

LF09:04:Layer I-II: Switch & ARP

1) Zur Erinnerung

- Ein Hub ist “ein Repeater mit mehr als zwei Schnittstellen.“ (vgl. Baun: Computernetze, 2022, S. 274)
- Repeater und Hub sind “dumm”, sie bekommen ein Signal, frischen es so weit technisch möglich auf und senden es weiter.
- Ein Hub-basiertes Netz läuft häufig in Kollisionen.
- Je mehr Geräte, desto häufiger Kollisionen.
- Worstcase: HUB/Netz nur noch mit Kommunikationsresets beschäftigt.

Kriterien einer Verbesserung auf Layer zwei wären [→ ZP:Sheet:2]

• beibehalten:

- Netzwerkkarten adhoc austauschbar (ohne Rekonfiguration anderer Netzwerkkarten/Rechner)
- Anzahl der Netzwerkkarten im Netz adhoc erweiterbar (ohne Rekonfiguration anderer Netzwerkkarten/Rechner)
- Direkter (Re)Start der Kommunikation (ohne ‘Abstimmung’ mit anderen Netzwerkkarten/Rechner)
- 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.

2) Die Bridge [→ ZP:Sheet:3]

Wünschenswert wäre eine Komponente, die

- ein HUB-basiertes Netz in 2 Teile splittet, indem sie
- Nachrichten an Rechner in demselben Teil **nicht** an die im anderen Teil weiterleitet

Dazu müsste diese Komponente,

- vermerken, welche MAC-Adressen auf welcher Seite ‘aktiv’ sind, indem sie
 - bei jeder eingehenden Nachricht deren Absender MAC-Adresse seitenspezifisch ‘cached’
 - und diesen Cache alle 300ms löscht
- analysieren, ob sie bereits weiß, dass Sender und Empfänger eine Nachricht auf derselben Seite positioniert sind und
 - Nachrichten an Rechner auf derselben Seite nicht weiterreichen, alle anderen aber wohl
- zwischen zwei Hubs stehen

Solche eine Komponente nennt man **eine Bridge**:

Ein Hub [...] verstärkte alle Signale, die er aufnimmt, und gibt sie ”[uninterpretiert] durch alle Ports wieder aus“, eine Bridge nicht:

- Ein(e) Bridge hat mehrere, in zwei Gruppen eingeteilte Anschlüsse.
- Sie interpretiert jedes Paket, das sie empfängt, und [...] merkt sich (anhand) der Source-Adresse“ welche MAC-Adresse auf welcher ihrer Seiten aktiv ist, indem sie Anschluss und Mac-Adresse [...] in eine ” Tabelle einträgt“.
- Hat sie eine Adresse gelernt, werden Messages an sie, die von derselben Seite kommen, nur noch an diese Seite gesendet. Nach 300ms soll(te) die Bridge die Tabelle verwerfen und die Zuordnung neu lernen.

Eine Bridge

- reduziert damit die Verkehrslast,
 - sofern die viel miteinander kommunizierenden Rechner auf derselben Seite positioniert sind
- kann mit geeigneten Transformatoren / variabler Sendetechnik auch 100MBit-Netze mit 10MBit-Netzen (etc.) verknüpfen
- vgl. Schreiner: Computernetzwerke, 2014, S. 45ff

2) Der (Layer-II)-Switch [→ ZP:Sheet:4]

ÜBUNG lf09:05:ARP&Switch:01

- Finden Sie eine Lösung dafür, wie man das ‘verbridgte’ Netz [→ ZP:Sheet:3] auf Layer-II umbauen müsste, damit gar keine überflüssigen Nachrichten mehr versendet werden.

Hinweis: Gehen Sie weiter davon aus, dass MAC-Adressen nach 300ms ‘verworfen’ werden müssen und dass also weiterhin das ARP genutzt werden muss.

Lösung:

Die Idee der ‘Verbridgung’ ist erweiterbar: Man könnte sich ja einen Hub konstruieren, der vor jedem Port eine Bridge hat, die

- aus den eingehenden Nachrichten die Source-MAC-Adresse in ihrem Cache vermerkt
- Nachrichten nach draußen über diesen Port nur dann nach außen sendet, wenn
 - eine Broadcast-Message vorliegt oder
 - die Ziel-MAC-Adresse in Adresscachedatei gelistet ist.

Denkt man sich um diese Gebilde ein Gehäuse, hat man einen **Layer-II-Switch** [→ ZP:Sheet:5]

Ein Switch ist sozusagen ein Hub, der auf jedem Anschlussport eine ”Bridge vorgeschaltet hat“:

und

- enthält intern einen Hochleistungsbust oder eine Matrix, die die Datenleitung regeln,
- ‘bridged’ technisch tatsächlich nach innen und außen und verbindet ggf. auch Netze mit verschiedenen Geschwindigkeiten.
- gibt - außer den Broadcast-Paketen nach einem ARP-Request - Daten [...] nur noch zu dem Port aus, an dem die Station angeschlossen ist, ”für die die Daten [im Paket] bestimmt sind“.
- vgl. Schreiner: Computernetzwerke, 2014, S. 45ff

Wenn 1 End-Gerät pro Port (Keine Hubs hinter Port), dann “voll geswitchtes Netzwerk” (vgl. Schreiner: Computernetzwerke, 2014, S. 51,53)

Tatsächlich:

- wissen wir nicht, wie Switches intern wirklich gebaut sind (geheimes Produktwissen)
- kennen wir nur ihr Verhalten als Black-Box

Vorteile:

- Massive Abnahme von Kollisionen

- Minimierung der Gefahr des Mitlesens
- In einem voll verswitchten Netzwerk kann die mangelnde Kooperation ignoriert werden
- vgl. Schreiner: Computernetzwerke, 2014, S. 52

Gefahren:

- Loops / verwirrte Bridges (= geräteinterne falsche Verschaltung)
- überreizte Spanning-Trees
- vgl. Schreiner: Computernetzwerke, 2014, S. 54 u. S. 59ff

(→ Spanning Tree = Teilgraph eines ungerichteten Graphen, der ein Baum ist und alle Knoten dieses Graphen enthält. (→ <https://de.wikipedia.org/wiki/Spannbaum>))

Konsequenz:

Auch in einem voll verswitschten Netz kennt die Kommunikationskomponenten

- dauerhaft nur den Computernamen (IP-Adresse)
- sollen Netzwerkkarten adhoc austauschbar sein (MAC-Adressen sollen nach 300ms vergessen werden)

ALSO: Auch ein verswitschtes Netz nutzt auf Layer-II-Ebene den ARP-Request.

ARP Sequenzdiagramm/explizit: [→ ZP:Sheet:6]

Nachdem wir das (vermutete) Innere eines Switches als verbundene Bridges expliziert und verstanden haben, dürfen wir bei komplexeren Darstellungen den Switch als Einheit darstellen und von seiner inneren Struktur abstrahieren:

ARP Sequenzdiagramm/verdichtet: [→ ZP:Sheet:7]

ARP Aktivitätsdiagramm: [brächte nur mehr ‘Schreibkram’ ohne wesentlichen Erkenntnisgewinn, deshalb hier weggelassen]