

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16, Bratislava

PKS Zadanie 2 – UDP komunikátor – Návrh riešenia

Fedor Viest

Termín odovzdania: 24.11.2021

Prednášajúci: prof. Ing. Ivan Kotuliak, PhD.

Cvičiaci: Ing. Lukáš Mastil'ak

Obsah

Zadanie	3
Návrh hlavičky	4
Typ správy (1B)	4
Číslo packetu (3B)	4
Checksum (4B)	5
ARQ metóda	6
Keep Alive metóda	6
Dôležité knižnice a funkcie	6
Flowchart	7

Zadanie

Navrhните a implementujte program s použitím vlastného protokolu nad protokolom UDP (User Datagram Protocol) transportnej vrstvy sieťového modelu TCP/IP.

Program umožní komunikáciu dvoch účastníkov v lokálnej sieti Ethernet, teda prenos textových správ a ľubovoľného súboru medzi počítačmi (uzlami).

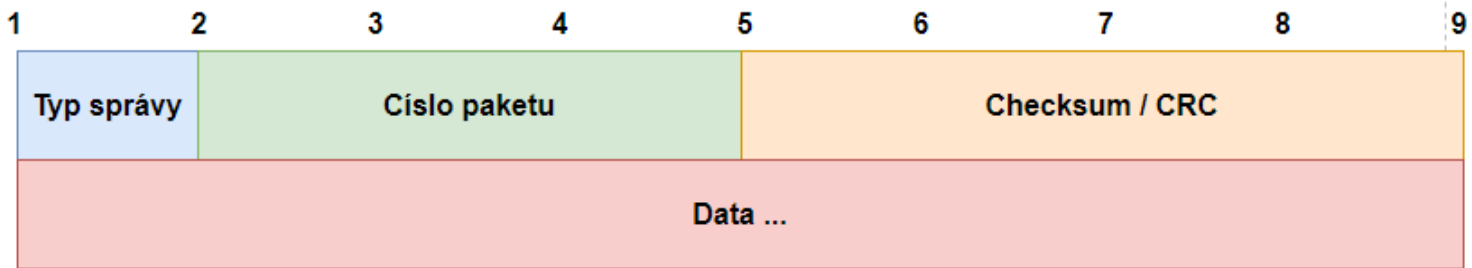
Program bude pozostávať z dvoch častí – vysielacej a prijímacej. Vysielací uzol pošle súbor inému uzlu v sieti. Predpokladá sa, že v sieti dochádza k stratám dát. Ak je posielaný súbor väčší, ako používateľom definovaná max. veľkosť fragmentu, vysielajúca strana rozloží súbor na menšie časti - fragmenty, ktoré pošle samostatne. Maximálnu veľkosť fragmentu musí mať používateľ možnosť nastaviť takú, aby neboli znova fragmentované na linkovej vrstve.

Ak je súbor poslaný ako postupnosť fragmentov, cieľový uzol vypíše správu o prijatí fragmentu s jeho poradím a či bol prenesený bez chýb. Po prijatí celého súboru na cieľovom uzle tento zobrazí správu o jeho prijatí a absolútnu cestu, kam bol prijatý súbor uložený.

Program musí obsahovať kontrolu chýb pri komunikácii a znovuvyžiadanie chybných fragmentov, vrátane pozitívneho aj negatívneho potvrdenia. Po prenesení prvého súboru pri nečinnosti komunikátor automaticky odošle paket pre udržanie spojenia každých 5-20s pokiaľ používateľ neukončí spojenie. Odporúčame riešiť cez vlastne definované signalizačné správy.

Program bude implementovaný v jazyku Python, kvôli ľahkej prístupnosti k pamäti a množstvu preddefinovaných funkcií, ktoré mi v konečnom dôsledku ušetria čas.

Návrh hlavičky



Typ správy (1B)

V tejto časti hlavičky sa nachádza informácia o aký typ správy ide. Správy sú reprezentované ako čísla v decimálnom formáte

- **0 – INIT** - Začiatok komunikácie
- **1 – MSG** - Inicializácia posielania správy, posiela klient serveru
- **2 – FILE** - Inicializácia posielania súboru, posiela klient serveru
- **3 – ACK** – potvrdenie o doručení packetu, posiela server klientovi
- **4 – NACK** – potvrdenie o doručení packetu s chybou, posiela server klientovi
- **5 – KPA** - Keep alive
- **6 – SWAP** – Výmena rolí
- **7 – END** - Ukončenie komunikácie

Číslo packetu (3B)

V tomto poli sa nachádza poradové číslo packetu. Zvolil som veľkosť 3B, lebo je potrebné preniesť 2MB súbor

Checksum (4B)

Na výpočet checksum som sa rozhodol použiť metódu **crc32** z knižnice **zlib**. Táto funkcia využíva 32 bitový polynóm vo formáte:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1, \text{ čö}$$

v binárnom formáte vyzerá takto:

1 0000 0100 1100 0001 0001 1101 1011 0111

Tento polynóm sa využíva na XOR so vstupom. V metóde crc sa využíva binárny posun vľavo a XOR.

Algorithmus:

1. Pridaj nakoniec 32 núl
2. XOR s polynómom, ak je prvý bit 1
3. Bitshift vľavo
4. Opakuj 4-5 pokým prvých 8 bitov nie je 0

[illegible]

Najväčšia veľkosť fragmentu môže byť 1463B, lebo:

$$1518 - 18(eth2) - 20(ip) - 8(udp) - 8(moja\ hlavička) = 1464 \text{ bajtov}$$

ARQ metóda

Ako ARQ metódu som si vybral **Stop-and-Wait (Bloková metóda)**. Klient vždy čaká na ACK od servera aby poslal ďalší packet. Pokiaľ server pošle NACK, klient odošle packet znova, až pokiaľ odpoveď servera nebude ACK, čiže packet dostal v poriadku.

Keep Alive metóda

V prípade nečinnosti na strane klienta, program prejde do fázy pre udržanie spojenia. Najprv klient pošle serveru správu **KPA**, následne v danom intervale (každých 5/10 sekúnd) server pošle **ACK** správu klientovi.

Dôležité knižnice a funckie

Nižšie je zoznam knižníc a niektorých funckii, ktoré určite použijem. Tento zoznam sa ešte určite rozrastie

Knižnice:

- Socket
- Zlib
- Threading

Funckie:

- sendto() – funckia z knižnice socket, pošle dáta
- recvfrom() – funckia z knižnice socket, prijme dáta
- crc32() – funckia z knižnice zlib, vypočíta checksum 32 bitov dlhý

V programe použijem vlastné funkcie na poslanie správy/súboru a ďalšie pomocné, v prípade potreby, keep alive budem riešiť v novom threade.

Flowchart

