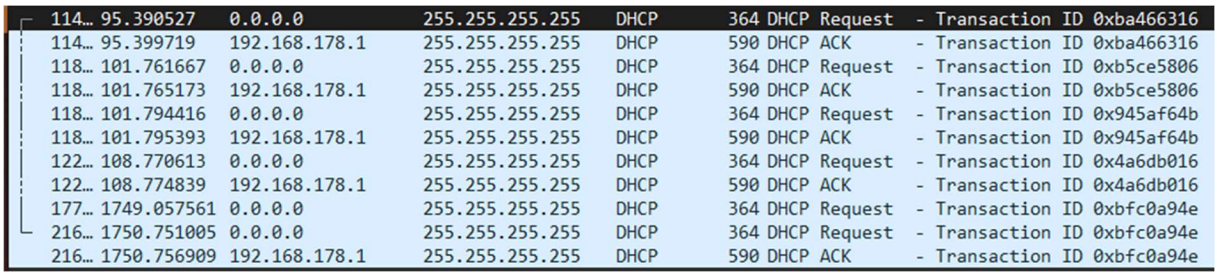
Aufgabe 1:

Das Sliding-Window-Protokoll wird verwendet, um die Speicherfähigkeit des Mediums maximal zu nutzen. Werden Pakete über das Internet gesendet, so werden diese quittiert, wenn sie ankommen. Am Beispiel der Voyager wurde gezeigt, dass so die Speicherfähigkeit des Mediums nicht ausgenutzt werden kann, wenn das nächste Paket erst nach der Quittierung des vorherigen Paketes gesendet wird. Daher wurde das Sliding-Window-Protokoll eingeführt. Dieses gibt an, wie viele unquittierte Pakete am Stück gesendet werden können. Diese Anzahl wird durch einen Parameter n angegeben. Somit kann der Sender maximal n Nachrichten unquittiert senden. Die n+1 Nachricht wird so lange vom Sender blockiert, bis der Empfänger eine Nachricht quittiert. Wird die 2. Nachricht des Fensters quittiert, jedoch nicht die erste, so fordert der Empfänger das erste Paket vom Sender erneut an, dieser kann dann per „Go-Back-N“ oder per selektiver Wiederholung Pakete erneut senden. Bei letzterem handelt es sich um eine komplexere Methode. Um ein passendes n zu finden, haben sich drei Methoden etabliert: TCP Tahoe, TCP Reno und TCP Vegas.
Bei TCP Tahoe wird die Fenstergröße bei jedem akzeptierten Paket linear (um 1) erhöht. Zunächst startet es bei 1 und steigt dann immer weiter an, bis das Senden eines Paketes fehlschlägt. Dies erkennt TCP indem es annimmt, dass bei 3 duplizierten ACKs vom gleichen Paket ein Paket verloren gegangen ist. Passiert dies, so wird die Fenstergröße wieder auf 1 zurückgesetzt und der Ablauf beginnt von neuem (zudem wird das Paket, welches verloren gegangen ist, erneut gesendet).
Bei TCP Reno wird die Fenstergröße zunächst, wie bei TCP Tahoe linear erhöht, bis es zum ersten Mal fehlschlägt. Dann wird die Fenstergröße halbiert und der Ablauf beginnt von neuem, zudem werden die fehlenden Pakete erneut gesendet.
TCP Vegas ist die am meisten fortgeschrittene Methode zur Flusskontrolle und Stauvermeidung. Hierbei wird die Round Trip Time für jedes Segment ermittelt (Abstand Send und dessen ACK). Aus der RTT kann dann die Fenstergröße ermittelt werden.

Abb. 1: TCP Vegas

Protokollbezeichnung	Ebene	Begründung
UDP / TCP	4	Datenübertragung zwischen Endgeräten
IP	3	Routing durch das Internet, Host to Host
ICMP	3	Generiert dann Fehlermeldung, wenn IP bspw. nicht den Host erreichen kann (bswp. TTL abgelaufen)
ARP/ARPR	2	Kommunikation in einem Netzwerk geht über Ebene 2, ARP ermittelt MAC Adressen und dient somit zur Kommunikation in einem Netzwerk
DNS	7	Übersetzung von IP-Adressen in Domainnamen, benötigt für Webbrowsing etc. -> Anwendungsschicht
DHCP	7	Sendet Konfigurationen wie IP-Adressen, DNS-Server etc. an Clienten (bspw. Handy) -> Anwendungsschicht
FDMA, TDMA, CDMA, CSMA/CD	2	Bestimmen, wie Geräte auf Übertragungsmedium zugreifen können bzw wie die Kommunikation zwischen 2 Knotenpunkten abläuft -> Data Link Layer

Aufgabe 2:



```
114… 95.390527   0.0.0.0         255.255.255.255   DHCP    364 DHCP Request  - Transaction ID 0xba466316
114… 95.399719   192.168.178.1   255.255.255.255   DHCP    590 DHCP ACK      - Transaction ID 0xba466316
118… 101.761667  0.0.0.0         255.255.255.255   DHCP    364 DHCP Request  - Transaction ID 0xb5ce5806
118… 101.765173  192.168.178.1   255.255.255.255   DHCP    590 DHCP ACK      - Transaction ID 0xb5ce5806
118… 101.794416  0.0.0.0         255.255.255.255   DHCP    364 DHCP Request  - Transaction ID 0x945af64b
118… 101.795393  192.168.178.1   255.255.255.255   DHCP    590 DHCP ACK      - Transaction ID 0x945af64b
122… 108.770613  0.0.0.0         255.255.255.255   DHCP    364 DHCP Request  - Transaction ID 0x4a6db016
122… 108.774839  192.168.178.1   255.255.255.255   DHCP    590 DHCP ACK      - Transaction ID 0x4a6db016
177… 1749.057561 0.0.0.0         255.255.255.255   DHCP    364 DHCP Request  - Transaction ID 0xbfc0a94e
216… 1750.751005 0.0.0.0         255.255.255.255   DHCP    364 DHCP Request  - Transaction ID 0xbfc0a94e
216… 1750.756909 192.168.178.1   255.255.255.255   DHCP    590 DHCP ACK      - Transaction ID 0xbfc0a94e
```

Damit DHCP-Pakete gesendet werden, muss sich beim Router registriert werden, damit einem Gerät eine IP-Adresse und weitere Daten zugesendet werden. Damit also DHCP-Pakete gesendet werden, kann das Internet aus und wieder eingeschaltet werden, dann wird eine Konfigurationsdatei angefordert und eine IP-Adresse wird zugewiesen.

```
› Frame 11474: 364 bytes on wire (2912 bits), 364 bytes captured (2912 bits) on interface \Device\NPF_{DABA4F92-5DAC-4457-ACCF-367E77D81BAA},
› Ethernet II, Src: Intel_58:1f:3d (98:43:fa:58:1f:3d), 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
    Source Port: 68
    Destination Port: 67
    Length: 330
    Checksum: 0xd3e0 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 54]
  › [Timestamps]
    UDP payload (322 bytes)
˅ Dynamic Host Configuration Protocol (Request)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xba466316
    Seconds elapsed: 0
  › Bootp flags: 0x8000, Broadcast flag (Broadcast)
    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_58:1f:3d (98:43:fa:58:1f:3d)
    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 (192.168.178.191)
  › Option: (12) Host Name
  › Option: (81) Client Fully Qualified Domain Name
  › Option: (60) Vendor class identifier
  › Option: (55) Parameter Request List
  › Option: (255) End
```

Aus den Package Details lässt sich erkennen, dass beim Request eine bestimmte IP-Adresse angefordert wird (Die IP-Adressen, die dieser Rechner auch sonst hat). Die src-IP ist die 0.0.0.0, dies bedeutet, dass keine spezielle IP zugewiesen ist. Mit der Broadcastadresse 255.255.255.255 wird der DHCP-Server angesprochen und angefordert, sodass eine neue IP-Adresse zugewiesen wird.

```
› Frame 11475: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits) on interface \Device\NPF_{DABA4F92-5DAC-4457-ACCF-367E77D81BAA},
› Ethernet II, Src: AVM_14:b2:03 (9c:c7:a6:14:b2:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Internet Protocol Version 4, Src: 192.168.178.1, Dst: 255.255.255.255
˅ User Datagram Protocol, Src Port: 67, Dst Port: 68
    Source Port: 67
    Destination Port: 68
    Length: 556
    Checksum: 0xf06f [unverified]
    [Checksum Status: Unverified]
    [Stream index: 55]
  › [Timestamps]
    UDP payload (548 bytes)
˅ Dynamic Host Configuration Protocol (ACK)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xba466316
    Seconds elapsed: 0
  › Bootp flags: 0x8000, Broadcast flag (Broadcast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.178.191
    Next server IP address: 192.168.178.1
    Relay agent IP address: 0.0.0.0
    Client MAC address: Intel_58:1f:3d (98:43:fa:58:1f:3d)
    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 [truncated]: 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

Als Antwort erhält man dann die IP-Adresse und weitere wichtige Informationen, wie bspw. die Router-IP, die Renewal Time meiner IP, die Subnetzmaske und den DNS.

Aufgabe 3:

a: nmap -sn 192.168.178.0/24
Es gibt 256 IP-Adressen in meinem Netz mit momentan 9 Hosts.
-ns gibt an, dass nur nach IP-Adressen gesucht werden soll und nicht zusätzlich alle Ports untersucht werden sollen

b: sudo -O nmap scanme.nmap.org
-O gibt an, dass das OS untersucht werden soll
Antwort: Linux 4.15 - 5.19

c: whois nmap.org
gibt sehr viele Informationen über nmap.org aus, unter anderem den Tag der Registrierung
Antwort: 18. Januar 1999.

d: nmap -T4 192.168.178.1-254
-T4 gibt an, dass agressiver gesucht werden soll (dadurch erhöht sich die Netzwerklast), aber wesentlich schneller als nmap 192.168.178.1-254

e: nmap -sS 192.168.178.1-254
-sS für SYN Scan
Der SYN-Scan kann dazu genutzt werden, um herauszufinden, welche TCP-Ports auf einem Zielsystem geöffnet sind. Dabei wird keine TCP-Verbindung aufgebaut, was es zu einem schnellen Verfahren macht. Es wird nur ein SYN Paket gesendet und auf das SYN/ACK gewartet, danach wird ein RST gesendet um die Verbindung abzubrechen.

f: 80/tcp http
443/tcp https

Aufgabe 4:

Betrachtung Router A:

Initialisierung: Jeder Router trägt in eine Tabelle ein, wie die Entfernung zu den Nachbarn ist

Aktualisierung: Von jedem Nachbarn (B und C) erhält A die Matrizen, diese Information verarbeitet A und fügt diese in die eigene Tabelle ein. Bspw. erkennt A, dass zu B der kürzeste Weg der direkte Weg ist, daher wird A -> B als minimal eingetragen. Die gelben Flächen zeigen, dass hier noch nicht sicher ist, ob es sich momentan um den minimalen Weg handelt.

Aktualisierung: Das Gleiche aus obigem Schritt gilt in diesem Schritt.

Aktualisierung und endgültiges Ergebnis: A hat zu jedem Knoten den kürzesten Weg gefunden, der Vorgang terminiert.

Algorithmus:

1. Bilde eine Matrize und vermerke, wie weit die Distanz zu deinen Nachbarn ist, welche du in einem Hop erreichen kannst.
2. Teile diese Information mit den anderen Routern. Gleichzeitig erhältst du Routinginformationen der anderen Router.
3. Beziehe diese Information der anderen Router in deine Matrix mit ein, wenn sich keine Änderungen mehr ergeben, terminiere.

Der Unterschied zum Link-State-Verfahren ist, dass Link-State die komplette Netzwerktopologie kennt, also von jedem Knoten die beste Route über diverse andere Knoten kennt. Die komplette Route mit ihren Knoten ist erkennbar. Bei distanzvektorbasiertem Routing ist dies nicht der Fall, hier werden nur die direkten Nachbarn in der Distanzmatrix eingezeichnet, die Route ist für den momentanen Knoten nicht bekannt, er weiß ausschließlich, zu welchem Nachbarn er das Paket schicken muss, damit es den schnellsten Weg nimmt.