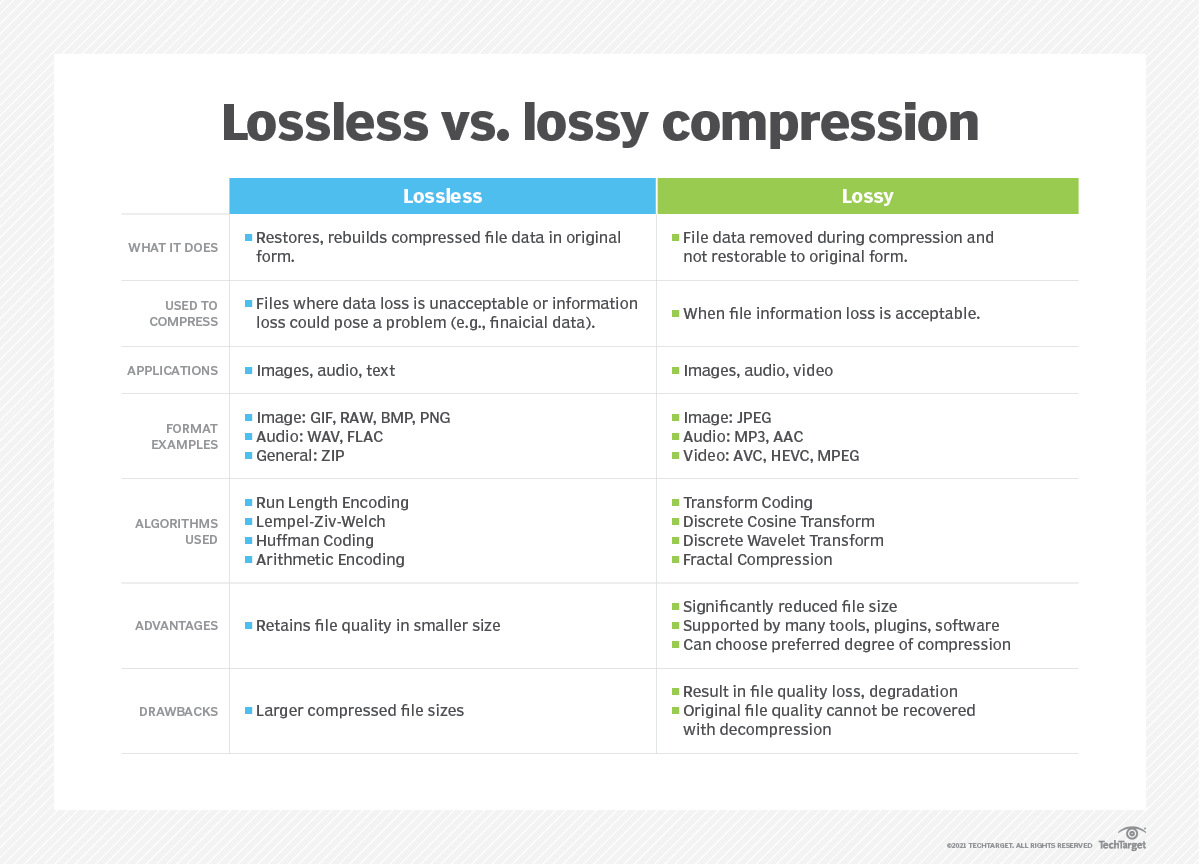
a) Compare IPv4 vs IPv6

|  |  |  |
| --- | --- | --- |
| Feature | IPv4 | IPv6 |
| Address Length | 32-bit | 128-bit |
| Address Format | Decimal (e.g., 192.168.0.1) | Hexadecimal (e.g., 2001:0db8::1) |
| Number of Addresses | ~4.3 billion | ~3.4×10^38 |
| NAT Requirement | Yes | No |
| Security | Optional (IPSec) | Mandatory (IPSec) |

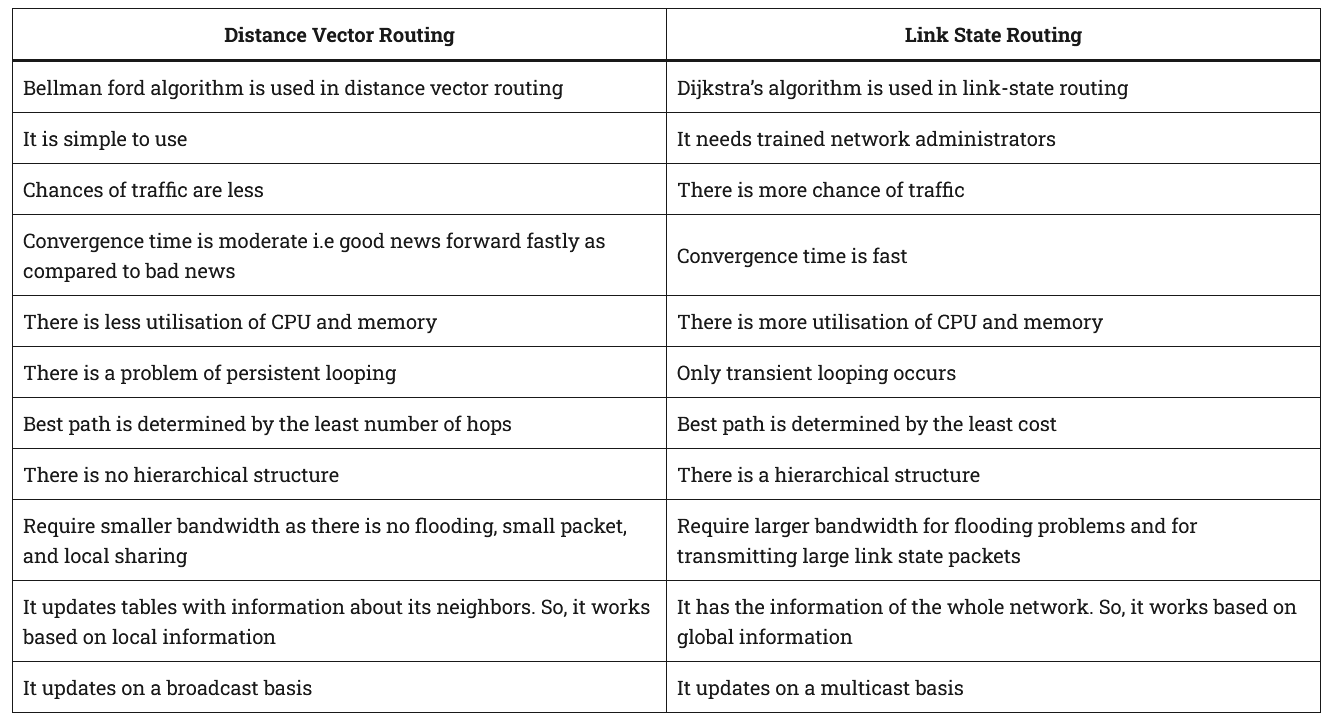
b) Compare Lossy compression vs Lossless compression

|  |  |  |
| --- | --- | --- |
| Feature | Lossy Compression | Lossless Compression |
| Data Integrity | Not preserved | Preserved |
| Compression Ratio | Higher | Lower |
| Use Cases | Images, audio, video | Text, executable files, PNG images |
| File Size | Smaller | Larger |
| Reversibility | Irreversible | Reversible |



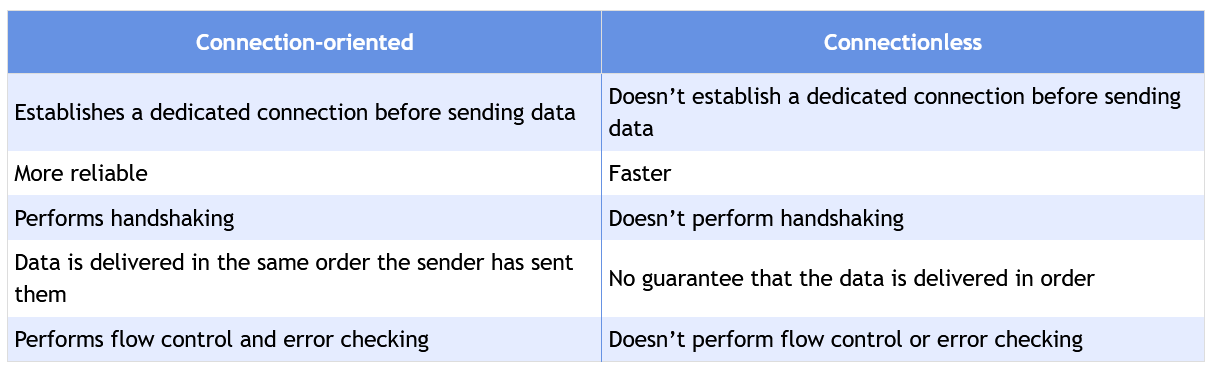
c) Distance Vector vs Link-State Routing

|  |  |  |
| --- | --- | --- |
| Feature | Distance Vector Routing | Link-State Routing |
| Information Shared | Entire routing table | Topology information |
| Algorithm Used | Bellman-Ford | Dijkstra’s algorithm |
| Convergence Time | Slower | Faster |
| Accuracy | Less accurate | More accurate |
| Scalability | Lower | Higher |



d) Connection-oriented vs Connectionless Protocols

|  |  |  |
| --- | --- | --- |
| Feature | Connection-Oriented (e.g., TCP) | Connectionless (e.g., UDP) |
| Setup Required | Yes | No |
| Reliability | Reliable | Unreliable |
| Data Ordering | Ordered | Unordered |
| Speed | Slower | Faster |
| Use Cases | File transfer, web browsing | Streaming, VoIP |



**2. Short Answer/Definition Questions**

**What is NAT?**

Network Address Translation (NAT) is a technique used in networking to map private IP addresses to a public IP address before transmitting data over the internet. NAT is commonly implemented in routers and firewalls to allow multiple devices on a local network to share a single public IP address. This helps conserve IPv4 addresses and enhances security by hiding internal network structures.

**Types of NAT:**

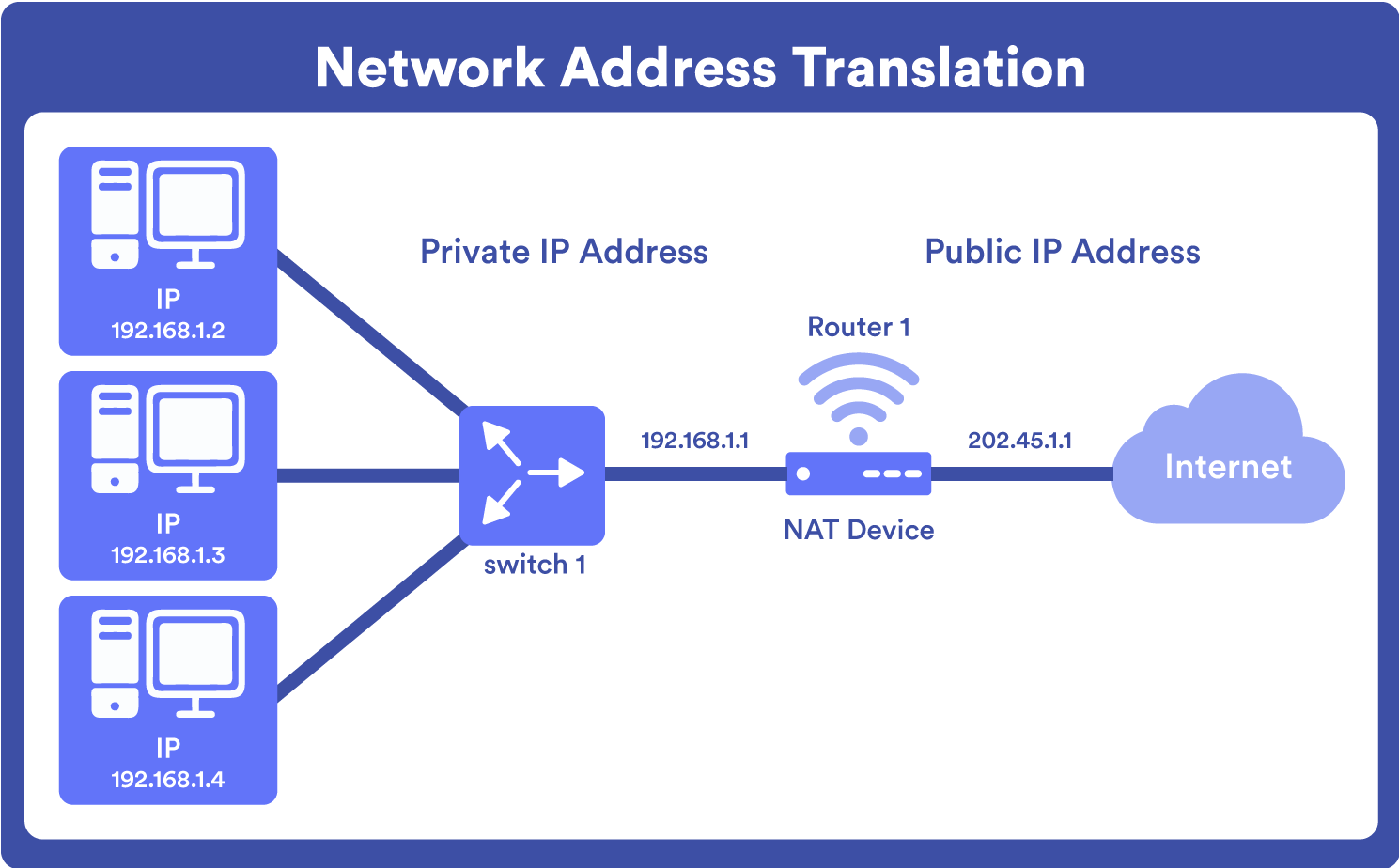
1. **Static NAT (1:1 Mapping):**
   * A private IP is permanently mapped to a public IP.
   * Used for hosting servers (e.g., web servers) inside a private network.
2. **Dynamic NAT (Pool-Based):**
   * A pool of public IPs is shared among private devices.
   * The router assigns a public IP from the pool temporarily.
3. **PAT (Port Address Translation / NAT Overload):**
   * Multiple private IPs share a single public IP using unique port numbers.
   * Most common in home routers.

**Advantages of NAT:**

* **Conserves IPv4 Addresses:** Reduces the need for unique public IPs.
* **Security:** Hides internal network topology from external threats.
* **Flexibility:** Simplifies network management (e.g., changing ISPs without reconfiguring internal IPs).

**Disadvantages of NAT:**

* **Breaks End-to-End Connectivity:** Some protocols (e.g., IPsec, FTP) require additional NAT traversal techniques.
* **Performance Overhead:** Translation processes can introduce latency.
* **Complexity:** Troubleshooting becomes harder due to address masking.



However, NAT can complicate peer-to-peer applications and requires additional configuration for services like online gaming or video conferencing. IPv6 adoption reduces reliance on NAT by providing abundant unique addresses, but NAT remains crucial for IPv4 networks.

**Explain Congestion Control**

Congestion control refers to techniques used in networking to prevent network

congestion, which occurs when too much data is sent over a network, causing

delays and packet loss. It involves controlling the rate at which data is sent to

avoid overwhelming the network

Key Mechanisms:

Flow Control: Manages how much data a sender can send before waiting for

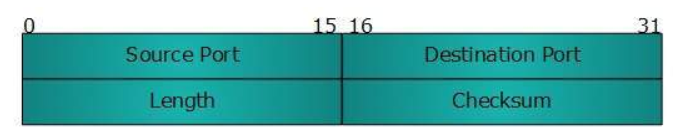
acknowledgment.

Congestion Window: TCP uses this to control the amount of data in transit.

**Describe the UDP Header Format**

UDP header contains four main parameters:

* **Source Port**  - This 16 bits information is used to identify the source port of the packet.
* **Destination Port**  - This 16 bits information, is used identify application level service on destination machine.
* **Length**  - Length field specifies the entire length of UDP packet (including header). It is 16-bits field and minimum value is 8-byte, i.e. the size of UDP header itself.
* **Checksum**  - This field stores the checksum value generated by the sender before sending. IPv4 has this field as optional so when checksum field does not contain any value it is made 0 and all its bits are set to zero.



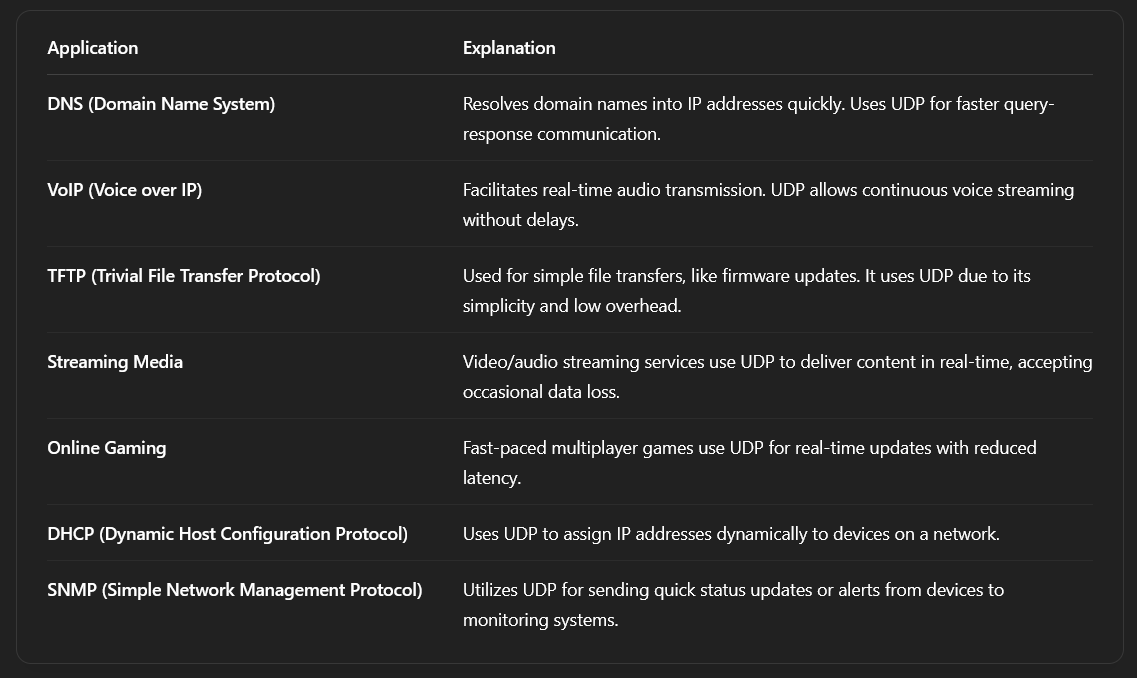
**What is the Leaky Bucket Algorithm?**

The Leaky Bucket algorithm shapes traffic by outputting data at a fixed rate, regardless of input bursts. Imagine a bucket with a small hole: water (data) leaks at a constant rate, and excess spills over (packet drops). This method smooths traffic spikes, preventing network congestion.

Unlike the **Token Bucket**, which permits bursts if tokens are available, the Leaky Bucket strictly enforces a steady rate. It’s used in QoS (Quality of Service) to regulate bandwidth and in ISP throttling. While effective for smoothing traffic, it can introduce delays for bursty applications like video streaming.



**List and Explain UDP Applications**



**What is RPC?**

A Remote Procedure Call (RPC) is a software communication protocol that one program uses to request a service from another program located on a different computer and network, without having to understand the network's details. Specifically, RPC is used to call other processes on remote systems as if the process were a local system. A procedure call is also sometimes known as a function call or a subroutine call.

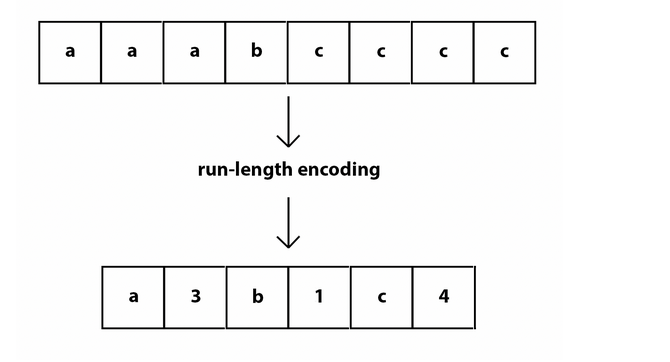
**Types of RPC**

* **Synchronous.** This is the standard method of RPC. The client makes a call and waits for a reply from the server.
* **Nonblocking.** The client makes a call, but instead of waiting for a reply, it continues with its own processing.
* **Batch-mode.** A client sends multiple nonblocking calls in a group.
* **Broadcast.** A client sends a message to multiple servers and then receives all the resulting replies.
* **Callback.** A client makes a nonblocking client-server call.

**Explain RLE**

RLE (Run-Length Encoding) compresses data by replacing repeated sequences with (count, value) pairs. For example, "AAAABBBCC" becomes "4A3B2C". It’s efficient for simple graphics (TIFF, BMP) or text with long repeats.

However, RLE performs poorly on non-repetitive data (e.g., encrypted files). Advanced variants use escape characters for mixed sequences, but modern formats like PNG prefer more sophisticated algorithms (e.g., DEFLATE).



**What is DNS?**

DNS (Domain Name System) translates human-readable domain names (e.g., google.com) into machine-readable IP addresses. The resolution process involves recursive queries to resolvers, root servers, TLD servers, and authoritative servers, with caching at each step to improve efficiency.

DNS records include **A** (IPv4), **AAAA** (IPv6), **MX** (mail servers), and **CNAME** (aliases). Security extensions like **DNSSEC** prevent spoofing by digitally signing responses.

**3. Protocol-Based Questions**

**FTP**

FTP (File Transfer Protocol) facilitates file exchanges between clients and servers. It operates on ports **21 (control)** and **20 (data)**, with two modes: **Active** (server initiates data connection) and **Passive** (client initiates, better for firewalls). Common commands include **USER**, **PASS**, **RETR** (download), and **STOR** (upload).

Security concerns led to encrypted alternatives: **FTPS** (FTP over SSL/TLS) and **SFTP** (SSH File Transfer Protocol, a different protocol). FTP lacks modern features like efficient partial transfers, prompting shifts to HTTP-based methods.

**SNMP**

Simple Network Management Protocol (SNMP) is an Internet Standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behavior. Devices that typically support SNMP include cable modems, routers, network switches, servers, workstations, printers, and more

**HTTP**

HTTP (Hypertext Transfer Protocol) underpins web communication via request-response cycles. Key methods include **GET** (retrieve resources), **POST** (submit data), **PUT** (update), and **DELETE**. Stateless by design, it uses **cookies** and sessions for continuity.

**HTTPS** adds TLS encryption, preventing eavesdropping. Modern versions (**HTTP/2**, **HTTP/3**) improve performance via multiplexing and UDP-based QUIC. Status codes like **200 (OK)**, **404 (Not Found)**, and **500 (Server Error)** convey request outcomes.

**OSPF**

OSPF (Open Shortest Path First) is a link-state IGP that uses **Dijkstra’s algorithm** to compute the shortest paths. It divides networks into **smaller networks** for scalability. It calculates the smallest path that the data packets will need to travel or hop to reach the destination or receiver.

**4. Advanced Networking Questions**

**IPv6 Header vs IPv4**

The IPv6 header simplifies routing with a fixed 40-byte structure, eliminating IPv4’s variable-length options. Key fields include **Version**, **Traffic Class** (prioritization), **Flow Label** (QoS), and **Hop Limit** (TTL replacement). IPv6 moves fragmentation to extension headers and removes the checksum (relying on higher layers).

Extension headers enable features like routing, security (IPsec), and mobility without bloating the base header. IPv6’s design improves performance and scalability, though adoption still lags behind IPv4 due to legacy infrastructure.

|  |  |  |
| --- | --- | --- |
| Feature | IPv4 | IPv6 |
| Address Length | 32-bit | 128-bit |
| Address Format | Decimal (e.g., 192.168.0.1) | Hexadecimal (e.g., 2001:0db8::1) |
| Number of Addresses | ~4.3 billion | ~3.4×10^38 |
| NAT Requirement | Yes | No |
| Security | Optional (IPSec) | Mandatory (IPSec) |

**TCP Header Structure**

The TCP header (20–60 bytes) includes **source/destination ports**, **sequence/acknowledgment numbers** for reliability, and **flags** (**SYN**, **ACK**, **FIN**, etc.) for connection control. The **window size** field implements flow control, while **options** support features like **MSS (Maximum Segment Size)** and **SACK (Selective ACK)**.

TCP’s checksum ensures data integrity, and the **urgent pointer** flags high-priority data. Header length varies due to options, with timestamps and window scaling enhancing performance on modern networks.

**GIF & JPEG Compression**

**Explain GIF Compression and its Specifications**

**GIF (Graphics Interchange Format)** uses **LZW (Lempel-Ziv-Welch)** compression, which is a **lossless** compression technique. This means the image data is compressed without losing any quality.

**Key Features and Specifications:**

* **Compression Type**: Lossless (LZW algorithm)
* **Color Support**: Up to **256 colors** (8-bit color depth)
* **Transparency**: Supports **1-bit transparency** (one color can be transparent)
* **Animation**: Can store multiple frames to create **simple animations**
* **Best Suited For**: Logos, icons, line art, and simple graphics with limited colors

**Limitations:**

* Not suitable for **high-resolution images** or photographs due to limited color palette.

**JPEG Compression (Joint Photographic Experts Group)**

**Overview**

JPEG is a lossy compression method designed primarily for photographic images and complex graphics with smooth color transitions.

**Compression Process**

1. **Color Space Conversion**:
   * RGB to YCbCr conversion (separates luminance from chrominance)
   * Human eye is more sensitive to brightness (Y) than color (Cb, Cr)
2. **Chrominance Downsampling**:
   * Typically 4:2:0 subsampling (reduces color resolution by 2x horizontally and vertically)
   * Discards 75% of color information with minimal perceptual impact
3. **Discrete Cosine Transform (DCT)**:
   * Image divided into 8x8 pixel blocks
   * Each block transformed from spatial domain to frequency domain
   * Concentrates visual information into fewer coefficients
4. **Quantization**:
   * Frequency coefficients divided by quantization values
   * Higher frequencies (less visible details) get more aggressive quantization
   * This is where most information loss occurs
5. **Entropy Coding**:
   * Zig-zag pattern reorders coefficients from low to high frequency
   * Run-length encoding of zeros
   * Huffman coding for final compression

**Characteristics**

* Lossy compression (quality decreases with each save)
* No transparency support
* No animation support
* Typical compression ratios: 10:1 to 20:1 with minimal quality loss
* 24-bit color (16.7 million colors)
* Best for: Photographs, realistic images with smooth gradients

**TCP Congestion Control**

TCP congestion control mechanism is an algorithmic approach to manage the flow of data in a TCP network. The congestion control mechanism regulates the rate at which data is transmitted, while maintaining a balance between network utilization and reliability.

The mechanism works by adjusting the sending rate based on various feedback mechanisms during data transmission. It ensures that the available bandwidth is shared fairly among all the users without causing network congestion.

Some best practices for managing TCP timers include:

* Regularly monitoring timer values using built in networking tools or third party software
* Analyzing network traffic patterns to identify potential issues such as packet loss or congestion
* Tuning retransmission and persistence timers based on latency, bandwidth, packet loss rates, etc.
* Maintaining consistency in timer configurations across all devices in a particular network
* Making gradual adjustments rather than sudden changes to avoid negative impacts on network performance

**TCP Timers**

TCP timers are mechanisms used by the protocol to manage and control various aspects of the data transmission process. Essentially, these timers are implemented by a device's operating system and are used to track different stages of a TCP connection. They ensure that packets are promptly delivered between devices and help avoid issues such as packet loss or congestion.

**Retransmission Timer**

The retransmission timer is a critical component in providing reliable data transfer over the network. Its primary function is to ensure that packets reach their destination by resending packets that may have been lost or corrupted during transmission.

When a packet is sent over the network, an acknowledgement (ACK) is expected from the receiver. If no ACK is received within a specified time frame set by the retransmission timer, the sender assumes that the packet has been lost and will resend it.

**Persistence Timer**

The Persistence Timer (PT) is another important aspect of TCP timers that helps manage network congestion. When there are too many unacknowledged packets on the network, it can lead to congestion and eventually result in packet loss or delay.

The PT prevents this by temporarily holding back new transmissions until previously sent packets are acknowledged. The PT works by periodically sending probes to check whether there are any unacknowledged packets on the network.

If there is no response from these probes before a specified time period elapses (set by PT), then new transmission attempts will be held back until existing transmissions have been acknowledged. PTs play a crucial role in managing congestion on networks with significant traffic loads.

**Keepalive Timer**

The Keepalive Timer (KT) is used to detect inactive connections. When a connection is idle for an extended period, it can be challenging to know whether the session has been terminated or not. The KT solves this by sending probes at regular intervals to check the status of the connection.

If there is no response from these probes before a specified time period elapses (set by KT), then the connection is assumed to be dead and will be dropped. The KT ensures that resources are not wasted on inactive sessions, freeing up network resources for active ones.