

Slovenská technická univerzita
Fakulta informatiky a informačných technológií

Počítačové a komunikačné siete

Komunikátor pre výmenu dát nad protokolom UDP

Meno: René Bukovina

Cvičiaci: Bc. Branislav Jančovič

Cvičenie: štvrtok 16:00

Implementačné prostredie

Komunikátor bude realizovaný v programovacom jazyku Python, nakoľko je to odporúčaný jazyk, ktorý disponuje knižnicami vhodnými na realizáciu zadania. Program bude fungovať ako konzolová aplikácia. Testovanie bude prebiehať na operačnom systéme Windows 11.

Popis funkcionality programu

Program bude obsahovať dve hlavné časti a to stranu **klienta** a stranu **servera**. Pri zvolení možnosti strany servera, je používateľ povinný zadať IP adresu, na ktorej bude server prístupný a taktiež aj port pre server. Server následne začne počúvať na inicializačné správy od klienta, pre vytvorenie spojenia, ktoré sa bude vykonávať 3-way handshakom. Až po úspešnom nadviazaní spojenia, bude klient schopný posilať správy/súbory serveru. Predtým ako bude môcť posilať správy, bude musieť zvoliť, **maximálnu akceptovateľnú veľkosť fragmentu**, ak tak neurobí, program zvolí maximálnu možnú veľkosť pre fragment. Ak sa nebude vykonávať žiadna komunikácia, server si bude s klientom vymieňať, **keep-alive** správy, každých 30 sekúnd. Ak sa nedostaví odpoveď do 30 sekúnd, server ukončí spojenie s klientom. Ak neprebíha posielanie dát, klient môže zažiadať o výmenu roli so serverom, t.j. klient sa stane serverom a server sa stane klientom bez toho aby sme vypli program.

Checksum/CRC

Kontrolu správnosti prijatých dát budem riešiť metódou CRC, na základe polynómu CRC-32, ktorý je štandardom pre rámce Ethernet II. Na získanie polynómu použijem knižnicu **zlib**, ktorá tento polynóm obsahuje. Zvolil som túto metódu, nakoľko samotný checksum, môže naraziť na problémy, keď označí dáta za korektné aj napriek tomu, že korektné nemusia byť.

CRC (Cyclic Redundancy Check) je kontrolný algoritmus používaný na detekciu chýb v dátach pri ich prenose cez sieť. Tento algoritmus je široko používaný pri posielaní paketov v rôznych komunikačných protokoloch, ako napríklad Ethernet, USB, Bluetooth a mnoho ďalších.

Fungovanie výpočtu CRC pri posielaní paketov je nasledovné:

- Odosielané dáta rozdelíme do fragmentov, ak je to potrebné
- Pri každom fragmente pridáme nakoniec 32 núl k jeho binárnej hodnote
- Následne toto číslo použijeme modulo polynómom CRC-32, ktorý je definovaný ako štandard
- Dostaneme zvyšok, ktorý následne odčítame od polynómu CRC-32
- Dostaneme polynóm 0 a 1 teda 32 miestne binárne číslo teda našu hodnotu CRC, ktorým nahradíme prvotne pridaných 32 núl na konci nášho paketu
- Toto nám zabezpečí, že ak nám dáta došli v poriadku, tak po použití modula na naše dáta aj s CRC dostávame číslo deliteľné našim polynómom teda zvyšok 0, ak máme iný zvyšok ako 0, tak dáta boli chybné

Výpočet CRC je rýchly a spoľahlivý spôsob detekcie chýb v prenášaných dátach. V prípade, že sa zistí chyba, môže byť paket opätovne poslaný, aby sa zabezpečila správna komunikácia medzi zariadeniami v sieti.

ARQ Selective Repeat

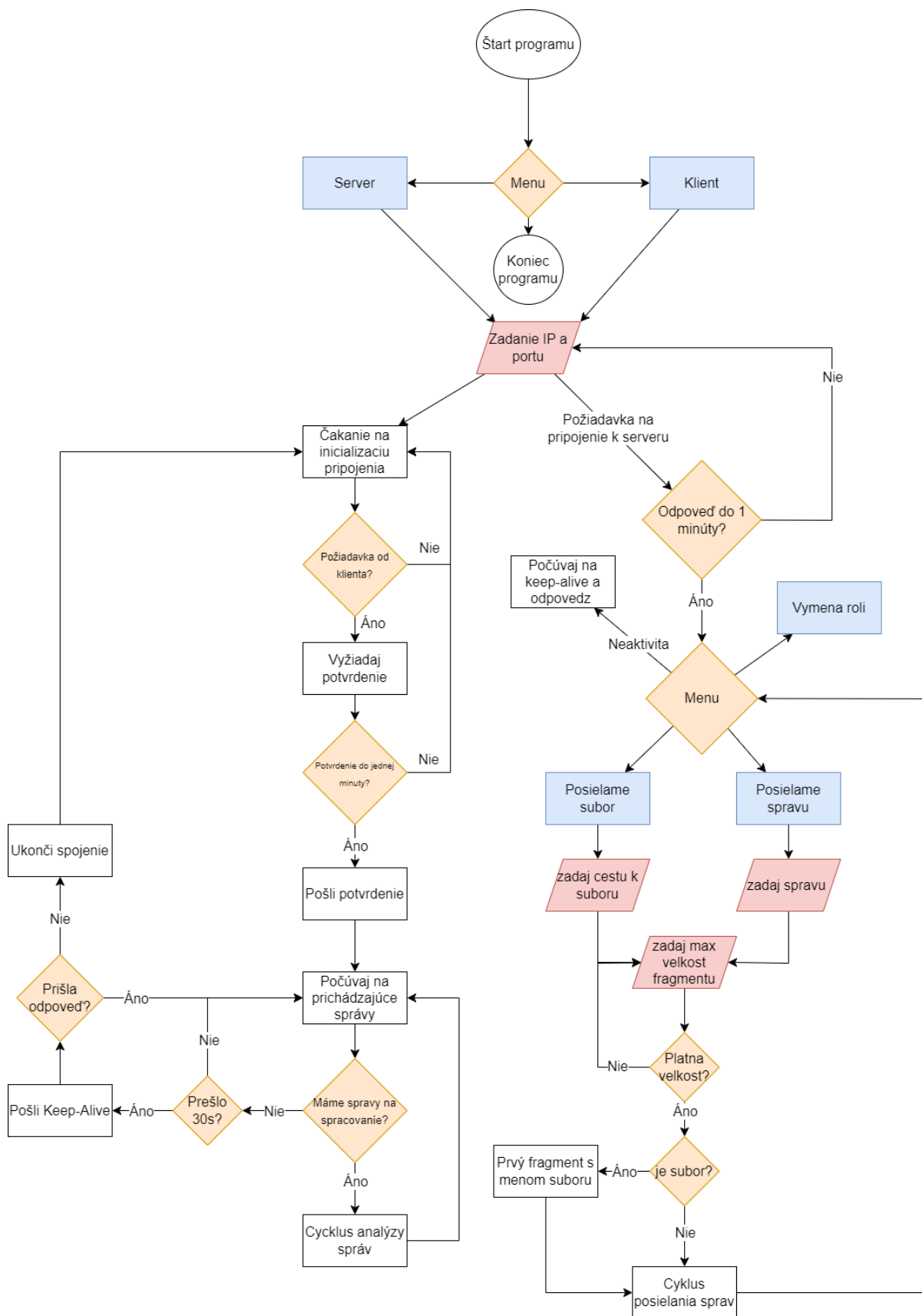
Prenos dát budem riešiť metódou **ARQ Selective Repeat**, čiže klient odošle pošle všetky alebo N správ bez ohľadu na to, či to server prijme, potom si server skontroluje, ktoré rámce boli chybné a vyžiada si znovu, iba tie pakety, v ktorých bola chyba. Tento postup sa opakuje až kým nemáme všetky pakety. Následne server odošle správu o tom, že už získal všetky parkety a klient tak bude vedieť, že nemá posilať žiadne ďalšie pakety.

Návrh hlavičky vlastného protokolu

flag 1B	fragment_size 2B	fragment_number 3B	data x < 1463B	crc 4B
------------	---------------------	-----------------------	-------------------	-----------

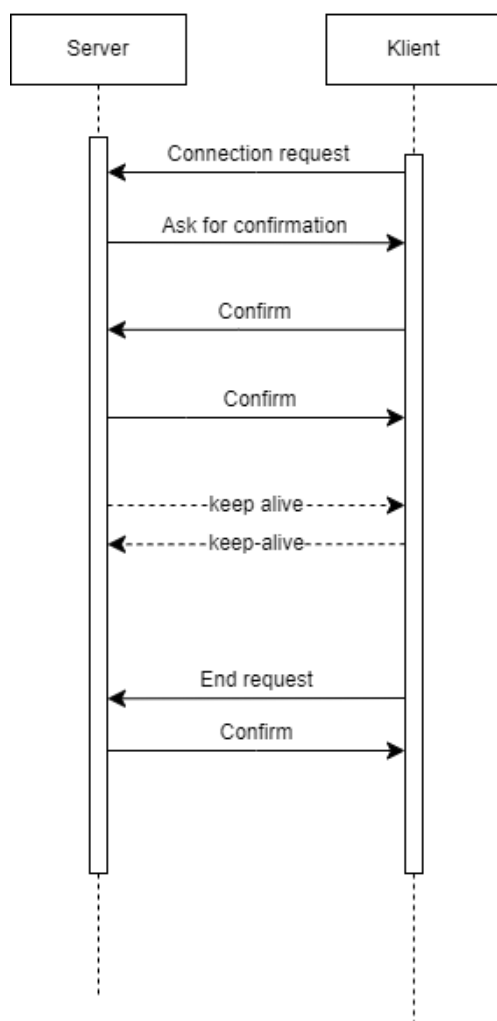
flag (dec)	flag (bin)	Popis funkcionality
1	0000 0001	FIN = 1
2	0000 0010	FIN = 1, SYN = 0
3	0000 0011	SYN = 1
4	0000 0100	Posielanie dát
5	0000 0101	Odpoveď na správu ACK = 1
6	0000 0110	Odpoveď na správu ACK = 0 vyžiadať zlý packet
7	0000 0111	Ukončenie odosielania správy
8	0000 1000	Keep Alive
9	0000 1001	Výmena rolí klient <=> server
10	0000 1010	Packet s názvom súboru

Flowchart programu



Sekvenčné diagramy

Otváranie a zatváranie spojenia



Priebeh komunikácie

