

## LF09:04:HUB & ARP

### 1) Definitionen

#### 1.A) Repeater [→ ZP:Sheet:2]

Netzwerkgerät der Bitübertragungsschicht mit zwei Schnittstellen, [das] das "Netz vergrößert, indem es die Signale weiterleitet, ohne sie zu analysieren" (vgl. Baun: Computernetze, 2022, S. 276)

#### 1.B) Hub [→ ZP:Sheet:2]

"[...] ein Repeater mit mehr als zwei Schnittstellen." (vgl. Baun: Computernetze, 2022, S. 274)

Konsequenz:

"Jedes Endgerät in einem Netzwerksegment, das durch Konzentratoren (Hubs) aufgebaut ist, sieht alle Datenpakete." (vgl. Schreiner: Computernetzwerke, 2014, S. 43)

### 2) Nachrichtenaustausch in einem Hub-basierten Netz [→ ZP:Sheet:3]

Auf der Layer-II-Ebene

1. wird die Kommunikation nur über die MAC-Adressen der Netzwerkkarten/-interfaces organisiert.
2. dienen IP-Adressen als Namen der Rechner.
3. gelten MAC-Adressen max. 300ms (um das Wiederanlaufen einer Kommunikation beim Austausch einer Netzwerkkarte (= neue MAC-Adresse) zu ermöglichen)
4. sendet der Hub (als "Repeater mit mehreren Anschlüssen" (Ports)) eine über einen Port eingegangenen Nachricht **immer** an alle anderen Ports und damit an alle je dort angeschlossenen Geräte weiter.

#### Addressierungsregeln:

Man darf sich die Zusammenarbeit zwischen Layer II (MAC-Adresse) und Layer III (IP-Adresse) wie beim Adressieren eines Briefes vorstellen:

- Die **IP-Adresse** ist der **Name** des Rechners.
- Die **MAC-Adresse** die Adresse des Rechners (seiner Netzwerkkarte).

Und wir beim Schreiben/Beantworten eines Briefes

1. *muss man den Namen des Empfängers wissen und dessen Adresse kennen.*
2. *schreibt man ...*

- 
- seine eigene Adresse (aus Adressfeld des erhaltenen Briefes ;-) ) in den Absenderbereich des Antwortbriefes
  - die Absenderadresse vom erhaltenen Brief in das Adressfeld des Antwortbriefes
- 

### ÜBUNG LF09:04:HUB&ARP:01

- Erfinden Sie - wie beim Fadennetzwerk - ein passendes Protokoll, mit dem Nachrichten in einem (Hub-basierten) Layer-II-Netzwerk auch unter der Bedingung verlässlich übermittelt werden, dass die MAC-Adresse über längere Zeiträume nicht verlässlich konstant ist.
- 

### Übliche Lösungen:

1. Jeder Rechner sendet - unter Ausnutzung der Hub-Funktionalität - seine Adressdaten in kurzen Abständen an alle anderen Rechner: diese updaten ihr ‘Telefonbuch’. [Dank an Jörn Unverzagt 11IA24] [Nachteil: Viel ‘Telefonbuchtraffic’]
2. Jeder Rechner, dessen Netzwerkkarte getauscht wird, sendet (per Hub) einen ‘Adressbuch-Update-Befehl’ an alle anderen Rechner, der seine IP-Adresse, seine alte MAC-Adresse und seine neue MAC-Adresse enthält. [Dank an Tobias Häuser, 11IA23] (Vorteil: viel weniger ‘Telefonbuchtraffic’, im Worstcase aber trotzdem.)
3. Wer eine Nachricht senden will, fragt zuerst alle nach ihrer MAC-Adresse, sofern sie der Zielrechner mit der gewünschten IP-Adresse sind.

Lösung 3 ist die Idee vom **ARP (Adress-Resolution-Protokoll)**:

Über/im ARP-Request und ARP-Response/-Reply

- erfährt ein Rechner die MAC-Adresse des Rechners, mit dem er kommunizieren will.
- referenziert der *Source-Rechner* den *Destination-Rechner* mit dessen Namen (= IP-Adresse).
- benutzt das (hier: vereinfachte!) Datenpaket:

[ **SIP-Address SMAC-Address DIP-Address DMAC-Address Payload** ]

### ARP-Algorithmus:

1. Der Rechner RX mit dem Wunsch, RY eine Nachricht zu schicken, sendet einen ARP-Request der Form [ **RX-IP RX-MAC RY-IP FF 0** ] an den Hub.
2. Der Hub sendet alle Nachricht (ARP-Request, ARP-Reply, Message) über jeden anderen Port an das je dort angeschlossene Gerät.

3. Die Empfänger empfangen und verstehen die Nachrichten:

- Wer seine eigene IP-Adresse (Namen) **nicht** im Destination-Bereich findet, ignoriert die Nachricht.
- Wer seine eigene IP-Adresse (Namen) im Destination-Bereich findet,
  - versteht die Ziel-MAC-Adresse FF als ARP-Request und sendet einen ARP-Reply der Form [ RY-IP RY-MAC RX-IP RX-MAC 0] an den Hub. [GOTO 2]
  - versteht die Nachricht [ RY-IP RY-MAC RX-IP RX-MAC 'Hello RY']
  - oder
    - \* entnimmt der Nachricht [ RY-IP RY-MAC RX-IP RX-MAC 0] die erfragte RY-MAC-Adresse UND
    - \* sendet seine Nachricht [ RY-IP RY-MAC RX-IP RX-MAC 'Hello RY'] an den Hub. [GOTO 2]

Das heißt:

- RX sendet [ RX-IP RX-MAC RY-IP FF 0] an Hub
- Hub sendet [ RX-IP RX-MAC RY-IP FF 0] an alle
- RZ ignoriert [ RX-IP RX-MAC RY-IP FF 0]
- RY antwortet mit [ RY-IP RY-MAC RX-IP RX-MAC 0] an Hub
- RZ ignoriert '[ RY-IP RY-MAC RX-IP RX-MAC 0]
- RX reagiert mit [ RY-IP RY-MAC RX-IP RX-MAC 'Hello RY']

#### **Disclaimer: [→ ZP:Sheet:4]**

Die Nachrichtenstruktur auf Layer-II-Ebene ist tatsächlich deutlich komplexer:

(1) Das **ARP-Paket** [https://de.wikipedia.org/wiki/Address\\_Resolution\\_Protocol](https://de.wikipedia.org/wiki/Address_Resolution_Protocol))

- hat im ARP-Request/Reply hat tatsächlich die Form:

```

1 Byte 00 - 07 :- Sonderinformationen
2 Byte 08 - 13 :- Quell-MAC-Adresse
3 Byte 14 - 17 :- Quell-IPv4-Adresse
4 Byte 18 - 23 :- Ziel-MAC-Adresse
5 Byte 24 - 27 :- Ziel-IPv4-Adresse
  
```

- ist **mit dem Ethertype 0x0806** (= 2054) in den Ethernet-Frame <https://de.wikipedia.org/wiki/Datenframe> **eingebettet**.

(2) In der eigentlichen Nachricht ist dann z.B. das IP-Paket mit Ethertype 0x0800 für das IP-Protokoll in den Ethernet-Frame eingebettet.

(3) Quell- und Ziel-Daten wechseln dabei Position und Reihenfolge

Wir reduzieren die Komplexität hier abstrahierend so, dass wir [ S-IP-Adress S-MAC-Address D-IP-Adress D-MAC-Address Payload] schreiben und der Payload beim ARP-Request/Reply 0 ist.

**Hinweis:**

Wichtige Aktivitäten beim Versenden und Empfangen sind:

- Die Empfängerin einer Nachricht findet Name (IP-Adresse) und Adresse (MAC-Adresse) des Sender im Source/Absender-Adressbereich der Nachricht.
- Bei einer Antwort setzt die Empfängerin ihre eigenen Daten in den Source/Absender-Adressbereich und füllt den Target/Adressat-Adressbereich mit den Daten, die sie vom Sender erhalten hat.

[ R1-IP-Adress R1-MAC-Address R4-IP-Adress R4-MAC-Address PL0]

wird zu

[ R4-IP-Adress R4-MAC-Address R1-IP-Adress R1-MAC-Address PL0]

---

**ÜBUNG LF09:04:HUB&ARP:02**

- Erstellen Sie ein Aktivitätsdiagramm für eine Layer-II-Nachricht von Rechner R1 zu R4 in einem Hub-basierten Netz mit den Rechner R1, R2, R3 und R4. Alle Rechner haben nur eine Netzwerkkarte (Netzwerkinterface)
- 

**Lösung: [→ ZP:Sheet:5]**

---

**ÜBUNG LF09:04:HUB&ARP:03**

- Erstellen Sie das entsprechende Sequenzdiagramm für eine Layer-II-Nachricht von Rechner R1 zu R4 in einem Hub-basierten Netz mit den Rechner R1, R2, R3 und R4. Alle Rechner haben nur eine Netzwerkkarte (Netzwerkinterface)
-

**Lösung: [→ ZP:Sheet:6]****ÜBUNG LF09:04:HUB&ARP:04****[→ ZP:Sheet:7]**

- Machen Sie einen ‘Schreibtischtest’ für eine Layer-II-Nachricht von Rechner R1 zu R4 in einem Hub-basierten Netz mit den Rechner R1, R2, R3 und R4
- Beobachten Sie, was passiert, wenn im Einzelnen Folgendes gilt:
  - Eine Schülerin agiert als Hub:
    - Sie sitzt an einem Schreibtisch.
    - Jede Ecke des Schreibtischs ist ein Port: Hier werden eingehende Nachrichten abgelegt und ausgehende Nachricht zum Abholen bereitgelegt.
    - Die Schülerin hat viele Zettel. Kommt eine Nachricht an einer Ecke herein, nimmt sie diese, kopiert sie 2 Mal und legt je einen auf eine Ecke, von der die Nachricht nicht hereingekommen ist.
  - 4 Schülerinnen sind die Rechner R1 - R2, mit je einem Netzwerkinterface.
    - Im ersten Durchgang schreibt R1 an R4 unter Nutzung des ARPs.
    - Im zweiten Durchgang schreibt R1 an R4 und - kurz nach deren Start - R2 an R3 - je unter Nutzung des ARPs.

-> <https://de.wikipedia.org/wiki/Schreibtischtest>

---

**Lösung: [→ ZP:Sheet:8]**

- Kollisionen und Hub-Überforderung.
  - Im Worstcase ist der Hub nur noch mit Resets beschäftigt.
- 

**ÜBUNG LF09:04:HUB&ARP:05**

- Listen Sie ARP-Eigenschaften auf, die in einer verbesserten LAYER-II-Kommunikation erhalten bleiben müssten
- Listen Sie ARP-Probleme auf, die in einer verbesserten LAYER-II-Kommunikation lösen müsste.

**Lösung: [→ ZP:Sheet:9]****• beibehalten:**

- Netzwerkkarten adhoc austauschbar (ohne Rekonfiguration anderer Rechner/Karten)
- Anzahl der Netzwerkkarten im Netz adhoc erweiterbar (ohne Rekonfiguration des Netzes)
- Direkter (Re)Start der Kommunikation (ohne 'Abstimmung' mit anderen Rechnern)
- Kostengünstige Lösung

**• verbessern:**

- Reduktion der Anzahl der Kollisionen.
- Reduktion der Anzahl der Nachrichten.
- Reduktion der 'Ich-bin-nicht-gemeint-also-ignorieren'-Berechnungen (= Rechner sollen nicht mehr aus Nachrichten herauslesen müssen, dass sie sie ignorieren können: sie sollen sie erst gar nicht bekommen.)
- Mehr Rechner im Netz.

**3) Einige Disclaimer in Sachen HUB-basierter Netze**

1. Frage: *Macht die hier gewählte Darstellung einer Kommunikation mit Nachricht und Antwort auf Layer II - nach der Beantwortung des ARP-Request - überhaupt Sinn?* **Antwort:** Nein - oder bestenfalls nur sehr begrenzt:

- Nachrichten werden auf höherer Ebene als Sequenzen interpretiert.
- Auf Level II ist alles (außer dem ARP Request) nur 'Fire-and-Forget'

2. Frage: *Sendet ein Hub die Eingangsnachricht an alle? Oder 'nur' an alle anderen?* **Antwort:**

- Dazu gibt es in der Literatur sprachlich doppeldeutige Aussagen, manchmal sogar innerhalb einer 'Definition':

“How does a network hub work?” Regarding Broadcasting Data: Upon receiving the data packet on one of its ports, the hub then broadcasts this packet out **through all other ports**. This means **every device connected to the hub receives the packet**, regardless of whether it is the intended recipient or not.” (→ <https://www.portnox.com/cybersecurity-101/network-hub/>)

- Logisch spielt die Unterscheidung keine Rolle! Wenn der Sender sein eigenes Paket vom Hub zurückbekommt, ignoriert er es, weil es nicht an ihn adressiert ist.

- Praktisch würde so ein Verhalten eines Hubs jedoch den Traffic erhöhen, wenn jeder Sender, sein eigenes Paket zurückhält.
- **Also:** allgemeine Interpretation: *Ein Hub empfängt eine Nachricht an einem Port und leitet sie an all seine anderen Ports weiter.*

### 3. Frage: *Wozu braucht in einem reinen HUB-Netz überhaupt ein ARP?*

- Antwort 1: Braucht man da in der Tat nicht:
  - Ein Hub schickt immer alle Nachrichten an alle. Man kann eine einzelne Nachricht gar nicht nur an einen spezifischen Teilnehmer schicken.
  - Also wäre es prozessual überflüssig, vorab nach der Mac-Adresse eines des gewünschten Partners zu fragen. Man könnte seine Nachricht gleich direkt schicken.
  - Was den ADHOC-Kartenaustausch betrifft:
    - \* man auch die MAC-Adressen als Computernamen verwenden
    - \* mit einem 'Klopf-Klopf'-Protokoll mit dem sich jeder neue Rechner in der Gesprächsrunde anmeldet. Jeder Rechner pflegt sein eigenes Telefonbuch.
    - \* Falls eine Netzwerkkarte ausgetauscht wird, meldet sich der Rechner mit neuer MAC-Adresse wieder an und sagt dazu: 'früher war ich der'
- Antwort 2:
  - Reine HUB-Netze können - der physikalischen Dämpfungen wegen - viel zu wenig Teilnehmer integrieren, als dass das Netz interessant wäre.
  - Je mehr Rechner im Netz, desto größer die Gefahr der Kollisionen. Zuletzt würde das Netz nur noch Kollisionen managen.
- Also: **Wir wollen gewiss nicht in reinen Hub-Netzen arbeiten!**