

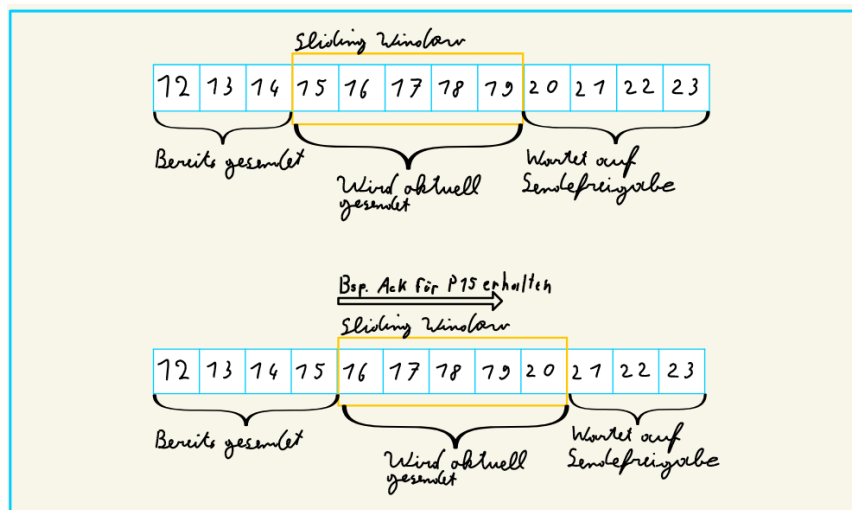
Übungsblatt 6

Aufgabe 1

Sliding-Window Protocol

Das Sliding-Window Protocol dient der Flusskontrolle bei TCP-Kommunikation. Um zu verhindern, dass der Sender dem Empfänger zu weit vorausseilt und diesen Überlastet, bestimmt das „Sliding-Window“, welche Pakete bereits gesendet werden dürfen und welche nicht.

Der Sender kann immer die Pakete versenden, die sich aktuell im Sliding-Window befinden, sollte es die Bestätigung bekommen, dass eins der Pakete angekommen ist, verschiebt sich das Sliding-Window um einen Schritt weiter.



TCP Tahoe

TCP Tahoe ist ein Protokoll zur Vermeidung von „Congestion“ zu vermeiden. Hierbei handelt es sich um ein älteres Protokoll und Vorgänger-Version des später diskutierten TCP Reno.

TCP Tahoe beginnt mit der „Slow Start“-Phase, hier ist das cwnd sehr klein und wird dann exponentiell bis zum ssthresh vergrößert. Anfangs ist der ssthresh typischerweise auf einen hohen Startwert gesetzt und wird anhand von Paketverlusten angepasst. Wenn das cwnd den Wert von ssthresh erreicht, kommen wir in die AIMD-Phase.

In der AIMD-Phase wächst das cwnd jetzt nur noch linear. Der ssthresh wird auf 50% seines vorherigen Wertes gesetzt

Die Fast-Retransmit-Phase wird beim Erkennen von 3 „Duplicate Acks“ getriggert und resettet cwnd auf seinen Initialwert. Nun beginnt wieder die „Slow Start“-Phase.

ssthresh: Slow Start Threshold

cwnd: Congestion Window

TCP Reno

Wie bereits erwähnt, handelt es sich bei TCP Reno um eine Weiterentwicklung von TCP Tahoe, es ist ebenfalls eine Technik zur „Congestion“ Vermeidung, allerdings mit einer schnelleren Reaktion auf Paketverlust.

Die „Slow Start“-Phase und AIMD-Phase sind gleich, es kommt eine „Fast Recovery“-Phase hinzu. Anstelle, dass nach dem Erhalt von 3 „Duplicate Acks“ wieder in die „Slow Start“-Phase gewechselt wird, wird nur das cwnd halbiert, denn der Erhalt dieser Acks deutet darauf hin, dass das Netzwerk nicht komplett überlastet ist. Falls kein ACK mehr ankommt, ist davon auszugehen, dass das Netzwerk überlastet ist, und jetzt erst wird die Fast-Retransmit-Phase wie bei TCP Tahoe ausgelöst.

TCP Vegas

Anders als die beiden vorherigen Varianten versucht TCP Vegas „Congestion“ zu erkennen, bevor es zu Paketverlusten kommt. Bei dieser Technik wird das cwnd anhand der Round-Trip Time (RTT) der Pakete gesteuert. Wenn also Pakete länger brauchen, um anzukommen, wird cwnd verkleinert und umgekehrt wieder vergrößert.

Protokolle

| | |
|--------------|----------------------|
| HTTP / HTTPS | 7: Application Layer |
| DNS | |
| DHCP | |
| NCP | |
| UDP | 4: Transport Layer |
| TCP | |
| IPv4 / IPv6 | 3: Network Layer |
| ICMP | |
| ARP | 2: Link Layer |
| RARP | |

Aufgabe 2

- a) `nmap -sn 192.168.178.0/24`
 - 1. `-sn` ist ein Ping-Scan
 - 2. Ergebnis: 256 IP addresses (18 hosts up) scanned in 3.76 seconds
- b) `nmap -O scanme.nmap.org`
 - 1. `-O` ist die OS-Erkennung
 - 2. Ergebnis:
 - i. Running: Linux 4.X|5.X
 - ii. OS CPE: `cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5`
 - iii. OS details: Linux 4.15 - 5.19
- c) `whois nmap.org`
 - 1. Ergebnis: Creation Date: 1999-01-18T05:00:00.OZ
- d) `nmap -Pn -T4 -p 1-1024 --open 192.168.178.0/24`
 - 1. `-Pn` Ping-Skip, wir führen kein Host-Discovery aus.
 - 2. `-T4` Timing-Template, schnell aber noch stabil.
 - 3. `-p` Hiermit können wir den Suchbereich auf einige Ports einschränken.
 - 4. `--open`, dient dazu nur die offenen Ports im Ergebnis anzuzeigen für bessere Lesbarkeit
 - 5. 192.168.178.0/24, Einschränkung auf die in meinem Netz tatsächlich vergebenen IP-Adressen.
- e) SYN-Scan kann, durch das senden von SYN-Paketen, schnell offene TCP-Ports erkennen.
- f) Die häufigsten Ports sind:
 - 1. 80/tcp für http
 - 2. 443/tcp für https
 - 3. 21/tcp für ftp
 - 4. 22/tcp für ssh
 - 5. 445/tcp für microsoft-ds

Aufgabe 3

Laut ChatGPT beruht DHCP auf dem älteren BOOTP-Protokoll, daher kann man diese Pakete mit dem Filter bootp in Wireshark finden.

Da mein PC neben einem Ethernet-Anschluss auch über integriertes WLAN verfügt, habe ich einfach das WLAN angeschaltet, was einen DHCP-Austausch getriggert hat. Andere Pakete wurden von Haushaltsgeräten, beim versuch sich mit der Fritzbox zu verbinden, getriggert.

| | | | | | | | |
|------|------------|---------------|-----------------|------|-----|---------------|----------------------------|
| 1128 | 42.739419 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover | - Transaction ID 0xa7eef2d |
| 1257 | 47.767137 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover | - Transaction ID 0xa7eef2d |
| 1258 | 47.767881 | 192.168.178.1 | 255.255.255.255 | DHCP | 590 | DHCP Offer | - Transaction ID 0xa7eef2d |
| 1378 | 52.798846 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover | - Transaction ID 0xa7eef2d |
| 1560 | 57.826266 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover | - Transaction ID 0xa7eef2d |
| 1561 | 57.827375 | 192.168.178.1 | 255.255.255.255 | DHCP | 590 | DHCP Offer | - Transaction ID 0xa7eef2d |
| 1729 | 62.861693 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover | - Transaction ID 0xa7eef2d |
| 1973 | 67.889196 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover | - Transaction ID 0xa7eef2d |
| 1974 | 67.890007 | 192.168.178.1 | 255.255.255.255 | DHCP | 590 | DHCP Offer | - Transaction ID 0xa7eef2d |
| 2602 | 89.910452 | 0.0.0.0 | 255.255.255.255 | DHCP | 342 | DHCP Discover | - Transaction ID 0xc846258 |
| 2603 | 89.916799 | 0.0.0.0 | 255.255.255.255 | DHCP | 352 | DHCP Request | - Transaction ID 0xc846258 |
| 5567 | 132.729638 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover | - Transaction ID 0xb3c48fb |

```

Frame 2603: 352 bytes on wire (2816 bits), 352 bytes captured (2816 bits) on interface
Ethernet II, Src: Intel_f4:07:c2 (0c:dd:24:f4:07:c2), 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 (Request)
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xc846258
  Seconds elapsed: 0
  Bootp flags: 0x0000 (Unicast)
    0... .. = Broadcast flag: Unicast
    .000 0000 0000 0000 = Reserved flags: 0x0000
  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: Intel_f4:07:c2 (0c:dd:24:f4:07:c2)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  Option: (53) DHCP Message Type (Request)
    Length: 1
    DHCP: Request (3)
  Option: (61) Client identifier
    Length: 7
    Hardware type: Ethernet (0x01)
    Client MAC address: Intel_f4:07:c2 (0c:dd:24:f4:07:c2)
  Option: (50) Requested IP Address (192.168.178.70)
    Length: 4
    Requested IP Address: 192.168.178.70
  Option: (54) DHCP Server Identifier (192.168.178.1)
    Length: 4
    DHCP Server Identifier: 192.168.178.1
  Option: (12) Host Name
    Length: 6
    Host Name: PC-Sam
  Option: (81) Client Fully Qualified Domain Name
    Length: 9
  Flags: 0x00
  A-RR result: 0
  PTR-RR result: 0
  Client name: PC-Sam
  Option: (60) Vendor class identifier
    Length: 8
    Vendor class identifier: MSFT 5.0
  Option: (55) Parameter Request List
    Length: 14
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (15) Domain Name
    Parameter Request List Item: (31) Perform Router Discover
    Parameter Request List Item: (33) Static Route
    Parameter Request List Item: (43) Vendor-Specific Information
    Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
    Parameter Request List Item: (46) NetBIOS over TCP/IP Node Type
    Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
    Parameter Request List Item: (119) Domain Search
    Parameter Request List Item: (121) Classless Static Route
    Parameter Request List Item: (249) Private/Classless Static Route (Microsoft)
    Parameter Request List Item: (252) Private/Proxy autodiscovery
  Option: (255) End
    Option End: 255

```

Es handelt sich um eine Anfrage

DHCP-Server hier in der Fritzbox integriert

Name meines PCs

Angefragte Parameter

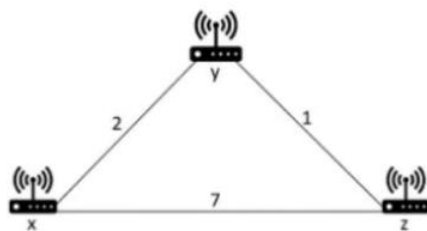
Aufgabe 4

Algorithmus:

Initialisierung: Knoten = [A, B, C, D]; Kanten = [(A, B, 3), (A, C, 23), (B, C, 2), (C, D, 5)]

- Für jeden Knoten
 - Neue Tabelle mit Dimensionen |Knoten| \times |Knoten| alle Felder auf NULL
 - Alle Spalten die keine direkten Nachbarn sind auf GREY
 - Zeile zu aktuellem Knoten auf GREY
 - Für jede Kante
 - Wenn aktueller Knoten in Kante, dann Gewicht zu Nachbarknoten eintragen
- Wiederhole bis keine NULL-Felder mehr da sind + 1
 - Für jeden Knoten // aktuelle Tabelle aktTab
 - Für jede Zeile
 - Suche und markiere Minimum
 - Für jedes NULL-Feld // Feld=(aktZu,aktVia)
 - Ist aktVia über seine entsprechende aktTab(x,aktVia) erreichbar?
 - Aktualisiere Wert aktTab(zuVia,aktVia)+ viaTab(aktZu,aktZu)

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 | NULL |
| Zu z | | NULL | 7 |

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

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

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

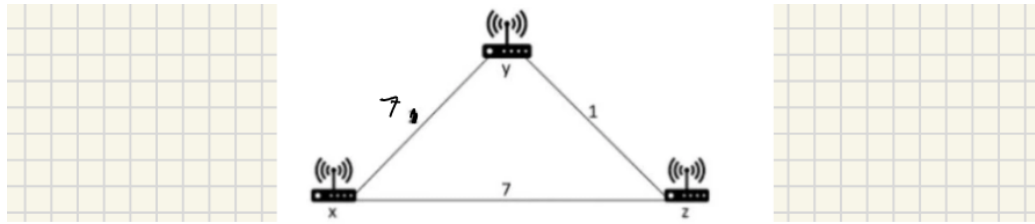
| Von y | Via x | Via y | Via z |
|-------|-------|-------|-------|
| Zu x | 2 | | 8 |
| Zu y | | | |
| Zu z | 9 | | 7 |

| 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 | | 7 |

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



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 | NULL |
| Zu z | | NULL | 7 |

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

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

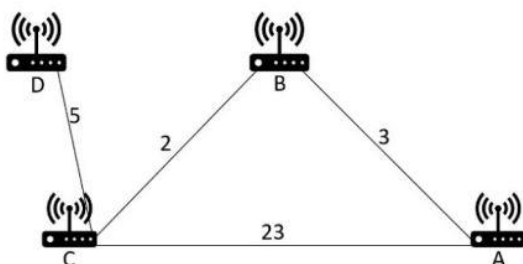
| Von z | Via x | Via y | Via z |
|-------|-------|-------|-------|
| Zu x | 7 | 8 | |
| Zu y | 14 | 7 | |
| Zu z | | | |

| 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 | | 7 |

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

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 Sofort bemerken, dass D wegfällt, da er ein direkter Nachbar ist, allerdings würden mit dem Distanzvektor Verfahren die anderen Router dies nicht mitbekommen, da diese immer nur Informationen über ihre direkten Nachbarn erhalten.