Tutoriumsblatt 6 mit Musterlösung

Aufgabe 6.1: IP-Adressen und CIDR

a) Sie haben die IP-Adressen 137.226.12.221 und 137.234.17.222 gegeben. Wie ist die *Subnetzmaske* zu wählen, damit *beide Rechner im gleichen Netz* liegen, das Netz allerdings *so klein wie möglich* ist?

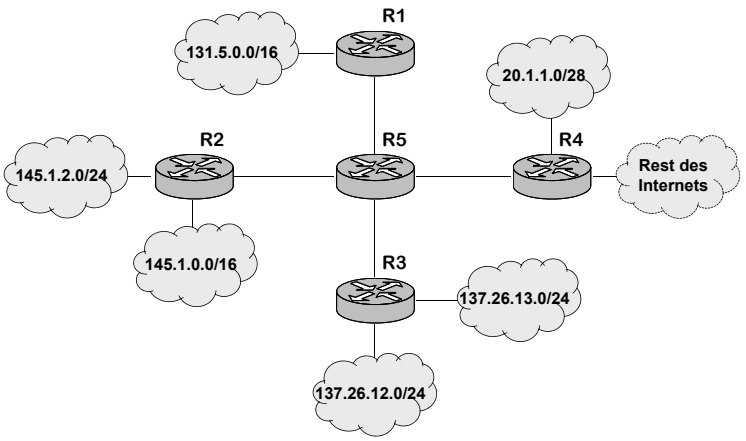
b) Es ist eine große Anzahl an aufeinander folgenden IP-Adressen verfügbar, die bei 137.226.0.0 beginnen. Angenommen, vier Organisationen W, X, Y, Z fordern in der folgenden Reihenfolge Adressbereiche für ihre Rechner an:

- W für 3990 Rechner
- X für 2020 Rechner
- Y für 4096 Rechner
- Z für 1853 Rechner

Vergeben wird jeweils die niedrigstmögliche Netzadresse. *Geben Sie für jede Organisation die zugewiesene Netzadresse mit Netzmaske an.* Geben Sie darüber hinaus jeweils an, welches die *erste und welches die letzte IP-Adresse* aus dem zugewiesenen IP-Adressbereich ist.

c) Angenommen, es würde klassenbasierte IP-Adressierung ohne Subnetzmasken verwendet. *Wie viele Einträge* müsste ein Router in seiner Routing-Tabelle vorhalten, damit er Daten an alle möglichen Zieladressen weiterleiten könnte?

d) Betrachten Sie den folgenden Netzaufbau, in dem die Router $R1$ - $R4$ jeweils ein oder zwei Netze verwalten. Der genaue Aufbau dieser Netze ist uninteressant – aber die Router $R1$ - $R4$ kennen die Adressbereiche der an sie angeschlossenen Netze. Lediglich Router $R4$ verfügt über eine Verbindung zum Rest des Internets. Sie haben nun den Router $R5$ installiert, um die Netze miteinander und mit dem Internet zu verbinden und müssen eine Routing-Tabelle erstellen. Geben Sie für Router R5 eine Routing-Tabelle mit so wenig Einträgen wie möglich an, damit alle Daten korrekt zwischen den Netzen (und dem Rest des Internets) weitergeleitet werden. Beschränken Sie sich bei der Tabelle auf Einträge der Form `Zielnetz, Next Hop`. `Next Hop` ist dabei einer der Router $R1$ - $R4$. Angaben wie Flags, Netzwerkkarten-Adressen oder weiteres sind hier nicht von Interesse.



Lösung 6.1

1.a) Wir stellen beide Adressen zunächst binär dar:

IP-Adresse	Binär
`137.226.12.221`	`1000 1001.1110 0010.0000 1100.1101 1101`
`137.234.17.222`	`1000 1001.1110 1010.0001 0001.1101 1110`

Wie man in der Tabelle sehen kann, sind die ersten 12 Bit (fett dargestellt) identisch. Also muss als Subnetzmakse `255.240.0.0` (entspricht dem Subnetz `137.224.0.0/12`) gewählt werden.

1.b) Wähle jeweils die nächstgrößere Zweierpotenz:

- 4096 (=12 Bit)
- 2048 (=11Bit)
- 8192 (=13 Bit)

In der folgenden Tabelle ist das letzte Bit des Netzanteils zur Netzunterscheidung rot dargestellt. Die erste und letzte IP-Adresse sind dann noch mal in der Dotted Decimal Notation angegeben.

Organssation	Subnetz	Erste IP-Adresse	Letzte IP-Adresse	# Adressen
W (3990)	137.226.0.0/20	137.226.[0000 0000].0 137.226.0.0	137.226.[0000 1111].255 137.226.15.255	4096
X (2020)	137.226.16.0/21	137.226.[0001 0000].0 137.226.16.0	137.226.[0001 0111].255 137.226.23.255	2048
Y (4096)	137.226.32.0/19	137.226.[0010 0000].0 137.226.32.0	137.226.[0011 1111].255 137.226.63.255	8192
Z (1853)	137.226.24.0/21	137.226.[0001 1000].0 137.226.24.0	137.226.[0001 1111].255 137.226.31.255	2048

Der Organisation Y wurden 8192 Adressen zugewiesen, obwohl sie eigentlich nur 4096 benötigt. Allerdings kann man nicht 4096 Rechner in einem Adressbereich der Größe 4096 adressieren, da die Netz- und die Broadcast-Adresse berücksichtigt werden müssen.

Man muss auch eine Lücke lassen: eine Basisadresse von `137.226.24.0` mit der Subnetzmaske `/19` ist nicht möglich, da die Zahl `24` nicht mit den ersten drei Bit dargestellt werden kann.

Bei Ogranisation Z wird die Lücke genutzt, da nach der niedrigestmöglichen Adresse gefragt war und die Lücke gerade Platz für die benötigte Zahl an Rechnern bietet.

1.c) Bei klassenbasierter Adressierung ohne Subnetzmasken kann man keine Adresseinträge aggregieren – ein Router muss für jedes mögliche Netz einen eigenen Eintrag haben. Man muss also schauen, wie viele Netze es gibt:

- Klasse A: $2^7 = 128$ Netze. Allerdings muss man zumindest `127.0.0.0` abziehen, da es für Loopback verwendet wird, und `10.0.0.0` als Bereich privater Adressen. Also nur 126.
- Klasse B: $2^{14} = 16.384$. Allerdings muss man auch hier private Adressen abziehen – und zwar 16 Netze (`172.16.0.0` bis `172.31.0.0`). Also bleiben 16.368 NEtze übrig.
- Klasse C: $2^{21} = 2.097.152$. Und auch hier gibt es einen Bereich privater Adressen: `192.168.0.0` bis `192.168.255.0`, also 256 Stück. Bleiben läppische 2.096.896 Netze.

Summiert man auf, kommt man auf 2.113.390 Netze und daher ebenso viele Einträge in der Routing-Tabelle. Zugegeben, diese Abschätzung ist wahnsinnig ungenau, da noch mehr Adressbereiche für spezielle Zwecke reserviert sind und die korrekte Zahl etwas niedriger liegt. Aber Fakt ist: dies sind deutlich zu viele Einträge, unsere Router könnten die Tabellen nicht mehr verwalten.

Die Reduktion der Tabellengröße betrachten wir in der nächsten Teilaufgabe.

1.d) Man kann jetzt einfach alle angeschlossenen Netze einsammeln und Einträge in R5 anlegen, siehe linker Teil der Abbildung. Das ist aber bei weitem nicht minimal. (Dies entspräche dem Vorgehen aus der vorherigen Teilaufgabe.)

Man kann die nötigen Einträge tatsächlich auf 4 reduzieren – dieses Verfahren nennt sich Route Aggregation. Router R2 schließt zwar zwei Netze an, aber eins ist vom Prinzip her Teilnetz des anderen Adressbereichs, und R5 interessiert nicht, wie es hinter R2 weitergeht, so dass ein Eintrag ausreicht. Router R3 schließt zwar zwei Netze an, aber diese liegen adressierungsmäßig direkt nebeneinander, so dass ein Eintrag mit einer kürzeren Subnetzmaske ausreicht. Router R4 ist unsere Default-Route – daher können wir die Information, dass sich in dieser Richtung auch noch ein konkretes Netz befindet, einfach unter den Tisch fallen lassen.

Destination	Next Hop
137.226.12.0/24	R3
137.226.13.0/24	R3
131.5.0.0/16	R1
145.1.2.0/24	R2
145.1.0.0/16	R2
20.1.1.0/28	R4
0.0.0.0/0	R4

Destination	Next Hop
137.226.12.0/23	R3
131.5.0.0/16	R1
145.1.0.0/16	R2
0.0.0.0/0	R4

Aufgabe 6.2: Network Address Translation (NAT)

NAT ist eine Möglichkeit, mit der Knappheit von IP-Adressen umzugehen.

a) Beschreiben Sie das *Prinzip von NAT*. Machen Sie dabei klar, ob durch dieses Prinzip irgendwelche *Vor- oder Nachteile* bei der Kommunikation zwischen Ihren eigenen Rechnern bzw. bei der Kommunikation Ihrer Rechner mit externen Rechnern entstehen.

b) Wie ändert sich die Situation durch die Einführung von *IPv6*?

Lösung 6.2

2.a) Prinzip von NAT: vergebe intern private Adressen, stelle nach außen hin nur eine einzige globale gültige Adresse bereit. Verlässt ein Paket das interne Netz, wird die Source-Adresse durch die global gültige Adresse ersetzt. Um eine spätere Rückübersetzung vornehmen zu können, wird ein Abbildungseintrag angelegt. Damit die Einträge eindeutig sind, muss auch noch der Absenderport mit angegeben werden – und eventuell auch ersetzt werden, falls bereits ein Eintrag mit diesem Port existiert.

Vor- und Nachteile bei der Kommunikation interner Rechner: Intern können die Rechner normal kommunizieren, da sie alle Adressen aus dem gleichen Adressbereich verwenden. Es entstehen also durch NAT keine Vorteile, aber auch keine Nachteile bei der Kommunikation interner Rechner.

Vor- und Nachteile bei der Kommunikation mit externen Rechnern: Kommunikation mit externen Rechnern ist erst nach dem Anlegen eines Eintrages möglich. Ein externer Rechner kann nicht direkt auf einen internen zugreifen – Einträge werden nur angelegt, wenn ein Paket das Netz verlässt, also nur auf Initiative eines internen Rechners. Um auch eine Initiierung der Verbindung von außen zuzulassen, muss eine statische Weiterleitung eingerichtet werden. Bei einem Webserver, der intern betrieben wird, müsste also z.B. Port 80 vom NAT auf die private Adresse weitergeleitet werden, die dem Webserver zugewiesen wurde. Bei Anwendungen, für die keine solchen standardisierten Ports existieren, muss man Hilfsprotokolle entwickeln. Das ist allerdings nicht Thema der Vorlesung.

Ebenso gibt es Anwendungsprotokolle, die die lokalen Adressinformationen auslesen (IP-Adresse und Port) und zur Initiierung einer Kommunikation an den Kommunikationspartner schicken. Da ist es nicht sehr hilfreich, wenn man die lokale private Adresse ausliest, die nach außen hin gar nicht gültig ist.

Ein NAT ist allerdings durch sein Verhalten automatisch auch ein Port-Filter, welcher unerwünschte Anfragen von außen nicht durchlässt. Einem potentiellen Angreifer bleibt die Struktur unseres Netzes verborgen und er findet keinen einfachen Ansatzpunkt, in unser Netz einzudringen. Diese Funktionalität erzielt man allerdings auch mit einer Firewall, so dass „Sicherheit“ kein Argument ist, unbedingt ein NAT aufzsetzen.

2.b) Bei IPv6 gibt es genügend Adressen. Daher wäre NAT nicht mehr nötig. Allerdings muss man manuell Port-Filter anlegen (Firewall), damit nicht alle internen Geräte von außen erreichbar sind. Denn jeder erreichbare Dienst ist auch ein potentielles Einfalltor. Viele Router/Modems von Providern bieten leider nur sehr schlechte Firewalls, die vielleicht sogar nur erlauben, alles von außen zu blocken oder alles rein zu lassen. NAT als zusätzliche Sicherheitsmaßnahme könnte man daher als sinnvoll ansehen. Aber auch hier kann man auch direkt überlegen, eine vernünftige Firewall anzuschaffen, die dann sogar noch mehr könnte.

Aufgabe 6.3: IP-Pakete und Fragmentierung

a) IP kann Pakete fragmentieren, um sie an die MTU (Maximum Transfer Unit) des auf der Sicherungsschicht verwendeten Protokolls anzupassen. Ebenso kann es die Fragmente zum ursprünglichen Paket zusammenfügen – allerdings erst beim Zielrechner. *Warum ist es sinnvoll, fragmentierte Pakete nicht schon in den zwischenliegenden Routern wieder zusammenzusetzen?*

b) Mittels IPv4 sollen 1690 Byte Nutzdaten verschickt werden. Die sendende IP-Instanz erzeugt aus diesen Daten ein Paket, indem sie den Standard-Header hinzugefügt; Optionen werden nicht verwendet. Zu versendende Pakete werden auf der Sicherungsschicht in Rahmen mit einem Header von 16 Byte Länge und einem Trailer (Prüfsumme) mit 4 Byte Länge gepackt. Die MTU der Sicherungsschicht sei 580 Byte, so dass eine Fragmentierung des IP-Pakets vorgenommen werden muss, bevor es an die Sicherungsschicht übergeben werden kann.

Wie viele Byte (inklusive aller Header und Prüfsummen) werden insgesamt über das Netzwerk übertragen? Skizzieren Sie für die Vermittlungs- und Sicherungsschicht alle PDUs (Payload mit Header und ggfs. Trailer) und geben Sie die jeweiligen Größen in Byte an. Sie brauchen keine konkreten Header-Felder für die Pakete/Fragmente bzw. Rahmen anzugeben; lediglich die für die Fragmentierung relevanten Informationen sollen pro Fragment korrekt angegeben werden.

c) Auf der Sicherungsschicht sei ein Rahmen der Übertragung aus Teil b) verfälscht worden. *Wie viele Rahmen müssen erneut versendet werden*, wenn die Nutzdaten zuverlässig übertragen werden sollen? Begründen Sie Ihre Aussage.

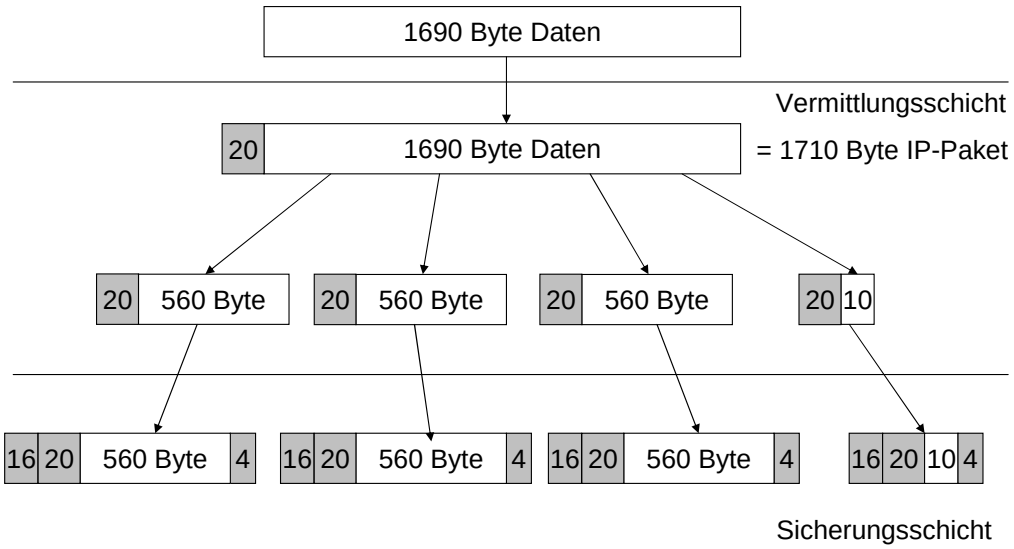
Anmerkung: Auf der Sicherungsschicht findet in diesem Fall – wie oft in der Praxis – nur Fehlererkennung, aber keine Fehlerbehandlung statt.

Lösung 6.3

a) Die Router würden mit deutlich mehr Arbeit belastet – sie müssten mit theoretisch bis zu 64 kByte an Fragmenten rechnen, die zwischengespeichert werden müssten, bevor ein vollständiges Paket weitergeleitet werden kann... und dies nicht nur für ein einzelnes Paket, sondern eventuell Tausende auf einmal. Hier würde also eine Menge Buffer und Verwaltungsaufwand benötigt. Und: auf dem nächsten Hop muss das Paket eventuell wieder fragmentiert werden, so dass die ganze Reassemblierung umsonst war.

Außerdem können Routen sich im Laufe der Zeit ändern, so dass ein Router, der breits einen Teil der zu reassemblierenden Dateneinheiten bekommen hat, nicht mehr auf dem Pfad der neuen Route liegt.

b) 1690 Byte heißt: das Paket ist zu groß und muss fragmentiert werden. Pro Rahmen können wir 580 Byte übertragen – und in jeden Rahmen muss ein komplettes Fragment inklusive IP-Header. Der ist 20 Byte groß, so dass pro Fragment noch 560 Byte Payload übertragen werden können. Dies ist ein Vielfaches von 8, also passen auch die 560 Byte Payload in ein Fragment. Damit werden insgesamt 4 Fragmente nötig:



Die Fragmentierungsinformationen für die vier Fragmente sind der Reihenfolge nach:

- ID=713, MF=1, Offset=0
- ID=713, MF=1, Offset=70
- ID=713, MF=1, Offset=140
- ID=713, MF=0, Offset=210

Die Identification kann beliebig gewählt werden – sie ist aber für jedes Paket eindeutig (solange der Nummernraum reicht) und muss in allen Fragmenten gleich sein, damit der Empfänger auch weiss, welche empfangenen Fragmente zu welchem Paket gehören (denn höchstwahrscheinlich muss ein ganzer Strom an Paketen fragmentiert werden und niemand garantiert uns die vollständige und reihenfolgetreue Übertragung).

MF ist in allen außer dem letzten Fragment 1, um anzuzeigen, dass weitere Fragmente folgen. Der Offset gibt in Vielfachen von 8 Byte an, an welcher Stelle im Originalpayload das aktuelle Fragment einzuordnen ist.

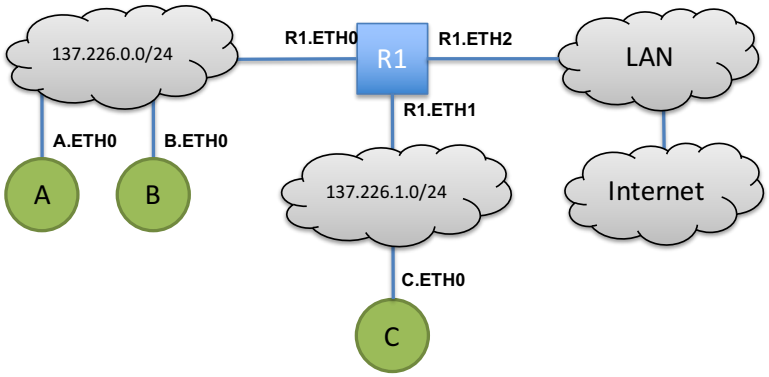
Im Prinzip relevant ist auch die `Total length` – denn die teilt dem Empfänger mit, wie viele Bytes ab dem Offset man denn empfangen hat, so dass der Empfänger auch beurteilen kann, ob er alle Fragmente empfangen hat oder ob zwischendurch eins fehlt. Aber das haben wir schon in der Abbildung stehen, das sind die 580 Byte Länge eines Fragments (Header + Payload).

Insgesamt übertragen werden in diesem Szenario: 1690 Byte Daten + 4 IP-Header (80 Byte) + 4 Sicherungsschicht-Header (80 Byte) = 1850 Byte.

c) Auf Schicht 3 sind mit IP keinerlei Sicherungsmaßnahmen gegen den Verlust von Paketen gegeben, und in Folge auch nicht gegen den Verlust von Fragmenten. Das bedeutet, dass ein auftretender Fehler nicht behandelt wird - geht ein Fragment verloren, ist das gesamte Paket beschädigt und wird verworfen! Es müssen also alle 4 Fragmente und somit alle 4 Rahmen erneut übertragen werden.

Aufgabe 6.4: Address Resolution Protocol (ARP)

Gegeben sei folgendes Netzwerk, in dem Router $R1$ zwei Subnetze `137.226.0.0/24` und `137.226.1.0/24` miteinander verbindet. In den Netzen befinden sich die drei Rechner A, B und C. Außerdem ist Router $R1$ mit einem Gateway `134.130.1.1` verbunden, welches ins Internet routet.



a) *Wie könnten die Routing-Tabellen des Routers und der drei Rechner aussehen?*

b) Alle ARP-Caches auf allen Systemen sind leer. Endsystem A möchte ein Paket an Endsystem C schicken. *Geben Sie alle ARP-Nachrichten und Paketübertragungen in der richtigen Reihenfolge an, die im Netzwerk übertragen werden, bis das Paket von A bei C angekommen ist.* Die MAC-Adresse einer Netzwerkkarte können Sie mit '`System.Interface`' angeben. Die MAC-Adresse der Ethernetkarte `ETH0` von Router $R1$ wäre z. B. `R1.ETH0`. Verwenden Sie folgende Formen für die Darstellung der Lösung:

ARP-Request: `Request <sender MAC> <sender IP> <receiver IP>` – <Inhalt/Zweck der Anfrage>

ARP-Reply: `Reply <sender MAC> <sender IP> <receiver MAC> <receiver IP>` – <Inhalt/Zweck der Antwort>

IP-Paket: `Data <sender MAC> <receiver MAC>`

Lösung 6.4

Wir können die IP-Adressen beliebig aus den entsprechenden Subnetzen wählen. Nur die IP-Adresse des Gateways steht fest.

a) Router $R1$ benötigt folgende Einträge, um Pakete korrekt weiterleiten zu können:

	Routing-Tabelle			
Router/IP-Adressen	Zielnetz	Interface	Gateway	Flags
R1				
ETH0:137.226.0.1	137.226.0.0/24	`ETH0`	*	U
ETH1:137.226.1.1	137.226.1.0/24	`ETH1`	*	U
ETH2:134.130.1.2	0.0.0.0/0	`ETH2`	134.130.1.1	UG

Alle Pakete, die ins eigene Netz gesendet werden, kann ein Endhost sofort auf den Link setzen (nachdem er die zugehörige MAC-Adresse herausgefunden hat). Jeglicher anderer Traffic wird zu $R1$ geschickt, da dieser das Gateway zu anderen Netzen (wie z.B. dem Internet) ist.

Beispiel für die Routing-Tabellen der Endsysteme:

	Routing-Tabelle			
Endsysteme/IP-Adressen	Zielnetz	Interface	Gateway	Flags
A				
ETH0:137.226.0.2	137.226.0.0/24	`ETH0`	*	U
	0.0.0.0/0	`ETH0`	137.226.0.1	UG
	127.0.0.0/8	`lo`	127.0.0.1	UH
B				
ETH0:137.226.0.3	137.226.0.0/24	`ETH0`	*	U
	0.0.0.0/0	`ETH0`	137.226.0.1	UG
	127.0.0.0/8	`lo`	127.0.0.1	UH
C				
ETH0:137.226.1.2	137.226.1.0/24	`ETH0`	*	U
	0.0.0.0/0	`ETH0`	137.226.1.1	UG
	127.0.0.0/8	`lo`	127.0.0.1	UH

b) Wir nehmen hier die Tabellen aus Aufgabenteil a).

Nach Routing-Tabelle von A muss das Paket über $R1$ geschickt werden, da sich C in einem anderen Netz befindet. Also benötigt A die MAC-Adresse des Gateways, bevor das Paket versendet werden kann:

- `Request A.ETH0 137.226.0.2 137.226.0.1` – A sucht MAC-Adresse zu 137.226.0.1 ($R1$)
- `Reply R1.ETH0 137.226.0.1 A.ETH0 137.226.0.2` – $R1$ teilt A seine MAC-Adresse mit
- `Data A.ETH0 R1.ETH0` (– Übertragung des Pakets von A an $R1$)

Nach Routing-Tabelle von $R1$ muss das Paket über `ETH1` geschickt werden. C befindet sich im selben Netz wie `R1.ETH1`:

- `Request R1.ETH1 137.226.1.1 137.226.1.2` – $R1$ erfragt zunächst die MAC-Adresse von C
- `Reply C.ETH0 137.226.1.2 R1.ETH1 137.226.1.1` – C teilt $R1$ seine MAC-Adresse mit
- `Data R1.ETH1 C.ETH0` (- Übertragung des Pakets von $R1$ an C)

Aufgabe 6.5: Netzwerkanalyse

In dieser Aufgabe sollen Sie Wireshark verwenden, um die Arbeit von IP und seinen ergänzenden Protokollen sowie die Interaktion mit anderen Schichten nachzuvollziehen. Wireshark für Linux, Windows oder MAC OS ist unter `http://www.wireshark.org/` erhältlich.

a) Machen Sie Sich zunächst mit Wireshark vertraut. Wie können Sie den Netzverkehr aufzeichnen? Was für Informationen enthalten die aufgezeichneten Daten? Wie finden Sie in der Menge der aufgenommenen Daten diejenigen, die Sie suchen?

b) Führen Sie nun einen `ping` auf `www.google.de` aus. Welche Auswirkungen hat dieser Befehl?

c) Führen Sie noch einmal einen `ping` aus, aber verändern Sie diesmal die Größe des Testpakets, das sie versenden, auf 2000 Byte. Was passiert? Wiederholen Sie diese Operation mit dem Ziel `www.rwth-aachen.de`. Was passiert nun?

d) Versuchen Sie als nächstes ein `traceroute` auf einen Rechner Ihrer Wahl. Wie arbeitet dieses Kommando?

e) Schauen Sie sich folgenden RFC an: https://tools.ietf.org/html/rfc3514. Glauben Sie, die Vorgehensweise erhöht die Sicherheit?

Einleitung des RFCs:

> „Firewalls [CBR03], packet filters, intrusion detection systems, and the like often have difficulty distinguishing between packets that have malicious intent and those that are merely unusual. The problem is that making such determinations is hard. To solve this problem, we define a security flag, known as the „evil“ bit, in the IPv4 [RFC791] header. Benign packets have this bit set to 0; those that are used for an attack will have the bit set to 1.“

Lösung 6.5

a) Wählt ein Interface aus (auf dem gerade was los ist) und loggt etwas Traffic mit. (Auswahl je nach Version entweder direkt im Startbildschirm oder über das Menü Aufzeichnen/Capture.)

Irgendeine ARP-Nachricht wird sicher vorbeikommen, also kann man dort schon mal genauer reinschauen: aha, hier stehen IP-Adressen und MAC-Adressen eingetragen. Und drumherum ist noch ein Ethernet-Frame mit Typ ARP, der an Broadcast geht. Als nächstes kann man mal eine Webseite aufrufen – dann gibt es auch IP-Pakete, in die man mal reinschauen und die einzelnen Header-Felder betrachten kann. Und danach kann man im Menü auf Analyze und dort auf Display Filters gehen, um sich anzuschauen, wie man all die dargstellten Pakete reduziert auf solche, die bestimmte Informationen enthalten.

b) Erst einmal wird DNS involviert, um den Namen auf eine IP-Adresse abzubilden. Das wird in der Vorlesung aber nicht behandelt. Dann kann man wunderbar sehen, dass ICMP-Nachrichten ausgetauscht werden, die wiederum in IP-Paketen übertragen werden.

c) Google reagiert nicht mehr auf die Ping-Anfrage. Schade. Aber eine Sache kann man sehen: die 2000 Byte werden in ein IP-Paket gesteckt, welches fragmentiert werden muss. Hier kann man also schön die Fragmente sehen.

   Dann nehmen wir uns den RWTH-Webserver vor, der reagiert nämlich auch auf Anfragen dieser Größe (zumindest aus dem RWTH-Netz). Und was wir mit Wireshark sehen, ist nun eventuell eine doppelte Fragmentierung: der Reply wurde etwas zu groß für unseren letzten Hop gewählt, drum hat man ein sehr kleines Fragment zwischendurch. Jedenfalls beim Ersteller dieser Lösung zu Hause (8 Byte Payload im zweiten von drei Fragmenten). Hier kann man sich also auch noch die Fragmentierung in der Praxis anschauen.

d) Nachverfolgung der Route eines Pakets gibt's ja prinzipiell schon über die IP-Optionen, aber in unbrauchbar. (Die Optionen können maximal 40 Byte an Daten aufnehmen, so dass nur 9 IP-Adressen aufgezeichnet werden können, was in der Praxis meist zu wenig ist. Traceroute (Windows: `tracert`) arbeitet hier, wie man schön in den Nachrichten sehen kann, mit einem Versenden eines Probepakets mit TTL=1; der erste empfangende Router zählt diese auf 0 runter, verwirft das Paket direkt und schickt eine ICMP-Fehlermeldung zurück. Damit kennen wir schon mal den ersten Hop. Nun geht es mit TTL=2, 3, usw weiter, bis wir das Ziel schließlich erreicht haben. Trace abgeschlossen.

e) Bei dem RFC handelt es sich natürlich um einen Aprilscherz. Bei der IETF hat man anscheinend zu viel Zeit. Daher gibt es sehr viele solcher RFCs ;).