



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Technical University Darmstadt

– Department of Computer Science –

Application-Layer Protocols on the Internet

A Summary of the Course Contents

Author:

Jonas Weißner

Teaching professor : Prof. Björn Scheuermann

Semester : Winter semester 2022/2023

CONTENTS

1	Filesharing, Overlays and Robustness	1
2	Basic Terms and Definitions	1

LIST OF ABBREVIATIONS

MAC	Medium Access Control
IP	Internet Protocol
HTTP	Hypertext Transfer Protocol
SSH	Secure Socket Shell
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
NTP	Network Time Protocol
NAT	Network Address Translation
RTT	Round Trip Time
POP ₃	Post Office Protocol version 3
IMAP	Internet Message Access Protocol
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
TLS	Transport Layer Security
PPP	Point to Point Protocol
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
EGP	exterior gateway protocol
IGP	interior gateway protocol
BGP	Border Gateway Protocol
OSPF	Open Shortest Path First
EIGRP	Enhanced Interior Gateway Routing Protocol
AS	autonomous system

1 FILESHARING, OVERLAYS AND ROBUSTNESS

2 BASIC TERMS AND DEFINITIONS

In this chapter, basic terms will be defined and explained.

1. **TCP/IP** Conceptual Layers:

- a) *Application*: Standardizes communication interfaces built on Transport Layer protocols for specific classes of applications.

Protocols: Hypertext Transfer Protocol (**HTTP**), Secure Socket Shell (**SSH**), Domain Name System (**DNS**), Dynamic Host Configuration Protocol (**DHCP**), Network Time Protocol (**NTP**), Post Office Protocol version 3 (**POP3**), Internet Message Access Protocol (**IMAP**), Simple Mail Transfer Protocol (**SMTP**), Secure Sockets Layer (**SSL**)/Transport Layer Security (**TLS**) (...)

- b) *Transport*: Provide process-to-process communication for application using ports. Features which might be given depending on the protocol are connection-oriented communication, reliability, error correction, flow-control and multiplexing (e.g. multicast / broadcast).

Protocols: Transmission Control Protocol (**TCP**), User Datagram Protocol (**UDP**) (...)

- c) *Internet*: Routing and transmission from source to destination.

Protocols: Internet Protocol (**IP**) (v4, v6) (...)

- d) *Network Interface*: Transmission of data between two computers in the same network via physical connection.

Protocols: Medium Access Control (**MAC**), Tunnels, Point to Point Protocol (**PPP**) (...)

- 2. **Datagram**: A datagram is a basic transfer unit in a packet-switched network. Datagrams consist of a header and payload. They contain address information and can be sent in connectionless communication.

- 3. **User Datagram Protocol (UDP)**: **UDP** provides a simple connectionless communication model. It provides checksums for verification of data integrity and port numbers for addressing specific services on the target machine. It does not have handshaking dialogues and requires no previous packets to be sent prior to communication. Messages are mapped directly to packets and are not split up into pieces. **UDP** does not provide a guarantee of delivery, order of packets, or duplicate protection. It therefore exposes the application layer to any unreliability of the network for the sake of less communication overhead. It is suitable for applications which do not require reliability but do require speed e.g. in video streaming or for applications which handle resubmission, ordering and duplicate checking themselves.

4. **Transmission Control Protocol (TCP):** TCP follows a connection-oriented communication model providing reliability, ordering and error checking and flow-control (not allowing one side to send too fast). Messages are handled as a stream of bytes which is split into packets of undefined size for transmission.
- a) *Connection establishment and termination:* The connection is established using a three-way handshake consisting of the packets SYN \rightarrow , SYN-ACK \leftarrow and ACK \rightarrow . The connection termination is done with a four-way handshake consisting of the packets FIN \rightarrow , ACK \leftarrow , FIN \leftarrow and ACK \rightarrow . It is possible to shorten this by sending a FIN-ACK as a reply to the FIN
 - b) *Reliability:* TCP uses sequence numbers to identify each byte of data, as shown in figure 1. The sequence number of the first byte is randomly chosen by the sender of the first packet in order to defend against TCP sequence prediction attacks. Each TCP-packet contains a sequence number and – if it is an ACK packet – an acknowledgement number. The sequence number of the packet is the sequence number of the first byte send in this message or the sequence number of the last byte send in previous messages in case the payload (data) of the packet has size zero (as for ACK packets). The acknowledgement number is the incremented sequence number of the last byte received. TCP uses cumulative ACKs, which means that an acknowledgement number of n acknowledges all bytes with sequence number $< n$.
 - c) *Resubmission:* If a single segment in a stream is lost, then the receiver cannot acknowledge subsequent packets, because of the cumulative acknowledgement semantics. TCP uses two mechanisms to trigger the resubmission of lost packets. The first one is *resubmission on duplicate ACKs*, which means the receiver will send ACKs for every subsequently received packet containing the last sequence number before missing packet. If the sender receives three ACKs for the same packet, it will trigger the resubmission of the missing packet. The other mechanism for resubmission is *timeout-based resubmission*, which means that packets will be resubmitted if they are not acknowledged within a given time frame. After a timeout has been triggered, the timeout for the resubmitted packet will be doubled.
 - d) *Flow Control:* TCP uses a sliding window flow control protocol. The receiver specifies in the *receive window* field of the packet how many bytes it is able to (or willing to) buffer for this connection. The sender can only send that amount of bytes without receiving an ACK, because the ACK signalizes that bytes up to the given acknowledgement number are out of the buffer. This prevents slower machines being overwhelmed with packets, which is important in the internet, because end devices can be very different.

- e) *Error Detection*: In order to detect submission errors, a checksum field is concluded. However, most errors are also corrected by protocols of underlying layers such as in the Ethernet frame.

5. **autonomous system (AS)**: The internet is a network of computer networks instead of just a huge network of computers. The big networks, which the internet consists of and which are handled independently of their internal structure are called **ASs**. This principle allows for scalability, because responsibilities are distributed. In today's world, **ASs** are administrative domains hosted by big companies, which commercially offer access to all computers that can be accessed inside or through this system to other **ASs**. The routing between **ASs** is organized using a so-called exterior gateway protocol (**EGP**), while routing inside of **ASs** is managed using so-called **IGPs**. The **EGP** used in today's internet is the Border Gateway Protocol (**BGP**). For **EGP** the host of the **AS** can choose from multiple protocols such as e.g. Open Shortest Path First (**OSPF**) or Enhanced Interior Gateway Routing Protocol (**EIGRP**) depending on the internal network topology. Only the edge gateways of an **AS** are responsible for the routing between **ASs**, such that internal routers only need to handle internal routing information.
6. **Border Gateway Protocol (BGP)**: The **BGP** is today's **EGP** connecting **ASs**. If a new IP address is registered inside of an **AS**, this will be reported to the edge router of the **AS** via an interior gateway protocol (**IGP**). This edge router then transmits the information about the accessibility of this IP address via this **AS** to the edge routers of all other **ASs** it is connected to. These edge routers then share this information to all other edge routers of their **AS**, which further share it to other routers of other connected **ASs**. As a result, each edge router of an **AS** will know what IP addresses are available inside or behind its own network. They however will not know the exact path, unless the target IP address is located inside of the own **AS**. When a computer inside of one **AS** wants to send a packet to a computer in another **AS**, an **IGP** will lead the packet to the edge router of the **AS**, which will then send it to another suitable **AS**. The receiving edge router then internally routes it to the computer, if it is located in that **AS**, or to another edge router which is connected to other **ASs**, which are closer to the destination IP address. This way, routers in an **AS** do not need to know the internal structure of other **ASs**, which makes this approach scalable.
7. **Round Trip Time (RTT)**: Is the transmission time for a packet from point A to point B plus the transmission time of the ACK-packet from point B back to point A. It can be determined using the ping command.
8. **Network Address Translation (NAT)**: In **IPv4** there are not enough different IP addresses such that every computer on the world could be

assigned a unique address. Therefore the internet is divided into private networks, which form the public network – the internet. Inside of private networks, private IP addresses are used, which are unique inside of that private network but not unique for the whole internet. Private IP addresses start with either 10.*, 172.16* or 192.168*. Only the gateway, which connects the private network with the internet, has a public IP address. If a computer inside of a private network wants to connect to a computer outside of that private network, it sends its packet to the default gateway, which is its connection to the public network. The gateway then opens a session for that connection attempt which maps the port of the private host to a given port of the gateway and stores that information in the so called [NAT-table](#). It then replaces the private IP address and port with its own public address and the port assigned to that connection and forwards it to its destination (possibly via borders of autonomous systems). If the receiver wants to reply to the host in the private network, it sends the reply to the global IP address of the gateway connected to the private network. Then the gateway router looks at the port of the incoming packet and translates the IP address back to the private IP address and port based on the port mapping specified in the [NAT-table](#). By the way, this means, that publicly available services must have their own public IP address or receive a dedicated static mapping in order to be available from outside of their private networks. [NAT](#) is criticized, because it mixes IP addresses with port numbers, which originate from different layers of the [TCP/IP](#) protocol stack – the *internet* and the *transport* layer.

9. Subnetting: TODO:

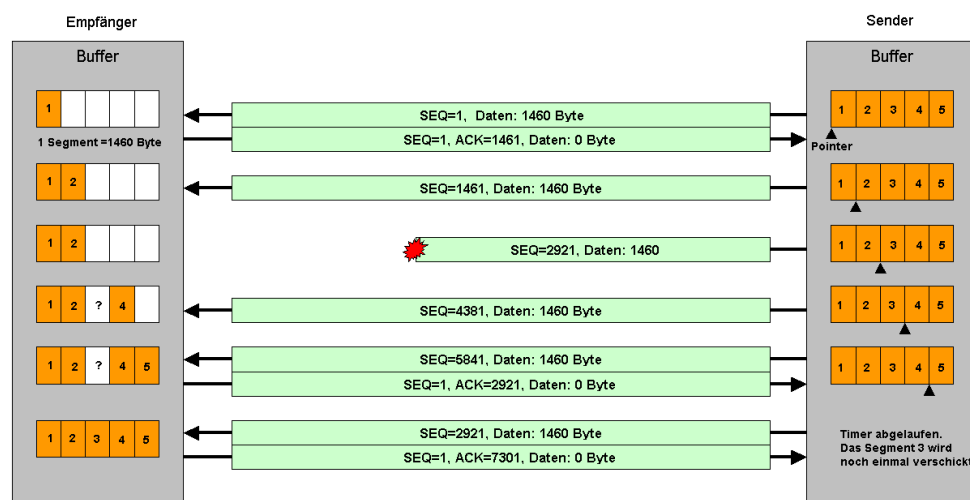


Figure 1: TCP data transmission with sequence numbers