Uebung 06 – Maxim Smirnov

Aufgabe 1: Konzepte)

Sliding-Window:

Sliding-Window wird in TCP verwendet um mehrere Pakete zu versenden, ohne für diese einen ACK-Nachricht zu erwarten. Mit der Fenstergröße wird bestimmt wie viele Pakete ohne ACK-Nachricht quittieren versendet werden. So kann man effizienter viele Daten versenden, ohne dass man für jedes Paket auf eine ACK-Nachricht warten muss. → Speicherfähigkeit des Mediums wird besser genutzt, da die Leitung während des Sendens durchgehend ausgelastet ist.

TCP-Tahoe:

TCP-Tahoe nutzt die Sliding-Windows und falls Pakete verloren gehen, wird dies durch ein Timeout erkannt, also wenn kein ACK-Nachricht empfangen wird. Es kann aber auch durch Duplicate ACKs erkannt werden, was schneller ist als über Timeouts.
Das funktioniert indem, für jedes Paket was in der richtigen Reihenfolge ankommt eine ACK-Nachricht mit der nächsten Position versendet wird. Wenn jetzt vier Pakete versendet werden aber Paket 2 fehlt, dann geschieht folgendes:

-Paket 1 wird empfangen → ACK (warte auf 2)
-Paket 2 fehlt
-Paket 3 wird empfangen → ACK (warte auf 2)
-Paket 4 wird empfangen → ACK (warte auf 2)

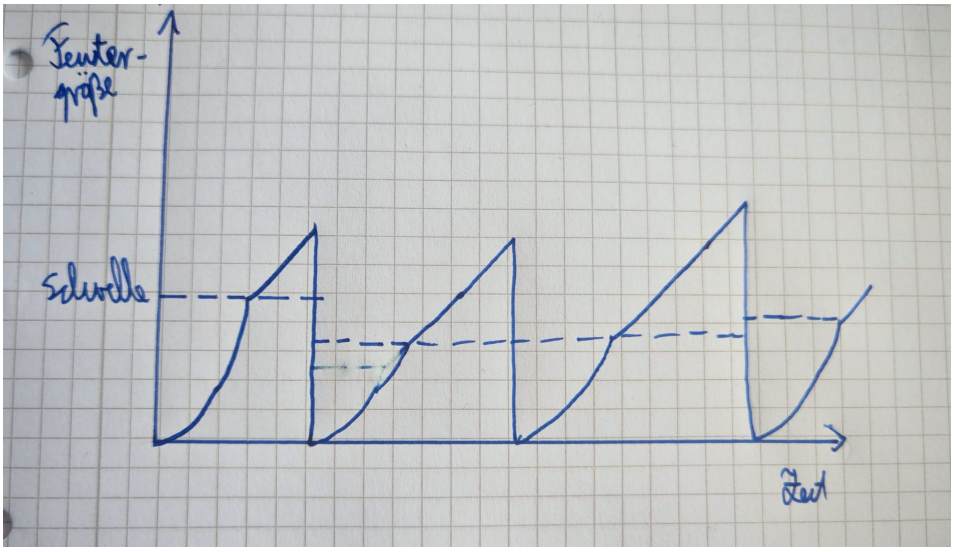
→ drei duplicated ACKs, Sender erkennt das Paket verloren ging.

Wenn jetzt das Paket 2 aber nach Paket 3 ankäme dann würde folgendes passieren:

-Paket 1 wird empfangen → ACK (warte auf 2)
-Paket 2 fehlt
-Paket 3 wird empfangen → ACK (warte auf 2)
-Paket 2 wird empfangen → ACK (warte auf 4)
-Paket 4 wird empfangen → ACK (warte auf 5)

→ Der Counter wird dann auf die aktuelle Position angehoben und es werden keine duplicate ACK's erkannt. Alles wurde richtig empfangen. Die ACK (warte auf 5) Nachricht bestätigt dann das alle Pakete bis 5 richtig empfangen wurde, da die Position aktualisiert wurde.

Dann verwendet TCP-Tahoe „Slow Start“, d.h. Die Fenstergröße beginnt bei 1. Nach jedem erfolgreichen versenden, wird die Größe verdoppelt, da nach jedem ACK die Fenstergröße um 1 erhöht wird.
Das passiert bis eine vordefinierte Schwelle erreicht ist. Ab dort wird die Größe nur linear um 1 für jeden RTT erhöht. Wenn ein Paketverlust erkannt wird, wird die Fenstergröße wieder auf 1 gesetzt und die Schwelle auf die Hälfte der bisherigen Fenstergröße gesetzt. → AIMD (lineares Wachstum/exponentielle Reduktion)



Hier sieht man wie sich die Fenstergröße verdoppelt bis die Schwelle erreicht wird und danach nur noch linear steigt. Bei Paketverlust fällt die Fenstergröße auf 1 und die Schwelle wird auf die Hälfte der bisherigen Fenstergröße gesetzt.

TCP-Reno:

Funktioniert ähnlich wie Tahoe, aber anstatt bei einem Paketverlust wieder zum Slow Start überzugehen wird hier die Schwelle und die Fenstergröße, beide auf die Hälfte der bisherigen Fenstergröße gesetzt. Also geht es sofort in AIMD → lineares Wachstum/exponentielle Reduktion → Die Verbindung wird nicht sofort komplett gedrosselt wie bei Tahoe.

TCP-Vegas:

TCP-Vegas misst die RTT der Pakete zum Empfänger und versucht daraus die optimale Fenstergröße zu ermitteln. Wenn sich zu viele Pakete stauen wird die Fenstergröße verringert, falls zu wenige Pakete im Netz sind, dann wird die Fenstergröße erhöht.

→ Paketverluste werden schon im Voraus vermieden und dadurch wird eine konstante Datenübertragung erzielt.

Protokolle	ISO/OSI Schicht	Begründung
IP (IPv4, IPv6)	Network-Layer	Zuständig für das Routing von Paketen im Netzwerk.
ICMP	Network-Layer	Wird verwendet um Fehlermeldungen/Diagnose-Nachrichten auszutauschen (z.B. Ping)
TCP	Transport-Layer	Verbindungsaufbauende Kommunikation mit Fehlerkorrektur
UDP	Transport-Layer	Verbindungslose und unzuverlässige Kommunikation
DNS	Application-Layer	Übersetzt Domainnamen in IP-Adressen
DHCP	Application-Layer	Vergibt IP-Adressen und andere Konfigurationsdaten an Clients.

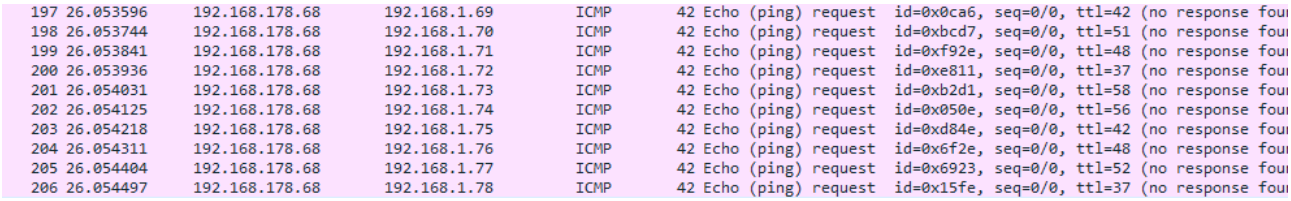
Ethernet	Physical-Layer	Definition von mechanischen Eigenschaften
Ethernet	Link-Layer	Definition wie ein Ethernet-Frame aufgebaut ist.

Aufgabe 2: Nmap)

a) Wie viele Hosts befinden sich in ihrem lokalen Klasse-C-Netz?
Benutzter Befehl: nmap -sn <lokale IP>/24
-sn: keinen Portscan machen nachdem Host gefunden wurde

Ergebnis: Nmap done: 256 IP addresses (15 hosts up) scanned in 52.89 seconds

In Wireshark sieht man wie eine Menge von ICMP Paketen versendet werden:



b) Welches Betriebssystem wird von scanme.nmap.org verwendet?

Befehl: nmap -O scanme.nmap.org

-O: Betriebssystem Erkennung

Ergebnis:
Aggressive OS guesses: Linux 2.6.32 (88%), Ubiquiti WAP (Linux 2.6.32) (88%).... usw.
No exact OS matches for host (test conditions non-ideal).

c) An welchem Datum wurde die Webseite nmap.org registriert?

Befehl: whois nmap.org

Domain Name: NMAP.ORG
Registry Domain ID: 5ed7a21fc9f74f97b55511f9857111f0-LROR
Registrar WHOIS Server: whois.dynadot.com
Registrar URL: http://www.dynadot.com
Updated Date: 2023-08-26T05:16:38.0Z
Creation Date: 1999-01-18T05:00:00.0Z

d) Wie kann man möglichst effektiv eine größere Menge an Adressen nach offenen TCP-Ports scannen?

Befehl: nmap -T4 -p 1-1024 <lokale IP>/24

-T4 : schnellere Timings, kürzere Timeouts usw.
-p 1-1024: Nur die meist genutzten Ports scannen

e) Wie funktioniert der SYN-Scan und für was kann man ihn verwenden?
Mit -sS macht man einen SYN-Scan und man verwendet ihn für schnelle Portscans ohne eine richtige Verbindung aufzubauen. Es wird eine SYN-Nachricht geschickt und auf ein SYN-ACK-

Nachricht gewartet, falls diese ankommt, ist der Port offen. Es wird aber keine ACK-Nachricht zurückgeschickt.

f) Welches sind die offenen Ports, die bei Ihren bisherigen Nmap-Scans am häufigsten auftreten, und wofür werden sie verwendet?

Bei mir Ports die häufig auftraten mit Erklärung von ChatGPT:

80 & 443: Jeder Webserver bietet mindestens Port 80 und 443 an.

21: Manche Geräte (Drucker, NAS) haben noch FTP aktiviert.

53: DNS-Server sind auf fast jedem Router oder Server aktiv.

135/139/445: Auf Windows-Rechnern für RPC und Datei-/Druckerfreigabe.

515 & 631: Auf Netzwerkdruckern häufig aktiv.

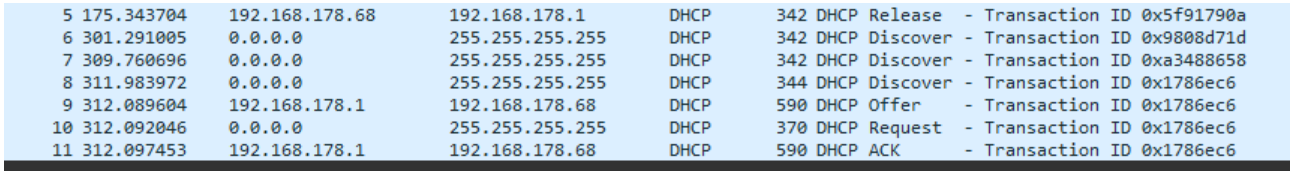
554: Typisch für Streaming-Geräte, Kameras oder Server mit Media-Streams.

Das klingt auch logisch, da ich im Netzwerk eine Überwachungskamera habe -> Port 554 und auch einen Drucker usw.

Aufgabe 3 – DHCP)

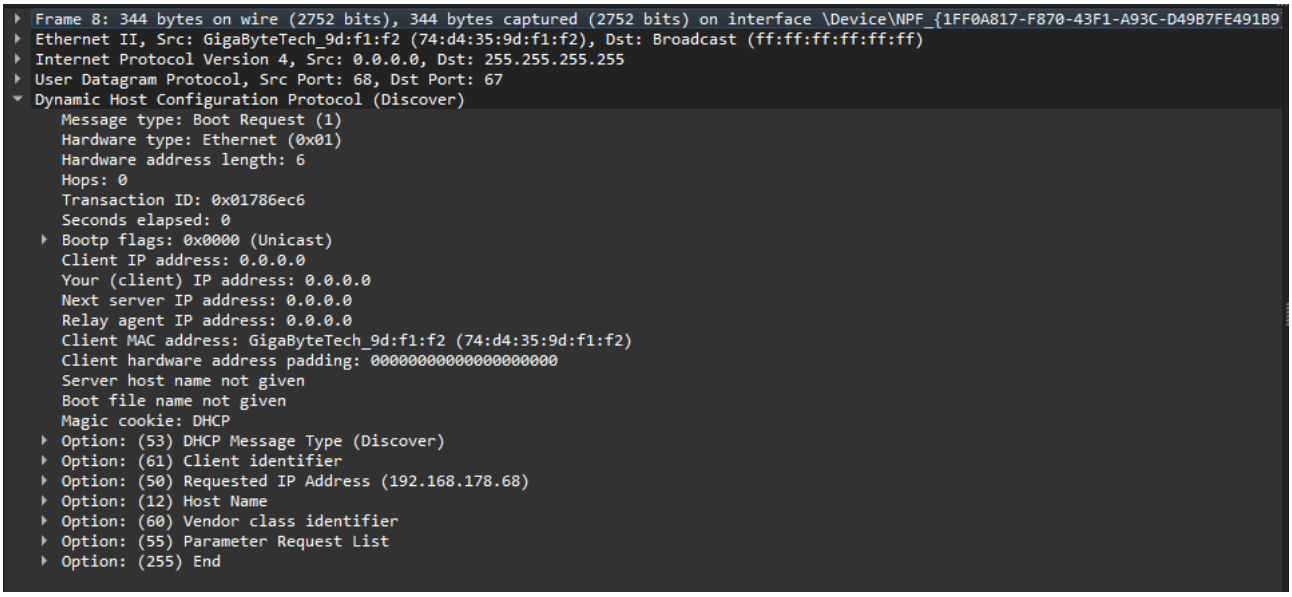
Wireshark Capture Filter: port 67 or port 68

Mit ipconfig /renew und ipconfig /release in der CMD hab ich dann 4 verschiedene Pakete in Wireshark abgefangen.



5	175.343704	192.168.178.68	192.168.178.1	DHCP	342 DHCP Release - Transaction ID 0x5f91790a
6	301.291005	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0x9808d71d
7	309.760696	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0xa3488658
8	311.983972	0.0.0.0	255.255.255.255	DHCP	344 DHCP Discover - Transaction ID 0x1786ec6
9	312.089604	192.168.178.1	192.168.178.68	DHCP	590 DHCP Offer - Transaction ID 0x1786ec6
10	312.092046	0.0.0.0	255.255.255.255	DHCP	370 DHCP Request - Transaction ID 0x1786ec6
11	312.097453	192.168.178.1	192.168.178.68	DHCP	590 DHCP ACK - Transaction ID 0x1786ec6

Das DHCP Discover Paket:



```
▸ Frame 8: 344 bytes on wire (2752 bits), 344 bytes captured (2752 bits) on interface \Device\NPF_{1FF0A817-F870-43F1-A93C-D49B7FE491B9
▸ Ethernet II, Src: GigaByteTech_9d:f1:f2 (74:d4:35:9d:f1:f2), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
▸ User Datagram Protocol, Src Port: 68, Dst Port: 67
▾ Dynamic Host Configuration Protocol (Discover)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x01786ec6
    Seconds elapsed: 0
  ▸ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: GigaByteTech_9d:f1:f2 (74:d4:35:9d:f1:f2)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▸ Option: (53) DHCP Message Type (Discover)
  ▸ Option: (61) Client identifier
  ▸ Option: (50) Requested IP Address (192.168.178.68)
  ▸ Option: (12) Host Name
  ▸ Option: (60) Vendor class identifier
  ▸ Option: (55) Parameter Request List
  ▸ Option: (255) End
```

Das Discover-Paket sagt aus, dass der Client nach einem DHCP-Server sucht.
Hier sieht man die Absender-IP: 0.0.0.0
und die Requested-IP 192.168.178.68, also die Wunsch-IP

Das DHCP Offer Paket:

```
User Datagram Protocol, Src Port: 67, Dst Port: 68
Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x01786ec6
    Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.178.68
    Next server IP address: 192.168.178.1
    Relay agent IP address: 0.0.0.0
    Client MAC address: GigaByteTech_9d:f1:f2 (74:d4:35:9d:f1:f2)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Offer)
  > Option: (54) DHCP Server Identifier (192.168.178.1)
  > Option: (51) IP Address Lease Time
  > Option: (58) Renewal Time Value
  > Option: (59) Rebinding Time Value
  > Option: (1) Subnet Mask (255.255.255.0)
  > Option: (3) Router
  > Option: (6) Domain Name Server
  > Option: (15) Domain Name
  > Option: (255) End
    Padding […]: 000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

Hier bekommt der Client die gewünschte IP-Adresse → 192.168.178.68
Wenn man die Lease-Time öffnet, sieht man, dass der Client die IP für 10 Tage hat.

Das DHCP Request Paket:

```
> Ethernet II, Src: GigaByteTech_9d:f1:f2 (74:d4:35:9d:f1:f2), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
v Dynamic Host Configuration Protocol (Request)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x01786ec6
    Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: GigaByteTech_9d:f1:f2 (74:d4:35:9d:f1:f2)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Request)
  > Option: (61) Client identifier
  > Option: (50) Requested IP Address (192.168.178.68)
  > Option: (54) DHCP Server Identifier (192.168.178.1)
  > Option: (12) Host Name
  > Option: (81) Client Fully Qualified Domain Name
  > Option: (60) Vendor class identifier
  > Option: (55) Parameter Request List
  > Option: (255) End
```

Der Client bestätigt die Nachricht vom Server und nimmt die IP-Adresse an.

Das DHCP ACK Paket:

```
▸ User Datagram Protocol, Src Port: 67, Dst Port: 68
▾ Dynamic Host Configuration Protocol (ACK)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x01786ec6
    Seconds elapsed: 0
  ▸ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.178.68
    Next server IP address: 192.168.178.1
    Relay agent IP address: 0.0.0.0
    Client MAC address: GigaByteTech_9d:f1:f2 (74:d4:35:9d:f1:f2)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▸ Option: (53) DHCP Message Type (ACK)
  ▸ Option: (54) DHCP Server Identifier (192.168.178.1)
  ▸ Option: (51) IP Address Lease Time
  ▸ Option: (58) Renewal Time Value
  ▸ Option: (59) Rebinding Time Value
  ▸ Option: (1) Subnet Mask (255.255.255.0)
  ▸ Option: (3) Router
  ▸ Option: (6) Domain Name Server
  ▸ Option: (15) Domain Name
  ▸ Option: (255) End
    Padding […]: 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

Damit bestätigt der Server die Anfrage vom Client und überträgt noch die Subnetzmaske.

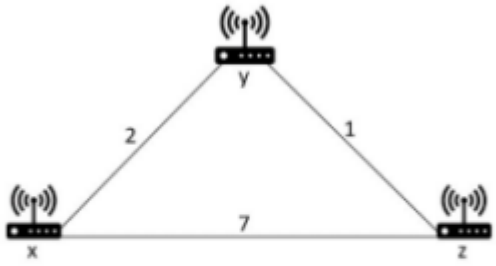
(siehe DHCP.pcapng)

Aufgabe 4: Routing)

Im ersten Schritt werden die Nachbarknoten jedes Routers und ihre Kosten initialisiert.
Mit jeder Aktualisierung werden die Tabellen mit den Nachbarn verglichen und die Kosten werden entsprechend aktualisiert. Das wird wiederholt bis keine Tabelle mehr aktualisiert werden kann.

Das Link-State-Verfahren teilt seine Tabelle mit allen Routern im Netzwerk und nicht wie hier nur mit seinen Nachbarknoten. Außerdem hat jeder Router eine identische vollständige Karte des gesamten Netwerks.

a) Bestimmen Sie die Routingtabellen in jedem Schritt, indem Sie Ihren zuvor entwickelten Algorithmus anwenden. Geben Sie danach den kostengünstigsten Weg von Router „z“ zu Router „x“ an. Sie können die Lösungen einfach direkt in die Tabellen der PDF einfügen und diese in den Pull Request hinzufügen.



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		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			

Der kostengünstigste Weg von Router „z“ zu Router „x“ ist über „y“ mit den Kosten 3.

b) Die Kosten zwischen „x“ und „y“ steigen nun von 2 auf 7. Berechnen Sie die Routingtabellen mit Hilfe des Algorithmus. Ändert sich der kostengünstigste Pfad von „z“ nach „x“?

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 Weg von Router „z“ zu Router „x“ ist über „x“ mit den Kosten 7 , also um 4 teurer. Und jetzt direkt über „x“ statt über „y“.

c) Sehen Sie sich den unteren Graphen an. Router „D“ fällt nun auf einmal aus. Beschreiben Sie, ob und wann die anderen Router merken, dass keine Verbindung mehr zu „D“ möglich ist.

C würde es als erstes merken, wenn er Pakete an D schickt und keine ACKs mehr bekommt, dieser Router würde dann D aus seiner Routing-Tabelle entfernen und seine Nachbarknoten informieren. Diese würden dann entsprechend ihre Routing-Tabellen aktualisieren.