SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Komunikátor s použitím UDP protokolu

NÁVRH IMPLEMENTÁCIE RIEŠENIA

Anna Yuová

Predmet: Počítačové a komunikačné siete

Cvičenie: Štvrtok 8:00

Akademický rok: 2020/2021

Semester: zimný

ZADANIE

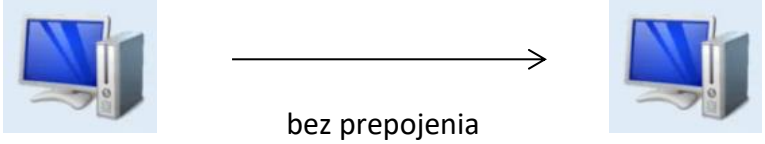
Nad protokolom UDP transportnej vrstvy sieťového modelu TCP/IP navrhnite a implementujte program s použitím vlastného protokolu, ktorý umožní komunikáciu dvoch účastníkov v lokálnej sieti Ethernet, teda prenos textových správ a ľubovoľného súboru medzi počítačmi (uzlami).
Program bude pozostávať z dvoch 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 maximálna 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 znovu vyžiadanie chybných fragmentov, vrátane pozitívneho aj negatívneho potvrdenia. Po prenesení prvého súboru pri nečinnosti komunikátor automaticky odošle paket pre udržanie spojenia každých 10-60s pokiaľ používateľ neukončí spojenie. Odporúčame riešiť cez vlastne definované signalizačné správy.

ANALÝZA UDP A TCP PROTOKOLOV

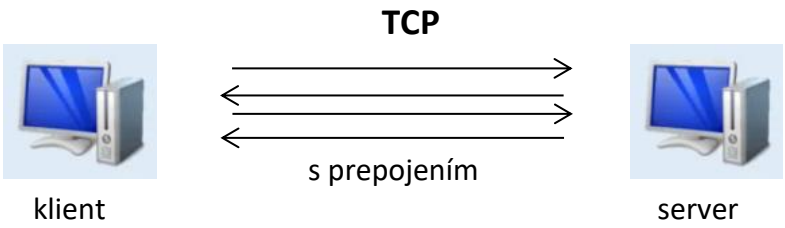
Komunikátor slúži na výmenu správ medzi 2 zariadeniami, ktoré sú vzájomne prepojené v počítačovej sieti. Nachádzam sa nad IP vrstvou (sieťovou) a TCP a UDP protokoly pracujú na transportnej vrstve modelu. Základný rozdiel medzi nimi je v spoľahlivosti. TCP protokol sa považuje za spoľahlivý a používa sa na prenos správ po internete. UDP protokol je naopak nespoľahlivý, no rýchlejší a používa sa na doručenie videa a hlasu.

Rozdiel medzi TCP a UDP protokolmi:

UDP



Pred odosielaním správ v UDP protokole nie je vytvorené prepojenie medzi klientom a serverom a nekontroluje ani to, či je server pripravený prijať dáta a rovno ich odošle. Ak by prišlo k strate nejakých údajov server príjme aj stratené (poškodené) dáta a nevypýta si nové. Má 8B hlavičku.



Naopak od UCP si TCP protokol vytvorí prepojenie medzi klientom a serverom. Ak by prišlo k strate nejakých údajov, vypýta si potvrdenie údajov a vie vyžiadať znovu poslanie údajov. Má 20B hlavičku.

NÁVRH RIEŠENIA

Program navrhnem tak, aby bežal na oboch počítačoch naraz, pričom jeden počítač bude odosielať informácie(textové a iné súbory) a druhý ich bude prijímať. Toto bude implementované tak, aby to fungovalo bez vypnutia celého programu a spustenia od začiatku.

Ďalej si urobím analýzu údajov, ktoré budem potrebovať. Potrebujem vytvoriť protokol, ktorý by vedel vytvoriť prepojenie medzi klientom a serverom. Toto spojenie by nemal stratiť a vedel by ho ukončiť. Protokol mi musí potvrdiť, že prijaté fragmenty prijal správne alebo ak ich neprijal správne (došlo k strate), musí ich vyžiadať znovu. Na zabezpečenie tejto požiadavky si vytvorím hlavičku:

TYP	POČET	PORADIE	VEĽKOSŤ	ODOSIELANÉ INFO	CHECKSUM

TYP HLAVIČKY:
Informácia o veľkosti fragmentu
Naviazanie spojenia – 0010
Ukončenie spojenia – 0100
Posielanie keep-alive – 0110
Prijatie chybného fragmentu – 0011
Poslanie správa – 0111
Poslanie súboru - 0101

POČET:
Počet poslaných fragmentov. Maximum je 65 536 (2^16).

PORADOVÉ ČÍSLO:

Informácia o tom, ktorý fragment bol poslaný. Maximálne poradie je 2^16 – 1. Ak boli odoslané všetky správne (neboli poškodené), hlavička si uloží počet správne prijatých fragmentov. Ak bol prijatý nejaký chybný fragment, uložím si index chybného fragmentu.

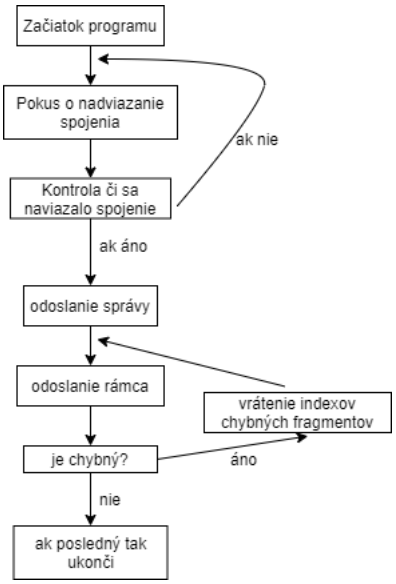
VEĽKOSŤ:
Informácia o odosielanom fragmente (jeho dĺžka). Najväčší fragment môže mať 1500B – veľkosť mojej hlavičky, pretože linková vrstva má najviac 1500B.

CHECKSUM/CRC:
Hashovacia funkcia, ktorá bude na konci správy. Ak by čísla nesedeli znamená to, že bola prijatá chybná správa a správu odošle tam, odkiaľ prišla. Tam ju zmením naspäť na dobrú a odošlem znovu.

Používateľské rozhranie:
Program implementujem v jazyku Python a v prostredí Pycharm (verzia 2020.2.3). Výstup bude cez konzolu, nie GUI.

Implementovať by som to chcela pomocou server – klient (nie peer to peer). Správy by som odosielala po blokoch (napr. pošlem 5 fragmentov naraz a ACK vráti error code a zároveň indexy neprijatých ak tam bude nejaký chybný). Stratu dát musím urobiť manuálne a to urobím tak, že na začiatku mám napr. ‚ABC', vypočítam jeho framecheck a uložím si ho do checksum. Následne zmením dáta z ‚ABC' na ‚ABD' a rovno pošlem do druhého počítača. Na druhej strane to prepočítam a ak sa tie hodnoty nebudú zhodovať, tak to bude vyhodnotené ako chybný fragment – ACK vráti error code s indexami chybných fragmentov a tá prvá strana zoberie tie isté dáta, ktoré prišli chybné a pošle ich znova už v správnom tvare. Ak bude posielaný súbor väčší ako je maximálna veľkosť – musím si ho rozdeliť na menšie časti – fragmenty. Nastavím veľkosť fragmentu a chcem, aby sa to na linkovej vrstve už nefragmentovalo (1500B – veľkosť mojej hlavičky). Ďalej budem potrebovať keep-alive pomocou klient-server, ktoré bude zabezpečovať odoslanie mojej správy. Ak je úspešné, tak čakám do 60-120sekúnd (potom čas vyprší). Ak je neúspešné tak končím. Ďalej ak mám úspešné spojenie, spustím poslanie správy, skontrolujem, či sa fragmenty zhodujú a skontrolujem checksum. Skontrolujem či prišli všetky a ak nie potrebujem vyžiadať tie, ktoré chýbajú. Inak ukončím spojenie a aj keep-alive.
Prijímanie je rovnako ako odosielanie, potvrdím prijatie fragmentov a ak sa nepodarí nadviazať spojenie tak skončí. Ak sa podarí nadviazať spojenie ale nezíska potvrdenie o prijatých fragmentoch tak sa o to bude pokúšať znova. Ak získa chybné správy, uloží si do pola indexy chybných fragmentoch a tie pošle naspäť odkiaľ prišli a vyžiada ich znova. Ak budú všetky v poriadku, ukončím spojenie aj program.

KLIENT:


Začiatok programu
Pokus o nadviazanie spojenia
ak nie
Kontrola či sa naviazalo spojenie
ak áno
odoslanie správy
odoslanie rámca
vrátenie indexov chybných fragmentov
je chybný?
áno
nie
ak posledný tak ukonči

SERVER:


Začiatok programu
Prijatie správy
ak nie znovu vyžiada dáta
je správa ok?
ak áno
odoslanie ack o prijatí
ak posledný tak ukonči