**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.

Obrázok, na ktorom je text, snímka obrazovky, písmo, číslo

Automaticky generovaný popis

* 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**

Obrázok, na ktorom je diagram, náčrt, plán, technický výkres

Automaticky generovaný popis

Používateľ si vyberie či chce byť klient alebo server. Klient je odosielateľ dát a server je prijímateľ dát.

Klient:

Klient musí zadať IP adresu a port servera kam má posielať dáta. Takto sa nadviaže spojenie a potom zadá či chce posielať správu alebo súbor. Pri súbore treba zadať jeho cestu, pri správe treba zadať text. Potom zadá fragmentovanie prípadne počet fragmentov. Potom sa stále posiela keep alive a zároveň sa začnú posielať packety. Pri posielaní čaká na potvrdenie a ak mu server pošle potvrdenie tak posiela 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 posielať správa na udržanie spojenia klientom. Server na ňu musí odpovedať. Ak neodpovie tak sa spojenie ukončí.