

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta informatiky a informačných technológií

Počítačové a komunikačné siete

Zadanie 2-návrh:

Komunikácia s využitím UDP protokolu

Tomáš Socha

ID: 110896

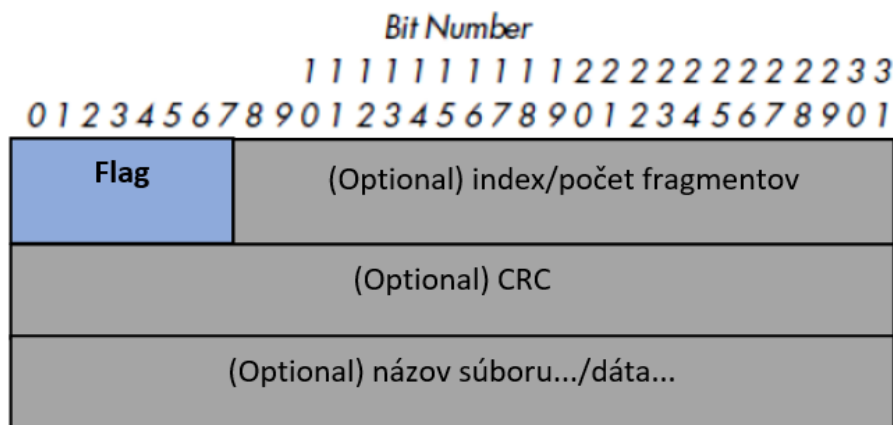
Akademický rok 2021/2022

1. Štruktúra hlavičky

Vlastná hlavička sa bude skladať z 2 častí. 1 bude povinná a druhá voliteľná na základe hodnoty v tej 1. časti, ktorá bude veľkosti 8bitov čiže 1B a bude obsahovať flag.

Ďalší blok má veľkosť 3B a obsahuje buď index, alebo počet fragmentov. 3B preto, aby bolo možné preniesť aj súbor o veľkosti 2MB pri malej veľkosti fragmentu. Tento blok je už voliteľný.

Ďalší voliteľný blok má veľkosť 4B a v ňom je uložená hodnota CRC pre prenášané dáta. Za ním sa už nachádzajú samotné dáta. Čiže hlavička. Maximálna veľkosť prenášaných dát bude 1500B - 20B(ip hlavička) – 8B(udp hlavička)– 8B(moja hlavička), čiže 1464B.



Obrázok 1- vlastný návrh hlavičky

Flagy:

S -> začatie spojenia(SYN), bez optional

F -> ukončenie spojenia(FIN), bez optional

K -> Keep-alive message, bez optional

C -> výmena rolí, bez optional

D -> inf. Správa o nadchádzajúcom odosielaní dát+ optional počet fragmentov + CRC+ názov súboru

M -> oznam o odosielaní message + počet fragmentov

A -> Acknowledgement správa(ACK) + index fragmentu

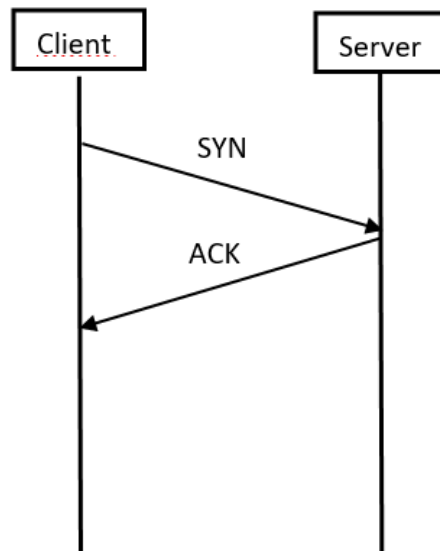
N -> Negative Acknowledgement(NACK) + index fragmentu

P -> posielanie packetu + index fragmentu + CRC + optional data

2. Komunikácia

a. Nadviazanie a ukončenie komunikácie

- Po zapnutí programu bude mať používateľ možnosť vybrať si či chce dáta posielať(client) alebo prijímať(server).
- Po prihlásení ako server používateľ zadá port, ktorý sa následne pripojí k vytvorenému socketu a bude počúvať na zadanom porte.
- Po prihlásení ako Client používateľ zadá IP adresu a port na ktorom server počúva a odošle mu SYN správu.
- Server prijme túto správu, uloží si adresu odkiaľ bola správa poslaná a na túto adresu odošle svoju SYN správu.
- Po doručení SYN správy clientovi bolo spojenie úspešne nadviazané a môže dôjsť k prenosu súborov.



Obrázok 2- nadviazanie komunikácie

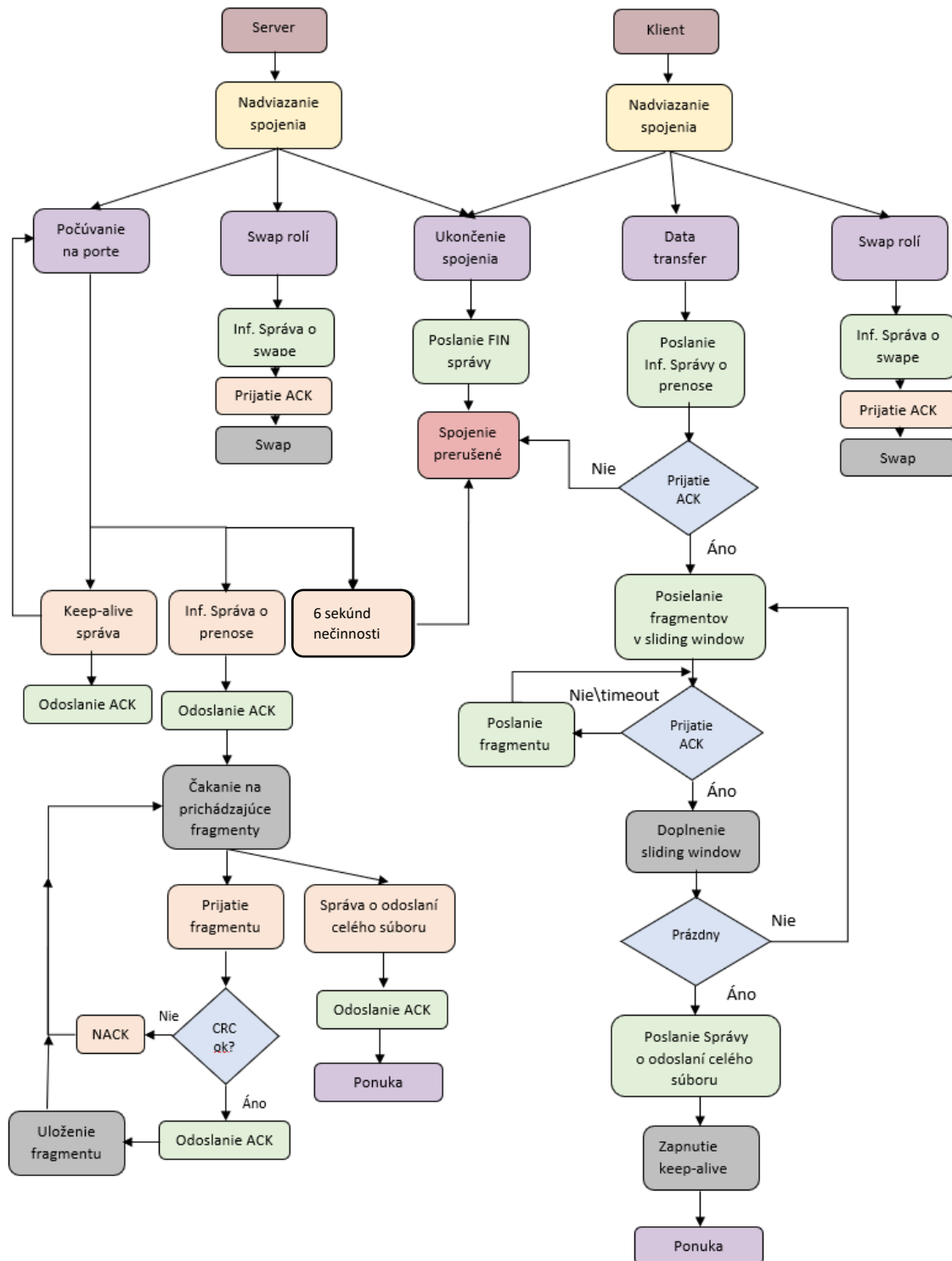
- Ku ukončeniu spojenia dôjde odoslaním FIN správy či už od vysielajúcej alebo prijímajúcej strany. Taktiež pri nepotvrdení Keep-alive správy v stanovenom časovom intervale.

b. Spracovávanie komunikácie

- Po úspešnom nadviazaní spojenia sa na oboch stranách zobrazí ponuka. V prípade odosielaajúcej strany si bude užívateľ môcť zvoliť či chce prenášať súbor alebo správu, vykonať výmenu rolí alebo ukončiť spojenie a ukončiť program.

Diagram nižšie predstavuje koncept spracovávanía komunikácie. V prípade zvolenia možnosti posielania si užívateľ bude môcť vybrať či chce poslať správu alebo súbor. Rozdiel bude v tom, že správa sa po prenesení vypíše na konzolu a pri súbore sa súbor uloží na požadovanú adresu. Taktiež užívateľ zadá ktorý súbor chce preniesť a akú chce mať veľkosť fragmentu.

Server po nadviazaní komunikácie a nezvolení nejakej z možností ponuky dostane informačnú správu o doručení súboru, ktorú môže prijať alebo odmietnuť. V prípade prejdenia v ponuke do stavu počúvania bude súbor prijatý automaticky.



Obrázok 3- diagram spracovávaní komunikácie

c. Výmena rolí

- pre výmenu rolí je potrebné aby jedna strana o to požiadala. Toto bude realizované Informačnou správou o swape(C). Výmena nastane na oboch stranách, no až po prijatí ACK správy od strany, ktorá výmenu inicializovala.

d. Keep-alive

- Po prenesení súboru sa spustí časovač na 3s ,kde vždy po ich uplynutí klient odošle Keep-alive správu a bude očakávať potvrdenie od servera. V prípade že nedostane odpoveď do 6s tak vypíše správu o prerušení spojenia a ukončí program.

e. simulácia chyby

- Chybu budem simulovať tým, že v každom nepárnom fragmente odpočítam od CRC číslo 10. Toto spravím v prvých 20 fragmentoch. Taktiež v prípade posielania viacerých fragmentov druhý a predposledný zámerne nepošlem vôbec.

3. CRC metóda

Ako metódu pre kontrolu, či dáta prišli nepoškodené som si zvolil crc32. Túto metódu som si zvolil hlavne preto, lebo v porovnaní s checksum dokáže detegovať viacej chýb kvôli komplexnejšej funkcii. CRC pracuje s dátami po jednotlivých bitoch. Jeho princíp spočíva vo vypočítaní zvyšku po vykonaní operácie XOR nad dátami a generujúcim polynómom¹. Generujúci polynóm(polynomial) má v tomto prípade tvar: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

Čo predstavuje binárne tvar: 1 0000 0100 1100 0001 0001 1101 1011 0111.

Výsledná hodnota crc bude mať veľkosť 32-bitov, čiže o 1 menej ako polynóm. Najskôr sa ku bitom predstavujúce dáta nakoniec prilepí 32 núl. Potom sa naplní posuvný register bitovým posunom doľava. Zvyšok vstupných dát sa nachádza napravo od tohto registra.

Po naplnení sa pozrieme na prvý bit v registri(zľava), a ak je jedna tak vykonáme posun vľavo, čiže prvý bit už bude mimo posuvného registra, ale ešte si ho zapamätáme. Následne vykonáme operáciu xor s polynómom a výsledok vložíme do registra. 1.bit,ten ktorý už bol mimo registra sa po aplikovaní XOR s 1 zmenil na 0, takže nám vypadol.

Novú hodnotu v posúvnom registri posúvame doľava, kým nám znovu na jeho začiatku nenachádza 1. potom opakujeme rovnaký postup(posun o 1,xor, posun).Keď sa už naľavo od posuvneho registra nebudú nachádzať žiadne bity, tak v posuvnom registri sa bude nachádzať výsledná hodnota CRC.

¹ http://www.sunshine2k.de/articles/coding/crc/understanding_crc.html

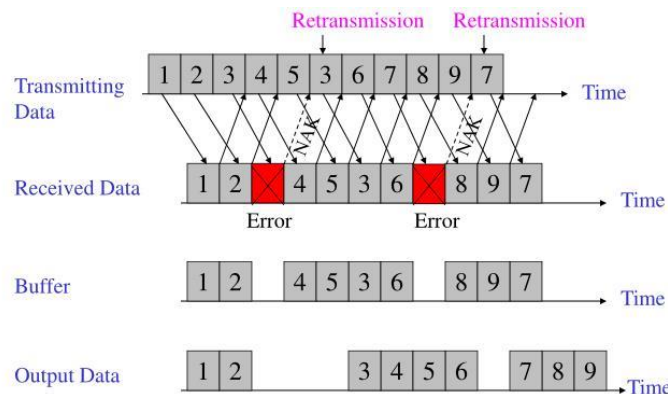
4. ARQ

Vo svojom komunikátore plánujem použiť Selective ARQ metódu na potvrdzovanie prijatých fragmentov. Jej hlavný princíp spočíva v tom, že vysielateľ opätovne posiela len tie fragmenty, ktoré neprišli korektné, alebo vôbec.

Ako prvé sa zistí počet fragmentov. Odosielajúca strana potom odošle informačnú správu o počte fragmentov, veľkosti fragmentu a názvu súboru. Po obdržaní ACK správy sa prenášaný súbor rozdelí na stanovený počet, každému fragmentu sa pred dáta vloží hlavička a uloží sa do listu. Následne sa určí veľkosť sliding window, ktorá bude zodpovedať celočíselnej hodnote po delení 4, no jeho najmenšia veľkosť bude 2. Samozrejme ak odosielame aspoň 5 fragmentov. Inak bude 1. Z listu fragmentov sa vždy vyberú prvé indexy ešte nevybraných fragmentov a vložia sa do sliding window.

Na začiatku sa zaradom odošlú všetky fragmenty ktorým prislúchajú indexy uložené v sliding window a zapne sa časovač. Vysielajúca strana čaká na ACK správy v poradí ako ich odoslala. V prípade že dostane NACK pre daný index, vyprší čas, alebo dostane ACK pre nasledujúci fragment, tak odošle daný fragment ešte raz a index vyberie zo začiatku a vloží ho na koniec sliding window. V prípade obdržania ACK zoberie v poradí prvý ešte nevybraný index z listu fragmentov, vloží na koniec sliding window a odošle. Takto pokračuje kým sa nevyprázdni sliding window, čo bude znamenať že prijímajúca strana obdržala všetky fragmenty. Následne odošle správu o doručení celého súboru.

Prijímajúca strana prijme správu o plánovanom prenose súboru, ktorú potvrdí ACK správou. Následne si vytvorí list, kde bude podľa indexov ukladať nepoškodené doručené fragmenty a odosielať ACK alebo NACK pre každý fragment. Po prijatí správy o doručení celého súboru ho poskladá naspäť do jedného celku a uloží na vopred používateľom zadanú adresu.



Copyright © 2002, Dr. Dharma P. Agrawal and Dr. Qing-An Zeng. All rights reserved.

30

Obrázok 4- Selective- Repeat ARQ ²

² <https://www.slideserve.com/marcie/chapter-4-3259668>

5. Popis častí zdrojového kódu

Vlastnú implementáciu budem programovať v jazyku Python vo verzii 3.9. Zdrojový kód bude obsahovať nasledujúce knižnice:

- **socket** -> funkciu *socket()* - na vytváranie socketov
- **struct** -> funkcie *pack()* a *unpack()* – na vytvorenie bajtového objektu, a následne jeho rozbalenie
- **time** -> funkciu *sleep()* – vo funkcii keep-alive na pravidelne posielanie Keep-alive správ
- **zlib** -> funkciu *crc32()* – funkcia pre kontrolu doručených dát
- **threading** -> funkciu *Thread()* – pre vytvorenie threadov pre Keep-alive a na dostávanie ack správ od prijímateľa pri posielaní dát.

Program bude organizovaný do 3 hlavných častí. Jedna časť bude obsahovať funkcionality klienta, ako funkciu pre keep-alive, funkciu na nadviazanie spojenia, funkciu zabezpečujúcu výmenu prijímajúcej a vysielajúcej strany, funkciu ktorá načíta dáta ktoré sa budú prenášať a funkciu pre zabalenie hlavičky. Ďalej bude obsahovať hlavnú funkciu klienta kde bude prebiehať samotná komunikácia a z ktorej sa budú volať vyššie spomenuté funkcie.

Ďalšia časť bude určená pre server. On bude obsahovať funkciu na nadviazanie spojenia, funkciu na výmenu rolí, na uloženie preneseného súboru a funkciu zabezpečujúcu rozbalenie hlavičky. Taktiež nebude chýbať jeho hlavná funkcia odkiaľ bude spracovávať prijímané dáta.

Tretia časť bude obsahovať hlavne vizuálnu časť programu, čiže funkciu pre zobrazenie úvodnej ponuky pre výber funkcionality prijímača/vysielača a následnej ponuky serveru a klienta.