CS 4457 Computer Networks Transport Layer Part 1 Lecture

4 October 2021

1 Introduction to the Transport Layer

The transport layer handles logical communication between *processes* (whereas the network layer handles communication between *hosts*). This means that the transport layer will place data packets that it receives from processes in the application layer in the correct order before sending it down to the NIC card. The transport layer does this ordering by using multiplexers and demultiplexers. On either side of the data transfer (both going from application to transport and from transport to application), a socket is needed. Sockets are an endpoint (or doorway) for receiving and sending data across the network, and there is a socket associated with every process. Sockets are located between the transport and application layers.

2 Transport Layer Protocols

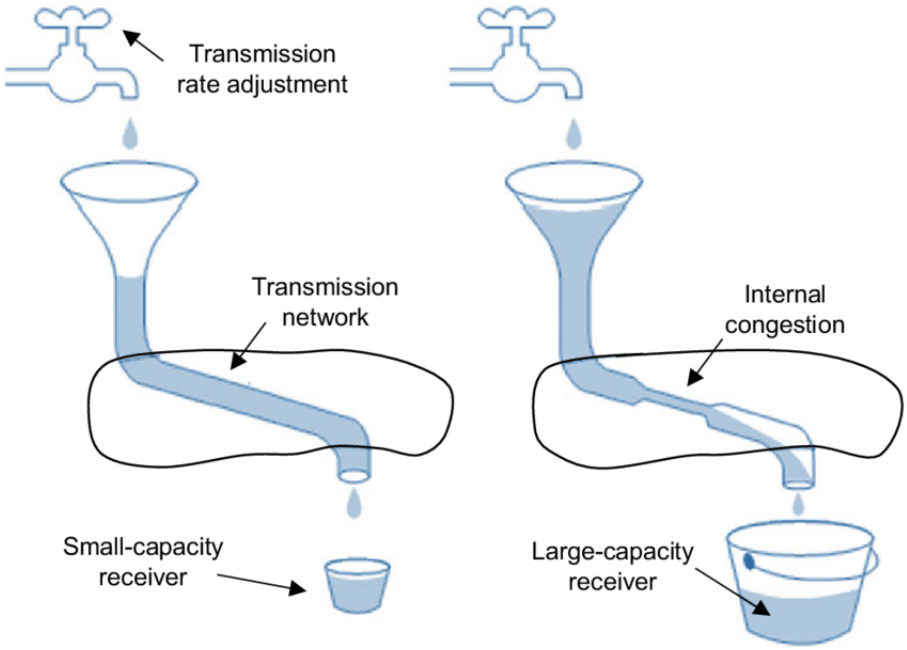
2 protocols used in the transport layer are UDP and TCP. In both of these protocols, there are no delay guarantees (there may be delays) and there are no bandwidth guarantees (there may be less bandwidth).

2.1 User Datagram Protocol (UDP)

- Unreliable (loses packets)
- Unordered delivery (will keep packets in the same order as they are received)
- No-frills extension of "best-effort" IP (meaning bare-bones delivery service (no extra features) and fastest service possible, so no reordering packets from the order in which they're sent and UDP segments may be lost due to this extra speed. Similar to the post office - sends letters as fast as possible but delayed when supplied too many letters at once).
- This protocol is faster but doesn't guarantee data delivery. It's used for gaming.

2.2 Transmission Control Protocol (TCP)

- Reliable (no packets are lost)
- In-order delivery (will reorder and reassemble packets when sent out of order).
- Flow control (avoids overloading the receiver node, meaning the sender node shouldn't send data faster than the receiver can consume. This is done by having the receiver node send the sender node data about its condition.)
- Congestion control (avoids too much congestion in the network, meaning sources are sending data too fast for the network to handle. This is done by giving the sender node a congestion window which indicates the maximum amount of data that can be sent on a connection without receiving an ACK from the receiver.)
- Connection setup (uses a 3-way handshake to establish a reliable connection. Client sends SYN, server sends SYN/ACK, client sends ACK.)



The figure above demonstrates flow control and congestion control for TCP. Flow control is shown on the left and congestion control is shown on the right.

3 Multiplexing and Demultiplexing

The transport layer uses multiplexing and demultiplexing to extend the host-to-host delivery by the network layer to process-to-process delivery for applications

running on the host. In simple terms, if 2 houses (hosts) send mail to each other, the postmen, who deliver all letters from one home to another, work in the network layer and the home-owners, who collect and disperse each letter from/to the household residents, work in the transport layer. Collecting and dispersing each letter from/to the different residents within the same house is done by multiplexing and demultiplexing.

Multiplexing is used for sending data. This process gathers data chunks from different sending node sockets in the host and converts them into data segments. Demultiplexing is used for receiving data. This process converts data segments back into data chunks and directs the data chunks to the correct receiving node sockets.

To multiplex/demultiplex, sockets must have unique identifiers and data segments must indicate the socket its being delivered to/from with source and destination port number fields.

3.1 Connectionless Multiplexing/Demultiplexing (UDP)

UDP is the only connectionless transport protocol. Here, a sender can directly send data to a receiver without first establishing a connection because this service is connectionless. Since there is no connection, the network is not checked for congestion, which is why data packets are sometimes lost. Thus, connectionless services are unreliable.

3.2 Connection-oriented Multiplexing/Demultiplexing (TCP)

TCP uses a connection-oriented transport protocol. Here, a connection (3-way handshake) is made between the sender and the receiver (sender sends SYN, receiver sends SYN/ACK, and sender sends ACK). Then, the sender can send its message to the receiver and the receiver can send a message back. After the messages have been sent, the sender will send a message to the receiver to terminate the connection. Thus, connection-oriented services are reliable because you know that the messages are being successfully transmitted to the receiver.

Here, a TCP socket is identified by a 4-tuple containing the source IP, source port number, destination IP and destination port number. You demultiplex with these values. Each socket is identified by its own 4-tuple and is associated with its own connection.

3.3 Addressing processes

There are several things that must be included for addressing another machine:

- To receive messages, a process must have an **identifier**. The identifier includes the IP address and port numbers associated with the host.

- Host device has a unique, 32-bit IP address.