

Sprawozdanie, W6

Uczestnicy zespołu: Tymon Zadara, Kinga Konieczna, Jan Czechowski

Zadanie 0 – Przygotowanie środowiska

Ćwiczenia 1 – 3 należy wykonywać w środowisku zwirtualizowanym. W tym celu należy pobrać VirtualBox <https://www.virtualbox.org/> a następnie utworzyć dwie maszyny wirtualne (VM). Przy tworzeniu VM warto użyć gotowego dysku z już zainstalowanym systemem operacyjnym (Ubuntu) dostępnego na stronie: <https://releases.ubuntu.com/focal/> (20.04.6, gdzie Desktop – z GUI (prościej), lub Server – bez GUI. Należy również skonfigurować sieć maszyny wirtualnej tak aby działała w trybie Bridge, ustawić odpowiedni interfejs (odpowiednio: Ethernet, PCI itp.) oraz umożliwić nasłuch/odbiór przez inne maszyny. Prawidłowo skonfigurowane maszyny powinny być widoczne z poziomu innych maszyn w lab. Prawidłową wstępną weryfikację można sprawdzić za pomocą polecenia *ping*

Obydwie maszyny muszą zarówno mieć kontakt ze sobą nawzajem, oraz z siecią Internet (np. ping 8.8.8.8)

Dwa tryby networkingu między maszynami:

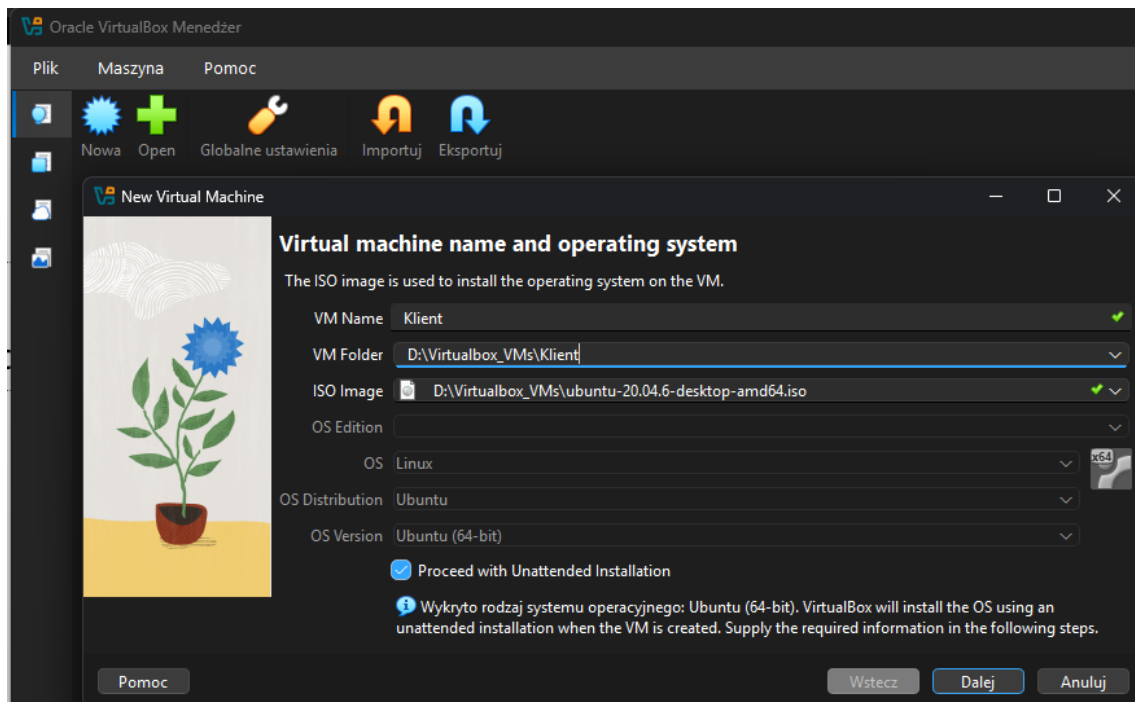
Bridge lub NAT (Network address translation) - Tutorial

<https://www.youtube.com/watch?v=vReAkOq-59I>

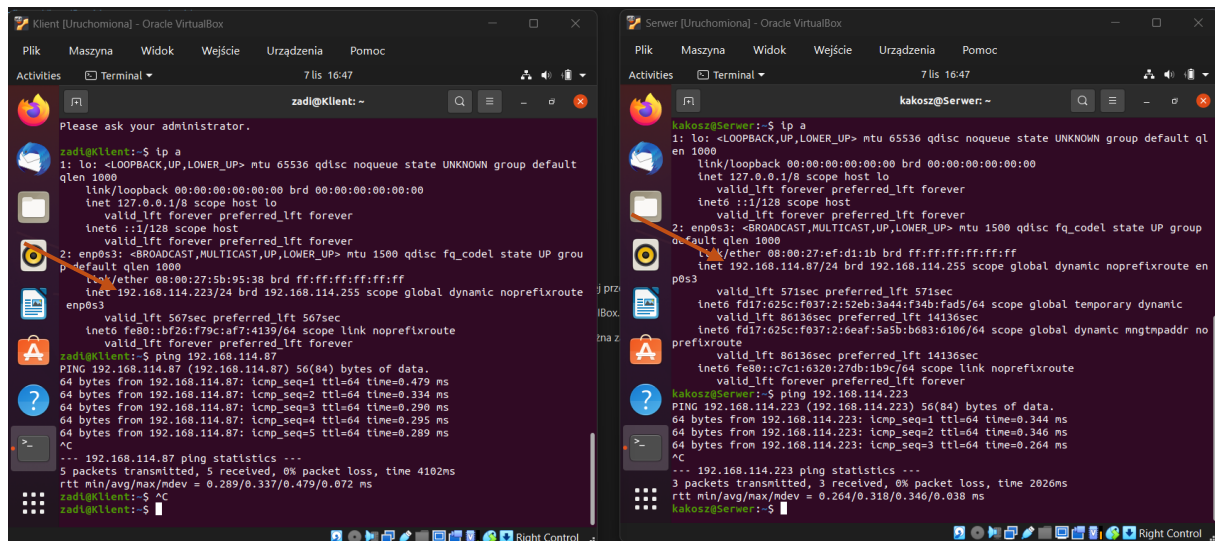
VirtualBox:

<https://www.virtualbox.org/wiki/Downloads>

Nowa maszyna wirtualna z obrazu ISO („Nowa” ->)



Zrzuty ekranu z ping między maszynami (2 zrzuty ekranu):



Na zrzucie ekranu przedstawiona jest adresacja IP maszyn (pomarańczowymi strzałkami) oraz ping między maszynami. Po lewej stronie znajduje się adresacja maszyny „Klient” (zwanej w dalszej części zadania jako „Klient”) oraz ping maszyny „Serwer” (zwanej w dalszej części zadania jako „Serwer”). Po prawej stronie znajduje się adresacja Serwera oraz ping Klienta. Proces pingowania maszyn odbył się pomyślnie w obie strony.

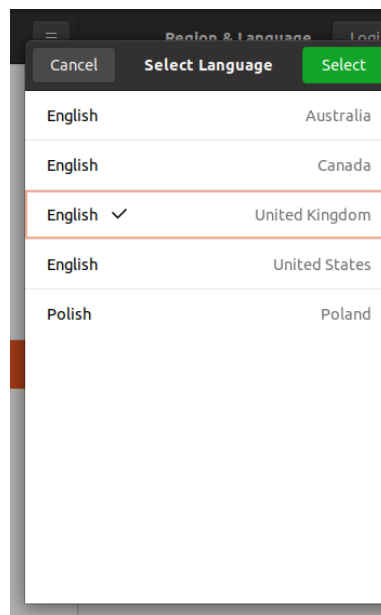
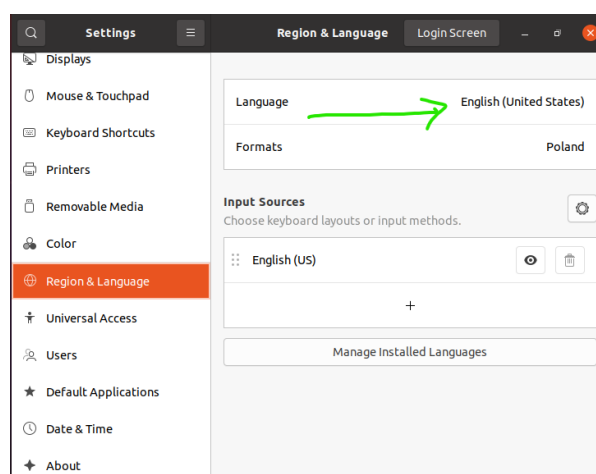
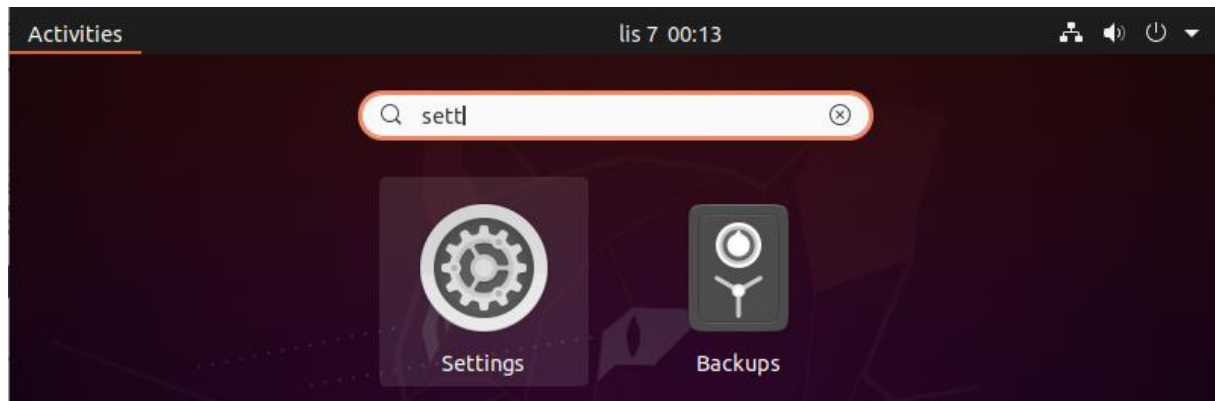
Adresacja IP maszyn (ipa / ifconfig na obydwu maszynach)

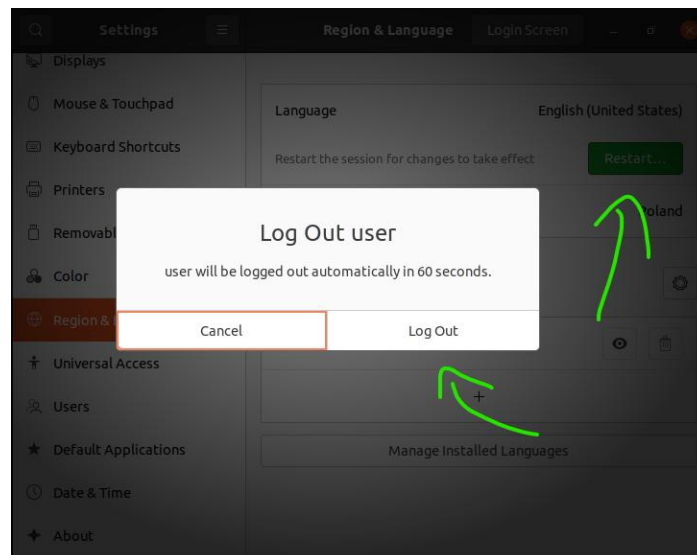
Adresacja IP maszyn znajduje się na zrzucie ekranu powyżej.

Adres IP Klienta – 192.168.114.223/24

Adres IP Serwera – 192.168.114.87/24

Jeśli terminal się nie otwiera (workaround):





Ćwiczenie 1 – Analiza ruchu sieciowego – TCP (Wireshark, Iperf)

1. Obie strony wykonują badanie maksymalnej przepływności łącza
 - Pobierają narzędzie iperf komendą: `apt-get install iperf3`
 - Uruchamiają pomiar
 - Jedna strona wywołuje komendę umożliwiającą odbieranie i analizę przychodzących pakietów: `iperf3 -s`
 - Druga strona wywołuje komendę pomiarową, która uruchamia całą procedurę testową `iperf3 -c <adresIP> -t 10`

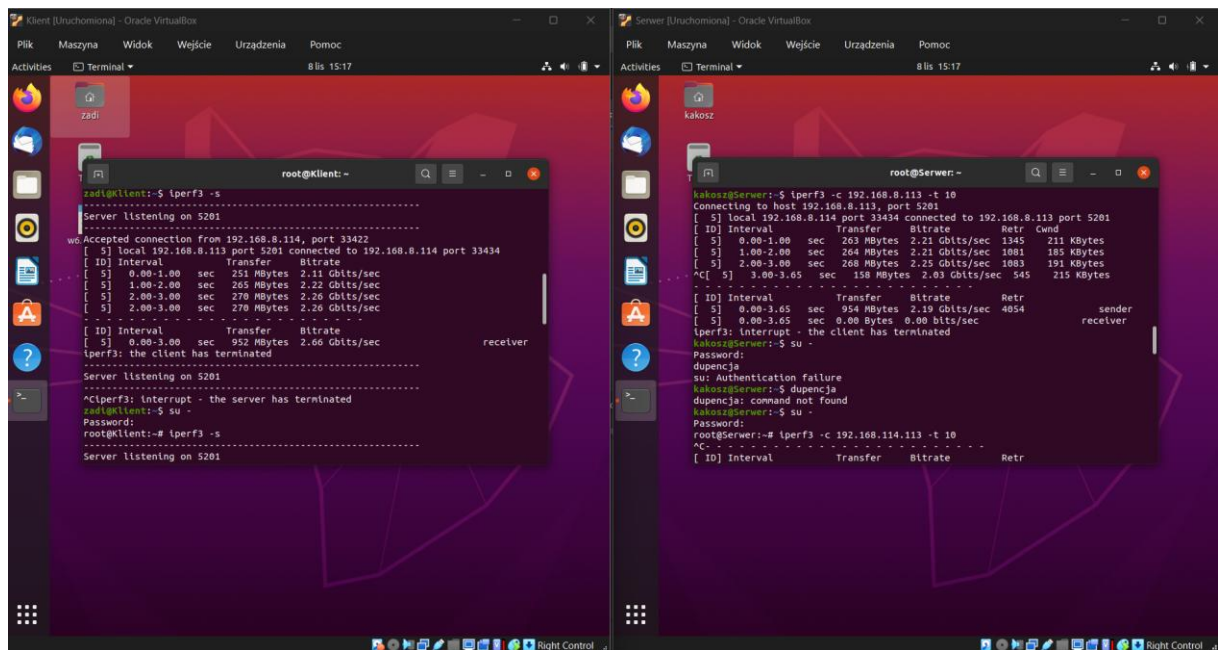
Zrzut ekranu z wyniku testu przepływności łącza iperf:

W procesie realizacji tego ćwiczenia (tylko część sprawdzania iperf3) zostało zmienione WiFi, do którego podłączone jest urządzenie, dlatego występują różnice w adresach IP w stosunku do poprzedniego zadania.

Adres IP Klienta – 192.168.8.113

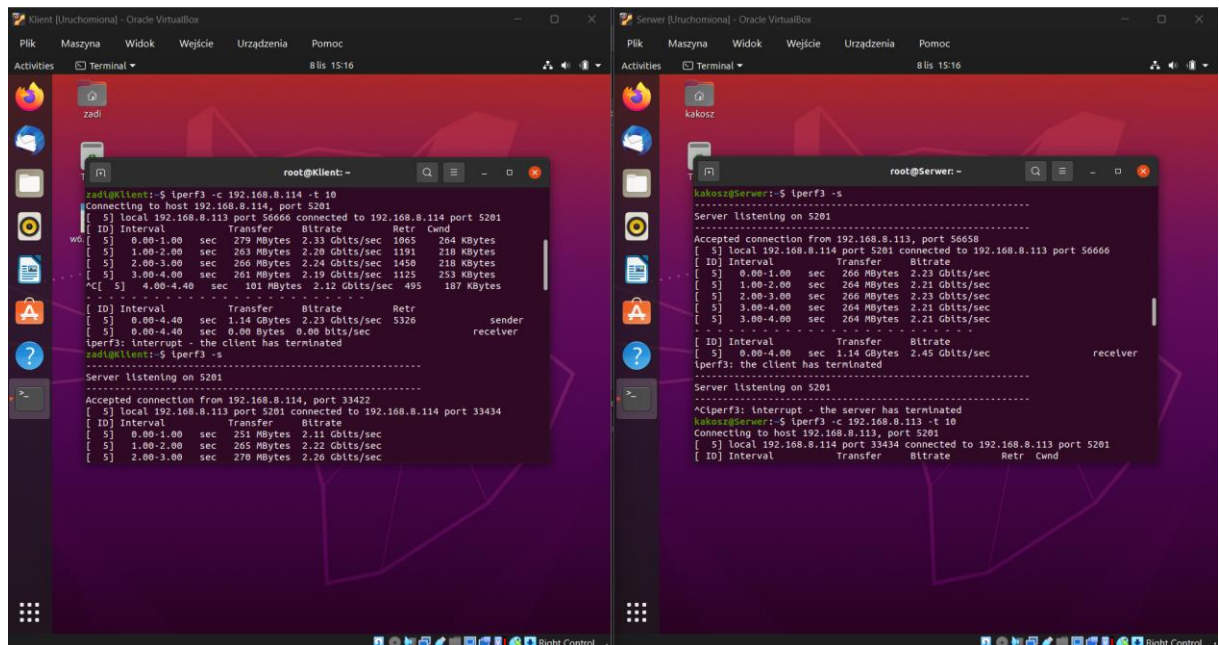
Adres IP Serwera – 192.168.8.114

Poprawność połączenia została sprawdzona za pomocą komendy ping na nowych adresach. Procedura pingowania przebiegła pomyślnie w obie strony.



W tej procedurze Klient odbiera i analizuje pakiety przychodzące od Serwera. Po lewej stronie znajduje się terminal Klienta, na którym widać, że przyjął on połączenie od Serwera i odbiera od niego transfer. Po prawej stronie znajduje się terminal Serwera, na którym widać, że Serwer połączył się z Klientem i rozpoczął wysyłanie danych. Połączenie zostało nawiązane pomiędzy obiema stronami.

Zrzut ekranu z wyniku testu przepływności łącza iperf w drugą stronę:



W tej procedurze Serwer odbiera i analizuje pakiety przychodzące od Klienta. Po prawej stronie znajduje się terminal Serwera, na którym widać, że przyjął on połączenie od Klienta i odbiera od niego transfer. Po lewej stronie znajduje się terminal Klienta, na którym widać, że Klient połączył się z Serwerem i rozpoczął wysyłanie danych. Połączenie zostało nawiązane pomiędzy obiema stronami.

Czy wyniki różnią się? Dlaczego?

Wyniki nie różnią się od siebie. Jest to spowodowane tym, że zarówno Klient jak i Serwer (pomimo ich mylącego nazewnictwa) posiadają takie same właściwości i prawa. Urządzenie nasłuchujące wciela się w rolę serwera (nie mylić z nazwą maszyny Serwer (pisane z dużej litery)) natomiast urządzenie nadające w rolę klienta. Powoduje to, że obie wirtualne maszyny mogą zarówno nadawać, jak i odbierać transmisję.

2. Jedna strona (klient iperf) instaluje Wireshark

Może być konieczne zrekonfigurowanie wireshark żeby non-root użytkownik mógł nasłuchiwać ruchu sieciowego:

```
su -  
dpkg-reconfigure wireshark-common  
usermod -a -G wireshark <nazwa_uzytkownika>  
exit
```

Na jednej maszynie uruchamiamy serwer, na drugiej maszynie uruchamiamy Wireshark (nasłuchiwanie – należy ustawić odpowiednie filtry, protokół TCP oraz adres ip serwera) oraz iperf klient.

Z jakich etapów składa się proces „three-way-handshake”? Wskaż na zrzucie ekranu wszystkie trzy w Wireshark, i opisz każdy z nich. Wskaż flagi i numery sekwencyjne i ACK

Zrzut ekranu Wireshark z faz nawiązywania połączenia three-way-handshake

Realizacja tego zadania przebiegła na tym samym WiFi, co zadanie 0.

Poniżej dla przypomnienia znajduje się adresacja urządzeń:

Adres ip Klienta: 192.168.114.87/24

Adres ip Serwera: 192.168.114.223/24

„Three-way-handshake” składa się z następujących trzech etapów pojawiających się w kolejności określonej poniżej:

1. [SYN] (czerwona strzałka):

flagi (zawarte pomiędzy nawiasami kwadratowymi []) = SYN;

numer sekwencyjny = 0 (Seq = 0);

numer ACK = brak;

Opis = wysyłany od klienta do serwera, klient inicjuje połączenie. Wysyła pakiet z flagą SYN (*Synchronize*) aby pokazać, że chce się połączyć z serwerem i

rozpocząć komunikację. Jego numer sekwencyjny to 0, ponieważ jest to jego pierwsza wiadomość do serwera.

2. [SYN, ACK] (fioletowa strzałka):

flagi = SYN, ACK;

numer sekwencyjny = 0 (Seq=0);

numer ACK = 1 (Ack = 1);

Opis = Wysyłany z serwera do klienta, serwer odpowiada, że przyjął od klienta jego pakiet SYN. Jego numer Ack to numer sekwencyjny pakietu SYN klienta, zwiększony o 1. Wysyła jednocześnie pakiet z flagą SYN do klienta w celu zainicjowania połączenia powrotnego od serwera do klienta. Numer sekwencyjny jest równy 0, ponieważ to pierwsza wiadomość serwera do klienta.

3. [ACK] (zielona strzałka):

flagi = ACK;

numer sekwencyjny = 1 (Seq = 1);

numer ACK = 1 (Ack = 1);

Opis = wysyłany z klienta do serwera, klient informuje, że otrzymał pakiet SYN od serwera i połączenie pomiędzy klientem a serwerem zostaje ustanowione.

Numer sekwencyjny oraz numer Ack zwiększa się o 1.

Ważne! Serwer oraz klient osobno śledzą swoje numery sekwencyjne oraz numery Ack. Zwiększają się one z następującą zasadą:

- Numer sekwencyjny zwiększa się o 1 w stosunku do numeru sekwencyjnego poprzedniej wiadomości tego samego nadawcy
- Numer Ack zwiększa się o 1 w stosunku do numeru sekwencyjnego odebranej wiadomości.

Klient [Uruchomiona] - Oracle VM VirtualBox

Plik Maszyna Widok Wejście Urządzenia Pomoc

Activities Wireshark 7 lis 17:26 Capturing from enp0s3

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 192.168.114.87

No.	Time	Source	Destination	Protocol	Length	Info
11	2.863581162	192.168.114.87	192.168.114.223	TCP	74	47520 → 5201 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
12	2.863616120	192.168.114.223	192.168.114.87	TCP	74	5201 → 47520 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460
13	2.864122431	192.168.114.87	192.168.114.223	TCP	66	47520 → 5201 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=31487025
14	2.864122551	192.168.114.87	192.168.114.223	TCP	103	47520 → 5201 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=37 TSval=31
15	2.864172247	192.168.114.223	192.168.114.87	TCP	66	5201 → 47520 [ACK] Seq=1 Ack=38 Win=65152 Len=0 TSval=2539984
16	2.864296749	192.168.114.223	192.168.114.87	TCP	67	5201 → 47520 [PSH, ACK] Seq=1 Ack=38 Win=65152 Len=1 TSval=25
17	2.864582790	192.168.114.87	192.168.114.223	TCP	66	47520 → 5201 [ACK] Seq=38 Ack=2 Win=64256 Len=0 TSval=3148702
18	2.864666947	192.168.114.87	192.168.114.223	TCP	70	47520 → 5201 [PSH, ACK] Seq=38 Ack=2 Win=64256 Len=4 TSval=31
19	2.905933711	192.168.114.223	192.168.114.87	TCP	66	5201 → 47520 [ACK] Seq=2 Ack=42 Win=65152 Len=0 TSval=2539984
20	2.906231389	192.168.114.87	192.168.114.223	TCP	166	47520 → 5201 [PSH, ACK] Seq=42 Ack=2 Win=64256 Len=100 TSval=
21	2.906243542	192.168.114.223	192.168.114.87	TCP	66	5201 → 47520 [ACK] Seq=2 Ack=142 Win=65152 Len=0 TSval=253998
22	2.907389118	192.168.114.223	192.168.114.87	TCP	67	5201 → 47520 [PSH, ACK] Seq=2 Ack=142 Win=65152 Len=1 TSval=2
23	2.907708184	192.168.114.87	192.168.114.223	TCP	74	47530 → 5201 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
24	2.907726359	192.168.114.223	192.168.114.87	TCP	74	5201 → 47530 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460
25	2.907905828	192.168.114.87	192.168.114.223	TCP	66	47530 → 5201 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=31487025

Frame 11: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp0s3, id 0

Ethernet II, Src: PcsCompu_ef:d1:1b (08:00:27:ef:d1:1b), Dst: PcsCompu_5b:95:38 (08:00:27:5b:95:38)

Internet Protocol Version 4, Src: 192.168.114.87, Dst: 192.168.114.223

Transmission Control Protocol, Src Port: 47520, Dst Port: 5201, Seq: 0, Len: 0

0000 08 00 27 5b 95 38 08 00 27 ef d1 1b 08 00 45 00 ...[.8... ..E-

0010 00 3c 0e a4 40 00 40 06 45 90 c0 a8 72 57 c0 a8 ...<.@E...rW...

0020 72 df b9 a0 14 51 11 47 37 1f 00 00 00 00 a0 02 r...Q.67.....

0030 fa f0 b1 31 00 00 02 04 05 b4 04 02 08 0a bb ad ...1.....

0040 63 50 00 00 00 00 01 03 03 07cP.....