

Лабораториска вежба 2

Марко Бунтески 129/2021

Одговорете ги следните прашања:

1. На кое TCP/IP ниво функционира протоколот TCP?
- транспортно ниво.
2. Какви сервиси обезбедува протоколот TCP при пренос на податоци?
 - Контрола на загушување - congestion control
 - Контрола на тек - flow control
 - Воспоставување на врска - connection setup
3. Да се набројат неколку апликации кои се TCP-базирани?
FTP, SMTP, WWW.
4. Дали при дефинирање на TCP мрежна комуникација треба посебно да се пишува програма за TCP сервер и TCP клиент? Објасни!
Да, бидејќи серверот мора да биде стартуван и да слуша пред клиентот да ја воспостави врската.
5. Кои се четирите параметра со кои единствено се дефинира една TCP конекција?
destination IP address ,source IP address , source port number, destination port number.

Извршување на клиент-сервер TCP комуникација

1. Каков излез добивте во двата терминални прозорци?
server:

```
Command Prompt
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ANE>cd Desktop

C:\Users\ANE\Desktop>python TCPserver.py
Connection address: ('127.0.0.1', 59654)
received data: Zdravo od klientot
```

client:

```
C:\Users\ANE\Desktop>python TCPclient.py
received data: Zravo od serverot
```

2. Променете ја пораката и испратете ја. Каков излез добивте во двата терминални прозорци?

сервер : Connection address: ('127.0.0.1', 59748) received data: Zdravo od klientot клиент :
received data: Python rocks!

3. На која машина се извршуваат TCP клиентот и TCP серверот?
- localhost .

Анализа на клиент-сервер TCP комуникација со помош на Wireshark packet sniffer

1. Колку TCP датаграми се испратени при мрежната комуникација помеѓу TCP клиентот и серверот?
- 10
2. Кои се специфичните TCP датаграми кои се испраќаат при воспоставување на TCP конекција?
- Три се: првиот од клиент кон сервер со бр. на секвенца=262624764 и SYN=1 , вториот од сервер кон клиент со бр на секвенца= 716060005 SYN=1 и ACK=1, и третиот од од клиент кон сервер со бр. на секвенца=262624765 со ACK=1.
3. Колку од испратените TCP датаграми се однесуваат на испраќање на податоци? Дали може да го определите нивниот секвентен број?
- Два:
- 4от: 0.000205 127.0.0.1 127.0.0.1 TCP 84 45428 → 5005 [PSH, ACK] Seq=1 Ack=1 Win=32800 Len=18 TSval=415204 TSecr=415204 со секв. бр. 262624765.
- 6от: 0.024144 127.0.0.1 127.0.0.1 TCP 83 5005 → 45428 [PSH, ACK] Seq=1 Ack=19 Win=32768 Len=17 TSval=415211 TSecr=415204 со секв. бр. 716060006.
- 7: 0.024192 127.0.0.1 127.0.0.1 TCP 66 45428 → 5005 [ACK] Seq=19 Ack=18 Win=32800 Len=0 TSval=415211 TSecr=415211

Анализа на заглавија на различни TCP/IP нивоа: Изберете еден од испратените TCP датаграми и направете анализа за него!

1. Како изгледа испратената рамка (број на бајти и сл)?
- Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
2. Како изгледа Ethernet заглавието (изворна и дестинациска MAC адреса)?
- Ethernet II:
Destination: 00:00:00_00:00:00 (00:00:00:00:00:00)
Source: 00:00:00_00:00:00 (00:00:00:00:00:00)
Type: IPv4 (0x0800)
3. Како изгледа IP заглавието (верзија, изворна и дестинациска IP адреса, TTL и сл)?
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Time to live: 64.
4. Како изгледа TCP заглавието (изворна и дестинациска порта, секвентен број, ACK број и сл)?

- Transmission Control Protocol, Src Port: 45428, Dst Port: 5005, Seq: 19, Ack: 18, Len: 0
Sequence number (raw): 262624783
- 5. Колку чисти податоци се испратени од апликацијата? Да се пресмета overhead-от кој се воведува при испраќање на заглавија.
 - Чисти податоци од апликацијата: 18 бајти + 17 бајти = 35 бајти.
 - Overhead само од избраниот датаграм: 32 бајти + 20 бајти = 52 bytes.

Механизми за надежен пренос на податоци, кај TCP

Во Wireshark податоците и заглавјата се прикажани во хексадесимален формат.

1. Да се анализираат секвентните броеви кај сите TCP датаграми испратени од клиентот кон серверот и обратно!
 - 0, 0, 1, 1, 1, 1, 19, 19, 18, 20 – relative.
2. Да се анализираат броевите на потврди (ACK) кај сите TCP датаграми испратени од клиентот кон серверот и обратно!
 - /, 1, 1, 1, 19, 19, 18, 18, 20, 19.