Introduction to Data Communications and Networks *(CYB 204)*

Presentation and Session Layers

Egena Onu, Ph.D.
Department of Computer Science
Bingham University
Karu.

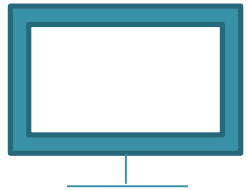
Icons

Router

Switch

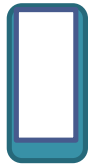
Desktop Computer

Server



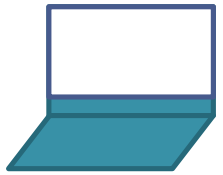
Monitor

Internet/Cloud



Mobile Device

Clock



Laptop

Mainframe

Presentation Layer

- The presentation layer is concerned with preserving the meaning of information sent across a network.
- The presentation layer may represent (encode) the data in various ways (e.g., data compression, or encryption), but the receiving peer will convert the encoding back into its original meaning.
- The presentation layer concerns itself with the following issues:
  - Data Formatting
  - Data Compression
  - Security and Privacy

Presentation Layer

- Data Formatting
  - The presentation layer converts the complex data structures used by an application – strings, integers, structures, etc., - into a byte stream transmitted across the network.
  - The layer represents information in such a way that communicating peers agree to the format of the data being exchanged. E.g., How many bits does an integer contain?, ASCII or EBCDIC character set?

Session Layer

- This layer is primarily concerned with coordinating applications as they interact on different hosts.
- Support the dialog between cooperating application programs
- The session layer offers provisions for efficient data transfer.
- The session layer decides when to turn communication on and off between two computer
- Provides duplex, half-duplex, or simplex communications between devices.

Session Layer

- The Session Layer provides services that allow to establish/manage/terminate a session-connection, to support orderly data exchange, to organize and to synchronize the dialogue and to release the connection in an orderly manner.

- The session's layer objective is to hide the possible failures of transport-level connections to the upper layer higher.

Session Layer

- Sessions offer various services, including:
  - Dialog Control
    - Keeping track of whose turn it is to transmit
  - Token Management
    - Preventing two parties from attempting the same critical operation simultaneously, and
  - Synchronization
    - Checkpointing long transmissions to allow them to pick up from where they left off in the event of a crash and subsequent recovery.
- In case of a connection loss this layer try to recover the connection.

Session Layer

- Session layer provides a name space that is used to tie together the potentially different transport streams that are part of a single application.
- For example, it might manage an audio stream and a video stream that are being combined in a teleconferencing application. Long story short, principal task of the session layer is to connect two processes together into a session.
- The session layer whose overall function is to ensure the end to end integrity of the applications that are being supported. Manages who can transmit data at a certain time and for how long.

Session Layer

- The Session layer should provide the following functionality to the Presentation layer.
  - Session-connection establishment
  - Session-connection release
  - Normal data transfer
  - Token management
  - Session-connection synchronization
  - Exception reporting
  - Activity management

Session Layer

- Session-connection Establishment
  - The Session Layer should enable two presentation-entities to establish a session-connection between them.
  - The presentation-entities are identified by session-addresses, and both sides negotiate session parameters.
- Session-Connection Release
  - The session-connection release service allows presentation-entities to release a session-connection without loss of data.
- Normal data Transfer
  - The ability to send data between presentation-entities.
- Token management
  - Allows the presentation-entities to control explicitly whose turn it is to carry out certain control functions.

Session Layer

- Session-Connection Synchronization
  - The presentation-entities should be able to define and identify synchronization points and to reset the session-connection to a defined state and agree on a resynchronization point.
  - The Session Layer is not responsible for any associated checkpointing or commitment action associated with synchronization.
- Exception Reporting
  - The Session Layer should provide exception reporting to inform the presentation-entities of exceptional situations.
- Activity Management
  - The user of the Session Layer should be able to divide logical pieces of work into activities.
  - A session could span several activities, and these activities can be interrupted and then resumed.

Session Layer

- Session Layer Functionality Includes
  - Virtual connection between application entities
  - Synchronization of data flow
  - Creation of dialog units
  - Connection parameter negotiations
  - Partitioning of services into functional groups
  - Acknowledgements of data received during a session
  - Retransmission of data if it is not received by a device

Session Layer

- Session Services
- The session layer must provide
  - For endpoints
    - Creation of endpoints.
    - Destruction of endpoints.
    - Provide one or several default endpoints which can be used by e.g. clients.
    - Rebinding of endpoints, e.g. change interface.
    - Load endpoints from disk.
    - Save endpoints to disk.

Session Layer

- Session Services
  - For services
    - Creation of services.
    - Destruction of services.
    - Accepting a client on a service.
  - For sessions
    - Connect to a service.
    - Suspend a session.
    - Resume a session.
    - Close a session.

Session Layer

- Session Layer Functions
  - Session takes multimedia data objects such as video, data, voice, image from multiple users and creates a conversational and collaborative environment for the users
  - Session Applications Development Environment, ADE, allows developers a common interface to "feed" multimedia data objects into the network and ensure that they are coordinated.
  - Whereas TCP (Transmission Control Protocol) ensures a one to one connection, Session ensures a many to many to many connection; many users to many users with many multimedia objects

Session Layer

- Session Layer Functions
  - Flow Management
    - Flow management is a high level session function which assures that all of the users and all of the elements are transported in a manner and quality as specified.
    - It sits atop TCP and does what TCP does for multimedia objects in the fully distributed environment controlled by the session layer.
    - Media Selection:
      - This control elements is set to ensure quality of the media element being sent.

Session Layer

- Session Layer Functions
  - User Management
    - User management is simply the session function which controls the user entry, control, and exit from a session.

    - The following details the key functions covered by user management.
      - Listing: A listing of all users by IP address. This is a dynamically reconfigurable list.
      - Change List: To minimize user identification the list is updated by adds and drops.
      - Verification and Authentication: Each user is verified and authenticated.
      - Security: There is a security level on a per users and per group basis.
      - Priority: Each user has a priority level.
      - Type: This specifies the type of user

Session Layer

- Session Layer Functions
  - Layer Controls
    - Layer control is the function which may be considered a bit different for this session layer protocol. It also provides data up to the service layer stack for the management and control of the network.

    - These controls are TCP Control , IP Control and MAC Layer Control
      - TCP Control:
        - This controls flow to Transmission Control Protocol (TCP) elements to manage delays and throughput which may be media dependent.
          a. Push
          b. Urgent
          c. Flags
          d. Delay Control

Session Layer

- Session Layer Functions
  - Layer Control
    - ☐2. IP Control:
      - This is the IP element which also controls router features and functions such as QoS (Quality of Service) and routing tables.
        a. Header Compression
        b. MPLS (Multiprotocol Label Switching) Control
        c. Router Table Control
    - MAC Layer Control:
      - The MAC layers can be controlled via Session layer such as RTS/CTS suppression.
        a. RTS/CTS (Request to Send / Clear to Send)
        b. Others

Session Layer

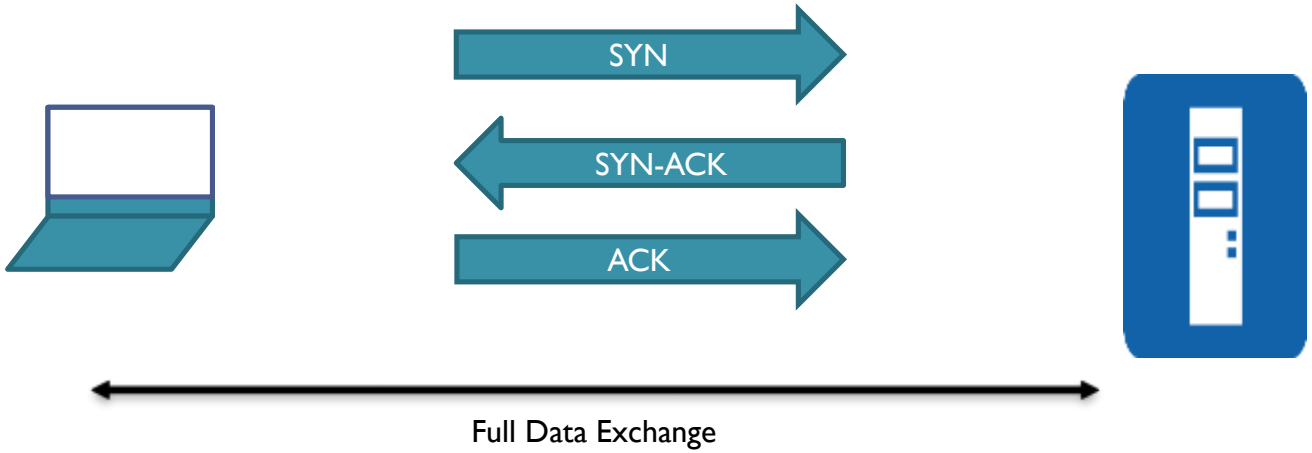
- Session Layer Functions
  - Resource Management
    - Resource management control is a higher layer management control function. It manages three key elements; media, router and events. They are described as following:
      1. Media Flow Control: Multimedia flow control at the session layer for multiple media elements can be managed via this mechanism.
      2. Router Management: The ongoing router management can be controlled via Session layer control elements.
      3. Event Management: The monitoring of performance, isolation of problems, and restoration of service is a key element of the session service.
    - The session layer functions are key to supporting the overall needs of a multimedia communications environment.

Session Layer

- Session Layer Functions
  - Creating a Connection
    - Session/communication session between an APPLICATION in one computer and another APPLICATION in another computer
    - THREE-WAY-HANDSHAKE:
      - A method used by the session layer to establish and end connections using the following steps:
        - Sender sends SYN message to request a session to the receiver
        - Receiver replies by sending ACK message to acknowledge the SYN message sent by the sender, and SYN message to request a session to the sender
        - Sender replies by sending ACK message to acknowledge the SYN message sent by the receiver

Session Layer

- Session Layer Functions
  - Creating a connection

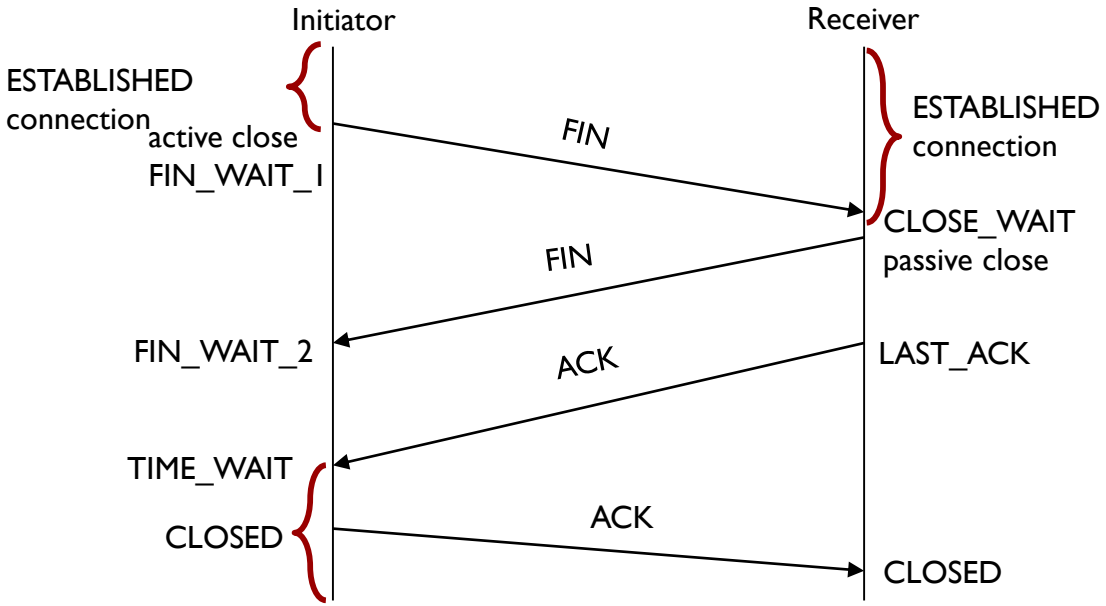


Session Layer

- Session Layer Functions
  - Managing Multiple Sessions
    - A computer can establish multiple sessions with several other computers
      - Session 1: exchanging information over the World Wide Web with www.binghamuni.edu.ng
      - Session 2: exchanging information over the World Wide Web with www.konga.com
      - Session 3: exchanging information over the World Wide Web with www.espn.com
      - Two computers can also establish multiple sessions,
        - Function 1: exchanging information over the World Wide Web;
        - Function 2: exchanging information over the FTP;
        - Function 3: exchanging information over the email

Session Layer

- Session Layer Functions
  - Ending a Session
    - THREE-WAY-HANDSHAKE: a method widely used to establish and end connection
    - Sender sends FIN message to close a session to the receiver
    - Receiver replies by sending ACK message to acknowledge the FIN message sent by the sender, and FIN message to close a session to the sender



    - Sender replies by sending ACK message to acknowledge the FIN message sent by the receiver

Session Layer

- Session Layer Protocols
  - ADSP, AppleTalk Data Stream Protocol
  - ASP, AppleTalk Session Protocol
  - DNA SCP, Digital Network Arcitecture Session Control Protocol
  - H.245, Call Control Protocol for Multimedia Communication
  - ISO-SP, OSI Session Layer Protocol (X.225, ISO 8327)
  - ISNS, Internet Storage Name Service
  - L2F, Layer 2 Forwarding Protocol
  - L2F, Layer 2 Forwarding Protocol
  - L2TP, Layer 2 Tunneling Protocol
  - NetBIOS, Network Basic Input Output System
  - NetBEUI, NetBIOS Enhanced User Interface
  - NFS, Network File System
  - NCP, NetWare Core Protcool

Session Layer

- Session Layer Protocols
  - PAP, Password Authentication Protocol
  - PPTP, Point-to-Point Tunnelling Protocol
  - RPC, Remote Procedure Call Protocol
  - RTCP, Real-time Transport Control Protocol
  - SMB Server Message Block
  - SMPP, Short Message Peer-to-Peer
  - SCP, Session Control Protocol
  - SCP, Secure Copy Protocol
  - SDP, Sockets Direct Protocol
  - SIP, Session Initiation Protocol
  - SOCKS, "SOCKetS" the SOCKS internet protocol
  - SQL, Structured Query Language
  - SSH, Secure Shell
  - ZIP, Zone Information Protocol
  - X Window System

Session Layer

- Session Layer Protocols
  - Apple Talk Protocols
    - The Apple Talk Protocol suite includes the following protocols:
      - AARP AppleTalk Address Resolution Protocol
      - DDP Datagram Delivery Protocol
      - RTMP Routing Table Maintenance Protocol
      - AEP AppleTalk Echo Protocol
      - ATP AppleTalk Transaction Protocol
      - NBP Name-Binding Protocol
      - ZIP Zone Information Protocol
      - ASP AppleTalk Session Protocol
      - PAP Printer Access Protocol
      - ADSP AppleTalk Data Stream Protocol
      - AFP AppleTalk Filing Protocol

Session Layer

- Session Layer Protocols
  - Apple Talk Protocols
    - Apple Computer developed the AppleTalk protocol suite to implement file transfer, printer sharing, and mail service among Apple systems using the LocalTalk interface built into Apple hardware.
    - AppleTalk ports to other network media such as Ethernet by the use of LocalTalk to Ethernet bridges or by Ethernet add-in boards for Apple machines.
    - AppleTalk is a multi-layered protocol providing internetwork routing, transaction and data stream service, naming service, and comprehensive file and print sharing.
    - In addition, many third-party applications exist for the AppleTalk protocols.

Session Layer

- Session Layer Protocols
  - Apple Talk Protocols
    - ASP
      - The AppleTalk Session Protocol (ASP) manages sessions for higher layer protocols.
      - ASP issues a unique session identifier for each logical connection and continuously monitors the status of each connection.
      - It maintains idle sessions by periodically exchanging keep alive frames in order to verify the session status.
    - ADSP
      - The AppleTalk Data Stream Protocol (ADSP) provides a data channel for the hosts. It is a connection-oriented protocol that guarantees in-sequence data delivery with flow control.
    - ZIP
      - The AppleTalk Zone Information Protocol (ZIP) manages the relationship between network numbers and zone names.
      - AppleTalk networks primarily implement ZIP in routers that gather network number information by monitoring RTMP (Real Time Messaging Protocol) frames.

Session Layer

- Session Layer Protocols
  - SCP
    - The Session Control Protocol (SCP) manages logical links for DECnet (DECnet is a suite of network protocols created by Digital Equipment Corporation) connections.
  - PAP
    - Password Authentication Protocol (PAP) provides a simple method for the peer to establish its identity.
    - This is done only upon initial link establishment.

Session Layer

- Session Layer Protocols
  - NetBIOS
    - NetBIOS (Network Basic Input/Output System) provides a communication interface between the application program and the attached medium.
    - It is a file sharing and name resolution protocol and the basis of file sharing with Windows.
    - All communication functions from the physical layer through the session layer are handled by NetBIOS
    - A NetBIOS session is a logical connection between any two names on the network.
    - It is described in IBM - Local Area Network Technical Reference 1990 DA-30/31 Protocol Operating Manual.

Session Layer

- Session Layer Protocols
  - L2F
    - The Layer 2 Forwarding protocol (L2F) permits the tunneling of the link layer of higher layer protocols.
  - L2TP
    - The L2TP (Layer 2 Tunneling Protocol) Protocol is used for integrating multi-protocol dial-up services into existing Internet Service Providers Point of Presence.
    - Used to support virtual private networks (VPNs)

Session Layer

- Session Layer Protocols
  - RTCP
    - The Real-time Transport Control Protocol (RTCP) is based on the periodic transmission of control packets to all participants in the session, using the same distribution mechanism as the data packets.
    - The underlying protocol must provide multiplexing of the data and control packets, for example using separate port numbers with UDP.
  - RTSP
    - The Real-Time Streaming Protocol (RTSP) is an application level protocols for control over the delivery of data with real-time properties.
    - RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video.
    - Sources of data can include both live data feeds and stored clips.
    - This protocol is intended to control multiple data delivery sessions.
    - Both an RTSP server and client can issue requests.
    - Data is carried out-of-band by a different protocol. (There is an exception to this.)

Transport Layer

- The transport layer manages the end-to-end transportation of packets across a network through transparent transfer of data between hosts. It provides end-to-end control and information transfer with the quality of service needed by the application program.
- The layer is the first true end-to-end layer, implemented in all End Systems (ES).
- Its role is to connect application processes running on end hosts as seamlessly as possible, as if the two end applications were connected by a reliable dedicated link, thus making the network "invisible."
- To do this, it must manage several nonidealities of real networks: shared links, data loss and duplication, contention for resources, and variability of delay.

Transport Layer

- Transport Layer Functions and Services
  - The transport layer is located between the network layer and the application layer.
  - The transport layer is responsible for providing services to the application layer; it receives services from the network layer.
  - The services that can be provided by the transport layer are
    1. Process-to-Process Communication
    2. Addressing : Port Numbers
    3. Encapsulation and Decapsulation
    4. Multiplexing and Demultiplexing
    5. Flow Control
    6. Error Control
    7. Congestion Control

Transport Layer

- Transport Layer Functions and Services
  - Process-to-Process Communication
    - The Transport Layer is responsible for delivering data to the appropriate application process on the host computers.
    - This involves multiplexing of data from different application processes, i.e. forming data packets, and adding source and destination port numbers in the header of each Transport Layer PDU.
    - Together with the source and destination IP address, the port numbers constitutes a network socket, i.e. an identification address of the process-to-process communication.

Transport Layer

- Transport Layer Functions and Services
  - Addressing: Port Numbers
    - Ports are the essential ways to address multiple entities in the same location.
    - Using port addressing it is possible to use more than one network-based application at the same time.
    - Three types of Port numbers are used:
      - Well-known ports - These are permanent port numbers. They range between 0 to 1023. These port numbers are used by Server Process.
      - Registered ports - The ports ranging from 1024 to 49,151 are not assigned or controlled.
      - Ephemeral ports (Dynamic Ports) - These are temporary port numbers. They range between 49152–65535. These port numbers are used by Client Process.

Transport Layer

- Transport Layer Functions and Services
  - Encapsulation and Decapsulation
    - To send a message from one process to another, the transport-layer protocol encapsulates and decapsulates messages.
    - Encapsulation happens at the sender site. The transport layer receives the data and adds the transport-layer header.
    - Decapsulation happens at the receiver site. When the message arrives at the destination transport layer, the header is dropped and the transport layer delivers the message to the process running at the application layer.

Transport Layer

- Transport Layer Functions and Services
  - Multiplexing and Demultiplexing
    - Whenever an entity accepts items from more than one source, this is referred to as multiplexing (many to one).
    - Whenever an entity delivers items to more than one source, this is referred to as demultiplexing (one to many).
    - The transport layer at the source performs multiplexing
    - The transport layer at the destination performs demultiplexing

Transport Layer

- Transport Layer Functions and Services
  - Flow Control
    - Flow Control is the process of managing the rate of data transmission between two nodes to prevent a fast sender from overwhelming a slow receiver.
    - It provides a mechanism for the receiver to control the transmission speed, so that the receiving node is not overwhelmed with data from transmitting node.
    - There are two categories of flow control:
      - Stop and wait: send one frame at a time.
      - Sliding window: send several frames at a time.

Transport Layer

- Transport Layer Functions and Services
  - Error Control
    - Error control at the transport layer is responsible for
      1. Detecting and discarding corrupted packets.
      2. Keeping track of lost and discarded packets and resending them.
      3. Recognizing duplicate packets and discarding them.
      4. Buffering out-of-order packets until the missing packets arrive.

    - Error Control involves Error Detection and Error Correction

Transport Layer

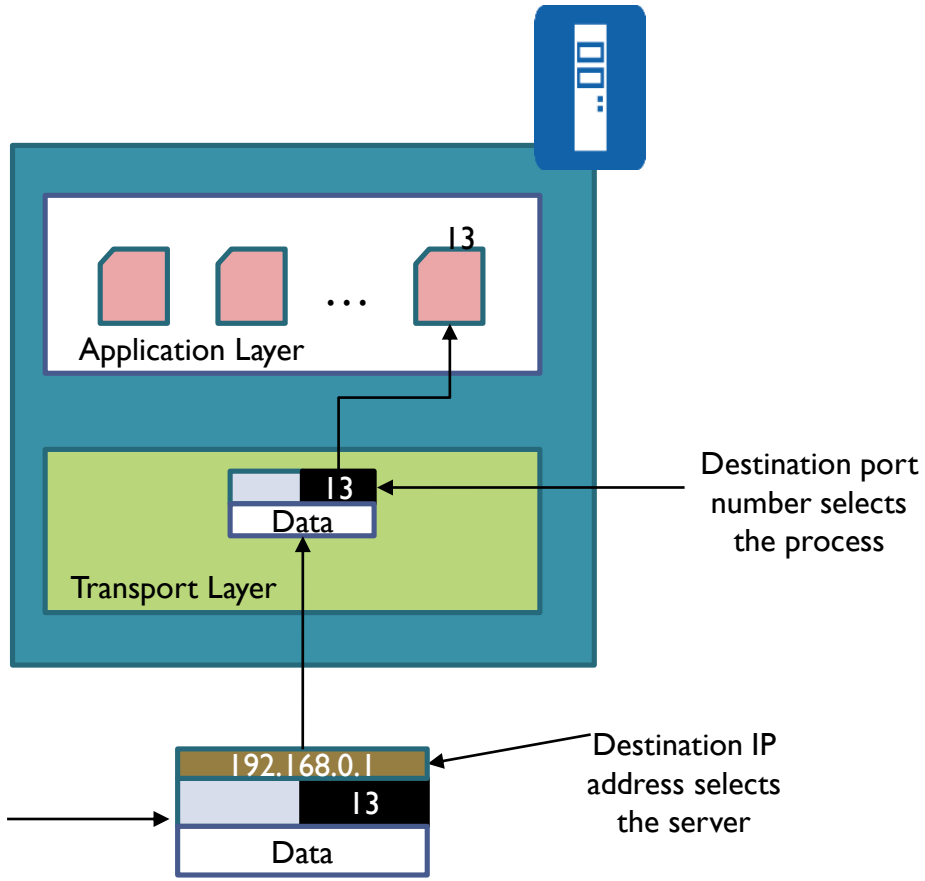
- Transport Layer Functions and Services
  - Congestion Control
    - Congestion in a network may occur if the load on the network (the number of packets sent to the network) is greater than the capacity of the network (the number of packets a network can handle).
    - Congestion control refers to the mechanisms and techniques that control the congestion and keep the load below the capacity.
    - Congestion Control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened
    - Congestion control mechanisms are divided into two categories,
      1. Open loop - prevent the congestion before it happens.
      2. Closed loop - remove the congestion after it happens.

Transport Layer

- Port Numbers
  - A transport-layer protocols usually have several responsibilities. One is to create a process-to-process communication.
  - Processes are programs that run on hosts. It could be either server or client. A process on the local host, called a client, needs services from a process usually on the remote host, called a server.
  - Processes are assigned a unique 16-bit port number on that host.
  - Port numbers provide end-to-end addresses at the transport layer.
  - They also provide multiplexing and demultiplexing at this layer.
  - The port numbers are integers between 0 and 65,535.

Transport Layer

- Port Numbers



- Internet Corporation for Assigned Names and Numbers(ICANN) has divided the port numbers into three ranges:
  - Well-known ports (0 – 1023)
  - Registered (1024 – 49151)
  - Ephemeral ports (Dynamic/Private Ports) (49152 – 65535)

Transport Layer

- Port Numbers
  - Well-Known Ports
    - These are permanent port numbers used by the servers.
    - They range between 0 to 1023.
    - This port number cannot be chosen randomly.
    - These port numbers are universal port numbers for servers.
    - Every client process knows the well-known port number of the corresponding server process.
    - For example, while the daytime client process, a well-known client program, can use an ephemeral (temporary) port number, 52,000, to identify itself, the daytime server process must use the well-known (permanent) port number 13.

Transport Layer

- Port Numbers
  - Well-Known Ports
    - These are permanent por
    - They range between 0 to
    - This port number cannot
    - These port numbers are
    - Every client process knov process.

Port	Protocol	Description
7	Echo	Echoes back a received datagram
11	Users	Active users
13	Daytime	Returns the date and time
17	Quote	Returns a quote of the day
20	FTP-data	FTP
21	FTP-21	FTP
23	TELNET	Terminal Network
80	HTTP	

    - For example, while the daytime client process, a well-known client program, can use an ephemeral (temporary) port number, 52,000, to identify itself, the daytime server process must use the well-known (permanent) port number 13.

Transport Layer

- Port Numbers
  - Ephemeral Ports (Dynamic Ports)
    - The client program defines itself with a port number, called the ephemeral port number.
    - The word ephemeral means "short-lived" and is used because the life of a client is normally short.
    - An ephemeral port number is recommended to be greater than 1023.
    - These port number ranges from 49,152 to 65,535 .
    - They are neither controlled nor registered. They can be used as temporary or private port numbers.
  - Registered Ports
    - The ports ranging from 1024 to 49,151 are not assigned or controlled.

Transport Layer

- Transport Layer Protocols
  - Three protocols are associated with the Transport layer.
  - They are
    1. UDP –User Datagram Protocol
    2. TCP – Transmission Control Protocol
    3. SCTP - Stream Control Transmission Protocol

  - Each protocol provides a different type of service and should be used appropriately.

  - UDP - UDP is an unreliable connectionless transport-layer protocol used for its simplicity and efficiency in applications where error control can be provided by the application-layer process.

  - TCP - TCP is a reliable connection-oriented protocol that can be used in any application where reliability is important.

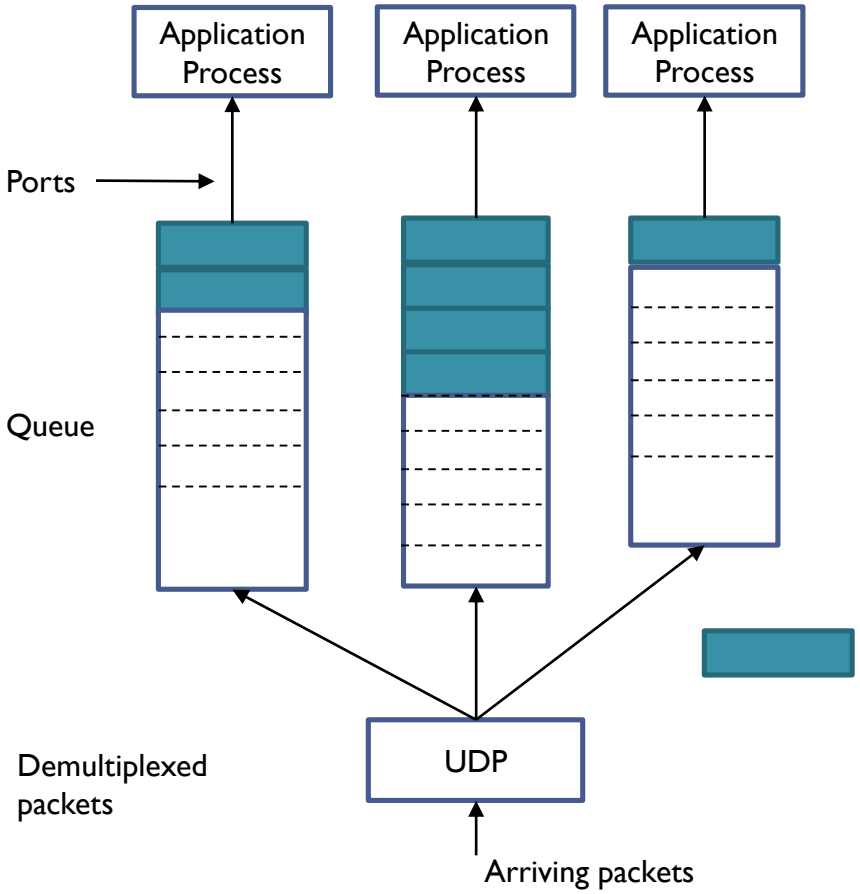
  - SCTP - SCTP is a new transport-layer protocol designed to combine some features of UDP and TCP in an effort to create a better protocol for multimedia communication.

Transport Layer

- Transport Layer Protocols
  - User Datagram Protocol (UDP)
    - User Datagram Protocol (UDP) is a connectionless, unreliable transport protocol.
    - UDP adds process-to-process communication to best-effort service provided by IP.
    - UDP is a very simple protocol using a minimum of overhead.
    - UDP is a simple demultiplexer, which allows multiple processes on each host to communicate.
    - UDP does not provide flow control , reliable or ordered delivery.
    - UDP can be used to send small message where reliability is not expected.
    - Sending a small message using UDP takes much less interaction between the sender and receiver.
    - UDP allow processes to indirectly identify each other using an abstract locator called port or mailbox

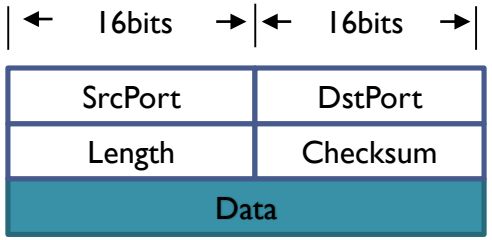
Transport Layer

- Transport Layer Protocols
  - User Datagram Protocol (UDP)
    - UDP Ports
      - Processes (server/client) are identified by an abstract locator known as port.
      - Server accepts message at well known port.
      - Some well-known UDP ports are 7–Echo, 53–DNS, 111–RPC, 161–SNMP, etc.
      - < port, host > pair is used as key for demultiplexing.
      - Ports are implemented as a message queue.
      - When a message arrives, UDP appends it to end of the queue.
      - When queue is full, the message is discarded.
      - When a message is read, it is removed from the queue.
      - When an application process wants to receive a message, one is removed from the front of the queue.
      - If the queue is empty, the process blocks until a message becomes available



Transport Layer

- Transport Layer Protocols
  - User Datagram Protocol (UDP)
    - UDP Datagram (Packet) Format
      - UDP packets are known as user datagrams .
      - These user datagrams, have a fixed-size header of 8 bytes made of four fields, each of 2 bytes (16 bits).



      - Source Port Number
        - Port number used by process on source host with 16 bits long.
        - If the source host is client (sending request) then the port number is an temporary one requested by the process and chosen by UDP.
        - If the source is server (sending response) then it is well known port number.

- Destination Port Number
  - Port number used by process on Destination host with 16 bits long.
  - If the destination host is the server (a client sending request) then the port number is a well known port number.
  - If the destination host is client (a server sending response) then port number is an temporary one copied by server from the request packet.
  - Length
    - This field denotes the total length of the UDP Packet (Header plus data)
    - The total length of any UDP datagram can be from 0 to 65,535 bytes.
  - Checksum
    - UDP computes its checksum over the UDP header, the contents of the message body, and something called the pseudoheader.
    - The pseudoheader consists of three fields from the IP header—protocol number, source IP address, destination IP address plus the UDP length field.
  - Data
    - Data field defines tha actual payload to be transmitted.
    - Its size is variable.

Transport Layer

- Transport Layer Protocols
  - User Datagram Protocol (UDP)
    - UDP Services
      - UDP services in communication process include:
        - Process-to-Process Communications
        - Connectionless Services
        - Flow Control
        - Error Control
        - Checksum
        - Congestion Control
        - Encapsulation and Decapsulation
        - Queueing
        - Multiplexing and Demulplexing

Transport Layer

- Transport Layer Protocols
  - User Datagram Protocol (UDP)
    - UDP Services
      - Process-to-Process Communication
        - UDP provides process-to-process communication using socket addresses, a combination of IP addresses and port numbers.
      - Connectionless Services
        - UDP provides a connectionless service.
        - There is no connection establishment and no connection termination .
        - Each user datagram sent by UDP is an independent datagram.
        - There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program.
        - The user datagrams are not numbered.
        - Each user datagram can travel on a different path.

Transport Layer

- Transport Layer Protocols
  - User Datagram Protocol (UDP)
    - UDP Services
      - Flow Control
        - UDP is a very simple protocol.
        - There is no flow control, and hence no window mechanism.
        - The receiver may overflow with incoming messages.
        - The lack of flow control means that the process using UDP should provide for this service, if needed.
      - Error Control
        - There is no error control mechanism in UDP except for the checksum.
        - This means that the sender does not know if a message has been lost or duplicated.
        - When the receiver detects an error through the checksum, the user datagram is silently discarded.
        - The lack of error control means that the process using UDP should provide for this service, if needed.

Transport Layer

- Transport Layer Protocols
  - User Datagram Protocol (UDP)
    - UDP Services
      - Checksum
        - UDP checksum calculation includes three sections: a pseudoheader, the UDP header, and the data coming from the application layer.
        - The pseudoheader is the part of the header in which the user datagram is to be encapsulated with some fields filled with 0s.

        - Optional Inclusion of Checksum
          - The sender of a UDP packet can choose not to calculate the checksum.
          - In this case, the checksum field is filled with all 0s before being sent.
          - In the situation where the sender decides to calculate the checksum, but it happens that the result is all 0s, the checksum is changed to all 1s before the packet is sent.
          - In other words, the sender complements the sum two times.

Transport Layer

- Transport Layer Protocols
  - User Datagram Protocol (UDP)
    - UDP Services
      - Congestion Control
        - Since UDP is a connectionless protocol, it does not provide congestion control.
        - UDP assumes that the packets sent are small and sporadic(occasionally or at irregular intervals) and cannot create congestion in the network.
        - This assumption may or may not be true, when UDP is used for interactive real-time transfer of audio and video.
      - Encapsulation and Decapsulation
        - To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages.

Transport Layer

- Transport Layer Protocols
  - User Datagram Protocol (UDP)
    - UDP Services
      - Queuing
        - In UDP, queues are associated with ports.
        - At the client site, when a process starts, it requests a port number from the operating system.
        - Some implementations create both an incoming and an outgoing queue associated with each process.
        - Other implementations create only an incoming queue associated with each process.
      - Multiplexing and Demultiplexing
        - In a host running a transport protocol suite, there is only one UDP but possibly several processes that may want to use the services of UDP.
        - To handle this situation, UDP multiplexes and demultiplexes.

Transport Layer

- Transport Layer Protocols
  - User Datagram Protocol (UDP)
    - Applications of UDP
      - UDP is used for management processes such as SNMP.
      - UDP is used for route updating protocols such as RIP.
      - UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software
      - UDP is suitable for a process with internal flow and error control mechanisms such as Trivial File Transfer Protocol (TFTP).
      - UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control.
      - UDP is normally used for interactive real-time applications that cannot tolerate uneven delay between sections of a received message..

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP is a reliable, connection-oriented, byte-stream protocol.
  - TCP guarantees the reliable, in-order delivery of a stream of bytes. It is a full-duplex protocol, meaning that each TCP connection supports a pair of byte streams, one flowing in each direction.
  - TCP includes a flow-control mechanism for each of these byte streams that allow the receiver to limit how much data the sender can transmit at a given time.
  - TCP supports a demultiplexing mechanism that allows multiple application programs on any given host to simultaneously carry on a conversation with their peers.
  - TCP also implements congestion-control mechanism. The idea of this mechanism is to prevent sender from overloading the network.
  - Flow control is an end to end issue, whereas congestion control is concerned with how host and network interact.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Services
    - The services provided by TCP include:
      - Process-to-Process Communication
      - Stream Delivery Service
      - Full-Duplex Communication
      - Multiplexing and Demultiplexing
      - Connection-Oriented Service
      - Reliable Service

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Services
    - Process-to-Process Communication
      - TCP provides process-to-process communication using port numbers.
    - Stream Delivery Service
      - TCP is a stream-oriented protocol.
      - TCP allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.
      - TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their bytes across the Internet.
      - The sending process produces (writes to) the stream and the receiving process consumes (reads from) it.

Transport Layer

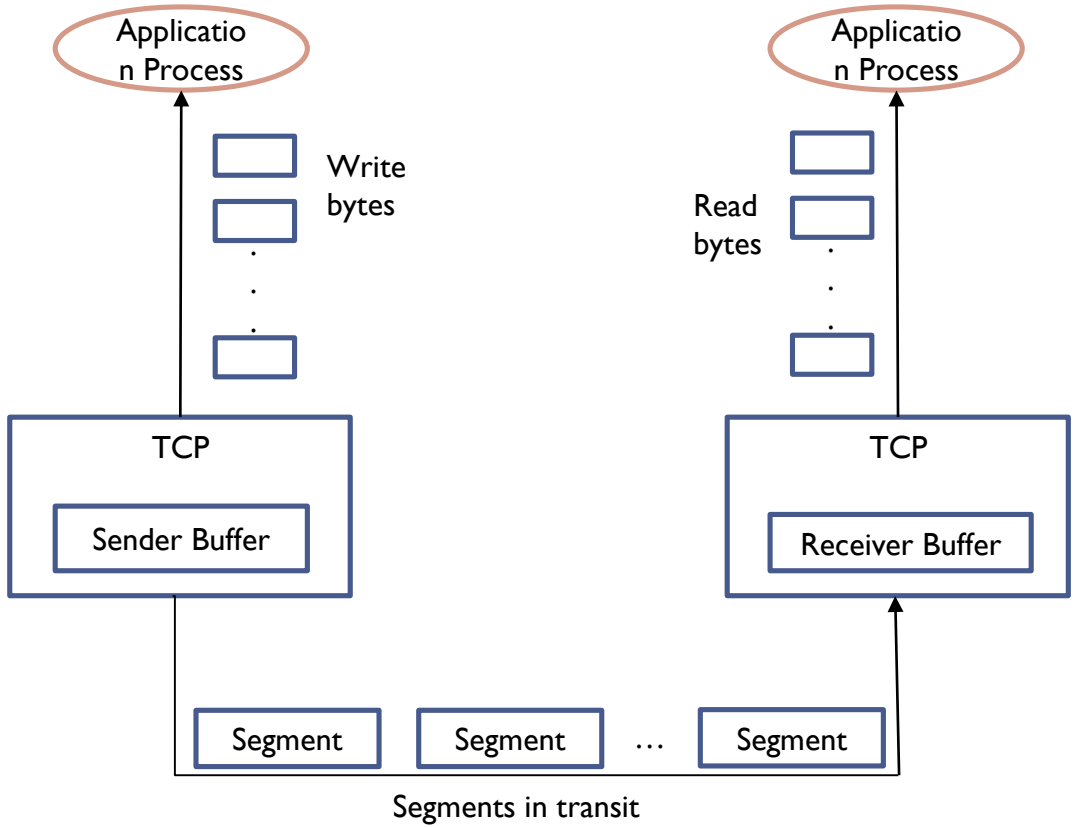
- Transmission Control Protocol (TCP)
  - TCP Services
    - Full-Duplex Communication
      - TCP offers full-duplex service, where data can flow in both directions at the same time.
      - Each TCP endpoint then has its own sending and receiving buffer, and segments move in both directions.
    - Multiplexing and Demultiplexing
      - TCP performs multiplexing at the sender and demultiplexing at the receiver.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Services
    - Connection-Oriented Service
      - TCP is a connection-oriented protocol.
      - A connection needs to be established for each pair of processes.
      - When a process at site A wants to send to and receive data from another process at site B, the following three phases occur:
        1. The two TCP's establish a logical connection between them.
        2. Data are exchanged in both directions.
        3. The connection is terminated.
    - Reliable Service
      - TCP is a reliable transport protocol.
      - It uses an acknowledgment mechanism to check the safe and sound arrival of data.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Segment
    - A packet in TCP is called a segment.
    - Data unit exchanged between TCP peers are called segments.
    - A TCP segment encapsulates the data received from the application layer.
    - The TCP segment is encapsulated in an IP datagram, which in turn is encapsulated in a frame at the data-link layer.

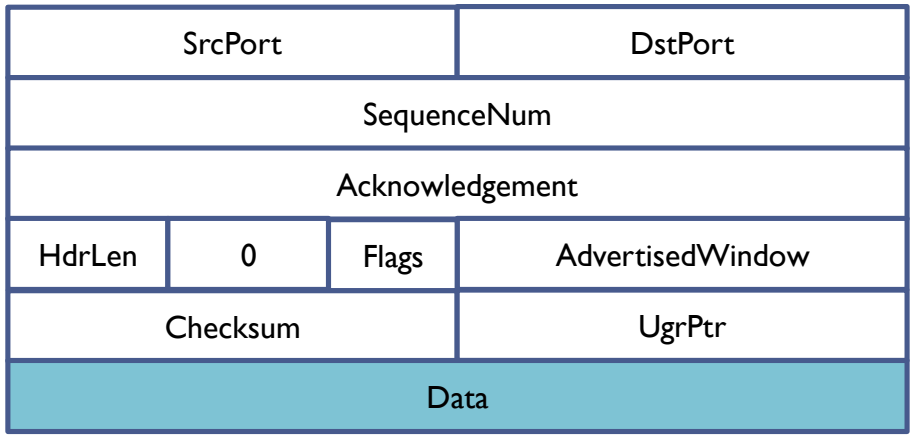


Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Segment
    - TCP is a byte-oriented protocol, which means that the sender writes bytes into a TCP connection and the receiver reads bytes out of the TCP connection.
    - TCP does not, itself, transmit individual bytes over the Internet.
    - TCP on the source host buffers enough bytes from the sending process to fill a reasonably sized packet and then sends this packet to its peer on the destination host.
    - TCP on the destination host then empties the contents of the packet into a receive buffer, and the receiving process reads from this buffer at its leisure.
    - TCP connection supports byte streams flowing in both directions.
    - The packets exchanged between TCP peers are called segments, since each one carries a segment of the byte stream.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Packet Format
    - Each TCP segment contains the header plus the data.
    - The segment consists of a header of 20 to 60 bytes, followed by data from the application program.
    - The header is 20 bytes if there are no options and up to 60 bytes if it contains options.



Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Packet Format
    - SrcPort and DstPort – port number of source and destination process.
    - SequenceNum – contains sequence number, i.e. first byte of data segment.
    - Acknowledgment – byte number of segment, the receiver expects next.
    - HdrLen – Length of TCP header as 4-byte words.
    - Flags – contains six control bits known as flags.
      - URG – segment contains urgent data.
      - ACK – value of acknowledgment field is valid.
      - PUSH – sender has invoked the push operation.
      - RESET – receiver wants to abort the connection.
      - SYN – synchronize sequence numbers during connection establishment.
      - FIN – terminates the TCP connection.

Transport Layer

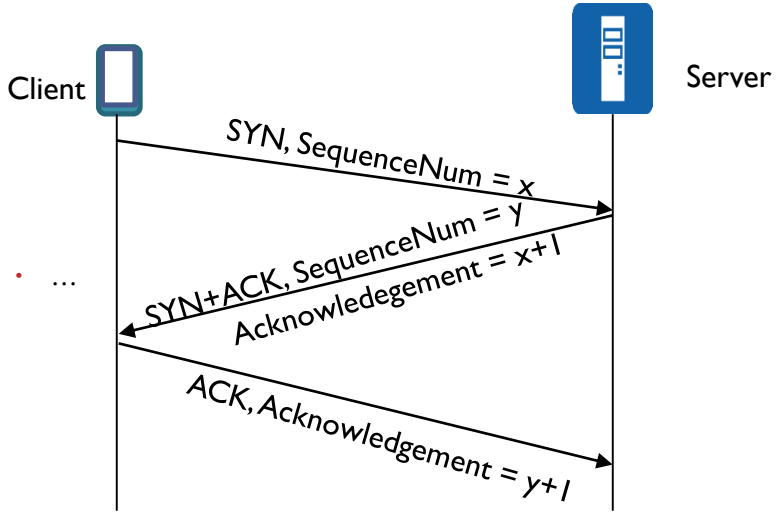
- Transmission Control Protocol (TCP)
  - TCP Packet Format
    - Advertised Window – defines receiver's window size and acts as flow control.
    - Checksum – It is computed over TCP header, Data, and pseudo header containing IP fields (Length, SourceAddr & DestinationAddr).
    - UrgPtr – used when the segment contains urgent data. It defines a value that must be added to the sequence number.
    - Options – There can be up to 40 bytes of optional information in the TCP header.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Connection Management
    - TCP is connection-oriented.
    - A connection-oriented transport protocol establishes a logical path between the source and destination.
    - All of the segments belonging to a message are then sent over this logical path.
    - In TCP, connection-oriented transmission requires three phases:
      - Connection Establishment
      - Data Transfer and
      - Connection Termination

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Connection Management
    - Connection Establishment
      - While opening a TCP connection the two nodes(client and server) want to agree on a set of parameters.
      - The parameters are the starting sequence numbers that is to be used for their respective byte streams.
      - Connection establishment in TCP is a three-way handshaking.



- …

1. Client sends a SYN segment to the server containing its initial sequence number (Flags = SYN, SequenceNum = x)
2. Server responds with a segment that acknowledges client's segment and specifies its initial sequence number (Flags = SYN + ACK, ACK = x + 1 SequenceNum = y).
3. Finally, client responds with a segment that acknowledges server's sequence number (Flags = ACK, ACK = y + 1).

- The reason that each side acknowledges a sequence number that is one larger than the one sent is that the Acknowledgment field actually identifies the "next sequence number expected,"
- A timer is scheduled for each of the first two segments, and if the expected response is not received, the segment is retransmitted.

Transport Layer

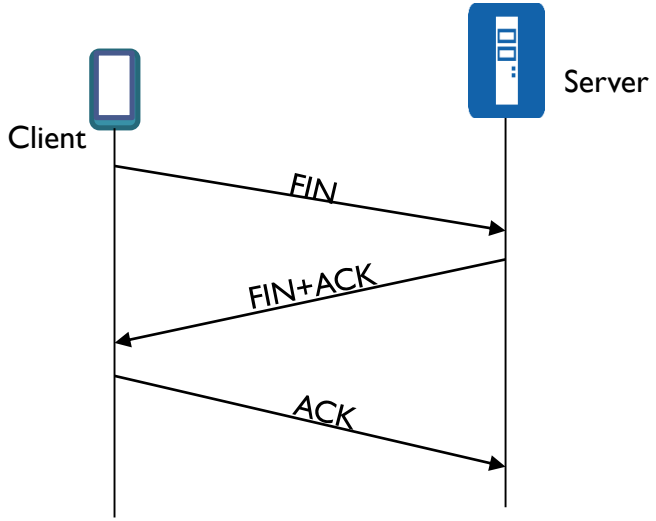
- Transmission Control Protocol (TCP)
  - TCP Connection Management
    - Data Transfer
    - After connection is established, bidirectional data transfer can take place.
    - The client and server can send data and acknowledgments in both directions.
    - The data traveling in the same direction as an acknowledgment are carried on the same segment.
    - The acknowledgment is piggybacked with the data.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Connection Management
    - Connection Termination
      - Connection termination or teardown can be done in two ways:
        - Three-way Close and
        - Half-Close

Transport Layer

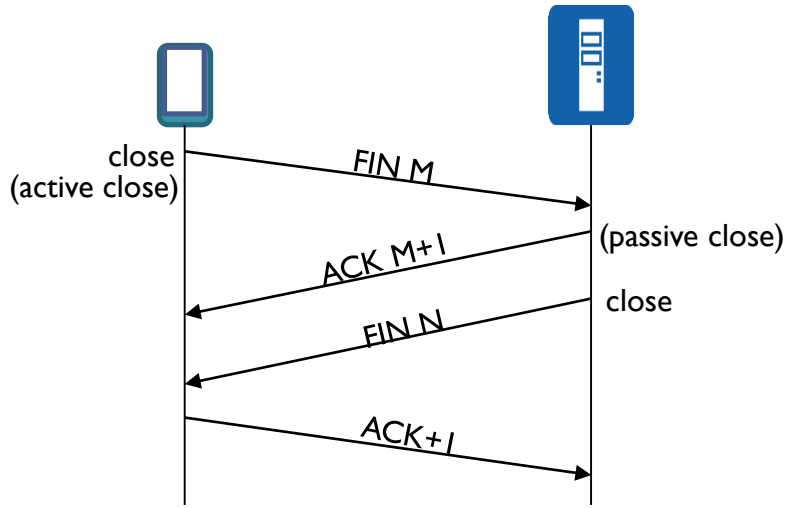
- Transmission Control Protocol (TCP)
  - TCP Connection Management
    - Connection Termination
      - Three-way Close
        - Three-way Close – Both client and server close simultaneously



- Client sends a FIN segment.
- The FIN segment can include last chunk of data.
- Server responds with FIN + ACK segment to inform its closing.
- Finally, client sends an ACK
- segment

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Connection Management
    - Connection Termination
      - Half-Close
        - Half-Close – Client stops sending but receives data.



- Client half-closes the connection by sending a FIN segment.
- Server sends an ACK segment.
- Data transfer from client to the server stops.
- After sending all data, server sends FIN segment to client, which is acknowledged by the client.

Transport Layer

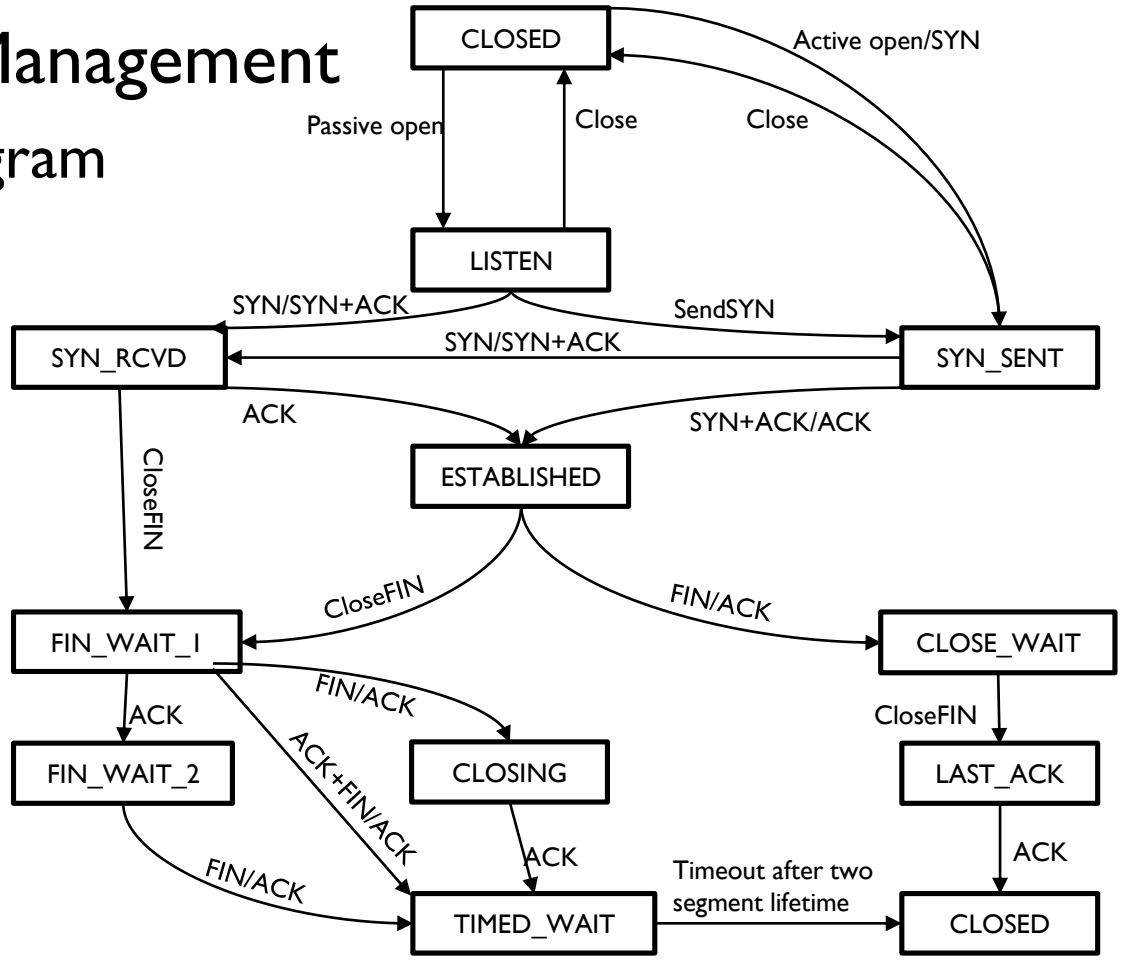
- Transmission Control Protocol (TCP)
  - State Transition Diagram
    - To keep track of all the different events happening during connection establishment, connection termination, and data transfer, TCP is specified as the finite state machine (FSM).
    - The transition from one state to another is shown using directed lines.
    - States involved in opening and closing a connection is shown above and below ESTABLISHED state respectively.

- States Involved in TCP:

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Connection Management
    - State Transition Diagram



Transport Layer

- Transmission Control Protocol (TCP)
  - Opening a TCP Connection
    1. Server invokes a passive open on TCP, which causes TCP to move to LISTEN state
    2. Client does an active open, which causes its TCP to send a SYN segment to the server and move to SYN_SENT state.
    3. When SYN segment arrives at the server, it moves to SYN_RCVD state and responds with a SYN + ACK segment.
    4. Arrival of SYN + ACK segment causes the client to move to ESTABLISHED state and sends an ACK to the server.
    5. When ACK arrives, the server finally moves to ESTABLISHED state.

Transport Layer

- Transmission Control Protocol (TCP)
  - Closing a TCP Connection
    - Client / Server can independently close its half of the connection or simultaneously.
    - Transitions from ESTABLISHED to CLOSED state are:
      - One side closes:
        - ESTABLISHED → FIN_WAIT_1 → FIN_WAIT_2 → TIME_WAIT → CLOSED
      - Other side closes:
        - ESTABLISHED → CLOSE_WAIT → LAST_ACK → CLOSED
      - Simultaneous close:
        - ESTABLISHED → FIN_WAIT_1 → CLOSING → TIME_WAIT → CLOSED

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP FLOW CONTROL
    - TCP uses a variant of sliding window known as adaptive flow control that:
      - Guarantees reliable delivery of data
      - Ensures ordered delivery of data
      - Enforces flow control at the sender
    - Receiver advertises its window size to the sender using AdvertisedWindow field.
    - Sender thus cannot have unacknowledged data greater than AdvertisedWindow.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Flow Control
    - Send Buffer
      - Sending TCP maintains send buffer which contains 3 segments
        1. Acknowledged data
        2. Unacknowledged data
        3. Data to be transmitted.

      - Send buffer maintains three pointers
        1. LastByteAcked,
        2. LastByteSent, and
        3. LastByteWritten
      - such that:

        $$\text{LastByteAcked} \leq \text{LastByteSent} \leq \text{LastByteWritten}$$

      - A byte can be sent only after being written and only a sent byte can be acknowledged.

      - Bytes to the left of LastByteAcked are not kept as it had been acknowledged.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Flow Control
    - Receive Buffer
      - Receiving TCP maintains receive buffer to hold data even if it arrives out-of-order.
      - Receive buffer maintains three pointers namely
        1. LastByteRead
        2. NextByteExpected, and
        3. LastByteRcvd
      - such that:

        $$\text{LastByteRead} \leq \text{NextByteExpected} \leq \text{LastByteRcvd} + 1$$

      - A byte cannot be read until that byte and all preceding bytes have been received.
      - If data is received in order, then NextByteExpected = LastByteRcvd + 1
      - Bytes to the left of LastByteRead are not buffered, since it is read by the application.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Flow Control
    - Flow Control in TCP
      - Size of send and receive buffer is MaxSendBuffer and MaxRcvBuffer respectively.
      - Sending TCP prevents overflowing of send buffer by maintaining
        - LastByteWritten − LastByteAcked ≤ MaxSendBuffer
      - Receiving TCP avoids overflowing its receive buffer by maintaining
        - LastByteRcvd − LastByteRead ≤ MaxRcvBuffer
      - Receiver throttles the sender by having AdvertisedWindow based on free space available for buffering.
        - AdvertisedWindow = MaxRcvBuffer − ((NextByteExpected − 1) – LastByteRead)

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Flow Control
    - Flow Control in TCP
      - Sending TCP adheres to AdvertisedWindow by computing EffectiveWindow that limits how much data it should send.
        - EffectiveWindow = AdvertisedWindow − (LastByteSent − LastByteAcked)
      - When data arrives, LastByteRcvd moves to its right and AdvertisedWindow shrinks.
      - Receiver acknowledges only, if preceding bytes have arrived.
      - AdvertisedWindow expands when data is read by the application.
        - If data is read as fast as it arrives then AdvertisedWindow = MaxRcvBuffer
        - If data is read slowly, it eventually leads to a AdvertisedWindow of size 0.
      - AdvertisedWindow field is designed to allow sender to keep the pipe full.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Transmission
    - TCP has three mechanism to trigger the transmission of a segment.
    - They are
      - Maximum Segment Size (MSS) – Silly Window Syndrome
      - Timeout – Nagle's Algorithm

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Transmission
    - Maximum Segment Size (MSS) – Silly Window Syndrome
      - When either the sending application program creates data slowly or the receiving application program consumes data slowly, or both, problems arise.
      - Any of these situations results in the sending of data in very small segments, which reduces the efficiency of the operation.
      - This problem is called the silly window syndrome.
      - The sending TCP may create a silly window syndrome if it is serving an application program that creates data slowly, for example, 1 byte at a time.
      - The application program writes 1 byte at a time into the buffer of the sending TCP.
      - The result is a lot of 1-byte segments that are traveling through an internet.
      - The solution is to prevent the sending TCP from sending the data byte by byte.
      - The sending TCP must be forced to wait and collect data to send in a larger block.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Transmission
    - Timeout – Nagle's Algorithm
      - If there is data to send but is less than MSS, then we may want to wait some amount of time before sending the available data
      - If we wait too long, then it may delay the process.
      - If we don't wait long enough, it may end up sending small segments resulting in Silly Window Syndrome.
      - The solution is to introduce a timer and to transmit when the timer expires
      - Nagle introduced an algorithm for solving this problem

when the application produces data to send

if(both the available data and the window) = MSS

send the segment

else

if(there is un_ACKed data)

Buffer the new data until an ACK arrives

else

Send all the new data now

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Control
    - Congestion occurs if load (number of packets sent) is greater than capacity of the network (number of packets a network can handle).
    - When load is less than network capacity, throughput increases proportionally.
    - When load exceeds capacity, queues become full and the routers discard some packets and throughput declines sharply.
    - When too many packets are contending for the same link
      - The queue overflows
      - Packets get dropped
      - Network is congested
    - Network should provide a congestion control mechanism to deal with such a situation.
    - TCP maintains a variable called CongestionWindow for each connection.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Control
    - TCP Congestion Control mechanisms are:
      1. Additive Increase / Multiplicative Decrease (AIMD)
      2. Slow Start
      3. Fast Retransmit and Fast Recovery

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Control
    - Additive Increase / Multiplicative Decrease (AIMD)
      - TCP source initializes CongestionWindow based on congestion level in the network.
      - Source increases CongestionWindow when level of congestion goes down and decreases the same when level of congestion goes up.
      - TCP interprets timeouts as a sign of congestion and reduces the rate of transmission.
      - On timeout, source reduces its CongestionWindow by half, i.e., multiplicative decrease. For example, if CongestionWindow = 16 packets, after timeout it is 8.
      - Value of CongestionWindow is never less than maximum segment size (MSS).

Transport Layer

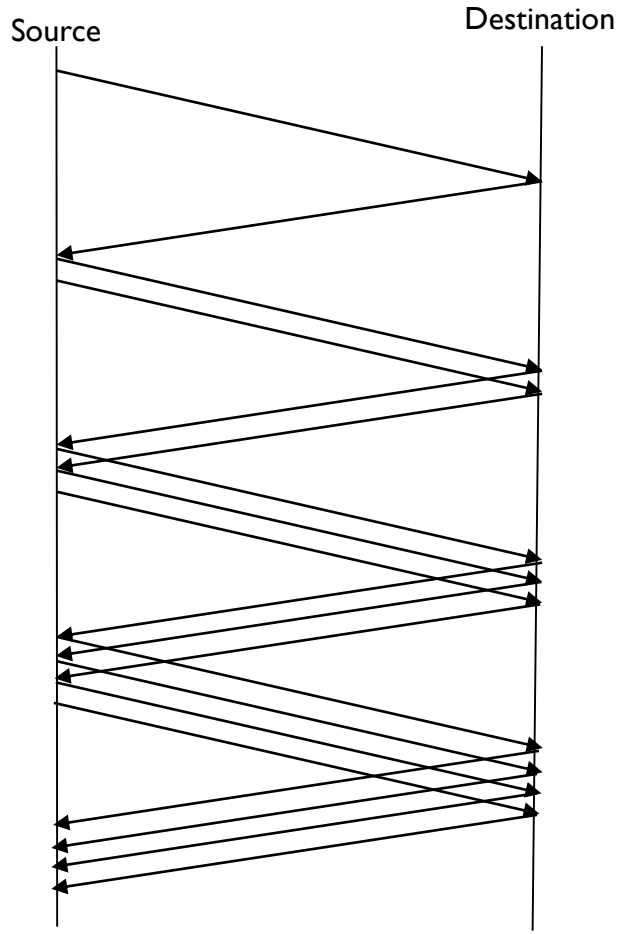
- Transmission Control Protocol (TCP)
  - TCP Congestion Control
    - Additive Increase / Multiplicative Decrease (AIMD)
      - When ACK arrives CongestionWindow is incremented marginally, i.e., additive increase.

        Increment = MSS × (MSS/CongestionWindow)

        CongestionWindow += Increment

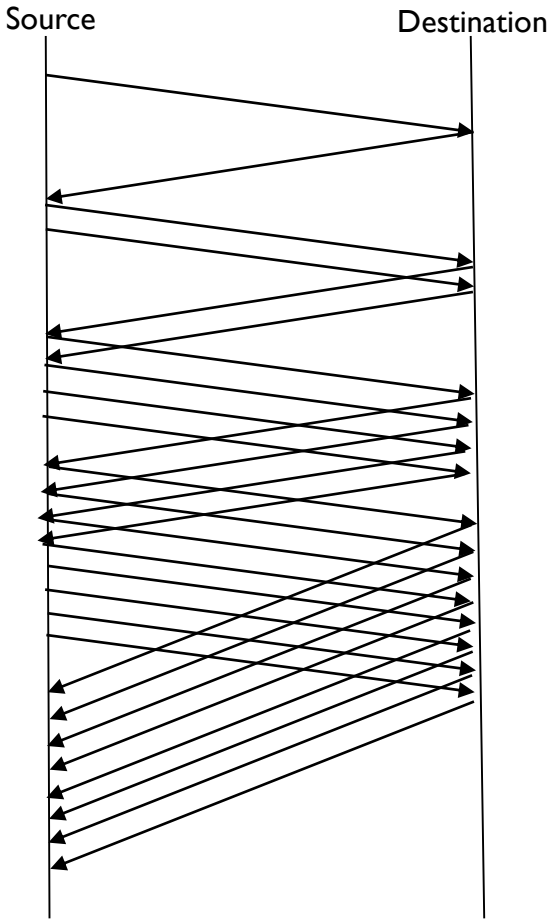
      - For example, when ACK arrives for 1 packet, 2 packets are sent. When ACK for both packets arrive, 3 packets are sent and so on.

      - CongestionWindow increases and decreases throughout lifetime of the connection.



Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Control
    - Slow Start
      - Slow start is used to increase CongestionWindow exponentially from a cold start.
      - Source TCP initializes CongestionWindow to one packet.
      - TCP doubles the number of packets sent every RTT on successful transmission.
      - When ACK arrives for first packet TCP adds 1 packet to CongestionWindow and sends two packets.
      - When two ACKs arrive, TCP increments CongestionWindow by 2 packets and sends four packets and so on.



Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Control
    - Slow Start
      - Instead of sending entire permissible packets at once (bursty traffic), packets are sent in a phased manner, i.e., slow start.
      - Initially TCP has no idea about congestion, henceforth it increases CongestionWindow rapidly until there is a timeout. On timeout:

        CongestionThreshold = CongestionWindow/ 2

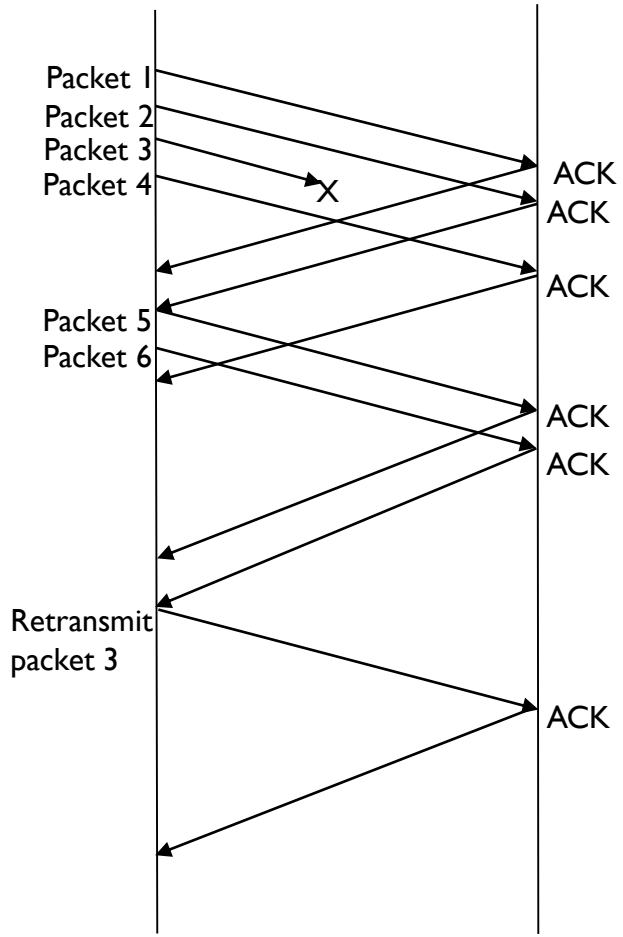
        CongestionWindow = 1
      - Slow start is repeated until CongestionWindow reaches CongestionThreshold and thereafter 1 packet per RTT.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Control
    - Fast Retransmit And Fast Recovery
      - TCP timeouts led to long periods of time during which the connection went dead while waiting for a timer to expire.
      - Fast retransmit is a heuristic approach that triggers retransmission of a dropped packet sooner than the regular timeout mechanism. It does not replace regular timeouts.
      - When a packet arrives out of order, receiving TCP resends the same acknowledgment (duplicate ACK) it sent last time.
      - When three duplicate ACK arrives at the sender, it infers that corresponding packet may be lost due to congestion and retransmits that packet. This is called fast retransmit before regular timeout.
      - When packet loss is detected using fast retransmit, the slow start phase is replaced by additive increase, multiplicative decrease method. This is known as fast recovery.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Control
    - Fast Retransmit And Fast Recovery
      - Instead of setting CongestionWindow to one packet, this method uses the ACKs that are still in pipe to clock the sending of packets.
      - Slow start is only used at the beginning of a connection and after regular timeout. At other times, it follows a pure AIMD pattern.
      - For example, packets 1 and 2 are received whereas packet 3 gets lost.
        - Receiver sends a duplicate ACK for packet 2 when packet 4 arrives.
        - Sender receives 3 duplicate ACKs after sending packet 6 retransmits packet 3.
        - When packet 3 is received, receiver sends cumulative ACK up to packet 6.



Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Avoidance
    - Congestion avoidance mechanisms prevent congestion before it actually occurs.
    - These mechanisms predict when congestion is about to happen and then to reduce the rate at which hosts send data just before packets start being discarded.
    - TCP creates loss of packets in order to determine bandwidth of the connection.
    - Routers help the end nodes by intimating when congestion is likely to occur.
    - Congestion-avoidance mechanisms are:
      - DEC bit - Destination Experiencing Congestion Bit
      - RED - Random Early Detection

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Avoidance
    - Dec Bit - Destination Experiencing Congestion Bit
      - The first mechanism developed for use on the Digital Network Architecture (DNA).
      - The idea is to evenly split the responsibility for congestion control between the routers and the end nodes.
      - Each router monitors the load it is experiencing and explicitly notifies the end nodes when congestion is about to occur.
      - This notification is implemented by setting a binary congestion bit in the packets that flow through the router; hence the name DECbit.
      - The destination host then copies this congestion bit into the ACK it sends back to the source.
      - The Source checks how many ACK has DEC bit set for previous window packets.
      - If less than 50% of ACK have DEC bit set, then source increases its congestion window by 1 packet

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Avoidance
    - Dec Bit - Destination Experiencing Congestion Bit
      - Otherwise, decreases the congestion window by 87.5%.
      - Finally, the source adjusts its sending rate so as to avoid congestion.
      - Increase by 1, decrease by 0.875 rule was based on AIMD for stabilization.
      - A single congestion bit is added to the packet header.
      - Using a queue length of 1 as the trigger for setting the congestion bit.
      - A router sets this bit in a packet if its average queue length is greater than or equal to 1 at the time the packet arrives.
      - Average queue length is measured over a time interval that includes the
        last busy + last idle cycle + current busy cycle.
      - It calculates the average queue length by dividing the curve area with time interval.

Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Avoidance
    - Red - Random Early Detection
      - The second mechanism of congestion avoidance is called as Random Early Detection (RED).
      - Each router is programmed to monitor its own queue length, and when it detects that there is congestion, it notifies the source to adjust its congestion window.
      - RED differs from the DEC bit scheme by two ways:
        - In DECbit, explicit notification about congestion is sent to source, whereas RED implicitly notifies the source by dropping a few packets.
        - DECbit may lead to tail drop policy, whereas RED drops packet based on drop probability in a random manner. Drop each arriving packet with some drop probability whenever the queue length exceeds some drop level. This idea is called early random drop.

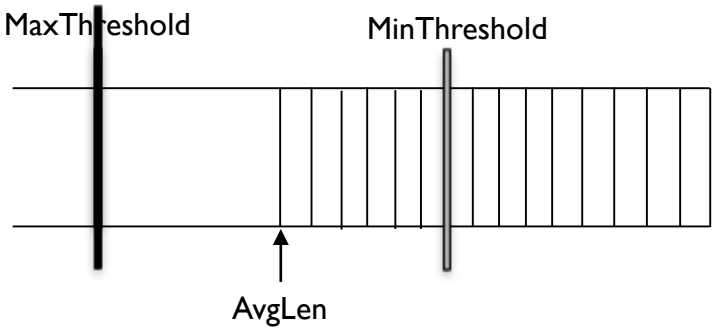
Transport Layer

- Transmission Control Protocol (TCP)
  - TCP Congestion Avoidance
    - Red - Random Early Detection
      - Computation of average queue length using RED

        AvgLen = (1 − Weight) × AvgLen + Weight × SampleLen

        where 0 < Weight < 1 and

        SampleLen – is the length of the queue when a sample measurement is made.
      - The queue length is measured every time a new packet arrives at the gateway.
      - RED has two queue length thresholds that trigger certain activity: MinThreshold and MaxThreshold
      - When a packet arrives at a gateway it compares Avglen with these two values according to these rules.



if AvgLen≤MinThreshold
→ queue the packet

if MinThreshold<AvgLen<MaxThreshold
→ calculate probability P
→ drop the arriving packet with probability P

if AvgLen≥MaxThreshold
→ drop the arriving packet

Transport Layer

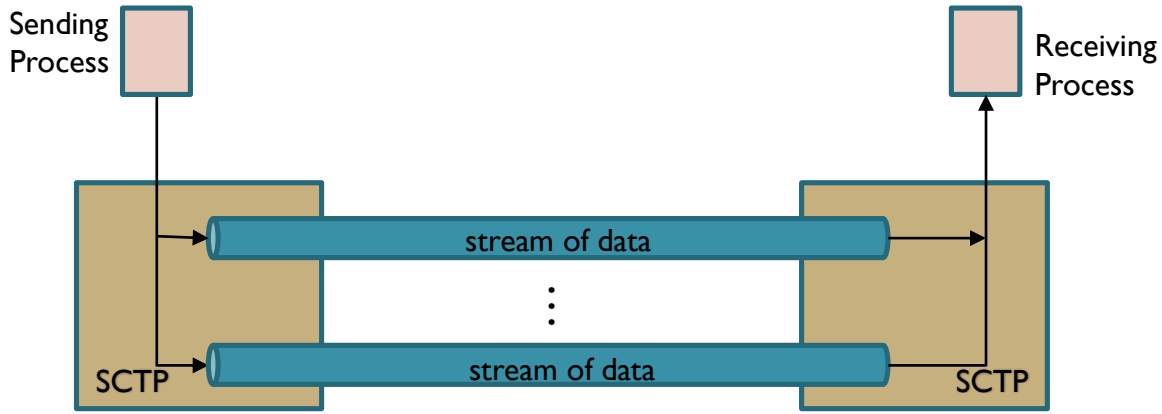
- Stream Control Transmission Protocol (SCTP)
  - Stream Control Transmission Protocol (SCTP) is a reliable, message-oriented transport layer protocol.
  - SCTP has mixed features of TCP and UDP.
  - SCTP maintains the message boundaries and detects the lost data, duplicate data as well as out-of-order data.
  - SCTP provides the Congestion control as well as Flow control.
  - SCTP is especially designed for internet applications as well as multimedia communication.

Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Services
    - The services provided by SCTP include:
      - Process-to-Process communications
      - Multiple Streams
      - Multihoming
      - Full-Duplex Communication
      - Connection-Oriented Service
      - Reliable Service

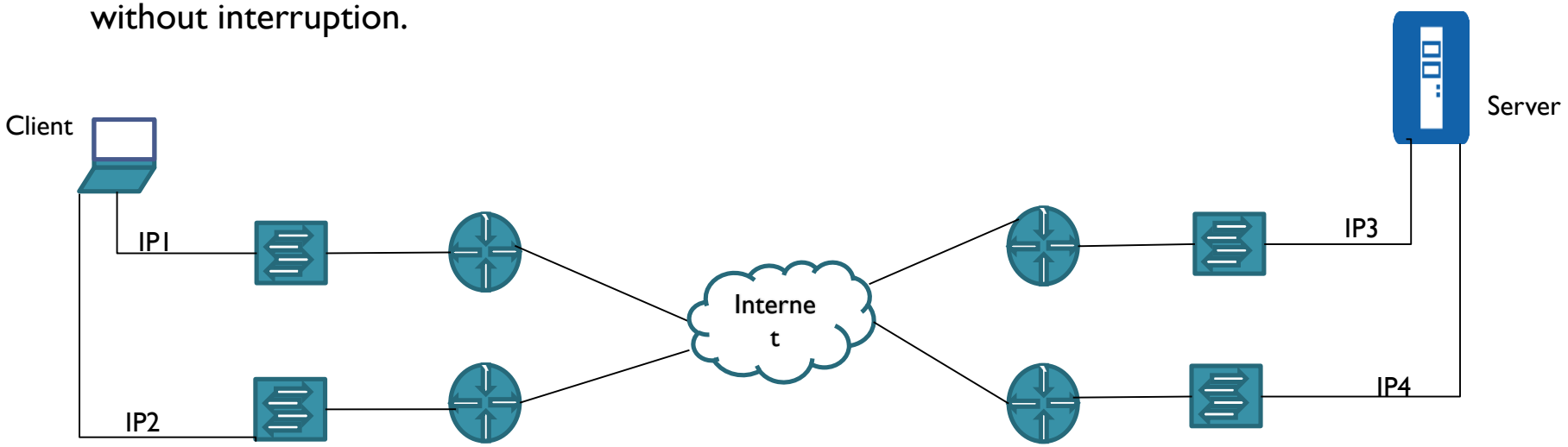
Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Services
    - Process-to-Process Communication
      - SCTP provides process-to-process communication.
    - Multiple Streams
      - SCTP allows multistream service in each connection, which is called association in SCTP terminology.
      - If one of the streams is blocked, the other streams can still deliver their data.



Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Services
    - Multihoming
      - An SCTP association supports multihoming service.
      - The sending and receiving host can define multiple IP addresses in each end for an association.
      - In this fault-tolerant approach, when one path fails, another interface can be used for data delivery without interruption.

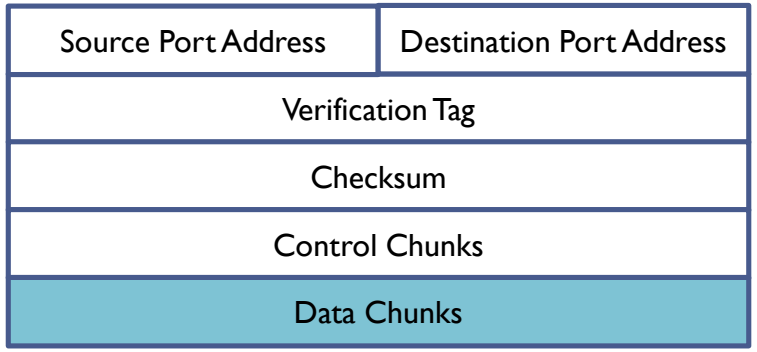


Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Services
    - Full-Duplex Communication
      - SCTP offers full-duplex service, where data can flow in both directions at the same time. Each SCTP then has a sending and receiving buffer and packets are sent in both directions.
    - Connection-Oriented Service
      - SCTP is a connection-oriented protocol.
      - In SCTP, a connection is called an association.
      - If a client wants to send and receive message from server , the steps are :
        - Step1: The two SCTPs establish the connection with each other.
        - Step2: Once the connection is established, the data gets exchanged in both the
        - directions.
        - Step3: Finally, the association is terminated.
    - Reliable Service
      - SCTP is a reliable transport protocol.
      - It uses an acknowledgment mechanism to check the safe and sound arrival of data.

Transport Layer

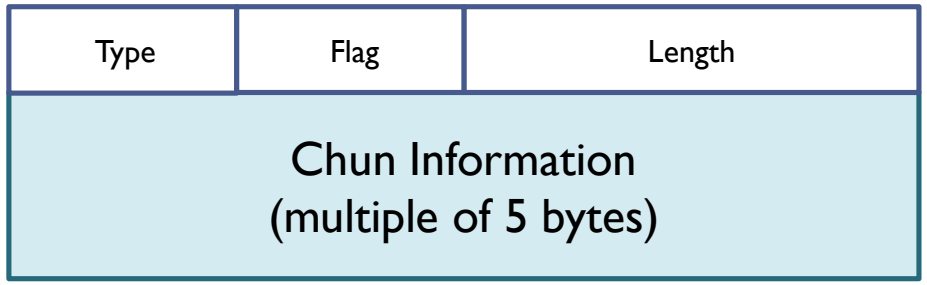
- Stream Control Transmission Protocol (SCTP)
  - SCTP Packet Format
    - An SCTP packet has a mandatory general header and a set of blocks called chunks.
    - The CTP packet is 20bytes in size



    - Verification tag
      - A 32-bit random value created during initialization to distinguish stale packets from a previous connection.
    - Checksum
      - The next field is a checksum. The size of the checksum is 32 bits. SCTP uses `CRC-32 Checksum.
  - General Header
    - The general header (packet header) defines the end points of each association to which the packet belongs
    - It guarantees that the packet belongs to a particular association
    - It also preserves the integrity of the contents of the packet including the header itself.
    - There are four fields in the general header.
  - Source port
    - This field identifies the sending port.
  - Destination port
    - This field identifies the receiving port that hosts use to route the packet to the appropriate endpoint/application.

Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Packet Format
    - Chunks
      - Control information or user data are carried in chunks.
      - Chunks have a common layout.
      - The first three fields are common to all chunks; the information field depends on the type of chunk.



- The type field can define up to 256 types of chunks. Only a few have been defined so far; the rest are reserved for future use.
- The flag field defines special flags that a particular chunk may need.
- The length field defines the total size of the chunk, in bytes, including the type, flag, and length fields.

Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Packet Format
    - Types of Chunks
      - An SCTP association may send many packets, a packet may contain several chunks, and chunks may belong to different streams.
      - SCTP defines two types of chunks - Control chunks and Data chunks.
      - A control chunk controls and maintains the association.
      - A data chunk carries user data.

Type	Chunk	Description
0	DATA	User data
1	INIT	Sets up an association
2	INIT ACL	Acknowledges INIT chunk
3	SACK	Selective acknowledgement
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 with establishment
11	COOKIE ACK	Acknowledges COOKIE ACK ECHO
14	SHUTDOWN COMPLETE	Third packet in association with termination
192	FORWARD TSN	For adjusting cumulating TSN

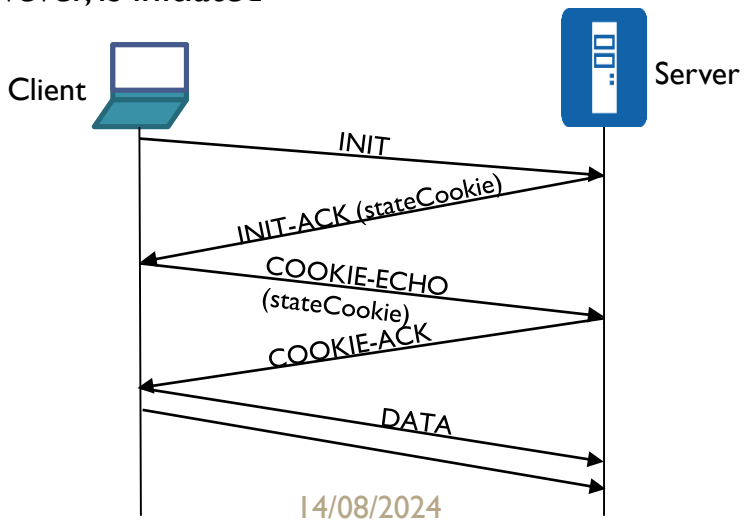
Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Association
    - SCTP is a connection-oriented protocol.
    - A connection in SCTP is called an association to emphasize multihoming.
    - SCTP Associations consists of three phases:
      - Association Establishment
      - Data Transfer
      - Association Termination

Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Association
    - Association Establishment
      - Association establishment in SCTP requires a four-way handshake.
      - In this procedure, a client process wants to establish an association with a server process using SCTP as the transport-layer protocol.
      - The SCTP server needs to be prepared to receive any association (passive open).
      - Association establishment, however, is initiated by the client (active open).
      - …``

- The client sends the first packet, which contains an INIT chunk.
- The server sends the second packet, which contains an INIT ACK chunk. The INIT ACK also sends a cookie that defines the state of the server at this moment.
- The client sends the third packet, which includes a COOKIE ECHO chunk. This is a very simple chunk that echoes, without change, the cookie sent by the server. SCTP allows the inclusion of data chunks in this packet.
- The server sends the fourth packet, which includes the COOKIE ACK chunk that acknowledges the receipt of the COOKIE ECHO chunk. SCTP allows the inclusion of data chunks with this packet.



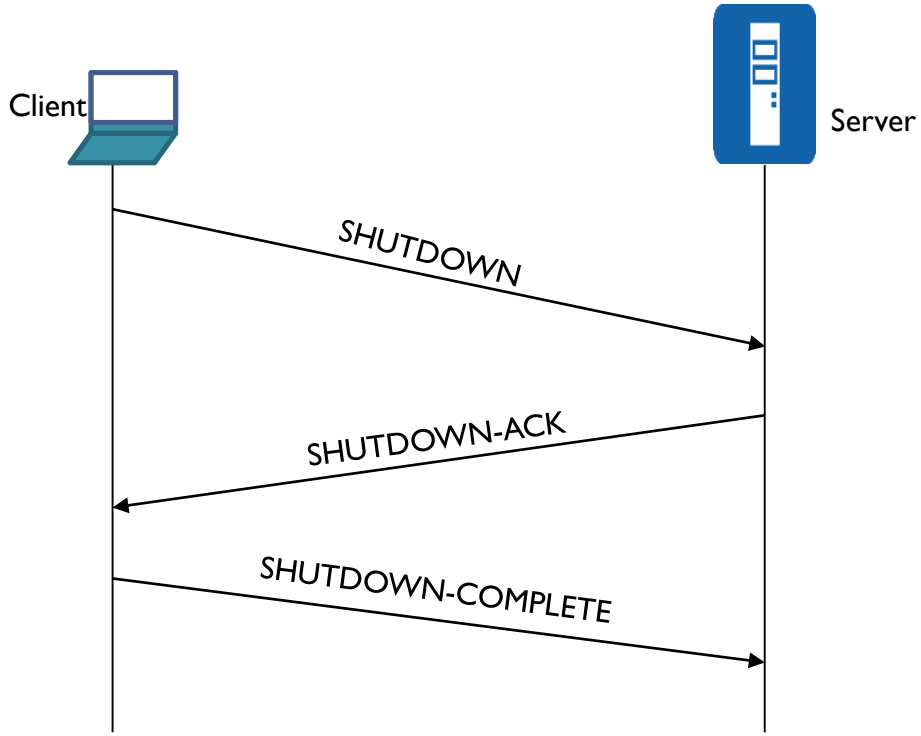
Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Association
    - Data Transfer
      - The whole purpose of an association is to transfer data between two ends.
      - After the association is established, bidirectional data transfer can take place.
      - The client and the server can both send data.
      - SCTP supports piggybacking

- Types of SCTP data Transfer :
  1. 1. Multihoming Data Transfer
     - Data transfer, by default, uses the primary address of the destination.
     - If the primary is not available, one of the alternative addresses is used.
     - This is called Multihoming Data Transfer.
  2. Multistream Delivery
     - SCTP can support multiple streams, which means that the sender process can define different streams and a message can belong to one of these streams.
     - Each stream is assigned a stream identifier (SI) which uniquely defines that stream.
     - SCTP supports two types of data delivery in each stream: ordered (default) and unordered.

Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Association
    - Association Termination
    - In SCTP, either of the two parties involved in exchanging data (client or server) can close the connection.
    - SCTP does not allow a "half closed" association. If one end closes the association, the other end must stop sending new data.
    - If any data are left over in the queue of the recipient of the termination request, they are sent and the association is closed.
    - Association termination uses three packets.

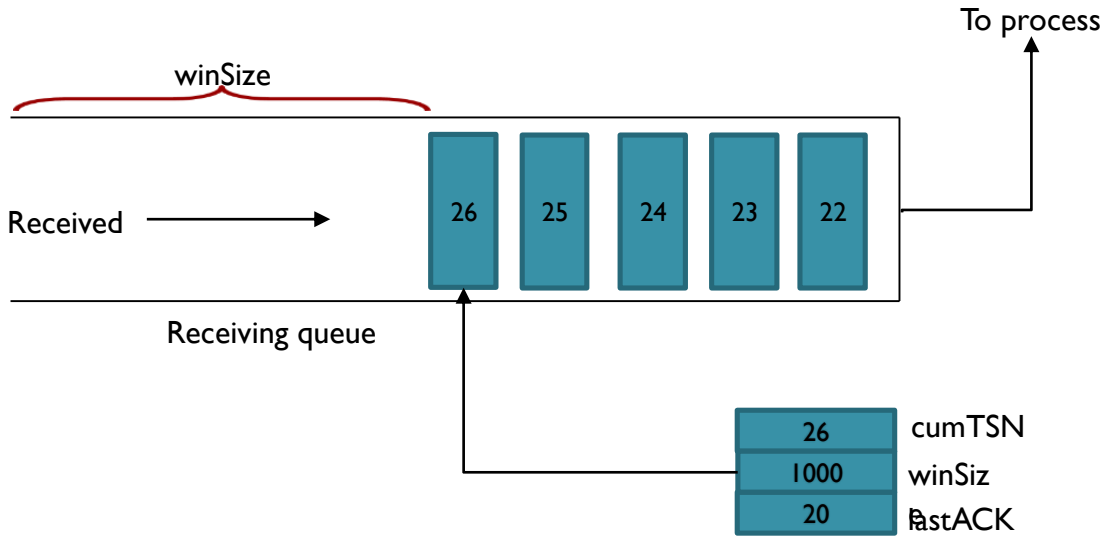


Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Flow Control
    - Flow control in SCTP is similar to that in TCP.
    - Current SCTP implementations use a byte-oriented window for flow control.

Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Flow Control
    - Receiver Site
      - The receiver has one buffer (queue) and three variables.
      - The first variable holds the last TSN received, cumTSN.
      - The second variable holds the available buffer size; winsize.
      - The third variable holds the last accumulative acknowledgment, lastACK.
      - The figure on the right shows the queue and variables at the receiver site.

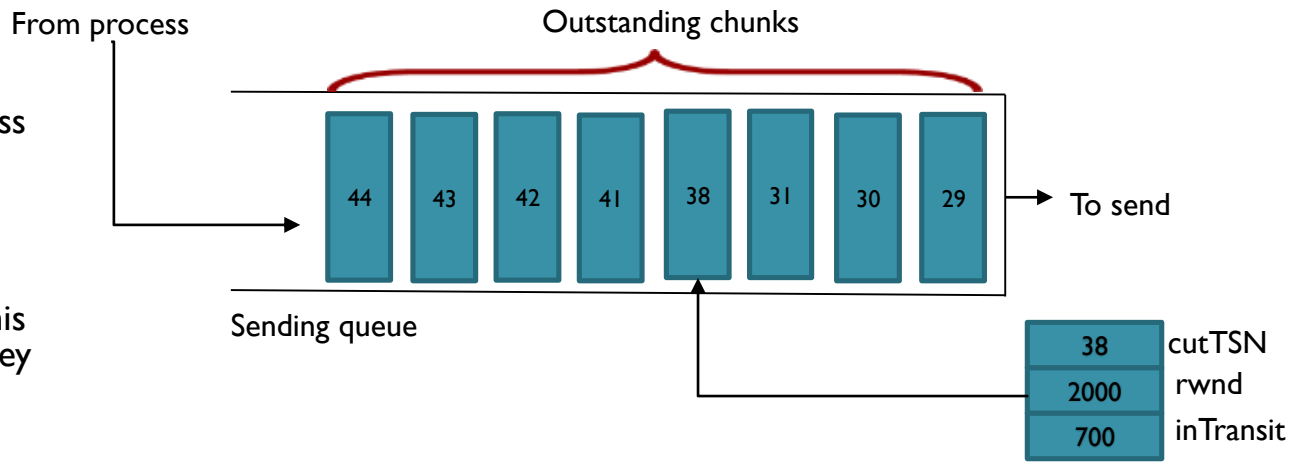


Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Flow Control
    - Receiver Site
      - When the site receives a data chunk, it stores it at the end of the buffer (queue) and subtracts the size of the chunk from winSize.
      - The TSN number of the chunk is stored in the cumTSN variable.
      - When the process reads a chunk, it removes it from the queue and adds the size of the removed chunk to winSize (recycling).
      - When the receiver decides to send a SACK, it checks the value of lastAck; if it is less than cumTSN, it sends a SACK with a cumulative TSN number equal to the cumTSN.
      - It also includes the value of winSize as the advertised window size.

Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Flow Control
    - Sender Site
      - The sender has one buffer (queue) and three variables: curTSN, rwnd, and inTransit.
      - We assume each chunk is 100 bytes long. The buffer holds the chunks produced by the process that either have been sent or are ready to be sent.
      - The first variable, curTSN, refers to the next chunk to be sent.
      - All chunks in the queue with a TSN less than this value have been sent, but not acknowledged; they are outstanding.
      - The second variable, rwnd, holds the last value advertised by the receiver (in bytes).
      - The third variable, inTransit, holds the number of bytes in transit, bytes sent but not yet acknowledged.
      - The following figure shows the queue and variables at the sender site.



Transport Layer

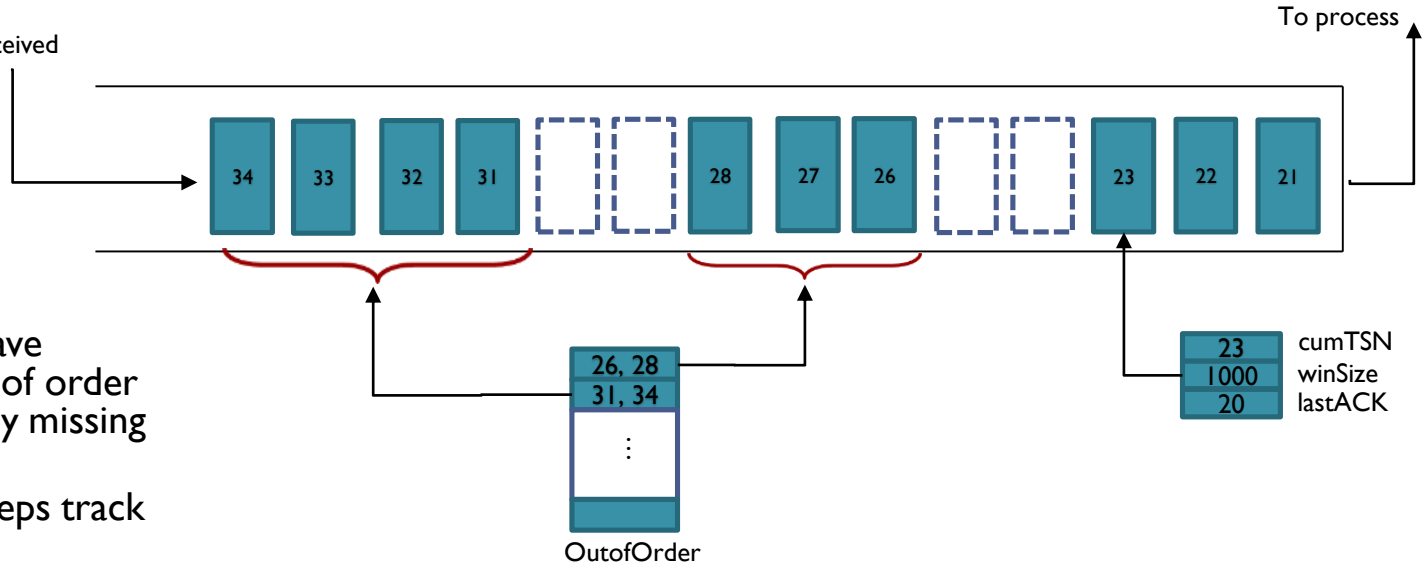
- Stream Control Transmission Protocol (SCTP)
  - SCTP Flow Control
    - Sender Site
      - A chunk pointed to by curTSN can be sent if the size of the data is less than or equal to the quantity rwnd - inTransit.
      - After sending the chunk, the value of curTSN is incremented by 1 and now points to the next chunk to be sent.
      - The value of inTransit is incremented by the size of the data in the transmitted chunk.
      - When a SACK is received, the chunks with a TSN less than or equal to the cumulative TSN in the SACK are removed from the queue and discarded. The sender does not have to worry about them anymore.
      - The value of inTransit is reduced by the total size of the discarded chunks.
      - The value of rwnd is updated with the value of the advertised window in the SACK.

Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Error Control
    - SCTP is a reliable transport layer protocol.
    - It uses a SACK chunk to report the state of the receiver buffer to the sender.
    - Each implementation uses a different set of entities and timers for the receiver and sender sites.

Transport Layer

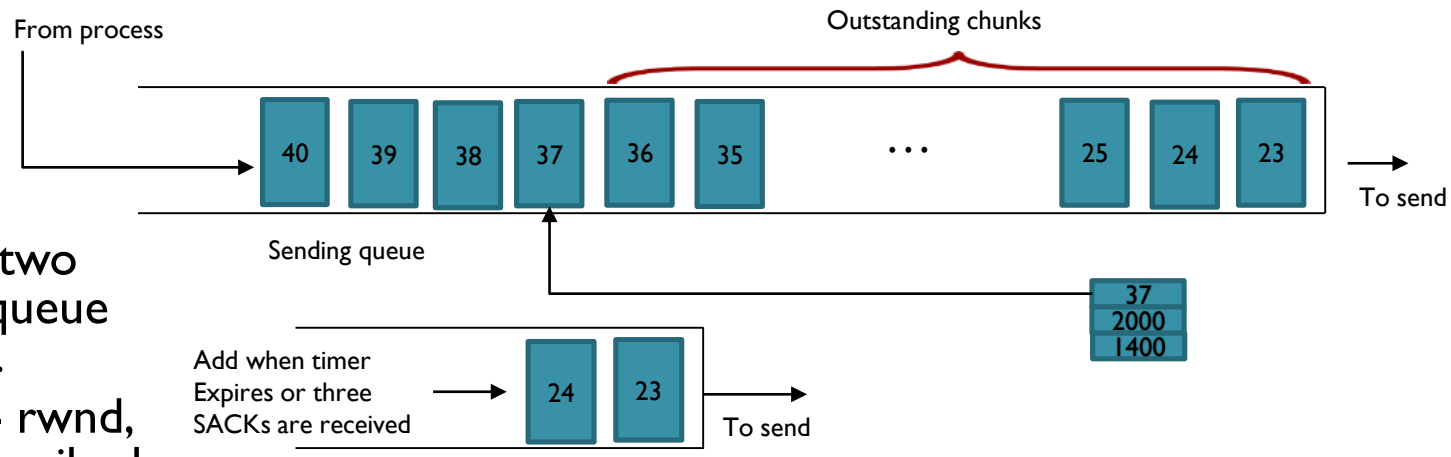
- Stream Control Transmission Protocol (SCTP)
  - SCTP Error Control
    - Receiver Site
      - The receiver stores all chunks that have arrived in its queue including the out of order ones. However, it leaves spaces for any missing chunks.
      - It discards duplicate messages, but keeps track of them for reports to the sender.
      - The following figure shows a typical design for the receiver site and the state of the receiving queue at a particular point in time
      - The available window size is 1000 bytes.
      - The last acknowledgment sent was for data chunk 20.
      - Chunks 21 to 23 have been received in order.
      - The first out-of-order block contains chunks 26 to 28.
      - The second out-of-order block contains chunks 31 to 34.



- A variable holds the value of cumTSN.
- An array of variables keeps track of the beginning and the end of each block that is out of order.
- An array of variables holds the duplicate chunks received.
- There is no need for storing duplicate chunks in the queue and they will be discarded.

Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP Error Control
    - Sender Site
      - At the sender site, it needs two buffers (queues): a sending queue and a retransmission queue.
      - Three variables were used - rwnd, inTransit, and curTSN as described in the previous section.
      - The following figure shows a typical design.
      - The sending queue holds chunks 23 to 40.
      - The chunks 23 to 36 have already been sent, but not acknowledged; they are outstanding chunks.
      - The curTSN points to the next chunk to be sent (37).



- We assume that each chunk is 100 bytes, which means that 1400 bytes of data (chunks 23 to 36) is in transit.
- The sender at this moment has a retransmission queue.
- When a packet is sent, a retransmission timer starts for that packet (all data chunks in that packet).
- Some implementations use one single timer for the entire association, but other implementations use one timer for each packet.

Transport Layer

- Stream Control Transmission Protocol (SCTP)
  - SCTP CONGESTION CONTROL
    - SCTP is a transport-layer protocol with packets subject to congestion in the network.
    - The SCTP designers have used the same strategies for congestion control as those used in TCP.

Questions!