

FIIT STU

Ilkovičova 2, 842 16 Karlova Ves

Komunikácia s využitím UDP protokolu

Autor: Martin Pirkovský

2020/2021

Návrh protokolu

Program je implementovaný v programovacom jazyku Python. Ako typ komunikácie som si zvolil server klient. Používateľ si môže vybrať či chce byť serverom alebo klientom. Keď si vyberie server, tak nastaví port, na ktorom má počúvať respektíve, z ktorého mu príde inicializačný packet. Server následne len prijíma počas posielania a keď dostane zlý packet, tak pošle správu aby ho poslal klient znovu. Po prenose server dostáva každých 60 sekúnd keep alive aby sa udržalo spojenie. Ak server nedostane keep alive tak ukončí spojenie. Ďalej môže server zmeniť svoju rolu a stať sa klientom. Ak si zvolí užívateľ klienta tak môže zadať port a IP servera, na ktorý chce posielať dáta, veľkosť fragmentu a text správy alebo súbor, ktorý chce poslať. Ďalej klient iba posiela dáta a po skončení posiela keep alive správu, pričom ak nedostane potvrdenie od servera každých 60 sekúnd, tak ukončí spojenie.

Veľkosť fragmentu

Veľkosť fragmentu si môže používateľ určiť od 1 do 1466. 1466 pretože od veľkosti 1500 B je potrebné odčítať veľkosť IP hlavičky (20 B), tiež veľkosť UDP hlavičky (8 B) a ešte veľkosť mojej hlavičky (6B). Do úvahy berieme 1500 B, pretože nechceme aby sa nám dáta pri posielaní neboli znova fragmentované na linkovej vrstve.

$$1500 - 20 - 8 - 6 = \mathbf{1466 \text{ B}}$$

Vlastná hlavička

Informačný paket:

<i>Bit Number</i>	
1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3	
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1	
Typ	Počet fragmentov (voliteľné)
Počet fragmentov	Názov súboru (voliteľné)

Veľkosť informačnej hlavičky: 1 B + 4 B (voliteľné) + názov súboru (voliteľné)

Dátový paket:

<i>Bit Number</i>	
1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3	
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1	
Poradové číslo fragmentu	
CRC	Dáta ...

Veľkosť dátovej hlavičky: 4 B + 2 B = 6 B

Informačný paket:

1) Typ paketu: 1 B

0	→	nadviazanie spojenia
1	→	potvrdenie úspešného doručenia fragmentu
2	→	Keep alive správa
3	→	text
4	→	súbor
5	→	chybný fragment
6	→	ukončenie spojenia

2) Počet fragmentov: 4 B

- Počet koľko bude poslaných dátových fragmentov.
- Potrebujem na to 2^{21} fragmentov, aby som vedel poslať súbor o veľkosti 2 MB po jednom byte.

3) Názov súboru (voliteľné):

- Tento údaj je voliteľný, nakoľko sa inicializuje iba ak posielam súbor.
- Veľkosť nie je potrebná, nakoľko sa iba dá nakoniec hlavičky ako posledný údaj.

Dátový paket:

1) Poradie fragmentu: 4 B

- Určuje poradie fragmentu, aby som nemohol mať pakety poprehadzované ak by prišli v inom poradí.

2) CRC: 2 B

- Na CRC, ktoré mi dá funkcia `crc_hqx(data, crc)` z knižnice `binascii`, mi stačia 2 B, nakoľko výstupom funkcie je maximálne 16 bitové číslo.

3) Dáta

- Dáta, ktoré sú posielané. Môže to byť text alebo súbor.

Selective Reject ARQ

Na zabezpečenie proti chybám som si vybral selective reject ARQ metódu. Táto metóda kontinuálne posiela packety po jednom za sebou. Prijímateľ na správne doručené packety odpovedá správou o úspešnom prijatí, no ak dostane chybný fragment, tak pošle správu o chybnom fragmente, čím požiada odosielateľa o znovu poslanie daného fragmentu. Odosielateľ takto pošle znova jeden daný chybný fragment, pričom neprestal posilať ďalšie fragmenty počas riešenia problému.

Keep Alive

Ako som už spomenul vyššie, tak na udržanie spojenia som sa rozhodol použiť keep alive časovač, ktorý je nastavený na 60 sekúnd. Teda každých 60 sekúnd, mimo posielania dát, pošle keep alive správu serveru aby udržal spojenie otvorené. Server pošle späť klientovi tiež keep alive správu, čím mu dá vedieť, že správu dostal a teda ostáva spojenie otvorené ďalších 60 sekúnd.

CRC (Cyclic Redundancy Check Code)

Na detekciu chybného fragmentu som sa rozhodol použiť vstavanú funkciu knižnice „binascii“, ktorá sa volá „crc_hqx(data, crc)“, kde data sú logicky naše dáta v byte-och, ktoré chceme poslať do CRC a crc nám predstavuje číslo, ktoré sa pridá pred dáta pri rátaní crc ako inicializačná hodnota. Ja som sa rozhodol použiť ako inicializačnú hodnotu „0xffff“, čo znamená, že pred dáta sa dá 16 jednotiek. Táto funkcia využíva CRC-CCITT s deliteľom zadaným ako polynóm „ $x^{16} + x^{12} + x^5 + 1$ “, čo je v binárnom tvare 1000 1000 0001 0000 1. Výpočet CRC „A“ by vyzeralo nejak takto:

- červenou sú inicializačné bity
- čiernou je ascii hodnota „A“ (65) v binárnom tvare
- modrou je (k-1) nulových bit-ov, ktoré sa pridávajú nakoniec, kde „k“ je dĺžka polynómu

```
1111 1111 1111 1111 0100 0001 0000 0000 0000 0000
1000 1000 0001 0000 1
----- XOR
 111 0111 1110 1111 11
.....
```

Takto by to pokračovalo až by sme nakoniec dostali hodnotu CRC = 1001 0100 0111 1001, ktorú následne pošleme spolu s dátami a poradovým číslom fragmentu serveru.

Diagram

