Transport Services and Protocols

Transport Services and Protocols

- The **transport layer** is responsible for providing **logical communication** between **application procceses** running on **different hosts**.
- There are two actions the transport layer needs to achieve:
  1. **Send action**: The transport layer needs to **divide application messages into segments**, and pass them to the **network layer**.
  2. **Recieve action**: The transport layer needs to **reassemble segments** from the **network layer** into **messages**, and pass them to the **application layer**.
- There are **two transport protocols** available: the **User Datagram Protocol (UDP)**, and the **Transmission Control Protocol (TCP)**.

The Application Layer, The Transport Layer, and The Network Layer

- The **application layer** is the **process** that is running on the **network host**.
- The **transport layer** is responsible for the **logical communcation** between **processes**.
- The **network layer** is responseible for the **logical communication** between **network hosts**.
- When a **message is sent** from an application over the network, it follow the following order: application layer → transport layer → network layer.
- When a **message is received** from the network, and sent to an application, it follows the following order: network layer → transport layer → application layer.

Transport Layer Actions

- When a message is **sent from an application**, the **transport layer** does the following:
  1. Recieves the application-layer message.
  2. Determines segment header field values.
  3. Creates the segment.
  4. Passes the segment to the internet protocol.
- When a message is **recieved from the network**, the **transport layer** does the following:
  1. Recieves the segment from the internet protocol.
  2. Checks the header values.
  3. Extracts the application-layer message.
  4. Demultiplexes message up to the application via a socket.

User Datagram Protocol and Transmission Control Protocol

- The **User Datagram Protocol (UDP)** is an **unreliable**, **unordered** delivery protocol. This protocol has **minimal overhead** but is **not reliable**.
- The **Transmission Control Protocol (TCP)** is a **reliable**, **in-order delivery** protocol that supports **congestion control**, **flow control**, and **connection setup**. This protocol has **significant overhead**, but is **very reliable**.
- Both **UDP** and **TCP** do not provide **delay guarantees** and **bandwith guarantees**.

Multiplexing

Multiplexing

- **Multiplexing** is a method used by **networks** to **consolidate multiple signals** into a **single composit signal** that is then **transported over a common medium**.
- When **sending data**, **multiplexing** is used to **transmit segments** from **several sockets** via **transport headers**.
- When **recieving data**, the **header info** is used to **demultiplex** the data, sending it to the **correct socket**.
- **Demultiplexing** works by receiving **IP datagrams** containing a **source and destination IP address and port as headers**. Each datagram contains a **single segment**.
  - The IP addresses and port numbers are used to route the segment to the correct socket.
  - This is why you must specify a port number when creating a socket.