
User Datagram Protocol (UDP)

Netzwerkgrundlagen (NWG2)

Markus Zeilinger¹

¹FH Oberösterreich
Department Sichere Informationssysteme

Sommersemester 2023



UNIVERSITY
OF APPLIED SCIENCES
UPPER AUSTRIA

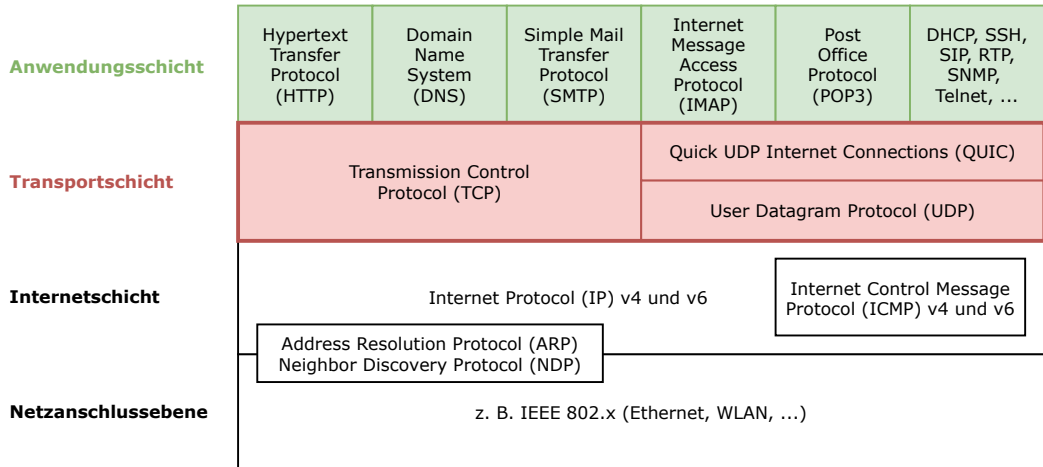
*Alle Materialien, die im Rahmen dieser LVA durch den LVA-Leiter zur Verfügung gestellt werden, wie zum Beispiel **Foliensätze**, **Audio-Aufnahmen**, **Übungszettel**, **Musterlösungen**, ... dürfen ohne explizite Genehmigung durch den LVA-Leiter **NICHT** weitergegeben werden!*

Transportschicht allgemein

Sockets, Ports & Transportdienste

User Datagram Protocol (UDP)

Transportschicht in der TCP/IP Protokollfamilie



Bedarfsgerechter Transportdienst zwischen zwei Prozessen (Anwendungen) auf (nicht notwendigerweise) verschiedenen Systemen (Ende-zu-Ende, End-to-End).

- ▶ Bedarfsgerecht = so wie es von der Anwendung gebraucht wird
- ▶ Ende-zu-Ende = es kommunizieren die absoluten Endpunkte (es gibt nur mehr den Nutzer vor dem Rechner)

Sockets & Ports

Anwendungen + Transportprotokoll + Port (Beispiele)

Anwendung	Transportprotokoll	Port
File Transfer Protocol (FTP)	TCP	21
Secure Shell (SSH)	TCP	22
Simple Mail Transfer Protocol (Secure) (SMTP(S))	TCP	25, 465, 587
Domain Name System (DNS)	UDP + TCP	53
Dynamic Host Configuration Protocol (DHCP)	UDP	67, 68
Hypertext Transfer Protocol (Secure) (HTTP(S)) ≤ 2	TCP	80, 443
Hypertext Transfer Protocol Secure (HTTPS/3)	UDP + QUIC	443
Post Office Protocol (Secure) (POP3(S))	TCP	110, 995
Network Time Protocol (NTP)	UDP	123
Internet Message Access Protocol (Secure) (IMAP4(S))	TCP	143, 993
Simple Network Management Protocol (SNMP)	UDP	161, 162
Realtime Transfer Protocol (RTP)	UDP	Dynamisch
Remote Desktop Protocol (RDP)	TCP + UDP	3389
Session Initiation Protocol (SIP)	TCP + UDP	5060, 5061

- Tools netstat (deprecated) bzw. ss: Anzeigen von Sockets und (aktiven) Verbindungen auf einem System.

```
$ sudo netstat -Wtaupen
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       User        Inode       PID/Program name
tcp      0      0 127.0.0.1:587           0.0.0.0:*                LISTEN      0           23349        2194/sendmail: MTA:
tcp      0      0 127.0.0.1:6379          0.0.0.0:*                LISTEN      110          33821        1587/redis-server 1
tcp      0      0 0.0.0.0:53              0.0.0.0:*                LISTEN      0           36045        2304/dnsmasq
tcp      0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      0           24572        1206/sshd
tcp      0      0 127.0.0.1:5432          0.0.0.0:*                LISTEN      107          1699        1385/postgres
tcp      0      0 127.0.0.1:25            0.0.0.0:*                LISTEN      0           23348        2194/sendmail: MTA:
tcp      0      1 185.252.75.38:22        218.92.0.211:42846     FIN_WAIT1   0            0            -
tcp      0      220 185.252.75.38:22        90.146.138.18:53328    ESTABLISHED 0           1561065      25305/sshd: monitor
tcp6     0      0 :::1:6379                :::*                   LISTEN      110          33822        1587/redis-server 1
tcp6     0      0 :::53                    :::*                   LISTEN      0           36047        2304/dnsmasq
tcp6     0      0 :::22                    :::*                   LISTEN      0           24574        1206/sshd
tcp6     0      0 :::1:5432                :::*                   LISTEN      107          1698        1385/postgres
tcp6     0      0 :::8443                  :::*                   LISTEN      0           16168        2322/python3
udp      0      0 0.0.0.0:53              0.0.0.0:*                0           0           2304/dnsmasq
udp      0      0 0.0.0.0:67              0.0.0.0:*                0           0           2304/dnsmasq
udp      0      0 0.0.0.0:68              0.0.0.0:*                0           0           919/dhclient
```


User Datagram Protocol (UDP) II

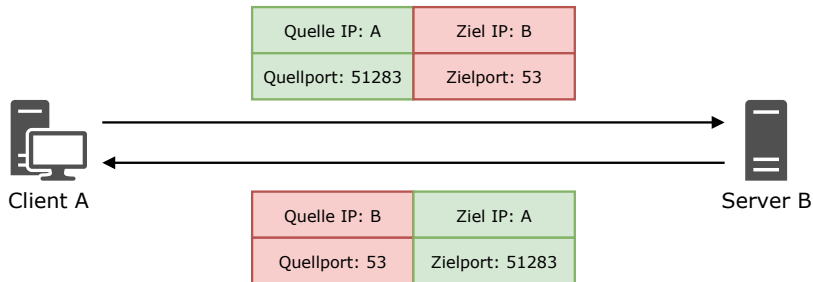
Header



- ▶ **Source (Quell-) Port:** Port des sendenden Prozesses am Quellsystem.
 - ▶ Client → Server: Dynamic/Private Port > 49152
 - ▶ Server → Client: Well-Known oder Registered Port des Services
- ▶ **Destination (Ziel-) Port:** Port des empfangenden Prozesses am Zielsystem.
 - ▶ Client → Server: Well-Known oder Registered Port des Services
 - ▶ Server → Client: Dynamic/Private Port > 49152
- ▶ **Length (Länge):** Länge des Datagramms mit dem Header in Bytes (mind. 8).

User Datagram Protocol (UDP) III

Quell-/Ziel-Information



User Datagram Protocol (UDP) IV

Wichtige Aspekte

- ▶ UDP betrachtet von der Anwendung kommende Daten (via `sendto()` Call) als eine **untrennbare Einheit** (ein **Byte-Block**) → Anwendungsdaten werden wie von dort kommend paketiert und an die Netzwerkschicht weitergegeben.
- ▶ UDP lässt der Anwendung die **Freiheit**, selbst über weitere Funktionalitäten zu entscheiden. UDP stellt eine **Andockmöglichkeit an IP** zur Verfügung.
- ▶ → Anwendungen können daher z. B. durchaus **Zuverlässigkeit über UDP** realisieren (z. B. **QUIC** für HTTP/3).

