Technische Universität München
Lehrstuhl Informatik VIII
Prof. Dr.-Ing. Georg Carle
Dipl.-Ing. Stephan Günther, M.Sc.
Johannes Naab, M.Sc.

Tutorübung zur Vorlesung Grundlagen Rechnernetze und Verteilte Systeme

Übungsblatt 12 (13. Juli – 17. Juli 2015)

Hinweis: Die mit * gekennzeichneten Teilaufgaben sind ohne Kenntnis der Ergebnisse vorhergehender Teilaufgaben lösbar.

Aufgabe 1 All in a nutshell

In dieser Aufgabe wollen wir noch einmal alles nachvollziehen, was geschieht, wenn Sie auf Ihrem Computer die Webseite `www.google.de` aufrufen. Wir treffen dabei lediglich die Annahme, dass in Ihrem privaten Netz ARP- und DNS-Caches noch leer sind. Die Netzwerktopologie ist in Abbildung 1 dargestellt. Ihr Router übersetzt bei Bedarf private in öffentliche IP-Adressen sowie Portnummern (NAT). Auf Ihrem Computer sei der Google-Resolver mit der IPv4-Adresse 8.8.8.8 konfiguriert, der rekursive Anfragen erlaubt.

Es sollen nun für **jeden Link** – also jeden Abschnitt zwischen jeweils zwei Geräten – einige ausgewählte Felder der Nachrichten notiert werden, die im jeweiligen Schritt über diesen Link versendet werden. Da dies etwas Schreibarbeit ist, kürzen wir Adressen mit der Bezeichnung `<Gerätename>.<Interface>.<Type>` wie in Abbildung 1 angegeben ab, z. B. stehe `RA.eth0.MAC` für die MAC-Adresse von Interface `eth0` an Router `RA` und `RA.eth0.IP4` für die entsprechende IPv4-Adresse.
Sie finden in den Abbildungen 2 – 4 vorgedruckte Tabellen. Eine Zeile entspricht dabei einer Nachricht, die über den jeweiligen Link gesendet wird. Die erste Spalte bezeichnet dabei den Link, also z. B. vom PC zum Switch oder vom Switch zum Router. Die übrigen Spalten entsprechen verschiedenen Schichten des ISO/OSI-Modells. Diese sind jeweils in die relevanten Headerfelder der üblicherweise verwendeten Protokolle unterteilt. Je nach Nachricht sind nicht alle Spalten oder Unterzeilen pro Spalte auszufüllen. **Streichen Sie deutlich nicht benötigte Felder.** Ein Beispiel ist bereits in der Tabelle eingetragen.

Einige Header verfügen über ein Protokoll-Feld, in dem das Protokoll der nächsthöheren Schicht angegeben wird. Üblicherweise stehen Zahlencodes für die jeweiligen Protokolle. Es ist **nicht** notwendig, diese Zahlencodes anzugeben. Stattdessen reicht es, das verwendete Protokoll anzugeben, z. B. IPv4, TCP oder UDP. Bei einigen Header-Feldern gibt es gewisse Freiheiten, z. B. bei Portnummern oder der initialen TTL. Wählen Sie in diesen Fällen **sinnvolle** Werte.

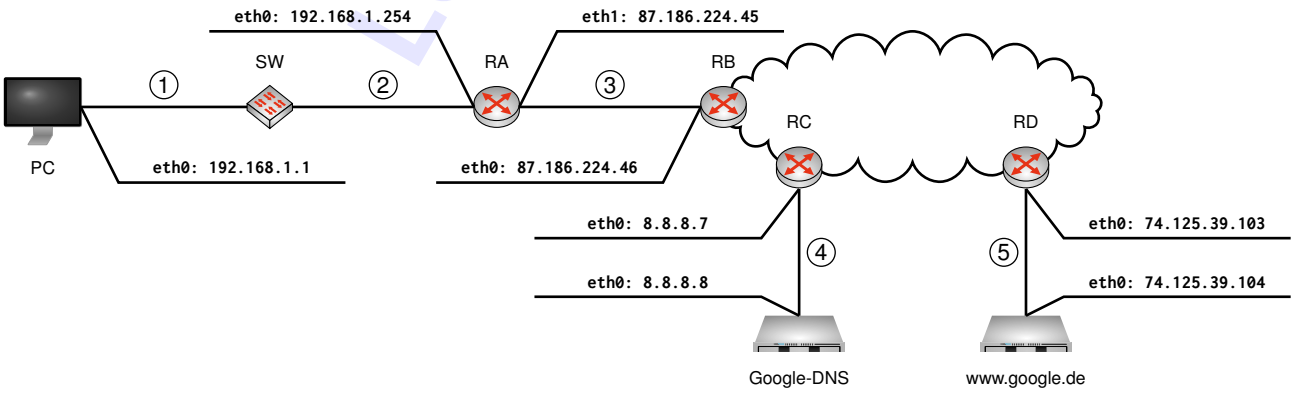


Abbildung 1: Netztopologie zu Aufgabe 1. Die relevanten Links sind mit den Ziffern 1 – 5 gekennzeichnet.

a)* Füllen Sie nun die Vordrucke in den Abbildungen 2 – 4 aus. Brechen Sie **nach dem ersten** an **`www.google.de`** übermittelten Paket auf Link 1 ab. **Hinweise:**

- Der Well-Known Port für DNS ist UDP 53.
- Wir nehmen an, dass sich zwischen Router RB und RC insgesamt 10 weitere Router befinden. Dies ist für die Bestimmung der TTL entscheidend.
- In die Spalte „Schicht 7" tragen sie einfach das Anwendungsprotokoll, ggf. den Typ der Nachricht (z. B. Request / Reply) sowie stichpunktartig den Inhalt der übermittelten Nachricht ein (z. B. „ DNS-Request" oder „DNS-Response").

Siehe Abbildungen 2 – 4.

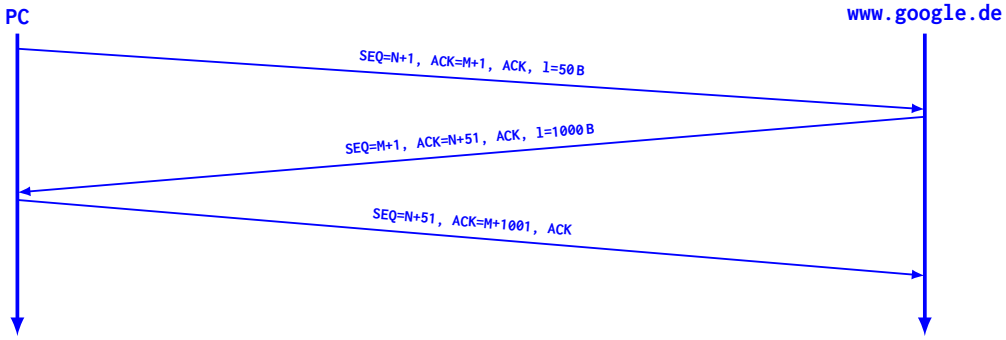
Die vorangegangene Teilaufgabe hat detailliert die Vorgänge bis zum Beginn des TCP-Verbindungsaufbaus dargestellt. Im Folgenden wollen wir uns auf die TCP-Verbindung und Datenübertragung konzentrieren. Aus diesem Grund betrachten wir ab jetzt nur noch die logische Verbindung zwischen dem PC und **`www.google.de`** in Form eines einfachen Weg-Zeit-Diagramms **ohne** die dazwischenliegenden Knoten. Sie können Serialisierungszeiten vernachlässigen. Gehen Sie außerdem davon aus, dass während der gesamten Übertragung keine Segmentverluste auftreten.

b)* Skizzieren Sie ein Weg-Zeit-Diagramm, welches den TCP-Verbindungsaufbau darstellt. Geben Sie für jedes übermittelte Segment die Sequenznummer, Bestätigungsnummer, die gesetzten[1] Flags sowie die Länge *l* der transportieren Nutzdaten an.

Die Bestätigungsnummer des ersten Segments hat keinerlei Auswirkung, da das ACK-Flag nicht gesetzt ist (was soll auch beim ersten Segment bestätigt werden?). Die initialen Sequenznummern beider Teilnehmer sind prinzipiell beliebig, d. h. $0 \leq N,M, \leq 2^{32} - 1$.

Der PC fordert nun die Webseite an, die auf **`www.google.de`** gehostet wird. Dazu sendet der PC eine **`HTTP-GET`**-Nachricht, welche aus Sicht von Schicht 4 eine Nutzdatenlänge von $l_1 = 50$ B habe. Der Webserver wird draufhin die Webseite an den PC senden, welche eine Länge $l_2 = 1000$ B habe. Die ausgehandelte MSS[2] sei größer als l_2.

c) Skizzieren Sie ein Weg-Zeit-Diagramm, welches die TCP-Verbindungsphase darstellt. Gehen Sie von den in Teilaufgabe b) ausgehandelten Sequenznummern aus. Geben Sie für jedes übermittelte Segment die Sequenznummer, Bestätigungsnummer, die gesetzten Flags sowie die Länge *l* der im Segment transportieren Nutzdaten an.

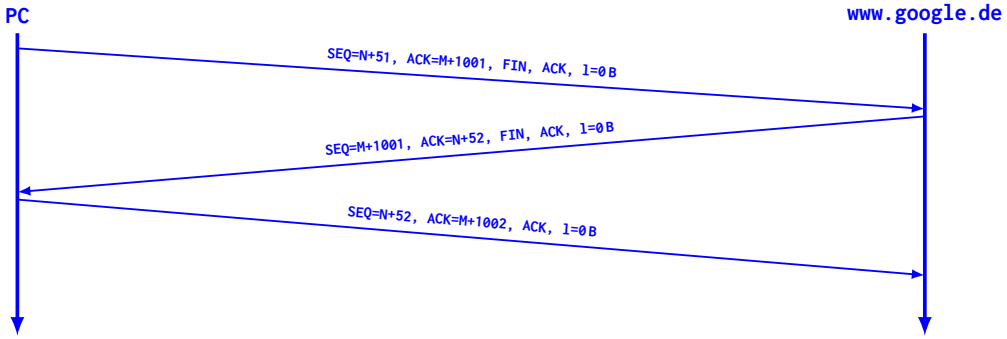


Hinweis: Will der PC die Verbindung unmittelbar nach dem Erhalt der Nachricht abbauen, so kann das dritte Segment bereits das FIN-Flag gesetzt haben.

[1] Ein Bit-Flag gilt als „gesetzt", wenn es logisch 1 ist.

[2] Die MSS (Maximum Segment Size) gibt die maximale Größe eines Segments an. Sie bezieht sich dabei lediglich auf die Nutzdaten. Bestätigungen beispielsweise sind Segmente der Länge null, welche lediglich aus einem TCP-Header bestehen.

d) Skizzieren Sie ein Weg-Zeit-Diagramm, welches den TCP-Verbindungsabbau darstellt. Dieser werde vom PC initiiert. Gehen Sie dabei von den in Teilaufgabe c) ausgehandelten Sequenznummern aus. Geben Sie für jedes übermittelte Segment die Sequenznummer, Bestätigungsnummer, die gesetzten Flags sowie die Länge *l* der im Segment transportieren Nutzdaten an.



An dieser und den vorherigen beiden Teilaufgaben ist noch einmal zu sehen, dass TCP einzelne Bytes, nicht aber Segmente bestätigt. Segmente, welche ein SYN oder FIN Flag gesetzt haben, werden dabei wie Segmente mit genau 1 B Nutzdaten behandelt.
Der Verbindungsabbau kann auch in Form von vier statt drei Nachrichten stattfinden, d. h. **`www.google.de`** bestätigt zunächst den erhalt des **`FIN`**-Flags ohne dabei selbst das **`FIN`**-Flag zu setzen. Dies könnte beispielsweise dann der Fall sein, wenn zwar der **`PC`** keine Daten mehr an **`www.google.de`** zu übertragen hat, **`www.google.de`** aber noch unbestätigte Segmente an den **`PC`** hat. Derartige TCP-Verbindungen werden als „halb-offen“ bezeichnet, da eine Datenübertragung in eine der beiden Richtungen noch immer möglich ist.

Link		Schicht 2		Schicht 3		Schicht 4		Schicht 7
From	PC	Src	PC.eth0.MAC	Src		Src		
		Dst	ff:ff:ff:ff:ff:ff	Dst		Dst		
To	SW	Prot	ARP	Prot		Flags		
		Op	Request	TTL		SEQ		
						ACK		
From	SW	Src	PC.eth0.MAC	Src		Src		
		Dst	ff:ff:ff:ff:ff:ff	Dst		Dst		
To	RA	Prot	ARP	Prot		Flags		
		Op	Request	TTL		SEQ		
						ACK		
From	RA	Src	RA.eth0.MAC	Src		Src		
		Dst	PC.eth0.MAC	Dst		Dst		
To	SW	Prot	ARP	Prot		Flags		
		Op	Reply	TTL		SEQ		
						ACK		
From	SW	Src	RA.eth0.MAC	Src		Src		
		Dst	PC.eth0.MAC	Dst		Dst		
To	PC	Prot	ARP	Prot		Flags		
		Op	Reply	TTL		SEQ		
						ACK		
From	PC	Src	PC.eth0.MAC	Src	192.168.1.1	Src	51827	DNS Request
		Dst	RA.eth0.MAC	Dst	8.8.8.8	Dst	53	Who is www.google.de?
To	SW	Prot	IPv4	Prot	UDP	Flags		
				TTL	64	SEQ		
						ACK		

Abbildung 2: Vordruck zu Aufgabe 1

Link		Schicht 2		Schicht 3		Schicht 4		Schicht 7
From	SW	Src	PC.eth0.MAC	Src	192.168.1.1	Src	51827	DNS Request
		Dst	RA.eth0.MAC	Dst	8.8.8.8	Dst	53	Who is www.google.de?
To	RA	Prot	IPv4	Prot	UDP	Flags		
				TTL	64	SEQ		
						ACK		
From	RA	Src	RA.eth1.MAC	Src	87.186.224.45	Src	38218	DNS Request
		Dst	RB.eth0.MAC	Dst	8.8.8.8	Dst	53	Who is www.google.de?
To	RB	Prot	IPv4	Prot	UDP	Flags		
				TTL	63	SEQ		
						ACK		
From	RC	Src	RC.eth0.MAC	Src	87.186.224.45	Src	38218	DNS Request
		Dst	DNS.eth0.MAC	Dst	8.8.8.8	Dst	53	Who is www.google.de?
To	Google DNS	Prot	IPv4	Prot	UDP	Flags		
				TTL	51	SEQ		
						ACK		
From	Google DNS	Src	DNS.eth0.MAC	Src	8.8.8.8	Src	53	DNS Reply
		Dst	RC.eth0.MAC	Dst	87.186.224.45	Dst	38218	www.google.de is
To	RC	Prot	IPv4	Prot	UDP	Flags		74.125.39.104!
				TTL	64	SEQ		
						ACK		
From	RB	Src	RB.eth0.MAC	Src	8.8.8.8	Src	53	DNS Reply
		Dst	RA.eth1.MAC	Dst	87.186.224.45	Dst	38218	www.google.de is
To	RA	Prot	IPv4	Prot	UDP	Flags		74.125.39.104!
				TTL	52	SEQ		
						ACK		

Abbildung 3: Vordruck zu Aufgabe 1

Link		Schicht 2		Schicht 3		Schicht 4		Schicht 7
From	RA	Src	RA.eth0.MAC	Src	8.8.8.8	Src	53	DNS Reply
		Dst	PC.eth0.MAC	Dst	192.168.1.1	Dst	51827	www.google.de is
To	SW	Prot	IPv4	Prot	UDP	Flags		74.125.39.104!
				TTL	51	SEQ		
						ACK		
From	SW	Src	RA.eth0.MAC	Src	8.8.8.8	Src	53	DNS Reply
		Dst	PC.eth0.MAC	Dst	192.168.1.1	Dst	51827	www.google.de is
To	PC	Prot	IPv4	Prot	UDP	Flags		74.125.39.104!
				TTL	51	SEQ		
						ACK		
From	PC	Src	PC.eth0.MAC	Src	192.168.1.1	Src	58392	
		Dst	RA.eth0.MAC	Dst	74.125.39.104	Dst	80	
To	SW	Prot	IPv4	Prot	TCP	Flags	SYN	
				TTL	64	SEQ	0	
						ACK	0	
From		Src		Src		Src		
		Dst		Dst		Dst		
To		Prot		Prot		Flags		
				TTL		SEQ		
						ACK		
From		Src		Src		Src		
		Dst		Dst		Dst		
To		Prot		Prot		Flags		
				TTL		SEQ		
						ACK		

Abbildung 4: Vordruck zu Aufgabe 1