Transport Layer
Study-Ready Notes

Compiled by Andrew Photinakis

November 15, 2025

Contents

1 Introduction to Transport Layer Services

- Transport layer provides logical communication between application processes running on different hosts

- Key services include:

  - Multiplexing and demultiplexing
  - Connectionless transport: UDP
  - Principles of reliable data transfer
  - Connection-oriented transport: TCP
  - Principles of congestion control
  - TCP congestion control
  - Evolution of transport-layer functionality

[Summary: The transport layer enables communication between application processes across networks, providing essential services like multiplexing, reliable data transfer, and congestion control through protocols like TCP and UDP.]

2 Transport Layer Overview

2.1 Learning Objectives

- Understand principles behind transport layer services:

  - Multiplexing and demultiplexing
  - Reliable data transfer
  - Flow control
  - Congestion control

- Learn about Internet transport layer protocols:

  - UDP: connectionless transport
  - TCP: connection-oriented reliable transport
  - TCP congestion control

3 Transport Services and Protocols

3.1 Core Functions

- Provides **logical communication** between application processes running on different hosts

- Transport protocols actions in end systems:

  - **Sender**: breaks application messages into **segments**, passes to network layer
  - **Receiver**: reassembles segments into messages, passes to application layer

- Two transport protocols available to Internet applications:

  - **TCP** (Transmission Control Protocol)
  - **UDP** (User Datagram Protocol)

3.2   Network Architecture Context

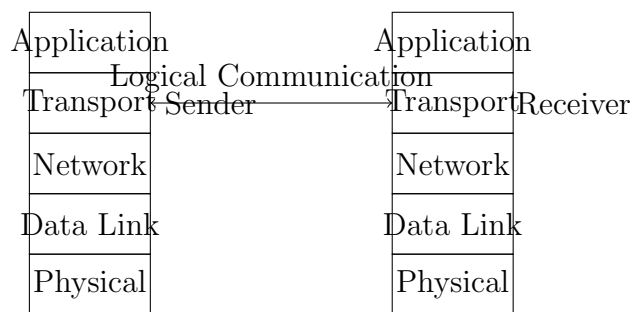| Application	Application
Transport	Transport
Network	Network
Data Link	Data Link
Physical	Physical

Logical Communication
Sender → Receiver

Figure 1: Transport layer in network protocol stack

[Summary: Transport protocols operate at end systems, segmenting application messages for transmission and reassembling them upon receipt, with TCP and UDP as the primary Internet protocols.]

4   Transport vs. Network Layer Services

4.1   Household Analogy

- **Hosts** = houses

- **Processes** = kids

- **Application messages** = letters in envelopes

- 12 kids in Ann's house sending letters to 12 kids in Bill's house

4.2   Layer Responsibilities

- **Network layer**: logical communication between hosts

- **Transport layer**: logical communication between processes

- Transport layer relies on and enhances network layer services

Network Layer	Transport Layer
Host-to-host communication	Process-to-process communication
Logical communication between hosts	Logical communication between processes
Provides basic datagram delivery	Enhances delivery with reliability, ordering, etc.

Table 1: Comparison of Network vs. Transport Layer Services

[Mnemonic: "Houses Host, Kids Process" - Network layer connects houses (hosts), Transport layer connects kids (processes) within houses.]

5 Transport Layer Actions

5.1 Sender Operations

1. Receives application-layer message

2. Determines segment header field values

3. Creates segment with header and payload
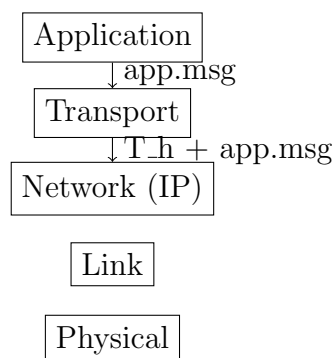
4. Passes segment to IP (network layer)

Application
↓ app.msg
Transport
↓ T_h + app.msg
Network (IP)

Link

Physical

Figure 2: Sender-side transport layer processing

5.2 Receiver Operations

1. Receives segment from IP (network layer)

2. Checks header values for correctness

3. Extracts application-layer message from segment

4. Demultiplexes message up to application via appropriate socket

Application

$\uparrow$ app.msg

Transport

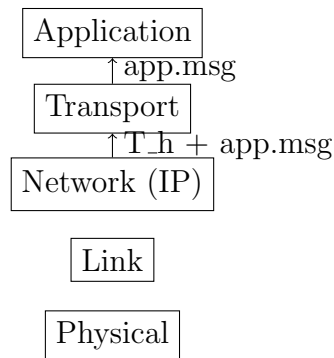$\uparrow$ T_h + app.msg

Network (IP)

Link

Physical

Figure 3: Receiver-side transport layer processing

[Summary: The transport layer segments application data at the sender and reassembles it at the receiver, handling header processing and message demultiplexing to the correct application process.]

6 Internet Transport Protocols

6.1 TCP: Transmission Control Protocol

- **Reliable, in-order delivery**: Ensures all data arrives correctly and in sequence

- **Congestion control**: Prevents network overload by adjusting transmission rate

- **Flow control**: Prevents overwhelming the receiver

- **Connection setup**: Requires establishment of connection before data transfer

6.2 UDP: User Datagram Protocol

- **Unreliable, unordered delivery**: No guarantees on delivery or ordering

- **No-frills extension of "best-effort" IP**: Minimal overhead beyond IP

- **Services not available**:

    - Delay guarantees
    - Bandwidth guarantees

[Concept Map: Transport Layer $\rightarrow$ TCP (reliable, connection-oriented, flow/congestion control) vs UDP (unreliable, connectionless, minimal overhead) $\rightarrow$ Applications choose based on reliability vs performance needs.]

TCP Features	UDP Features
Connection-oriented	Connectionless
Reliable delivery	Best-effort delivery
In-order delivery	No ordering guarantees
Flow control	No flow control
Congestion control	No congestion control
Higher overhead	Lower overhead
Use cases: Web, email, file transfer	**Use cases**: DNS, VoIP, streaming

Table 2: Comparison of TCP vs. UDP Protocols

7 Study Aids

7.1 Key Concepts

- Understand the difference between network layer (host-to-host) and transport layer (process-to-process) communication

- Memorize the characteristics and use cases for TCP vs UDP

- Be able to describe the segmentation process at sender and reassembly at receiver

- Know the household analogy for understanding layer responsibilities

7.2 Practice Questions

1. **Compare and contrast** the services provided by TCP and UDP. When would you choose one over the other?

2. Describe the process of **multiplexing and demultiplexing** at the transport layer. How does the transport layer ensure messages reach the correct application process?

3. Explain the **household analogy** for understanding the difference between network and transport layer services.

4. What are the key **transport layer actions** performed by the sender and receiver during data transmission?

5. Why is **congestion control** an important transport layer function, and which protocol provides it?

[Mnemonic: "TCP: Reliable Connection, UDP: Unreliable Datagram" - TCP establishes reliable connections while UDP sends unreliable datagrams.]