

Netzwerke – Übung WiSe2018/19

OSI-Stack & Wireshark

Benjamin.Troester@HTW-Berlin.de

PGP: ADE1 3997 3D5D B25D 3F8F 0A51 A03A 3A24 978D D673

Benjamin Tröster

Road-Map



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

1 Aktueller Stand

2 Ethernet + IP → ARP

- Ethernet
- IP & ARP
- ARP

3 IP

4 TCP

5 UDP

6 Ports

7 Sockets

8 Wireshark

Aktueller Stand



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

- Zwei LANs pro Tischreihe durch Backbone-Netz verbunden.
 - Alle Rechner können miteinander kommunizieren.
- Uplink ins Internet/DFN → Default-Gateway 10.10.10.254 samt DNS
 - Alle Rechner könne auch außerhalb Ihres LAN|WAN-Segments kommunizieren.

Ethernet + IP → ARP

- Switch & Ethernet: Link Layer
- Switch ist physikalisch ein (Multi-)Bus
 - direkt „über“ dem Physical Layer (wo die Bits fließen)
- Ethernet-Switches können (meistens) nur Ethernet-Frames versenden
- Zuordnung von MAC-Adressen zu IP-Adressen sorgt für Adressauflösung

Ethernet + IP → ARP



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

- IP: Network Layer
- IP → routing-fähig sorgt durch das Routing für Finden von Routen sowie Ausliefern der Pakete
- Woher weiß der Router an wen das Paket schlussendlich geht?
 - Router auf Layer 3 – muss also zwangsläufig die Layer 2 & 1 durchlaufen

Ethernet + IP → ARP



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

- IP: Network Layer
- IP → routing-fähig sorgt durch das Routing für Finden von Routen sowie Ausliefern der Pakete
- Woher weiß der Router an wen das Paket schlussendlich geht?
 - Router auf Layer 3 – muss also zwangsläufig die Layer 2 & 1 durchlaufen
 - → kein Rückgriff auf reines *IP* möglich

Ethernet + IP → ARP

- IP: Network Layer
- IP → routing-fähig sorgt durch das Routing für Finden von Routen sowie Ausliefern der Pakete
- Woher weiß der Router an wen das Paket schlussendlich geht?
 - Router auf Layer 3 – muss also zwangsläufig die Layer 2 & 1 durchlaufen
 - → kein Rückgriff auf reines *IP* möglich
 - Auflösung der physikalischen Adresse mithilfe *ARP*'s

ARP

- ARP gehört zum Link Layer
- Ergo: nur begrenzt auf das LAN-Segment
- Wie ermittelt ARP Zuordnung IP zu MAC-Adresse?
 - Nachschlagen im ARP-Table (ARP Cache): Ist IP-Adresse und somit MAC-Adresse bekannt?
 - Falls kein passender Eintrag: Broadcast ARP-Message – geht an alle Geräte im LAN
 - Wenn Rechner auf ARP-Request antwortet: Eintrag in ARP-Table

Internet Protocol (IP)

- IP: aktuell IPv6 & legacy IPv4
- Gehören zum Network Layer
- Paketerorientiert – Pakete könne unterschiedliche Routen nehmen, kann dazu führen, dass Pakete in anderer Reihenfolge ankommen als sie abgeschickt wurden
- Zwei Teile:
 - IP-Header: Inkl. Quell IP-Adr., Ziel IP & Metadaten zum routen| ausliefern
 - Payload: Eigentlicher Inhalt – Nutzdaten aus höherer Schicht → TCP|UDP|... Datagram

Transmission Control Protocol (TCP)

- *TCP*:
- Gehört zum Transport Layer
- Definiert Art und Weise des Datenaustausches
- Verbindungsorientiert: Üblicherweise *Three-Way-Handshake* – *SYN,SYN-ACK,ACK*
- Zwei Teile:
 - Header: Inkl. Quell Port, Ziel Port, Seq. No., ACK-No.,...Window-Size, Checksum,...
 - Payload: Eigentlicher Inhalt (Content höherer Schichten – http, XML,...)

User Datagram Protocol (UDP)

- *UDP:*
- Gehört zum Transport Layer
- Definiert Art und Weise des Datenaustausches
- Verbindungslos: kein Aufbau einer virtuellen Verbindung („Fire & Forget“)
- Zwei Teile:
 - Header: Inkl. Quell Port, Ziel Port, Length, Checksum
 - Payload: Eigentlicher Inhalt (Content höherer Schichten – http, XML,...)

Ports

- Endpunkt einer Kommunikation im Betriebssystem
- Betriebssystem ordnet Anwendung (Prozess)/Netzwerkdienst Port zu
- Bestimmt somit Quelle & Ziel einer Kommunikation
- Well-Know-Ports:
 - 0-1024 – Ports für Dienste die viel genutzt werden (Standardports)
 - HTTP: 80/ HTTPS: 443
 - SSH: 22
 - Telnet: 23
 - DNS: 53
 - SMTP: 25, IMAPS: 585/993

Sockets



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

- Da auf einem System mehrere Prozesse über Netzwerkschnittstelle(n) kommunizieren können
- Sockets kombinieren IP-Adresse und Port
- → Tupel: (IP, Port)
 - Bspw.: (141.45.146.48, 22) – SSH auf dem Uranus-Server
- Mithilfe der Sockets kann also genau angegeben werden wo sich die Endpunkte der Kommunikation befinden

Road-Map
Aktueller Stand
Ethernet + IP → ARP
IP
TCP
UDP
Ports
Sockets
Wireshark



- Network Sniffer - setzt auf *libcap* auf
- Erlaubt Mitschneiden und Auswerten des Netzwerkverkehrs
- <https://www.wireshark.org/>
- Doku: https://www.wireshark.org/docs/wsug_html_chunked/ → ab Chpt. 3.3 wird es interessant

