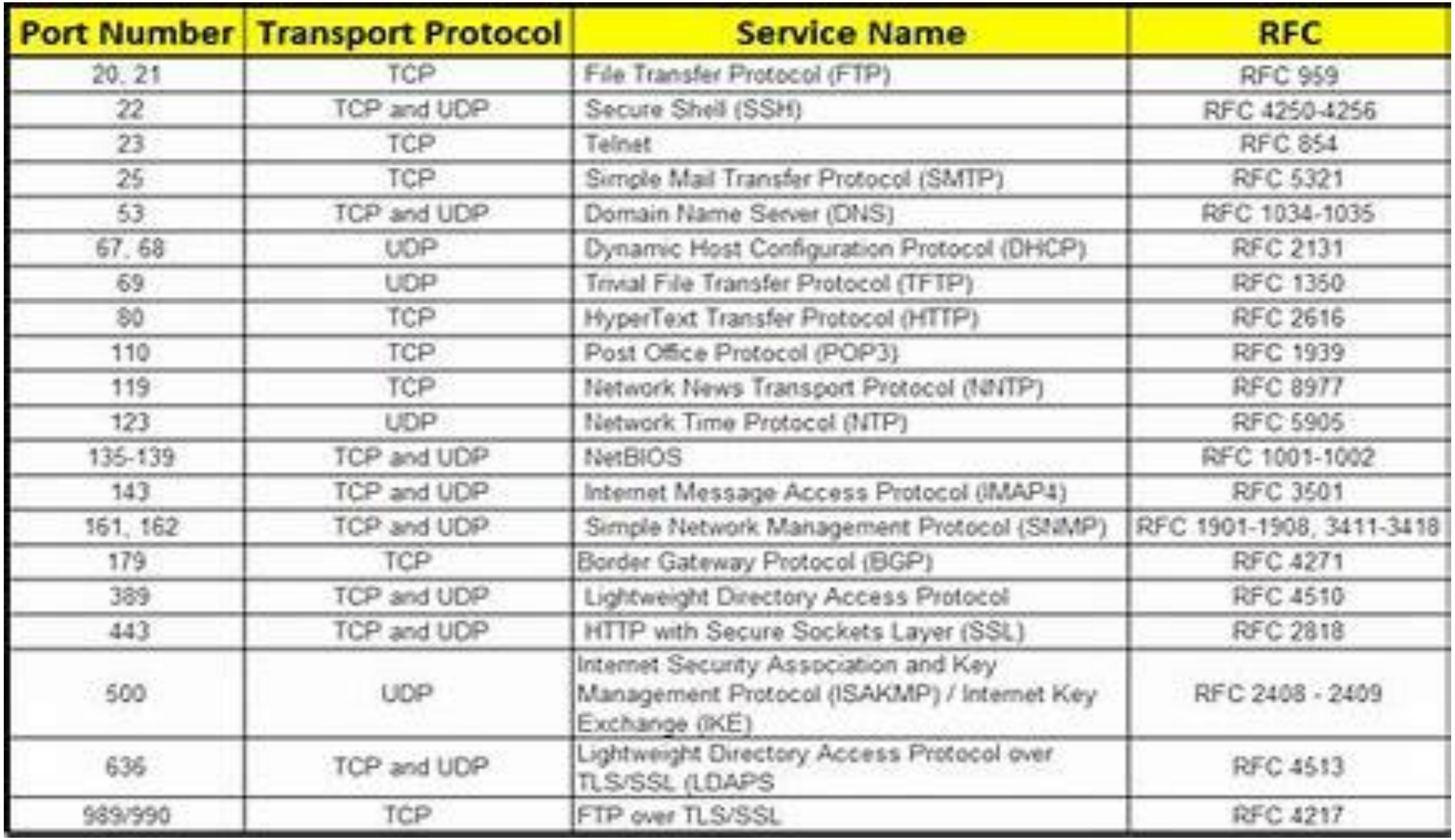
- An SCTP packet has a mandatory general header and a set of blocks called chunks.
- There are two types of chunks: **control chunks and data chunks.**
- A control chunk controls and maintains the association; a data chunk carries user data.
- In a packet, the control chunks come before the data chunks.
- Figure above shows the general format of an SCTP packet.

Chunks



Type	*Chunk*	*Description*
0	DATA	User data
1	INIT	Sets up an association
2	INIT ACK	Acknowledges INIT chunk
3	SACK	Selective acknowledgment
4	HEARTBEAT	Probes the peer for liveliness
5	HEARTBEAT ACK	Acknowledges HEARTBEAT chunk
6	ABORT	Aborts an association
7	SHUTDOWN	Terminates an association
8	SHUTDOWN ACK	Acknowledges SHUTDOWN chunk
9	ERROR	Reports errors without shutting down
10	COOKIE ECHO	Third packet in association establishment
11	COOKIE ACK	Acknowledges COOKIE ECHO chunk
14	SHUTDOWN COMPLETE	Third packet in association termination
192	FORWARD TSN	For adjusting cumulating TSN

Port Numbers



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, 3411-3418
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

Summary

In this section we have discussed the following:

- ✓ARP, IP, TCP, UDP and SCTP working and frame formats