What Happens When We Type GOOGLE.com and Hit Enter

IN TRANSPORT LAYER

A Comprehensive Analysis Based on the TCP/IP Model

Submitted By:
Aayush Adhikari,
Roshan Tiwari,
Shishir Sharma Rijal,
Sudip Acharya

July 22, 2024

1 Introduction

When a user types "google.com" into a web browser's address bar and presses Enter, the transport layer plays a crucial role in ensuring reliable, ordered, and error-checked delivery of data between the user's device and Google's servers. This document provides an in-depth analysis of the processes occurring at the transport layer during this interaction.

2 Transport Layer Overview

Transport Layer Functions

The transport layer, the second layer , is responsible for end-to-end communication between applications. Its primary functions include:

- Segmentation and reassembly of data
- Connection-oriented and connectionless communication
- Reliability through error detection and recovery
- Flow control
- Congestion control
- Multiplexing and demultiplexing of application data

The two main protocols in the transport layer are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

3 Detailed Transport Layer Processes

3.1 Protocol Selection

Protocol Selection Process

1. **Application Requirements Analysis:**
   - Web browsing typically requires reliable, ordered delivery
2. **Protocol Choice:**
   - TCP is chosen for HTTP/HTTPS connections to google.com
   - UDP might be used for some background services or DNS lookups
3. **Port Number Assignment:**
   - Client OS assigns an ephemeral source port (typically > 1024)
   - Destination port is set to 80 (HTTP) or 443 (HTTPS) for google.com

3.2 TCP Connection Establishment (Three-Way Handshake)

TCP Handshake Process

1. **SYN Packet:**
   - Client sends SYN packet with initial sequence number
   - SYN flag set, ACK flag clear
2. **SYN-ACK Packet:**
   - Server responds with SYN-ACK packet
   - Acknowledges client's sequence number
   - Sends its own initial sequence number
3. **ACK Packet:**
   - Client sends ACK packet
   - Acknowledges server's sequence number
4. **Connection Establishment:**
   - After ACK, connection is established
   - Both sides can begin sending data

3.3 Data Segmentation

Segmentation Process

1. **Receiving Application Data:**
   - Transport layer receives data stream from application layer
2. **Maximum Segment Size (MSS) Determination:**
   - Typically negotiated during connection setup
   - Often based on MTU (Maximum Transmission Unit) of the network
3. **Segmentation:**
   - Divides data into segments of appropriate size
   - Adds TCP header to each segment
4. **Sequence Number Assignment:**
   - Assigns sequence numbers to ensure proper ordering

3.4 Flow Control

Flow Control Mechanisms

- **Window Size Advertisement:**
  - Receiver advertises available buffer space in window field
  - Sender limits data in flight to this window size
- **Dynamic Window Sizing:**
  - Window size can change during transmission
  - Allows receiver to slow down sender if necessary
- **Zero Window Handling:**
  - If receiver advertises zero window, sender stops transmission
  - Sender periodically sends window probe packets

3.5 Error Detection and Correction

Error Handling Mechanisms

1. **Checksum Calculation:**
   - Calculates checksum over segment header and data
   - Includes pseudo-header with IP addresses for additional protection
2. **Acknowledgment System:**
   - Receiver acknowledges successfully received data
   - Uses cumulative ACKs: ACK number indicates next expected byte
3. **Retransmission:**
   - Sender retransmits unacknowledged segments after timeout
   - Fast Retransmit: retransmit after receiving duplicate ACKs

3.6 Congestion Control

Congestion Control Mechanisms

- **Slow Start:**
  - Begins with small congestion window
  - Exponentially increases window size until threshold
- **Congestion Avoidance:**
  - Linear increase of congestion window after threshold
  - Aims to find equilibrium point
- **Fast Recovery:**
  - After packet loss, reduces window but not to initial size
  - Allows quicker recovery from congestion
- **Explicit Congestion Notification (ECN):**
  - If supported, allows routers to signal congestion
  - Helps prevent packet drops due to congestion

3.7 Data Transmission

Data Transmission Process

1. **Segment Preparation:**
   - Adds necessary headers (sequence number, checksum, etc.)
2. **Transmission:**
   - Passes segments to IP layer for transmission
3. **Acknowledgment Handling:**
   - Processes incoming ACKs
   - Updates sliding window and congestion window
4. **Retransmission Timer Management:**
   - Sets and adjusts retransmission timers
   - Uses adaptive algorithms to estimate appropriate timeout values

3.8 Data Reception and Reassembly

Data Reception Process

1. **Segment Reception:**
   - Receives segments from IP layer
2. **Checksum Verification:**
   - Verifies segment integrity using checksum
3. **Sequence Number Check:**
   - Ensures segments are in correct order
   - Handles out-of-order segments
4. **Acknowledgment Generation:**
   - Sends ACKs for received segments
   - May use delayed ACKs to reduce overhead
5. **Data Reassembly:**
   - Reassembles segments into complete data stream
   - Passes reassembled data to application layer

3.9 Connection Termination

TCP Connection Termination Process

1. **FIN Packet (Initiator):**
   - Either client or server sends FIN packet
2. **ACK Packet (Receiver):**
   - Other side acknowledges FIN
3. **FIN Packet (Receiver):**
   - Other side sends its own FIN when ready to close
4. **ACK Packet (Initiator):**
   - Original initiator acknowledges final FIN
5. **Time-Wait State:**
   - Initiator enters TIME-WAIT state
   - Ensures all packets have died out in network

4 UDP Considerations

UDP Operations

While TCP is the primary protocol for web browsing, UDP may be used for some aspects:

- **DNS Lookups:**
  - UDP is typically used for DNS queries (port 53)
  - Provides faster, lightweight communication for name resolution
- **QUIC Protocol:**
  - Google's QUIC protocol uses UDP as its base
  - Provides TCP-like reliability with reduced latency
- **Stateless Nature:**
  - No connection establishment or termination
  - Each datagram handled independently
- **Simple Header:**
  - Contains only source and destination ports, length, and checksum

5 Conclusion

The transport layer plays a vital role in the process of accessing "google.com" by providing reliable, ordered, and error-checked delivery of data between the user's device and Google's servers. It handles tasks such as connection establishment, data segmentation, flow control, error detection, and congestion control. Understanding these processes is crucial for network administrators, developers, and anyone involved in optimizing network performance or troubleshooting connectivity issues.