Computer Networks

BCST -502 BCSP- 502

B.Tech (CSE) 5th Semester

Course Instructor: Dr Bishwajeet Pandey

New 2020 Syllabus

Unit –I

Computer Network: Definitions, goals, components, Architecture, Classifications & Types.Layered Architecture: Protocol hierarchy, Design Issues, Interfaces and Services, ConnectionOriented & Connectionless Services, Service primitives, Design issues & its functionality. ISOOSI Reference Model: Principle, Model, Descriptions of various layers and its comparison with TCP/IP. Principals of physical layer: Media, Bandwidth, Data rate and Modulations

Unit-II

Data Link Layer: Need, Services Provided, Framing, Flow Control, Error control. Data Link Layer Protocol: Elementary &Sliding Window protocol: 1-bit, Go-Back-N, Selective Repeat, Hybrid ARQ. Protocol verification: Finite State Machine Models & Petri net models. ARP/RARP/GARP

Unit-III

MAC Sub layer: MAC Addressing, Binary Exponential Back-off (BEB) Algorithm, Distributed Random Access Schemes/Contention Schemes: for Data Services (ALOHA and Slotted- ALOHA), for Local-Area Networks (CSMA, CSMA/CD, CSMA/CA), CollisionFree Protocols: Basic Bit Map, BRAP, Binary Count Down, MLMA Limited Contention Protocols: Adaptive Tree Walk, Performance Measuring Metrics. IEEE Standards 802 series & their variant.

New 2020 Syllabus

Unit-IV

Network Layer: Need, Services Provided, Design issues, Routing algorithms: Least CostRouting algorithm, Dijkstra's algorithm, Bellman-ford algorithm, Hierarchical Routing,Broadcast Routing, Multicast Routing. IP Addresses, Header format, Packet forwarding,Fragmentation and reassembly, ICMP, Comparative study of IPv4 & IPv6

Unit-V

Transport Layer: Design Issues, UDP: Header Format, Per-Segment Checksum, CarryingUnicast/Multicast Real-Time Traffic, TCP: Connection Management, Reliability of DataTransfers, TCP Flow Control, TCP Congestion Control, TCP Header Format, TCP TimerManagement. Application Layer: WWW and HTTP, FTP, SSH, Email (SMTP, MIME, IMAP), DNS, Network Management (SNMP).

About Course Instructor

- PhD from Gran Sasso Science Institute, Italy
- PhD Supervisor Prof Paolo Prinetto from Politecnico Di Torino, World Rank 13 in Electrical Engineering
- MTech from Indian Institute of Information Technology, Gwalior
- Scopus Profile: https://www.scopus.com/authid/detail.uri?authorId=57203239026
- Google Scholar: https://scholar.google.com/citations?user=UZ_8yAMAAAAJ&hl=hi
- Contact: gyancity@gyancity.com, +91-7428640820 (For help in this Subject @ BIAS and Guidance for future MS from Europe and USA after BIAS)

About Course Outline

- UNIT 1:
    - Theory Lecture No 1-4, Lecture 29
    - Lab on Vivado: Lecture 9-11
- UNIT 2: Theory Lecture No 5-8
- UNIT 3: Theory Lecture No 14-18
- UNIT 4:
    - Theory Lecture No 12-13, 19-21, 36
    - Lab on Packet Tracer and C: Lecture 24-28
- UNIT 5: Theory Lecture No 30-35
- Student Assignment Presentation: 22-23
- Lecture No 37-42: Discuss Previous Year Question of UTU

OUTLINE OF LECTURE 31

- Transport Layer:
  - Design Issues
- UDP:
  - Header Format,
  - Per-Segment Checksum,
  - Carrying Unicast/Multicast Real-Time Traffic,

Design Issues with Transport Layer

- Accepting data from Session layer, split it into segments and send to the network layer.
- Ensure correct delivery of data with efficiency.
- Isolate upper layers from the technological changes.
- Error **control** and flow **control**.

What are the responsibilities of transport layer?

- The main **role** of the **transport layer** is to provide the communication services directly to the application processes running on different hosts.

- The **transport layer** provides a logical communication between application processes running on different hosts.

What are the two main transport layer protocols?

- The **two** most important **protocols** in the **Transport Layer** are Transmission Control **Protocol** (TCP) and User Datagram **Protocol** (UDP).
- TCP provides reliable data delivery service with end-to-end error detection and correction.
- UDP provides low-overhead, connectionless datagram delivery service.

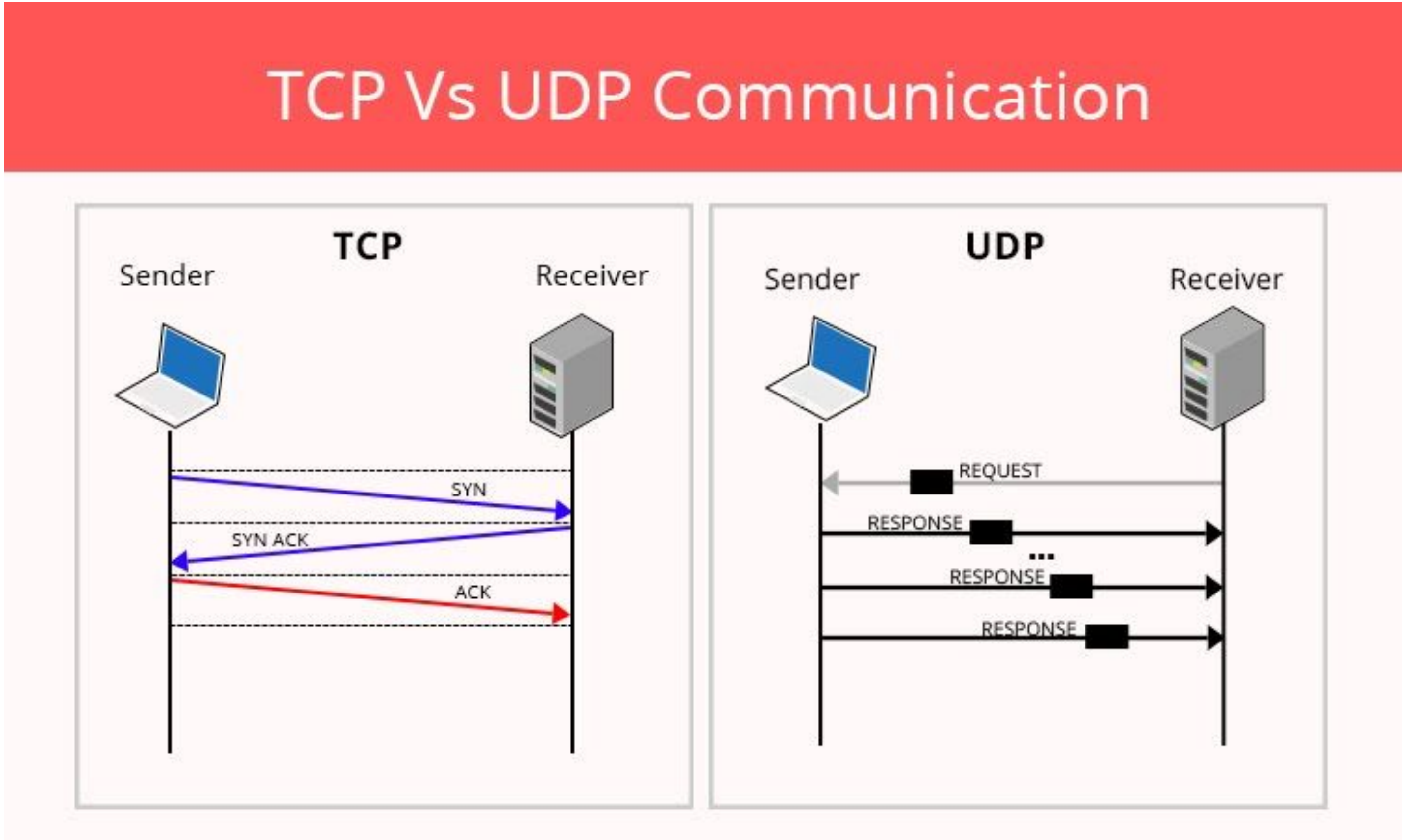
TCP VERSUS UDP

TCP	UDP
Reliable	Unreliable
Connection-oriented	Connectionless
Segment retransmission and flow control through windowing	No windowing or retransmission
Segment sequencing	No sequencing
Acknowledge segments	No acknowledgement

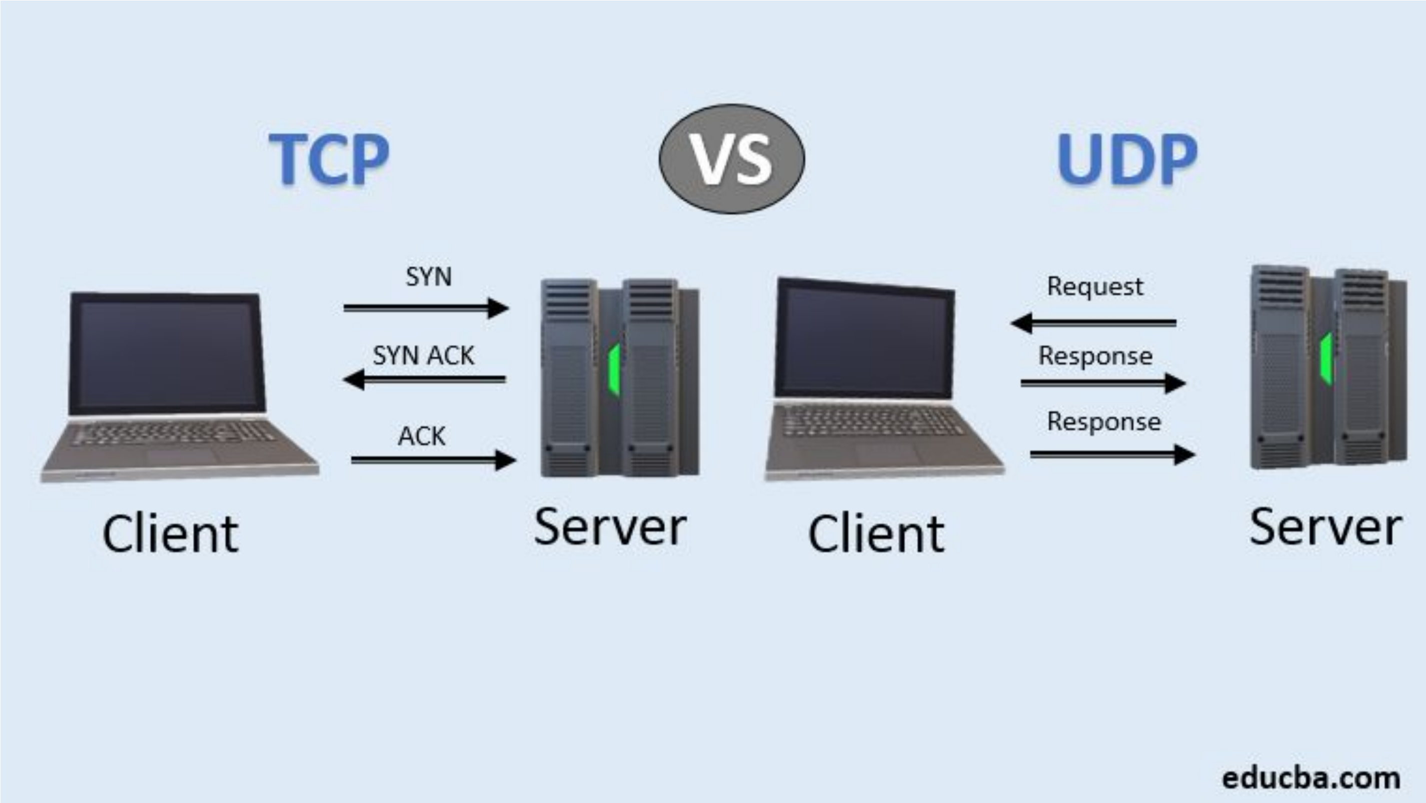
TCP VERSUS UDP

Item	TCP	UDP
Stands For	Transmission Control Protocol	User Datagram Protocol
Protocol	Connection Oriented	Connectionless
Security	Makes Checks For Errors And Reporting	Makes Error Checking But No Reporting
Data Sending	Slower	Faster
Header Size	20 Bytes	8 Bytes
Segments	Acknowledgement	No Acknowledgement
Typical Applications	- Email	- VoIP

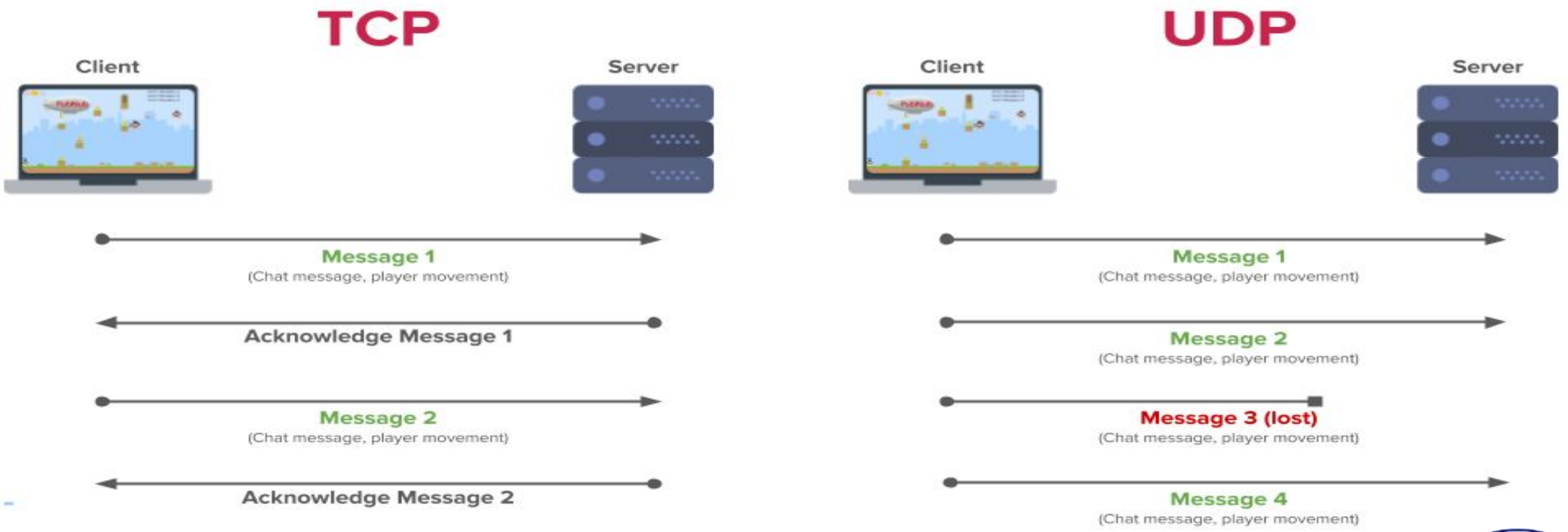
TCP VERSUS UDP



TCP VERSUS UDP



TCP VERSUS UDP



TCP & UDP: Header Format

TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset / Res / Flags		Window Size	
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

UDP Datagram Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

User Datagram **Protocol** (UDP)

- In computer networking, the **User Datagram Protocol** (**UDP**) is one of the core members of the Internet protocol suite.
- The protocol was designed by David P. Reed in 1980 and formally defined in RFC 768.
- With UDP, computer applications can send messages, in this case referred to as *datagrams*, to other hosts on an Internet Protocol (IP) network.
- Prior communications are not required in order to set up communication channels or data paths.

UDP: Header Format

UDP Datagram Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

- The header added to messages by the Transport layer includes more than just the source and destination port numbers.
- UDP header requires less information and overhead in comparison to TCP header.

UDP: Header Format

UDP Datagram Header Format

Bit #	0 7	8 15	16 23	24 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

- A UDP datagram consists of a datagram *header* and a *data* section. The UDP datagram header consists of 4 fields, each of which is 2 bytes (16 bits).The data section follows the header and is the payload data carried for the application.
- The use of the *checksum* and *source port* fields is optional in IPv4. In IPv6 only the *source port* field is optional.

UDP: Header Format

UDP Datagram Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

- **Source port number**
- This field identifies the sender's port, when used, and should be assumed to be the port to reply to if needed. If not used, it should be zero. If the source host is the client, the port number is likely to be an ephemeral port number. If the source host is the server, the port number is likely to be a well-known port number.

Which ports are ephemeral ports?

- Random port numbers (sometimes called ephemeral port numbers) have values greater than **1024**

- **Ephemeral ports**, which are usually dynamic **ports**, are the set of **ports** that every machine by default will have them to make an outbound connection. Well-known **ports**are the defined **port** for a particular application or service. For example, file server service is on **port** 445, HTTPS is 443, HTTP is 80, and RPC is 135.

UDP: Header Format

UDP Datagram Header Format

Bit #	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Length				Header and Data Checksum			

Destination port number

- This field identifies the receiver's port and is required. Similar to source port number, if the client is the destination host then the port number will likely be an ephemeral port number and if the destination host is the server then the port number will likely be a well-known port number.

UDP: Header Format

UDP Datagram Header Format

Bit #	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Length				Header and Data Checksum			

- Length field specifies the length in bytes of the UDP header and UDP data. The minimum length is 8 bytes, the length of the header. The field size sets a theoretical limit of 65,535 bytes (8 byte header + 65,527 bytes of data) for a UDP datagram. However the actual limit for the data length, which is imposed by the underlying IPv4 protocol, is 65,507 bytes (65,535 − 8 byte UDP header − 20 byte IP header).
- Using IPv6 jumbograms it is possible to have UDP datagrams of size greater than 65,535 bytes.RFC 2675 specifies that the length field is set to zero if the length of the UDP header plus UDP data is greater than 65,535.

UDP: Header Format

<table>
<tr><th colspan="5">UDP Datagram Header Format</th></tr>
<tr><td>Bit #</td><td>0 7</td><td>8 15</td><td>16 23</td><td>24 31</td></tr>
<tr><td>0</td><td colspan="2">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td>32</td><td colspan="2">Length</td><td colspan="2">Header and Data Checksum</td></tr>
</table>

- The checksum field may be used for error-checking of the header and data.
- This field is optional in IPv4, and mandatory in IPv6.
- The field carries all-zeros if unused.

User Datagram **Protocol** (UDP): Per-Segment Checksum

- Checksum computation
    - Checksum is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.
    - In other words, all 16-bit words are summed using one's complement arithmetic. Add the 16-bit values up. On each addition, if a carry-out (17th bit) is produced, swing that 17th carry bit around and add it to the least significant bit of the running total. Finally, the sum is then one's complement to yield the value of the UDP checksum field.
    - If the checksum calculation results in the value zero (all 16 bits 0) it should be sent as the one's complement (all 1s) as a zero-value checksum indicates no checksum has been calculated.

User Datagram **Protocol** (UDP): Per-Segment Checksum

- The difference between IPv4 and IPv6 is in the pseudo header used to compute the checksum and the checksum is not optional in IPv6.

- **IPv4 pseudo header**
- **IPv6 pseudo header**

User Datagram **Protocol** (UDP): Per-Segment Checksum

- **IPv4 pseudo header**
- UDP checksum computation is optional for IPv4. If a checksum is not used it should be set to the value zero.

IPv4 pseudo header format

<table>
<tr><th>Offsets</th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0</td><td>0</td><td colspan="32">Source IPv4 Address</td></tr>
<tr><td>4</td><td>32</td><td colspan="32">Destination IPv4 Address</td></tr>
<tr><td>8</td><td>64</td><td colspan="8">Zeroes</td><td colspan="8">Protocol</td><td colspan="16">UDP Length</td></tr>
<tr><td>12</td><td>96</td><td colspan="16">Source Port</td><td colspan="16">Destination Port</td></tr>
<tr><td>16</td><td>128</td><td colspan="16">Length</td><td colspan="16">Checksum</td></tr>
<tr><td>20</td><td>160+</td><td colspan="32">Data</td></tr>
</table>

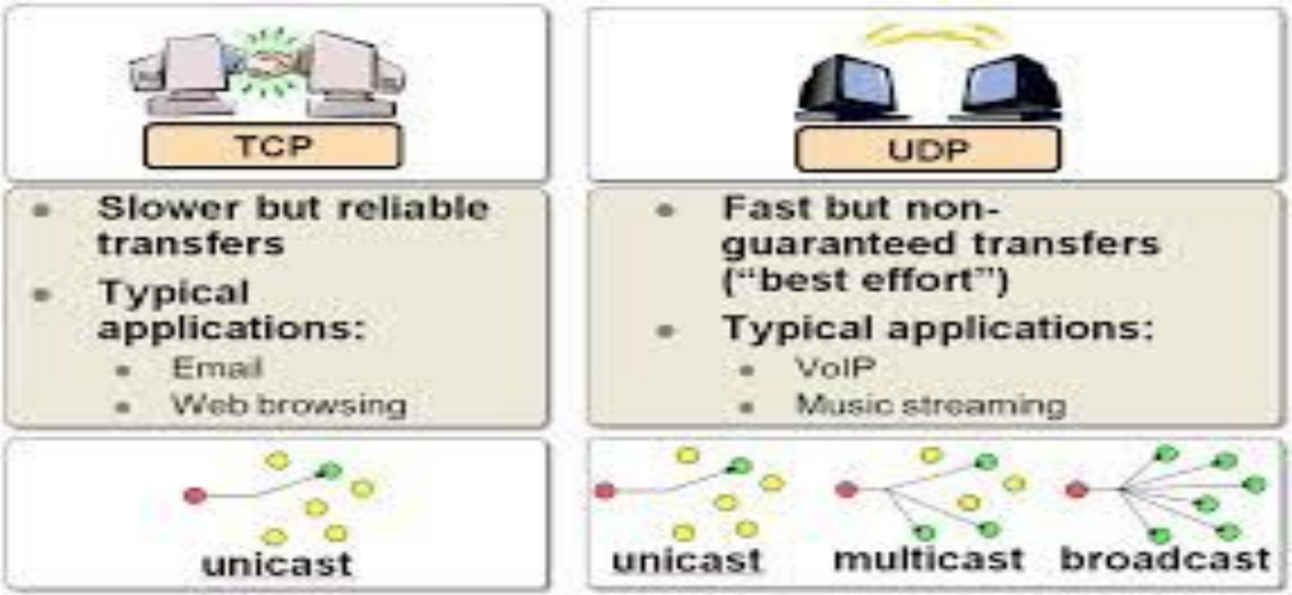
User Datagram **Protocol** (UDP): Per-Segment Checksum

- **IPv6 pseudo header**
- When UDP runs over IPv6, the checksum is mandatory.

IPv6 pseudo header format

<table>
<tr><th>Offsets</th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0</td><td>0</td><td colspan="32" rowspan="4">Source IPv6 Address</td></tr>
<tr><td>4</td><td>32</td></tr>
<tr><td>8</td><td>64</td></tr>
<tr><td>12</td><td>96</td></tr>
<tr><td>16</td><td>128</td><td colspan="32" rowspan="4">Destination IPv6 Address</td></tr>
<tr><td>20</td><td>160</td></tr>
<tr><td>24</td><td>192</td></tr>
<tr><td>28</td><td>224</td></tr>
<tr><td>32</td><td>256</td><td colspan="32">UDP Length</td></tr>
<tr><td>36</td><td>288</td><td colspan="24">Zeroes</td><td colspan="8">Next Header = Protocol[10]</td></tr>
<tr><td>40</td><td>320</td><td colspan="16">Source Port</td><td colspan="16">Destination Port</td></tr>
<tr><td>44</td><td>352</td><td colspan="16">Length</td><td colspan="16">Checksum</td></tr>
<tr><td>48</td><td>384+</td><td colspan="32">Data</td></tr>
</table>

User Datagram **Protocol** (UDP): Carrying Unicast/Multicast Real-Time Traffic



Is multicast TCP or UDP?

- Since **TCP** supports only the unicast mode, **multicast** applications must use the **UDP** transport protocol.

- Unlike broadcast transmission (which is used on some local area networks), **multicast** clients receive a stream of packets only if they have previously elect to do so (by joining the specific **multicast** group address).