Übungsblatt 4 – Routing & Traffic Analysis

Aufgabe A – Wiederholung: OSI-Modell & Transport Layer + IP

Dieser Aufgabenteil dient der Wiederholung der Vorlesung [1]

1. Erläutern sie was in Netzwerken unter Datenkapselung verstanden wird.
2. Schauen Sie folgendes Video: https://youtu.be/iDCi_CJAyXs (Transport Layer)
3. Recherchieren Sie die Funktion, sowie den Aufbau des *TCP*-Protokolls.
   https://youtu.be/_WP9be9W3xE
4. Lesen sie [KR12, Kap. 3.1, 3.4]
   a) Auf welcher Ebene im OSI-Modell arbeitet *TCP*?
   b) Welche Aufgabe übernimmt das oben genannte Protokoll?
   c) Aus welchen Segmenten besteht ein *TCP*-Datagram?
   d) Zeigen Sie beispielhaft den Aufbau eines *TCP*-Datagram.
5. Recherchieren Sie die Funktion, sowie den Aufbau des *UDP*-Protokolls.
   https://youtu.be/xWsD6a3KsAI
6. Lesen sie [KR12, Kap. 3.3]
   a) Auf welcher Ebene im OSI-Modell arbeitet *UDP*?
   b) Welche Aufgabe übernimmt das oben genannte Protokoll?
   c) Aus welchen Segmenten besteht ein *UDP*-Datagram?
   d) Zeigen Sie beispielhaft den Aufbau eines *UDP*-Datagram.
7. Worin unterscheiden sich *TCP* und *UDP* grundlegend?
8. Recherchieren Sie die Funktion, sowie den Aufbau des IP-Protokolls.
9. Lesen sie [KR12, Kap. 4.1]
   a) Auf welcher Ebene im OSI-Modell arbeitet IP?
   b) Welche Aufgabe übernimmt das oben genannte Protokoll?
   c) Aus welchen Segmenten besteht ein IP-Paket im allgemeinen?
10. Recherchieren Sie die Funktion, sowie den Aufbau von Ethernet (*IEEE 802.3* Protokollfamilie).

[1] Das heißt, wenn sie fit sind einfach überspringen. Kann aber auch als Klausurvorbereitung dienen.

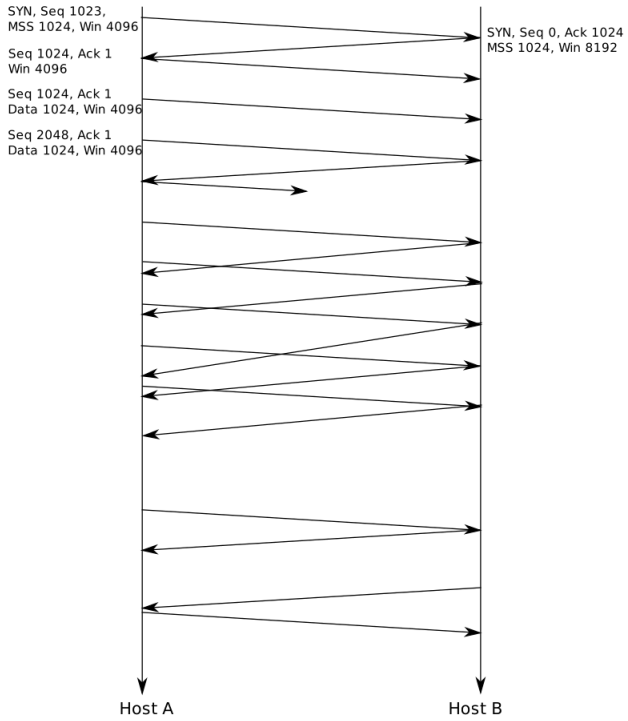
11. Lesen sie [KR12, Kap. 5ff]
    a) Auf welcher Ebene im OSI-Modell arbeitet *Ethernet*?
    b) Welche Aufgabe übernimmt das oben genannte Protokoll?
    c) Aus welchen Segmenten besteht ein Ethernet-Frame?

Aufgabe B – TCP: 3-Way-Handshake

Da TCP ein verbindungsorientiertes Protokoll ist, ist der Aufbau eines Sockets etwas komplizierter. Beide Seiten müssen sichergehen, dass die Verbindung korrekt funktioniert.

1. Recherchieren Sie wie der 3-Way-Handshake bei TCP funktioniert [KR12, Kap. 3.5]

2. Gegeben sei Folgendes Sequenzdiagramm einer TCP Verbindung.



   Die horizontalen Pfeile repräsentieren die Zeit. Die Beschriftungen sollen die Header-Felder der TCP-Segmente beschreiben. Eine 3-Way-Handshake wird von Host A initialisiert.

3. Erläutern sie den Austausch der ersten drei Segmente und Werte der Header-Felder.

4. Host A übermittelt 7 Segmente mit einer Nutzlast (Payload) von 1024 Byte an Host B, anschließend schließt A die Verbindung. Die ersten beiden Segmente samt Nutzdaten sind im Sequenzdiagramm bereits beschriftet. Vervollständigen Sie die restlichen Segmente samt Werte anhand folgender Informationen:
   a) Eines der Segmente ging verloren (signalisiert durch eine Pfeil der nicht die rechte Seite erreicht)
   b) Nehmen sie an, das Host A den Fast-Retransmit unterstützt und keine Timeouts durch ein verlorenes Segment auftritt.

Aufgabe C – Wiederholung: ICMP

Bevor es zum Thema Routing geht, soll im Folgenden noch das ICM-Protokoll betrachtet werden. Dieses dient in vielen Fällen als Diagnoseprotokoll.

1. Lesen sie den Abschnitt 4.4.3 zu ICMP [KR12, S. 353].
2. Was ist die Funktion des Internet Control Message Protocol (ICMP)?
3. ICMP hat verschiedene Message-Codes (einige brauchen wir in den Übungen 4 und 5!). Erläutern sie was diese Nachrichten kodieren sollen – was ist der Zweck der Message-Codes?
4. Recherchieren sie welchen Hinweis Ihnen die verschiedenen *ICMP*-Messages geben.
   i) Echo
   ii) Echo Reply

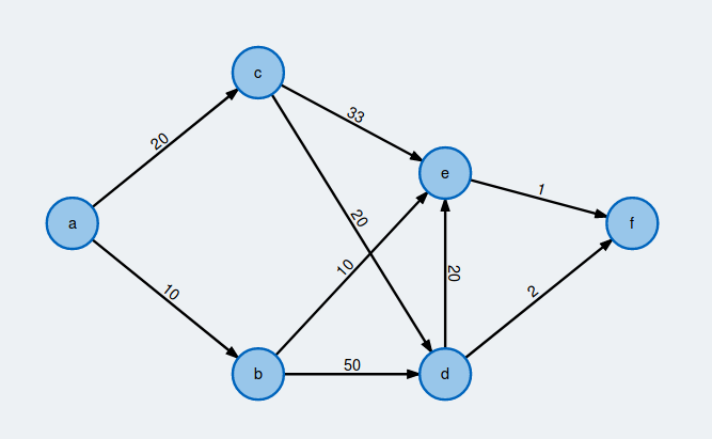
Aufgabe D – Wiederholung: Routing-Algorithmen

1. In der Vorlesung haben sie zwei Routing-Algorithmen kennen gelernt. Dies sind das Distanz-Vektor- und Link-State-Routing. Beide ermöglichen es den kürzesten Weg durch einen Graphen zu finden (Shortest-Path-Problem).
   Für gewöhnlich wird für das Distanz-Vektor-Routing der Bellman-Ford-Algorithmus verwandt, das Link-State-Routing nutzt den Dijkstra-Algorithmus. [KR12, S. 363ff]
   a) Recherchieren sie wie das Link-State-Routing unter Nutzung des Dijkstra-Algorithmus funktioniert [KR12, S. 366].
   b) Recherchieren sie wie das Distanz-Vektor-Routing unter Nutzung des Bellman-Ford-Algorithmus funktioniert [KR12, S. 371].
   c) In welchen Protokollen finden diese beiden Protokollen Verwendung? Ist diesen Protokollen etwas gemein?
   d) Erläutern sie die fundamentalen Unterschiede beider Lösungsansätze.

e) Warum wird keines der beiden Verfahren für das Exterior-Gateway-Protokoll genutzt?

f) Diskutieren sie, ob der Bellman-Ford-Algorithmus für das Link-State-Routing und der Dijkstra-Algorithmus für das Distanz-Vektor-Routing genutzt werden könnte.

2. Gegeben sei folgender Graph:



Finden sie den kürzesten Weg vom Knoten a zum Knoten f!

a) Nutzen sie zunächst den Dijkstra-Algorithmus.

b) Nutzen sie den Bellman-Ford-Algorithmus.

Aufgabe E – Traceroute

1. Lesen sie die folgenden Artikel:
https://en.wikipedia.org/wiki/Traceroute,
https://linux.die.net/man/8/traceroute.
Beantworten sie anschließend folgende Fragen:

a) Wofür wird Traceroute genutzt?

b) Wie wird Traceroute umgesetzt, d.h. wie läuft eine Routen-Verfolgung ab?

c) Welche ICMP-Messages werden für die Realisierung genutzt?

d) Welche Limitationen ergeben sich aus dieser Umsetzung?

e) Dokumentieren sie die Syntax, sowie die Bedeutung von Traceroute beispielhaft.

2. Lesen sie folgendes Paper zu Paris-Traceroute [Aug+06] von der ACM International Measurement Conference (IMC) 2006:
http://conferences.sigcomm.org/imc/2006/papers/p15-augustin.pdf

a) Warum ist eine „neue“ Traceroute-Applikation notwendig?

b) Nennen sie drei Topologie-Anomalien die durch Paris-Traceroute erkannt werden kann.

c) Recherchieren sie wie *paris-traceroute* zu nutzen ist! Notieren sie sich entsprechend die Kommandos und deren Bedeutung.

Aufgabe F – Wireshark

Wireshark ist eine Open-Source-Software, mit dem Analysen, Fehlerbehebungen, Software- und Protokollkommunikation untersucht werden können. *Wireshark* ähnelt *tcpdump* in gewisser Weise, jedoch bietet *Wireshark* ein grafisches Frontend (GUI), sodass die Analysen visuell ansprechend dargestellt werden können.[2]

1. Lesen Sie folgendes Tutorial in Hinblick auf die Fragen in Aufgabenteil 3.): https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packe , alternativ gibt es ein Wireshark 101 unter: https://youtu.be/f4zqMDzXt6k [3]

2. Nachdem Sie die Tutorials abgearbeitet haben:
   a) Was ist ein *Network-Sniffer*?
   b) Wozu kann ein Netzwerk-Sniffer genutzt werden?
   c) Recherchieren sie wozu Filter in *Wireshark* eingesetzt werden.
   d) Bringen sie in Erfahrung wie Filter genutzt werden.
   e) Welche beiden unterschiedlichen Mitschnitt-Modi (Caputre Modes) bietet *Wireshark*? Worin unterscheiden sich diese?

3. Erläutern sie anhand von Beispielen den grundlegende Umgang mit *Wireshark*.
   a) Wie stellen Sie Netzwerkinterfaces ein – auf welchem Interface soll der Mitschnitt laufen.
   b) Wie filtern Sie nach Protokollen?
   c) Wie filtern Sie *MAC*-Adressen?
   d) Wie filtern Sie *IP*-Adressen?

Aufgabe G – Wiederholung: Address Resolution Protocol (ARP) & Neighbor Discovery Protocol (NDP)

[2] Wireshark kann natürlich auch als *CLI* genutzt werden.
[3] Lohnenswert ist das Wireshark 101 Buch im PDF Format (eine Suche bei Google hilft bestimmt).

Es sollte ihnen aufgefallen sein, dass in der zweiten Übung (Geswitchte Netze) ihr Netzwerk in der Planung zwar IP-Adressen nutzt, aber kein Router Verwendung fand. Im Labor würde ein Switch zum Einsatz kommen. Switches sind OSI-Layer 2 Geräte und kommen ohne IP-Adressen zurecht. Ihre VMs verlangen jedoch zwingend eine IP-Adresse von Ihnen.

1. Recherchieren Sie mithilfe der Literatur was *ARP* ist [KR12, Kap. 5.4f]
2. Wie adressiert ein Switch die Frames zwischen den Endknoten (also den VMs)? **Hinweis:** Wie oben bereits erwähnt geschieht dies nicht mittels IP-Adressen.
3. Erläutern Sie das *MAC*-Adressschema. Kann dieses Adressschema auch zu Problemen führen?
4. Unter *IPv6* gibt es kein *ARP*, wie wird dies dort gehandhabt? Buw. wie funktioniert *NDP*?
5. Recherchieren Sie wozu die Werkzeuge *arp* und *ip neigh* in unixoiden Betriebssystemen genutzt werden können.
6. Recherchieren Sie die grundlegende Syntax und Semantik von *arp* sowie *ip neigh*.

Aufgabe H – Wiederholung: ARP-Cache & MITM

Da sie in den Übungen bis dato hauptsächlich Infrastruktur aufgebaut haben, soll dieser Teil als Vorbereitung für einen ersten Einblick in die Netzwerksicherheit bieten. Aufgrund ihres Wissen ahnen sie schon, dass Netzwerke leicht manipulierbar sind.

1. Vergegenwärtigen sie sich, was ein Cache ist und wozu dieser eingesetzt wird. Anschließend daran, recherchieren sie was es mit dem ARP-Cache auf sich hat.
2. Erläutern sie wie der ARP-Cache-Mechanismus funktioniert.
3. Recherchieren sie was ein Man-In-The-Middle-Angriff (*MITM*) ist.
4. Da der ARP-Cache keinerlei Validierungsmöglichkeiten hat, ist ein Manipulation des ARP-Caches möglich. Überlegen sie sich zunächst, welche Schritte hierfür notwendig wären, wenn sie als Angreifer, den Cache eines anderen Systems verändern wollen.
   - Welche Voraussetzungen müssen gegeben sein?
   - Welche Informationen über das Angriffsziel benötigen sie?
   - Welche Schritte muss ein Angriff auf den ARP-Cache folgen?
5. Im Moodle (sowie auf den Raspberry Pis) steht ein Angrifftool für das sogenannte ARP-Spoofing bzw. ARP-Cache-Poisoning bereit. Lesen dieses Skript und versuchen Sie es weitestgehend zu verstehen. Notieren Sie sich den Ablauf! Bzw. Fragen zu den Stellen im Quellcode, die Sie nicht verstehen.

Literatur

[Aug+06] Brice Augustin u. a. „Avoiding Traceroute Anomalies with Paris Traceroute“. In: *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*. IMC '06. Rio de Janeriro, Brazil: ACM, 2006, S. 153–158. ISBN: 1-59593-561-4. DOI: 10.1145/1177080.1177100. URL: http://doi.acm.org/10.1145/1177080.1177100.

[KR12] James F. Kurose und Keith W. Ross. *Computer Networking: A Top-Down Approach (6th Edition)*. 6th. Pearson, 2012. ISBN: 0132856204, 9780132856201.