Rechnernetze und verteile Systeme: Block 7

Aufgabe 5:

a)
15892

b)
897,982 bytes

c)
00:0c:29:b6:b5:48
00:50:56:c0:00:08
00:50:56:f3:f2:f6
ff:ff:ff:ff:ff:ff
33:33:00:01:00:03
01:00:5e:00:00:fc

d)
53 insgesamt
52 ipv4
1 ipv6

e)

f)
Internet Protocol Version 4: 99,68%
Internet Protocol Version 6: 00,01%

g)
Transmission Control Protocol: 98,23%

h)
SSL
HTTP

i)
DNS
NetBIOS Name Service
Dropbox LAN sync Discovery Protocol

Link-local Multicast Name Resolution (LLMNR)

j)
Internet Protocol Version 4
Internet Protocol Version 6
Adress Resolution Protocol

k)
Ethernet

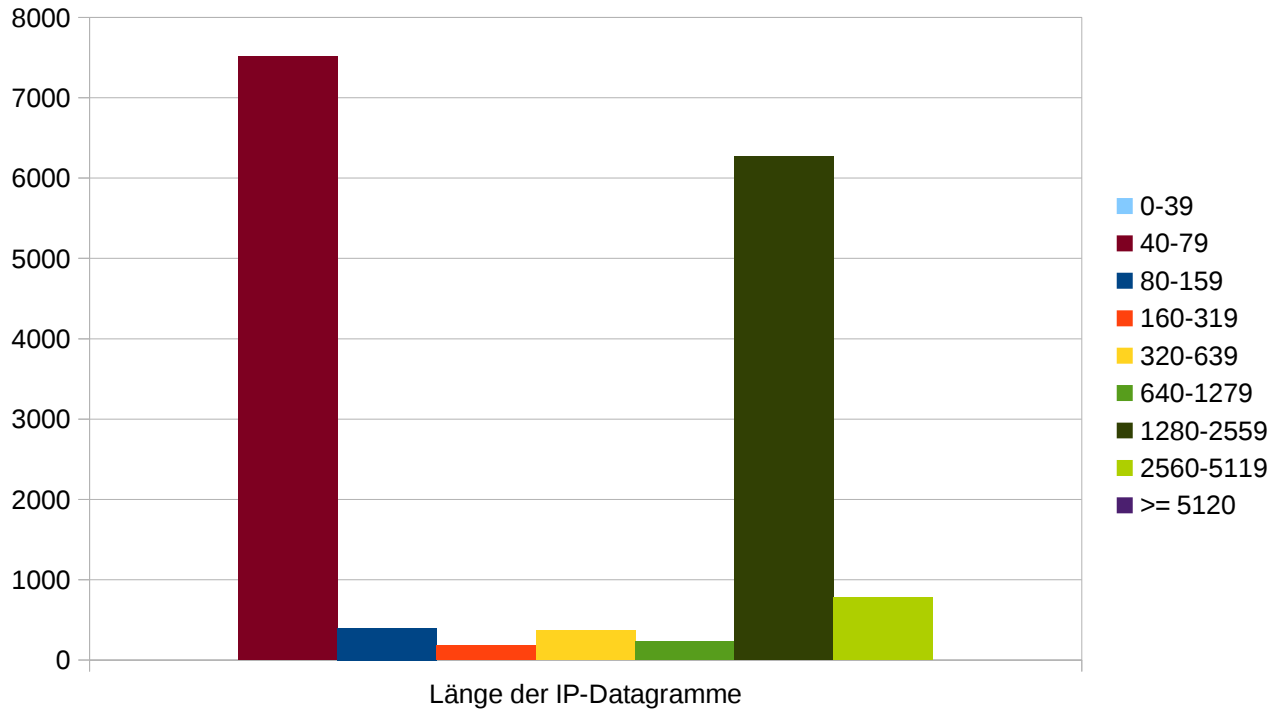
l)
195

m)
TTL > 200 : 0
TTL = 128 : 15833
TTL = 64 : 6

Interpretation: Vermutlich hat der Großteil der Pakete eine TTL von 128, da es sich hier um einen standardisierten Wert handelt. Die TTL ist somit groß genug, damit ein IP-Paket in der Regel sein Ziel erreicht.

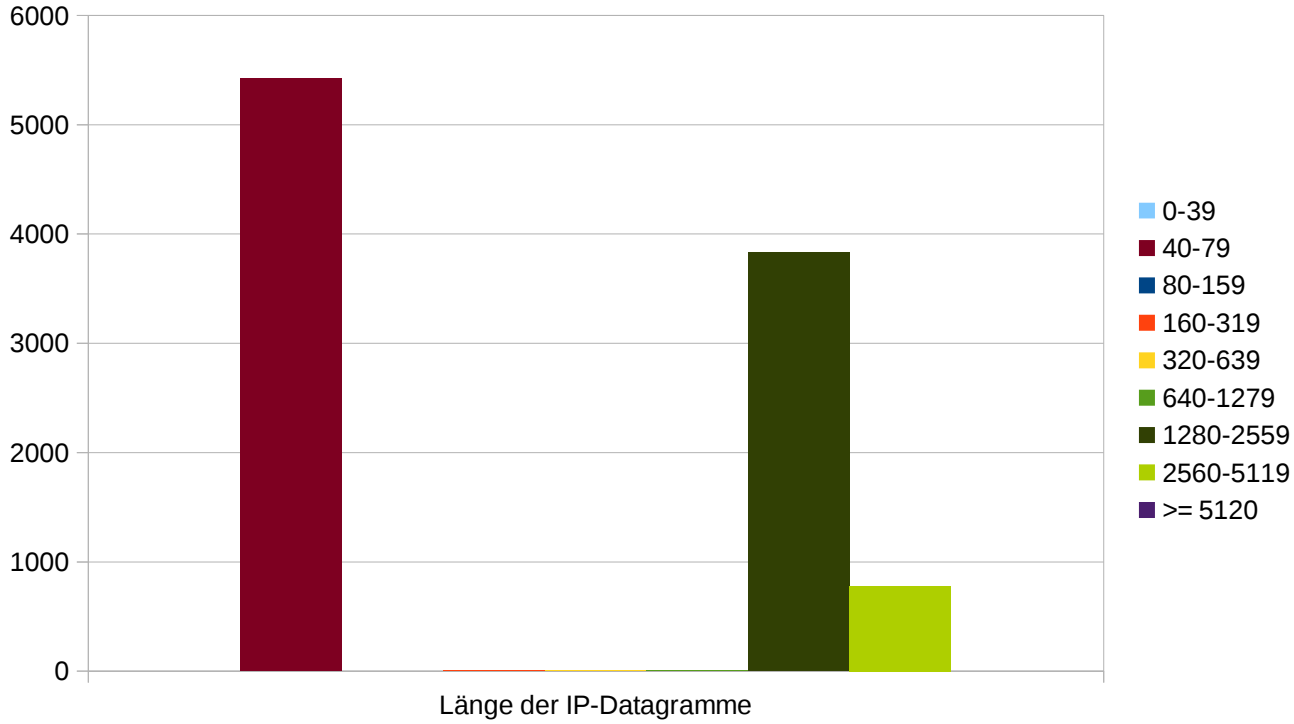
n)
Größe Ethernet-Header:207-193 = 14
Größe IP-Header: 20 bytes
Größe IP-Datagram: 193 gesamt / 173 Nutzteil
Größe TCP-Header: 20 bytes
Größe TCP-Segment: 173 - 20 = 153

o)



Auffällig ist, dass besonders häufig IP-Datagramme der Längen 40-79 und 1280-2559 versendet werden. Eine Ursache hierfür könnte ein häufig im Verlauf des Trace verwendetes Protokoll sein, das oftmals IP-Datagramme gleicher Länge versendet.

p)
Zwischen 81.166.122.238 und 172.16.254.128 werden die meisten Bytes ausgetauscht. Insegesamt 10012519 bytes in 10124 Pakteten.



Auch hier ist auffällig, dass häufig Datagramme der Längen 40-79 und 1280-2559 versendet werden. Auch hier ist es denkbar, dass ein bestimmtes Protokoll, das eine bestimmte Paketgröße besonders häufig verwendet, zwischen den beiden IP-Adressen Anwendung findet. Auch ein Austausch größerer Datenmengen, die in Paketen fester Größe versendet werden, ist denkbar.

q)
81.166.122.238 und 172.16.254.128
10124 packets

r)
ja mehrere. u.a.: 216.58.208.238 mit 172.16.254.128
über TLSv1.2

s)
Ja es wurde Google-Chrome verwendet, da User-agent in trace 621:
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.89 Safari/537.36\r\n

Aufgabe 6

a)
Das DHCP Protokoll ist fdafür verantwortlich, dass der Rechner eine IP-Adresse bekommt.
Es finden sich zwei Pakete des DHCP Protokolls im Trace:
ein DHCP Request- und ein DHCP ACK-Paket.
Im Request Paket ist als Source-Adresse 0.0.0.0 eingetragen, da der Rechner noch keine IP-Adresse zugewiesen bekommen hat.
Als Destination-Adresse ist 255.255.255.255 eingetragen, um im lokalen Netz einen Broadcast durchzuführen.
Im DHCP-ACK-Paket ist als Source-Adresse 192.168.1.1 und als Destination-Adresse 192.168.1.109 eingetragen.
Der Rechner möchte die IP-Adresse 192.168.1.109 erhalten, da diese im Request-Paket als Requested Ip Adress" steht.
Dieser Wunsch wird erfüllt, da als Zieladresse des ACK-Pakets eben diese Adresse eingetragen ist.
Die Adresse ist 12 Stunden gültig, da dies der in "IP Address Lease Time" eingestellte Wert ist.
Der in "Domain Name Server" Feld des Ack-Pakets zuständige DNS-Server ist 192.168.1.1

b)
Die Pakete 1 und 4-7 sind Pakete des ARP Protokolls und dienen der Auflösung von IP-Adressen in MAC-Adressen im lokalen Netz.
Dabei sind die Pakte 1,4 und 6 Anfragen des betrachteten Rechners, der bestimmte Adressen auflösen möchte. In den Paketen 5 und 7 beantwortet er zwei eingehende Anfragen mit je einer MAC-Adresse.

c)
Die Anfrage per DNS findet in Paket 8 des Trace statt und enthält den Hostnamen "www.tkn.tu-berlin.de"
Die Antwort des DNS-Servers befindet sich in Paket 9 und enthält die Antwort:
www.tkn.tu-berlin.de: type CNAME, class IN, cname ace-hauptblock4.tubit.tu-berlin.de
ace-hauptblock4.tubit.tu-berlin.de: type A, class IN, addr 130.149.7.204

d)
In Paket 13 findet die Anfrage statt:
Es wird die URL: "www.tkn.tu-berlin.de" angefragt.
Der Eintrag "User-Agent: Lynx/2.8.8dev.15 libwww-FM/2.14 SSL-MM/1.4.1 GNUTLS/2.12.20\r\n" lässt auf den Web-Browser Lynx schließen.

Die Antwort findet sich in Paktet 19:
Es antwortet die Apache Server-Software.
Es handelt sich um keine persistente bzw. Keep-Alive Verbindung, da in diese Paket der Eintrag "Connection: close" steht und nicht "Connection: keep-alive".

e)
In der zweiten Verbindung wird "hyperion.tkn.tu-berlin.de" aufgelöst zu der Adresse 130.149.49.153
Die Verbindung über TCP wird beim Client auf dem Port 39506 eingerichtet und beim Server auf dem Port 22.

Dies entspricht dem Secure Shell (SSH) Protokoll. Im anschließenden Datenaustausch lassen sich keine sinnvollen Daten erkennen, da eine verschlüsselte Verbindung eingerichtet wurde.

f)
Bereits in diesem Trace wurde ersichtlich, dass wir die Nutzdaten, die im Rahmen des HTTP-Protokolls übertragen wurden, unverschlüsselt einsehen konnten.
Auch bei anderen Protokollen wie FTP oder SMTP ist es (sofern diese auf unverschlüsselten Verbindungen basieren) ein Risiko, dass jede Person, die Zugriff auf das Netzwerk hat, auch auf sensible Daten im Rahmen dieser Kommunikation zugreifen kann, da diese unverschlüsselt übertragen werden.