

Počítačové a komunikačné siete

Zadanie č.2 - Komunikácia s využitím UDP protokolu

Autor: Marek Čederle
AIS ID: 121193 (xcederlem)
Cvičiaci: Ing. Matej Janeba
(Utorok 16:00)

Návrh

Implementáciu budem robiť v programovacom jazyku python. Je to z dôvodu že má je jednoduchší ako C/C++ s tým že má knižnice na veľa vecí a celkovo sa s ním jednoduchšie pracuje. Síce jeho nevýhodou je že je pomalší ale to v našom prípade nebude tak vadiť. Pôjde o console program, to znamená že výstupy aj vstupy budú robené cez konzolu. Bude to v podstate také TUI (Terminal User Interface). Keďže pracujem na počítači s Windows tak aj následné testovanie bude prebiehať na Windows PC prípadne na localhoste. Môj protokol sa nazýva DNP.

Vlastná hlavička

Horná hlavička reprezentuje Ethernet hlavičku s vnorenými IPv4 a UDP protokolmi a PAYLOADOM, v ktorom sa nachádza môj protokol respektíve moja hlavička a následne dáta.

Preamble	Delimiter	Ethernet	IPv4	UDP	PAYLOAD	Ethernet CRC	IPG
7	1	14	20	8	18-1472	4	12

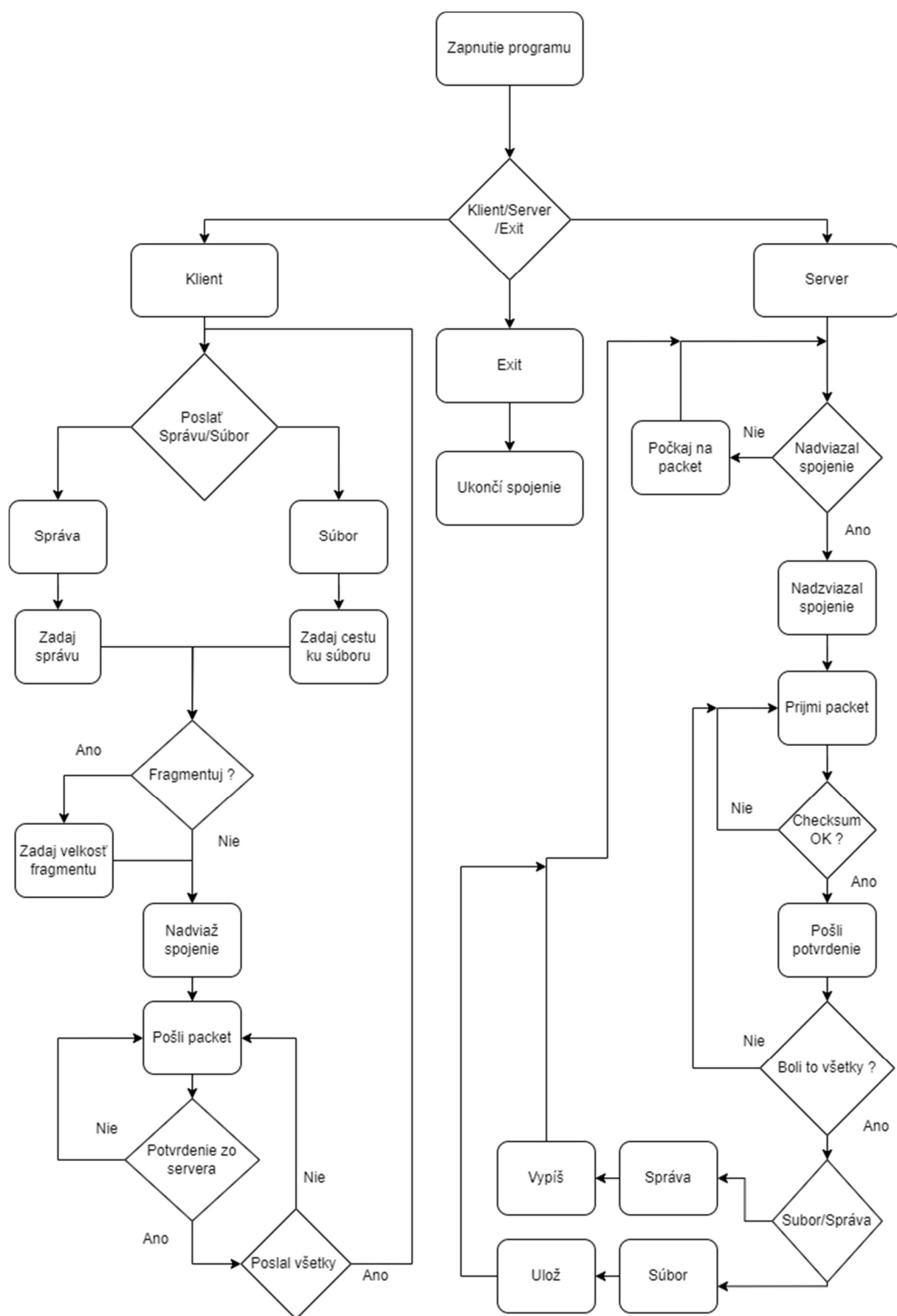
PAYLOAD								
DNP HEADER								
MSG TYPE	PACKET LENGHT	FRAG COUNT	FRAG OFFSET	CHECKSUM	DATA	without header	Ethernet minimal	60
1	2	2	2	1	10-1464	theoretical minimum	Ethernet max	1526
					24-1464	defined minimum		

- MSG TYPE – tip riadiacej správy (1B)
- PACKET LENGHT – veľkosť fragmentu (2B)
- FRAG COUNT – počet fragmentov (2B)
- FRAG OFFSET – offset kde majú pokračovať dáta (2B)
- CHECKSUM – kontrolný súčet pre dátovú časť (nazývam to ako checksum ale v skutočnosti ide o paritu) (1B)
- DATA – dáta vo vnútri packetu

Typy správ v MSG TYPE:

Decimal	Binary	Info
1	00000001	Keep alive
2	00000010	Nadviazanie spojenia
3	00000011	Ukončenie spojenia
4	00000100	Posielanie dát - správa
5	00000101	Posielanie dát - súbor
6	000000110	Chyba pri doručení dát
7	000000111	Doručenie dát bolo OK

Diagram a opis



Používateľ si vyberie či chce byť klient alebo server. Klient je odosielať dát a server je prijímať dát.

Klient:

Klient musí zadať IP adresu a port servera kam má posilať dáta. Takto sa nadviaže spojenie a potom zadá či chce posilať správu alebo súbor. Pri súbore treba zadať jeho cestu, pri správe treba zadať text. Potom zadá fragmentovanie prípadne počet fragmentov. Potom sa stále posila keep alive a zároveň sa začnú posilať packety. Pri posilaní čaká na potvrdenie a ak mu server pošle potvrdenie tak posila ďalej.

Server:

Server si nastaví IP adresu a port na ktorom bude počúvať. Čaká na inicializáciu spojenia. Ak sa klient snaží nadviazať tak server mu pošle správu o prijatí. Server pri prijímaní fragmentov vždy kontroluje každý jeden podľa checksum. Ak nastane chyba vypýta si znova. Keď sa pošlú všetky tak ak to bol súbor tak sa uloží ak správa tak ju vypíše. Potom zase počúva.

ARQ metóda

Budem používať STOP & wait ARQ metódu. Bude fungovať nasledovne:

Jej fungovanie v podstate vyplýva z jej názvu. Klient pošle dáta a ak server nepošle potvrdzovaciu správu že prišli dáta OK tak klient počká 8 sekúnd či náhodou nepríde potvrdzovacia správa. V prípade že príde tak pošle znova posledný packet.

Checksum (Parita)

Parita bude vypočítaná z dátovej časti a to nasledovne:

Keďže má 1B a to je 8 bitov tak sa dátová časť rozdelí na 8 častí a z každej sa vypočíta parita. Tá sa počíta spôsobom že ak v danej časti bol párny počet jednotiek tak dá nulu a ak nepárny počet jednotiek tak dá jednotku. Toto sa urobí pre každú časť a výsledných 8 bitov sa zapíše do hlavičky. Následne server po prijatí správy vypočíta podľa rovnakého postupu paritu a porovná s hlavičkou. Ak sa rovnajú tak prijme packet, ak nie tak ho zahodí a vypýta si ho znova.

Keep alive metóda

Metóda na udržanie spojenia funguje nasledovne:

Každých 12 sekúnd sa bude posilať správa na udržanie spojenia klientom. Server na ňu musí odpovedať. Ak neodpovie tak sa spojenie ukončí.