Versuch 4: Analyse von Abläufen im TCP

(Wireshark Version)

Versuchsvorbereitung

Lesen Sie die Versuchsunterlagen gründlich durch und wiederholen Sie anhand der Vorlesungsunterlagen zur Transportschicht die Adressierung mit Portnummern sowie die wesentlichen Abläufe des TCP-Verbindungsauf- und -abbaus und der Flusssteuerung zur zuverlässigen Datenübertragung. Machen Sie sich insbesondere noch einmal mit den Details der Quittierung von empfangenen Bytes unter Verwendung der Sequenz- und Quittungsnummern vertraut.

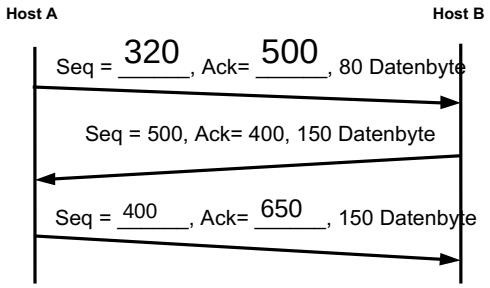
Informieren Sie sich anhand der Vorlesungsunterlagen zu Kapitel 10 über die grundlegenden Aufgaben und Funktionsweisen von DNS und HTTP.

Sehen Sie sich noch einmal Ihre Aufzeichnungen zu Versuch 1 an, um die grundlegende Nutzung von Wireshark zur Protokollanalyse zu wiederholen.

Beantworten Sie (bitte schriftlich) die folgenden Fragen:

1.1 a) Welche Seite leitet den TCP-Verbindungsaufbau grundsätzlich ein?
b) Wie viele Schritte umfasst der TCP-Verbindungsaufbau?
c) Welche Flags werden darin zur Steuerung verwendet?
d) Welche Information wird darin mithilfe der Sequenzfolge und Quittungsnummern zwischen Client und Server ausgetauscht?

1.2 Tragen Sie in das folgende Diagramm die fehlenden Sequenzfolge- (Seq) und Quittungsnummern (Ack) im Rahmen der Flusssteuerung für eine fehlerfreie TCP-Verbindung ein.



1.3 Warum nennt man den TCP-Verbindungsabbau auch „Zweifaches Halbschließen“? Wie viele Schritte umfasst er?

1.4 Welche Information wird mit Hilfe von DNS ermittelt?

1.1

a) Der Client leitet die TCP-Verbindung ein.

b) Der Verbindungsaufbau umfasst 3 Schritte.

c) SYN, ACK

d) Die Reihenfolge und Richtigkeit der TCP-Segmente.

1.3

Das Zweifache Halbschließen enthält 4 Schritte und wird so benannt, da beide Parteien jeweils einen Teil der Verbindung einleiten und sich selbige auch gegenseitig quittieren.

1.4

Das DNS-Protokoll dient zur Übersetzung von Domain Adressen in IP-Adressen.

Versuch: Analyse von TCP-, DNS- und HTTP-Abläufen

1. Aufgezeichnete Wireshark Pakete

In der nachstehenden Abbildung ist die für diesen Versuchsteil genutzte Topologie dargestellt:

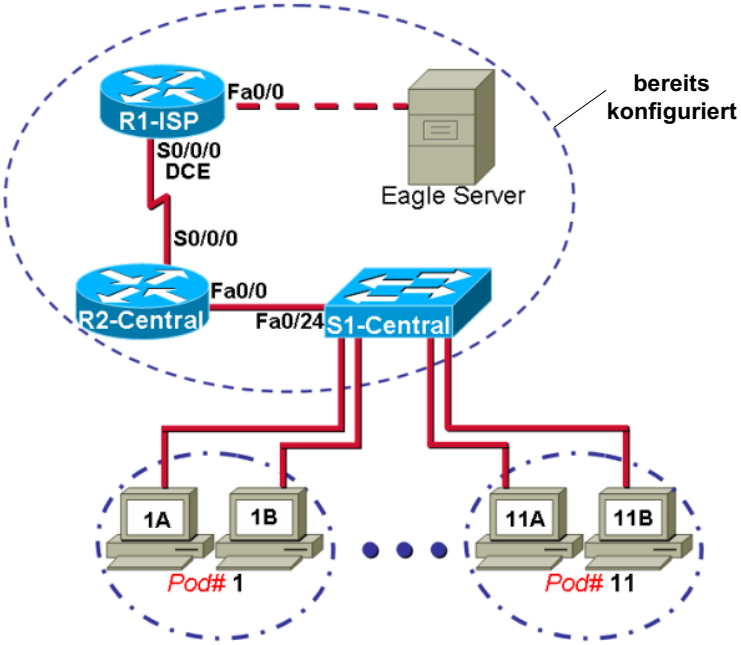


Abbildung 1.1: Netztopologie zu Versuchsteil 1

Diese Topologie stellt u.a. einen Server mit verschiedenen Anwendungen zur Verfügung. Dieser sogenannte „Eagle“-Server (= ein Server mit dem Domain-Namen „`eagle-server.example.com`“) beinhaltet u.a. einen Web-Server und DNS-Server.

Auf einem der PCs wurde ein Browser geöffnet und die Seite

`http://eagle-server.example.com/info`

aufgerufen. Mit Wireshark wurde der dabei ausgetauschte http- und DNS-Verkehr aufgezeichnet. Hierzu wurde in Wireshark folgender Capture-Filter (Mitschnittfilter) genutzt:

`(tcp port http or udp port 53) and host <IP-Adr. Des PCs>`

Die aufgezeichneten PDUs finden Sie in der Datei KN_Versuch4_1.pcapng. Öffnen Sie die Datei mit Wireshark.

2. Analyse der DNS-Abläufe und der Protokoll-Adressierung

Hinweis: Das erste (obere) Übersichtsfenster von Wireshark liefert immer nur eine Zusammenfassung der wichtigsten Daten der aufgezeichneten PDUs. Dabei wird in der Spalte „Protocol“ immer nur das Protokoll der höchsten OSI-Schicht angezeigt. Denken Sie daran, dass diese immer in den PDUs der darunterliegenden Schichten gekapselt werden. Sie müssen also häufig auch in die Detailinformationen (Zweites Fenster darunter) hineinsehen, um für die weiteren Aufgaben alle Informationen finden und ermitteln zu können.

a) Identifizieren Sie die Pakete, die zu DNS und die zu HTTP gehören. Weshalb ist vor dem HTTP-Nachrichtenaustausch mit dem Eagle Server eine DNS-Abfrage erforderlich?

Zuerst wird die IP-Adresse des Eagle-Servers benötigt.

b) Betrachten Sie die erste PDU der Aufzeichnung. Hierbei handelt es sich um die DNS-Abfrage (query) des PCs an den DNS-Server.

Welche IP-Adresse hat der anfragende PC? 172.16.254.0

Welche IP-Adresse hat der DNS-Server? 192.168.254.254

Schauen Sie in den DNS-Query hinein.

Die IP-Adresse welches Servers soll mit der DNS-Anfrage ermittelt werden (Unterpunkt Queries)?

eagle-server.example.com

Von welchem „type“ (=Ressource Type) ist die DNS-Abfrage? A

c) Die zweite PDU ist die Antwort des DNS-Servers. Schauen Sie in die Antwort (DNS response – Unterpunkt Answers).

Welche IP-Adresse hat der eagle-server gemäß dem Response? 192.168.254.254

d) Betrachten Sie nun wieder die mit Wireshark aufgezeichneten Frames. Welches Protokoll der OSI-Schicht 4 (Transportschicht) wird von DNS genutzt, welches von HTTP? Mit welchem Zahlenwert werden diese Protokolle jeweils im Protocol-Feld des IP-Headers codiert?

Protokoll der Anwendungschicht	Nutzt in der Transportschicht	Zahlenwert im Protocol-Feld
DNS	UDP	17
http	TCP	6

e) Welche Adressen verwendet der PC in der Transportschicht zum Transport der HTTP-PDUs und welche bei DNS? Wählen Sie jeweils eine geeignete Anfrage, die vom PC ausgeht, und vervollständigen Sie die folgende Tabelle:

Übertragungsrichtung: Eigener Rechner (Client) -> Server

Protokoll der Anwendungsschicht:	Ziel-Portnummer	Absender-Portnummer
DNS	53	50282
http	80	52363

Welche dieser Adressen sind feststehende (well-known) Portnummer? Ziel

und welche Portnummern wurden frei gewählt? Absender

3. Analyse des TCP Verbindungsauf- und abbaus

a) Identifizieren Sie die drei Segmente für den TCP-Verbindungsaufbau. Welche Flags (SYN, ACK) werden in diesen Segmenten jeweils gesetzt?

Segment 1: SYN Segment 2: SYN, ACK Segment 3: ACK

b) Wozu werden die Felder „Sequence Number“ und „Acknowledgement Number“ im Rahmen des Verbindungsaufbaus benutzt?

Zur initialen Synchronisierung und Quittierung der TCP-Verbindung.

Hinweis: Diese Sequenzfolge- und Quiitungsnummern werden von Wireshark zur Vereinfachung der Analyse bei der Anzeige in Wireshark in beiden Richtungen jeweils durch Subtraktion der Startwerte in Werte relativ zu einer Startsequenznummer 0 umgerechnet. Dieses Vorgehen ist üblicherweise voreingestellt und lässt sich unter Edit/Bearbeiten → Preferences/Einstellungen → Protocols → TCP als Option „Relative Sequence Numbers and Window Scaling“ ein- oder ausschalten. Sie können sich die Originalwerte einmal ansehen, indem Sie die Option dort ausschalten. Es empfiehlt sich aber für die weiteren Aufgaben diese Option einzuschalten.

c) Betrachten Sie nun die nach dem TCP-Verbindungsaufbau folgenden Segmente, mit denen die Anwendungsdaten zwischen Client und Server ausgetauscht werden. Um welche Informationen handelt es sich hier eigentlich?

Richtung Client an Server: Anforderung einer Webseite

Richtung Server an Client: Inhalt der Webseite

Hinweis: Im Frame 22 stellt Wireshark die http Antwort zusammengefasst dar. Erkennbar ist das unter TCP im mittleren Fenster: „[8 Reassembled TCP Segments ….]“. Im „Line-based text data“ finden Sie den gesamten Text der Webseite, der in den Frames 12,13,15,16,18,19,21,22 übertragen wurde.

d) Ab Frame No. 26 wird eine neue, zweite TCP-Verbindung aufgebaut. Davor wird die bisherige Sitzung abgebaut.

Identifizieren Sie die vier Segmente für den TCP Verbindungsabbau (Flags FIN, ACK) der ersten (!) TCP-Verbindung. Geben Sie die Nummern der beiden Frames an, in denen Client und Server das FIN zum Abbau der ersten TCP-Verbindung schicken:

Server TCP-Segment mit gesetzten FIN-Flag ist Frame Nr. 22

Client TCP-Segment mit gesetztem FIN-Flag ist Frame Nr. 24

4. Analyse der TCP-Flusssteuerung während der Datenübertragung

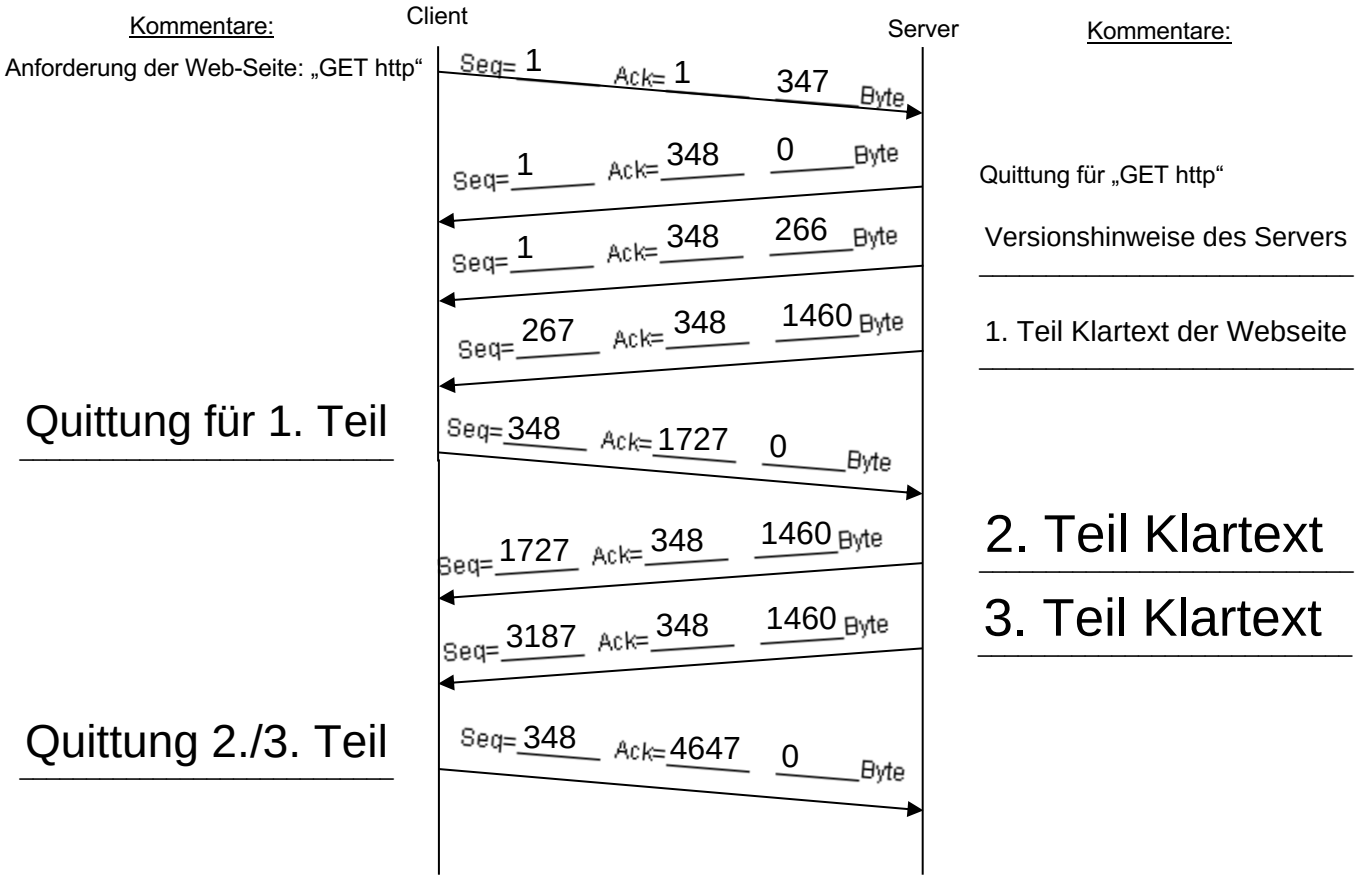
TCP transportiert Anwendungsdaten erst, nachdem der Verbindungsaufbau abgeschlossen ist. Die TCP-Flusssteuerung mit Sequenznummern, Quittungsnummern und Fenstergrößen wurde ausführlich in der Vorlesung erläutert. Zur Erinnerung hier noch einmal die Bedeutung der relevanten Werte des TCP-Headers:

- Sequenznummer (Seq) = Nummer des 1. Bytes des Segments
- Quittungsnummer (Ack) = Nummer des nächsten erwarteten Bytes

Analysieren Sie die Nummerierung und Quittierung der Anwendungsdaten beginnend mit dem ersten TCP-Segment nach dem Verbindungsaufbau (ist in diesem Falle das GET http). Ergänzen Sie dazu das folgende Diagramm. Berücksichtigen Sie alle Segmente, bis der Client zweimal die vom Server gesendeten Daten quittiert hat. Das sind die Frames Nr.10 bis Nr. 17.

Hinweise:

- *Die Anzahl der Bytes in den Segmenten wird von Wireshark in den Detailinformationen zu TCP als „Len=“ angegeben (Zweites Fenster). Sie finden sie nicht in dem ersten oberen Übersichtsfenster.*
- *Für das http Protokoll ist festgelegt, das eine „GET“ Anforderung eines Clients zunächst explizit quittiert werden muss, deshalb wird vom Server vor dem Senden des Inhaltes zunächst eine reine Quittungsnachricht ohne Nutzdaten (Len=0) verschickt.*



5. Analyse des http-Reloads der Seite vom Eagle-Server

Öffnen Sie die Datei KN_Versuch4_2.pcapng, in der der Kommunikationsverkehr eines Reloads der Webseite aufgezeichnet wurde. Der TCP-Verbindungsaufbau verteilt sich hier auf die Frames 1, 3 und 5.

Hinweis: Frame 2 enthält eine DNS-Abfrage vom Typ AAAA zur Ermittlung der IPv6-Adresse des Eagle-Servers. Da der Eagle-Server keine IPv6 hat, enthält der DNS-Response (Frame 4) keine Antwort.

Zurück zu http: In Frame 6 schickt der Client den http-GET-Request.

Der Server schickt jetzt NICHT erneut die Seite! Was schickt der Server dem Client anstelle der Webseite?

Einen "Not Modified" Hinweis.

Woher weiß der Server, dass die Seite nicht geändert wurde, seitdem der Client diese zum letzten Mal abgerufen hat? Speichert sich der Server, wann der letzte Seitenabruf durch den Client erfolgte? Nein, natürlich nicht, dann müsste der Server ja für sämtliche Abfragen aller Clients Zeitpunkte speichern! Also woher weiß der der Server das? Die Antwort liefert Ihnen der GET-Request des Clients. Welches Feld und welche Angabe sind hierzu relevant?

Die Angabe "Last-Modified-Since" in der HTTP Schicht.