

Übungsblatt 3 – Wireshark

Aufgabe A – OSI-Modell & Transport Layer + IP

Da Sie in der kommenden Übung mithilfe *Wiresharks* Ihren Netzwerkverkehr untersuchen sollen, müssen Sie zumindest grundlegend verstanden haben auf welche Protokolle Sie dort stoßen werden.

1. Schauen Sie folgendes Video: https://youtu.be/iDCi_CJAYXs (Transport Layer)
2. Recherchieren Sie die Funktion, sowie den Aufbau des *TCP*-Protokolls.
https://youtu.be/_WP9be9W3xE
 - a) Auf welcher Ebene im OSI-Modell arbeitet *TCP*?
 - b) Welche Aufgabe übernimmt das oben genannte Protokoll?
 - c) Aus welchen Segmenten besteht ein *TCP*-Datagramm?
 - d) Zeigen Sie beispielhaft den Aufbau eines *TCP*-Datagramm.
3. Recherchieren Sie die Funktion, sowie den Aufbau des *UDP*-Protokolls.
<https://youtu.be/xWsD6a3KsAI>
 - a) Auf welcher Ebene im OSI-Modell arbeitet *UDP*?
 - b) Welche Aufgabe übernimmt das oben genannte Protokoll?
 - c) Aus welchen Segmenten besteht ein *UDP*-Datagramm?
 - d) Zeigen Sie beispielhaft den Aufbau eines *UDP*-Datagramm.
4. Worin unterscheiden sich *TCP* und *UDP* grundlegend?
5. Recherchieren Sie die Funktion, sowie den Aufbau des IP-Protokolls.
 - 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?
6. Recherchieren Sie die Funktion, sowie den Aufbau von Ethernet (*IEEE 802.3* Protokollfamilie).
 - 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?

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

- [illegible]

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 Time-outs durch ein verlorenes Segment auftritt.

Aufgabe C – ICMP

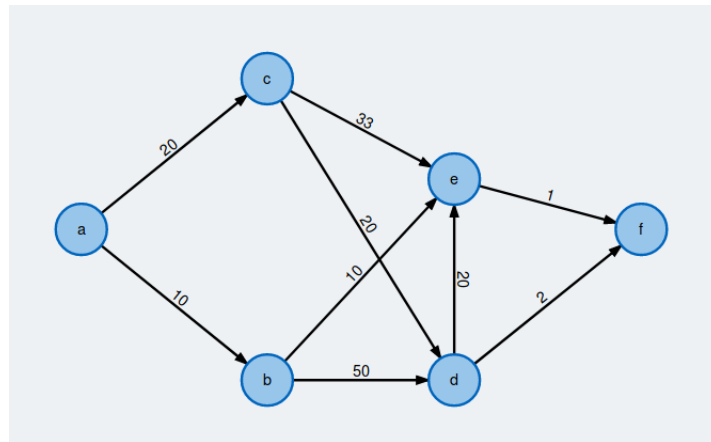
Bevor es zum Thema Routing geht, soll im Folgenden noch das ICM-Protokoll betrachtet werden. Dieses dient in vielen Fällen als Diagnoseprotokoll.

1. Lesen Sie den Abschnitt 4.4.3 zu ICMP [2, 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 – Routing

1. Im wesentlichen gibt es zwei fundamentale Routing-Algorithmen. Dies sind das Distanz-Vektor- und Link-State-Routing. Diese 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. [2, S. 363ff]
 - a) Recherchieren Sie wie das Link-State-Routing unter Nutzung des Dijkstra-Algorithmus funktioniert [2, S. 366].
 - b) Recherchieren Sie wie das Distanz-Vektor-Routing unter Nutzung des Bellman-Ford-Algorithmus funktioniert [2, 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 (EGP) 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*!

- Nutzen Sie zunächst den Dijkstra-Algorithmus.
- 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:

- Wofür wird Traceroute genutzt?
 - Wie wird Traceroute umgesetzt, d.h. wie läuft eine „Routen-Verfolgung“ ab?
 - Welche ICMP-Messages werden für die Realisierung genutzt?
 - Welche Limitationen ergeben sich aus dieser Umsetzung?
 - Dokumentieren Sie die Syntax, sowie die Bedeutung von Traceroute beispielhaft.
2. Lesen Sie folgendes Paper zu Paris-Traceroute [1] von der ACM International Measurement Conference (IMC) 2006:
- <http://conferences.sigcomm.org/imc/2006/papers/p15-augustin.pdf>
- Warum ist eine „neue“ Traceroute-Applikation notwendig?
 - Nennen Sie drei Topologie-Anomalien die durch Paris-Traceroute erkannt werden kann.
 - Recherchieren Sie wie *paris-traceroute* zu nutzen ist! Notieren Sie sich entsprechend die Kommandos und deren Bedeutung.

Literatur

- [1] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. Avoiding traceroute anomalies with paris traceroute. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, IMC '06*, pages 153–158, New York, NY, USA, 2006. ACM.
- [2] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach (6th Edition)*. Pearson, 6th edition, 2012.