Sascha Paulus Matrikelnr.: 1536783

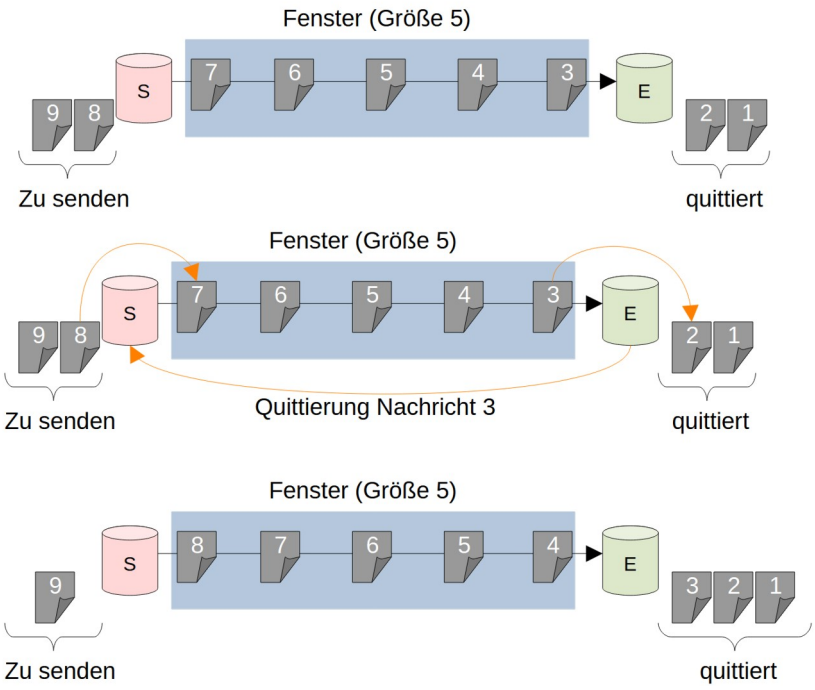
Rechnernetze

Übungsblatt 6

Aufgabe 1: *Konzepte*

Sliding-Window:

Bei Sliding-Window-Protokollen erfolgt die Kommunikation über einen Quittierungsbetrieb. Das heißt, dass der Empfänger den erfolgreichen Empfang dem Sender rückmeldet. Damit der Sender nicht immer ein Paket schickt und dann auf die Antwort wartet und es so zu möglichen Wartezeiten kommt, weil die Kapazität der Netzwerkleitung nicht optimal genutzt wird. Daher wird Sliding-Window mit einer festgesetzten Fenstergröße eingesetzt. Die Fenstergröße besagt, wie viele Pakete der Sender senden kann, ohne eine Quittierung zu erhalten. Ist die Größe bspw. auf 5 festgelegt, so kann der Sender 5 Pakete versenden, muss dann jedoch vor dem nächsten Versenden warten, bis mindestens ein Paket vom Empfänger quittiert wurde. Veranschaulicht man sich dies visuell, so kann man die Namensherkunft auch leicht erkennen:

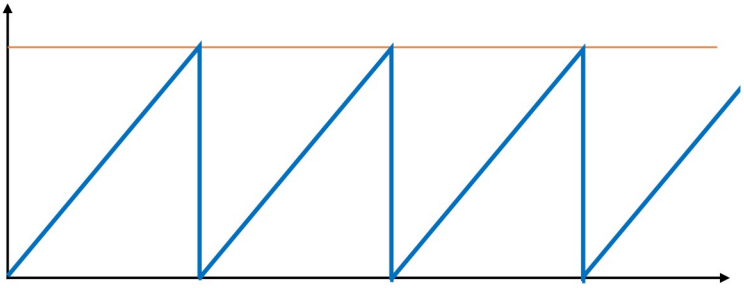


Die Größe des Sliding-Windows wird normalerweise dynamisch gestaltet, sodass die Kapazität der Netzwerkleitung wirklich optimal ausgenutzt werden kann. Die Größenanpassung ist dabei abhängig von der jeweiligen Protokollart. Im folgenden werden einige Beispiele genannt.

TCP Tahoe:

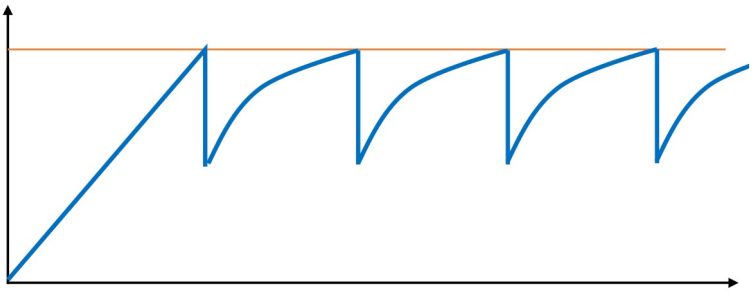
Bei TCP Tahoe wird die Fenstergröße auf 1 gesetzt und anschließend linear erhöht. Empfängt der Sender 3 duplicate acknowledgment, so wird die Fenstergröße wieder auf 1 reduziert und wie zu Beginn erhöht. Duplicate acknowledgment werden vom Empfänger übermittelt, wenn er Pakete in falscher Reihenfolge (bspw. 2,4,3) erhält. Dies kann entweder

durch die unterschiedlichen „Transportwege“ durchs Netzwerk oder durch tatsächliche Paketverluste auftreten. Die Arbeitsweise ist auch im folgenden Diagramm zu erkennen:



TCP Reno:

Bei TCP Reno wird die Fenstergröße auch mit 1 initialisiert und anschließend über die Zeit erhöht. Werden irgendwann 3 duplicate acknowledgment empfangen, so wird die Fenstergröße halbiert und anschließend wieder erhöht, wie das folgende Diagramm zeigt:



TCP Vegas:

Wie bereits beschrieben ist ein duplicate acknowledgment nicht unbedingt ausschlaggebend für den tatsächlichen Verlust von Paketen. TCP Vegas setzt daher auf eine zusätzliche Prüfung durch die Berechnung der Round-Trip Time (RTT), welche die Zeitdauer für das Senden eines Pakets von Sender zu Empfänger und die Zeitdauer für das Übermitteln des acknowledgment beinhaltet. Die Berechnung kann natürlich erst nach ein paar Sendungen erfolgen, wird dann aber verwendet, um eine möglichst optimale Fenstergröße daraus abzuleiten. Verkleinert wird die Fenstergröße wenn ein duplicate acknowledgment empfangen wird und zusätzlich auch die RTT überschritten ist. Dieses Verfahren nutzt die Kapazität der Netzwerkleitung ziemlich optimal aus (bspw. auch 40% mehr Durchsatz als TCP Reno).

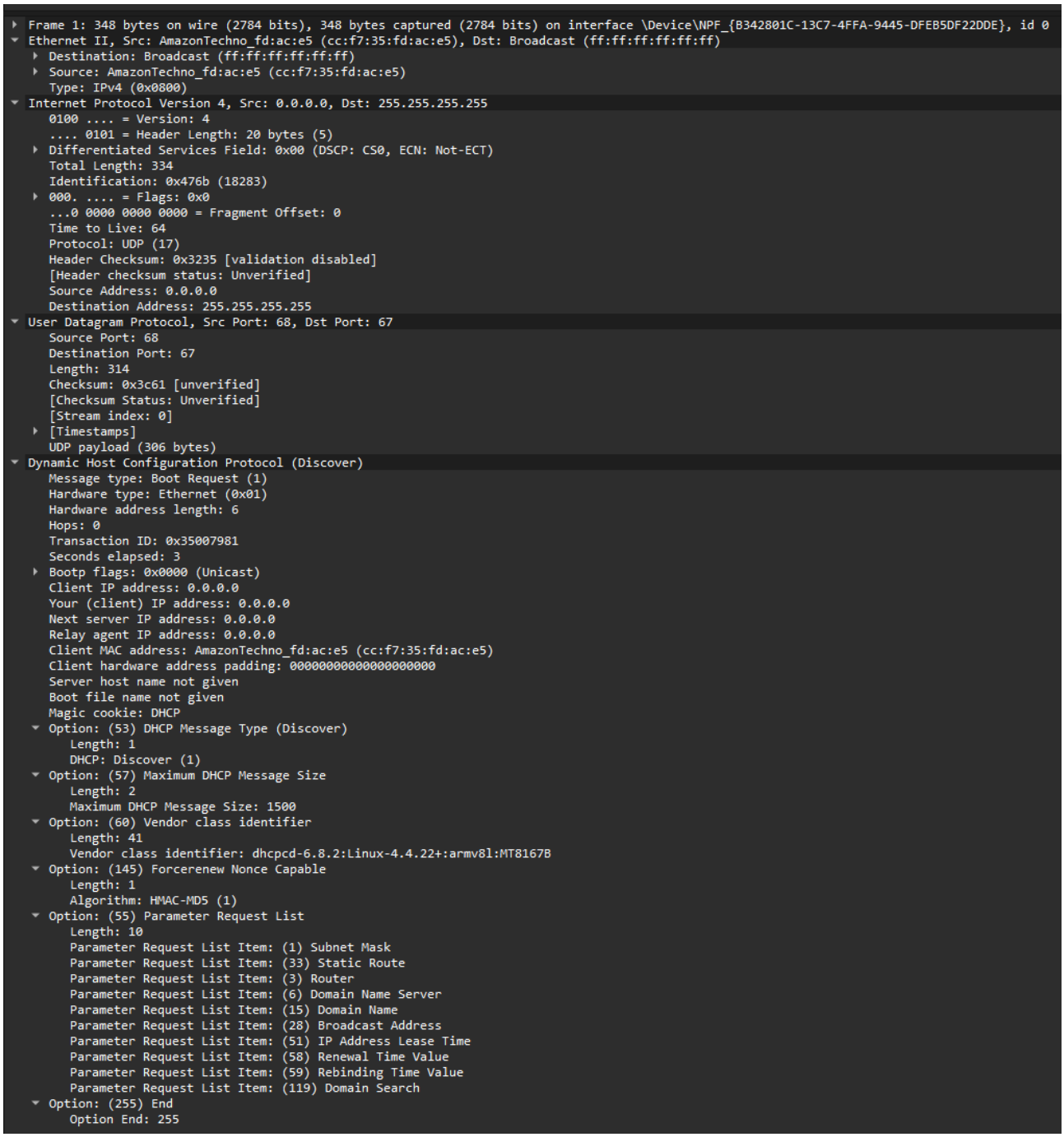
Einteilung von behandelten Protokollen ins ISO/OSI-Modell:

Protokoll	**ISO/OSI-Modell Schicht**	**Begründung**
Transmission Control Protocol (TCP)	4: Transport Layer	- Ende-zu-Ende-Protokoll - Flußkotnrolle - Reliable Stream - …
User Datagramm Protocol (UDP)	4: Transport Layer	- Ende-zu-Ende-Protokoll - Datagramm (verbindungslos) - …
Internet Protocol (IP)	3: Network Layer	- Host-to-Host Protocol - Adressierung …

Adress Resolution Protocol (ARP)	3: Network Layer (2: Link Layer)	- Adressierung → Umwandlung zu Mac-Adresse (teilw. Link-Layer) - …
Internet Control Message Protocol (ICMP)	3: Network Layer	- Management Funktionen - Netzwerk Diagnose - Austausch von Routing Infos - Fehler Behandlung - …
Domain Name System (DNS)	7: Application Layer	- Client-Server Funktion - Anwendungsprotokoll - Umwandlung zwischen IP Adressen lesbar für Mensch und Maschine - …
Dynamic Host Configuration Protocol (DHCP)	7: Application Layer	- Client-Server Funktion - Anwendungsprotokoll - Netzwerk Konfiguration - …
Routing Information Protocol (RIP)	3: Network Layer	- Routingprotokoll - Austausch von Routing Infos - …
Open Shortest Path First (OSPF)	3: Network Layer	- Routingprotokoll - Management Funktionen → Routing Optimierung - …
Border Gateway Protocol (BGP)	3: Network Layer	- Routingprotokoll - Austausch von Routing Infos - Management Funktionen → Optimierung Paketweiterleitung - …
Neighbor Discovery Protocol (NDP)	3: Network Layer	- ARP für IPv6 (+ Zusatz) - Adressierung - …

Aufgabe 2: *DHCP*

Um überhaupt DHCP Pakete zu empfangen/ zu senden, muss ich in den Netzwerkeinstellungen des PCs eine automatische Konfiguration einstellen (bspw. IP und DNS). Zum Filtern habe ich die festgelegten Ports von DHCP (67 und 68) verwendet. Folgende Informationen konnte ich aus einem Paket entnehmen:



Von oben nach unten konnte ich folgende Informationen herauslesen:
DHCP Pakete werden per Broadcast versendet.
Es wird (in meinem Netzwerk) IPv4 verwendet.
Zum Einsatz kommt UDP auf den spezifizierten Ports 67 (Server) und 68 (Client).
Passend zur MTU des Ethernet ist die maximale Paket Größe 1500 Bytes.
Es werden über DHCP verschiedene Parameter angefragt/ übermittelt:

- die Subnetzmaske

- die statische(n) Routen zum DHCP Clients (Eintrag in Routing-Tabelle)
- die IP Adresse des Routers des Netzwerks
- die IP Adresse des Domain Name Servers
- den Suffix der für DNS Resolutionen von unvollständigen Domain Namen verwendet werden soll.
- die Broadcast Adresse
- die Zeit wie lange die aktuelle IP Adresse genutzt werden kann (Gültigkeitsdauer)
- die Zeit wie lange gewartet wird, bevor der DHCP Server für eine Verlängerung der Gültigkeitsdauer der IP Adresse kontaktiert wird.
- die Zeit wie lange gewartet wird, bevor per Broadcast an alle verfügbaren DHCP Server eine Anfrage zur Verlängerung der Gültigkeitsdauer der IP Adresse versendet wird.
- eine Liste von Domain Namen, die der Client als DNS benutzen kann/ sollte.

Aufgabe 3: *Nmap*

a) Nmap-Befehl: nmap -sn 192.168.178.0/24
   -sn: ping scan ohne Ports
   192.168.178.0/24: Netzwerk range
   Zum Messzeitpunkt waren 20 Hosts im Netzwerk aktiv.
b) PowerShell-Befehl: nslookup scanme.nmap.org
   → liefert 192.168.178.1 (IP Adresse des Routers)
   Nmap-Befehl: nmap -O 192.168.178.1
   -O: Betriebssystem Erkennung
   Es wird das Betriebssystem Linux 4.19 verwendet.
c) Befehl: whois nmap.org
   Die Webseite wurde am 18.01.1999 registriert.
d) Es gibt verschiedene Flags die man zum effizienten Scan verwenden kann, z.B.:
   -TX ($X \in \{0,\ldots,6\}$): Anpassung der Geschwindigkeit (Vorsicht mit Netzwerkbelastung etc.)
   -Pn: Schaltet das Analysieren der Hosts aus
   -n: Verhindert eine reverse DNS Resolution
   → mit diesen Flags ist Nmap in der Lage einen parallelen Scan durchzuführen.
   Alternativ kann man vordefinierte Scan Techniken verwenden wie bspw. SYN Scan.
e) SYN Scan kann zum scannen von TCP Ports verwendet werden. Dabei arbeitet er wie folgt: Im Gegensatz zum normalen TCP Verbindungsaufbau (three-way-handshake) sendet die scannende Maschine nach Erhalt des acknowledgment der Zielmaschine bereits ein reset anstatt einem acknowledgment, sodass die Verbindung gar nicht erst eröffnet, sondern terminiert wird.
f) Folgende TCP Ports sind am häufigsten aufgetreten:
   Port 53: Domain Service
   Port 80: http Service
   Port 443: https Service
   Port 8080: http proxy Service

Aufgabe 4: *Routing*

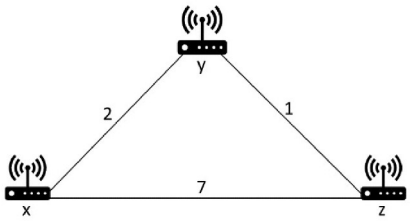
Algorithmus:

1) Initialisiere die Routing-Tabelle mit den bekannten Distanzen zu den direkten Nachbarn.
2) Schicke die Informationen an die direkten Nachbarn.
3) Terminiere, wenn du keine neuen Informationen von deinen Nachbarn erhalten hast und du keine neuen Informationen verschicken kannst.
4) Verarbeite die erhaltenen Informationen der Nachbarn, indem du (bestmögliche) Distanzen zu neu erreichbaren Routern in der Tabelle ergänzt.
   Berechne außerdem, ob für bereits bestehende Einträge nun Wege mit kürzerer Distanz bekannt sind und aktualisiere gegebenenfalls die Einträge.
5) Springe zu Schritt 2)

Unterschied zu Link-State-Verfahren:

Bei Link-State-Verfahren werden komplexere Informationen ausgetauscht. Außerdem werden Informationen meist nicht nur zum direkten Nachbarn weitergegeben, sondern per Multicast verschickt. Außerdem werden nicht jedes mal die kompletten Tabellen ausgetauscht, sondern nur Informationen über Änderungen.

a)



Von x	Via x	Via y	Via z
Zu x	---	---	---
Zu y	---	2	
Zu z	---		7

Von y	Via x	Via y	Via z
Zu x	2	---	
Zu y	---	---	---
Zu z		---	1

Von z	Via x	Via y	Via z
Zu x	7		---
Zu y		1	---
Zu z	---	---	---

Von x	Via x	Via y	Via z
Zu x	---	---	---
Zu y	---	2	8
Zu z	---	3	7

Von y	Via x	Via y	Via z
Zu x	2	---	8
Zu y	---	---	---
Zu z	9	---	1

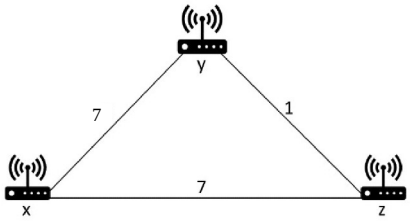
Von z	Via x	Via y	Via z
Zu x	7	3	---
Zu y	9	1	---
Zu z	---	---	---

Von x	Via x	Via y	Via z
Zu x	---	---	---
Zu y	---	2	8
Zu z	---	3	7

Von y	Via x	Via y	Via z
Zu x	2	---	4
Zu y	---	---	---
Zu z	5	---	1

Von z	Via x	Via y	Via z
Zu x	7	3	---
Zu y	9	1	---
Zu z	---	---	---

b)



Von x	Via x	Via y	Via z
Zu x	---	---	---
Zu y	---	7	
Zu z	---		7

Von y	Via x	Via y	Via z
Zu x	7	---	
Zu y	---	---	---
Zu z		---	1

Von z	Via x	Via y	Via z
Zu x	7		---
Zu y		1	---
Zu z	---	---	---

Von x	Via x	Via y	Via z
Zu x	---	---	---
Zu y	---	7	8
Zu z	---	8	7

Von y	Via x	Via y	Via z
Zu x	7	---	8
Zu y	---	---	---
Zu z	14	---	1

Von z	Via x	Via y	Via z
Zu x	7	8	---
Zu y	14	1	---
Zu z	---	---	---

Von x	Via x	Via y	Via z
Zu x	---	---	---
Zu y	---	7	8
Zu z	---	8	7

Von y	Via x	Via y	Via z
Zu x	7	---	8
Zu y	---	---	---
Zu z	14	---	1

Von z	Via x	Via y	Via z
Zu x	7	8	---
Zu y	14	1	---
Zu z	---	---	---

Der kostengünstigste Pfad von z nach x ändert sich von 3 auf 7 Kosten und verläuft nun direkt zu x anstatt wie vorher über y.

c) Einen Ausfall von Router D kann man so modellieren, dass die Pfadkosten von C nach D unendlich betragen. Das führt (nach obigen Algorithmus) dazu, dass C versucht über B nach D zu gelangen, da die Tabelle einen existierenden und kostengünstigeren Pfad beinhaltet. Die Pfadkosten von C nach D über B steigen nun aber auch an. Dies wird wieder mittels des Algorithmus den jeweiligen Nachbarn kommuniziert. C weiß jedoch, dass er auch über A zu D kommen kann und nimmt diese Route, doch auch da wird festgestellt, dass die Pfadkosten gestiegen sind und damit die Tabelle aktualisiert und den Nachbarn weitergeleitet. So werden immer größere indirekte Wege verfolgt und die Pfadkosten Stück für Stück erhöht. Da, wie zu Beginn erwähnt, der Ausfall von Router D durch unendlich große Pfadkosten modelliert wird, bricht die Erhöhung der Pfadkosten niemals ab (Algorithmus terminiert nicht). Diese endlose Schleife führt dazu, dass die Router den Ausfall von D nicht bemerken. Dieser Effekt ist auch als Count-to-infinity Effekt bekannt, dessen Name durch diese Modellierung gut ersichtlich wird.