

Počítačové a komunikačné siete

Zadanie 2

Komunikácie s využitím UDP protokolu

Obsah

Zadanie.....	3
Vlastnosti:.....	3
Návrh riešenia	3
Štruktúra hlavičky.....	3
Kontrolné pole - checksum (4B)	4
Poradové číslo fragmentu (4B)	4
Celkový počet fragmentov (4B)	4
Dĺžka fragmentu (2B).....	4
Identifikačný kód – Code (1B).....	4
Checksum	4
CRC	4
Automatic repeat request – ARQ	4
Stop-and-Wait	5
Udržanie spojenia.....	5
Spracovanie príliš veľkého súboru - fragmentovanie	5
Diagramy	5
Nadviazanie spojenia.....	5
Prenos správ a dát	6
Fungovanie prijímača	6
Fungovanie vysielacza.....	6

Zadanie

Navrhните a implementujte program s použitím vlastného protokolu nad protokolom UDP transportnej vrstvy sieťového modelu TCP/IP. Program umožní komunikáciu dvoch účastníkov v lokálnej sieti Ethernet, prenos ľubovoľných správ a súborov medzi zariadeniami (uzlami).

Program bude pozostávať z dvoch častí:

- Vysielacia
 - o Pošle súbor inému uzlu v sieti.
 - o Pri presiahnutí veľkosti musí zabezpečiť rozloženie na menšie fragmenty.
- Prijímacia
 - o Pri rozkladaní súboru poskladá súbor, potvrdzuje a kontroluje správnosť doručených rámcov.
 - o Vypisuje poradové číslo prijatia rámca a či bol prenesený bez chýb.
 - o Po prijatí celého súboru, vypíš umiestnenie súboru.

Program musí riešiť kontrolu chýb -> prijímací uzol si musí vyžiadať rámec pri zistení chybného rámca.

Program musí zabezpečovať udržanie otvorenej komunikácie medzi uzlami s využitím signalizačných správ.

Vlastnosti:

- Optimálna práca s dátami
- Určenie cieľovej IP adresy a portu
- Voľba maximálnej veľkosti fragmentu
- Zobrazenie názvu súboru a absolútnu cestu, ako aj veľkosť a počet fragmentov
- Simulovať chybu, odoslaním jedného chybného fragmentu
- Oznámenia o správnosti alebo nesprávosti doručených fragmentov
- Možnosť odoslať 2MB súbor

Uzly musia vedieť prepínať medzi funkciou prijímača a vysielča bez nutnosti reštartovať program.

Návrh riešenia

Program bude ponúkať dva módy, medzi ktorými bude možné prepínať. Prvý mód bude mód odosielateľa, v tomto móde bude mať používateľ možnosť odosielať na prijímacie uzly súbory, ktoré si vyberie. Druhý mód bude mód prijímateľa, v tomto móde bude server, ktorý bude spracovávať prijaté dáta a príslušne na ne reagovať.

Štruktúra hlavičky

Hlavička bude pozostávať z 15 bajtov, prvé 4 bajty budú určovať checksum fragmentu, nasledujúce 4 bajty budú reprezentovať poradové číslo dátových fragmentov. Ďalšie 4 bajty určujú počet fragmentovaných paketov. Nasledujúce 2 bajty určujú dĺžku dát. Posledný použitý bajt reprezentuje druh fragmentu.

1B		2B		4B		8B			
8B	checksum				Poradové číslo				
16B	Celkový počet				Dĺžka		Kód		

Kontrolné pole - checksum (4B)

Kontrolná hodnota, v ktorej bude zapísaná kontrolná informácia vypočítaná z funkcie na základe algoritmu. Samotná hodnota poľa nebude zaradená do výpočtu checksumu.

Poradové číslo fragmentu (4B)

Hodnota podľa ktorej majú byť jednotlivé fragmenty zoradené a poskladané, v prípade chýbajúceho alebo poškodeného fragmentu je na základe tohto údaju možné vyžiadať len konkrétny fragment a nie je nutné opakovať celé odosielanie.

Celkový počet fragmentov (4B)

Na základe tejto informácie bude jasné koľko fragmentov treba za ideálnych podmienok poslať. Rovnako aj prijímací uzol bude vedieť, kedy prijal posledný fragment a môže bez problémov za prejdenia určitého času ukončiť nadviazanú komunikáciu.

Dĺžka fragmentu (2B)

Informácia o dĺžke fragmentu.

Identifikačný kód – Code (1B)

Pomocný kód, ktorý nesie informáciu o druhu fragmentu.

Druhy fragmentov:

- 0 – Fragment na nadviazanie spojenia
- 1 – Dátový fragment
- 2 – Fragment potvrdenia prijatia fragmentu
- 3 – Fragment s informáciou, že fragment s príslušným poradovým číslom dorazil chybný
- 4 – Fragment so správou pre prijímací uzol
- 5 – Fragment udržiavajúci spojenie platné
- 6 – Fragment, ktorý oznamuje ukončenie spojenia
- 7 – Požiadavka o zmenu role

Checksum

V programe bude spravená funkcia, ktorá na základe vstupných dát pomocou algoritmu crc vypočíta hodnotu checksum. V checksume budú zahrnuté všetky dáta a všetky polia hlavičky okrem poľa checksum.

CRC

Cyclic redundancy check, je algoritmus ktorý vypočíta kontrolnú hodnotu po prejdení všetkých bitov. Na vstup potrebuje zadaný surový vstup a polynóm, ktorý bude postupne xorovať a upravovať výsledné dáta.

V prípade zmeny akéhokoľvek bitu algoritmus vygeneruje inú hodnotu a program bude vedieť, že došlo ku poškodeniu dát, súčasný fragment zahodí a vyžiada si nové.

Automatic repeat request – ARQ

Metóda na kontrolovanie správnosti prenosu dát. Je zabezpečený potvrdzovaním prijatia dát. Odosielateľ odošle fragment a čaká na potvrdenie, ak potvrdenie nedostane, alebo uplynie stanovený limit tak správu odošle znova. V prípade nepotvrdenia preddefinovaného počtu fragmentov sa komunikácia ukončí.

Stop-and-Wait

Zastav a čakaj je metóda potvrdzovania kedy odosielateľ pošle dáta a čaká na ich potvrdenie. V okamihu ako vysielateľ obdrží potvrdenie prijatia fragmentu, je pripravený odoslať ďalší fragment v poradí, prípadne znovu poslať predchádzajúci fragment na základe obdržanej správy.

Udržanie spojenia

Klient odosiela správy typu „keep alive“ na server v pravidelných intervaloch pokiaľ sa nevykonáva iný dátový prenos. V prípade, ak server odpovie na správu a potvrdí jej zachytenie, časový interval klienta sa resetuje. Ak server neodpovie do stanoveného intervalu, klient skúsi udržať komunikáciu ešte 3 krát. Ak na žiadny z týchto rámcov nedostane odpoveď. Klient začne považovať spojenie za ukončené.

Server bude na správy typu „keep alive“ reagovať a pošle klientovi príslušnú informáciu, aby potvrdil prijatie rámca.

Spracovanie príliš veľkého súboru - fragmentovanie

V prípade príliš veľkého súboru alebo správy je nutné súbor rozdeliť medzi viaceré fragmenty. Rovnako na druhej strane je potrebné takéto fragment poskladať naspäť dohromady. Na tú budú vytvorené samostatné funkcie, ktorá podľa zadanej dĺžky fragmentu rozdelia dáta na príslušný počet fragmentov.

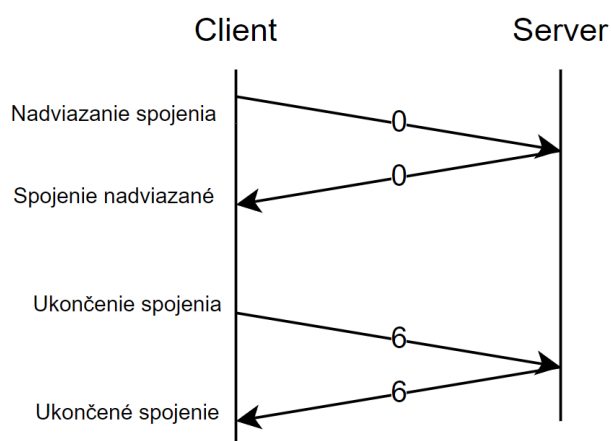
Druhá strana bude na základe hlavičiek vedieť zoradiť fragmenty a následne poskladať súbor do pôvodného stavu.

V prípade spracovania súboru sa v prvom fragmente odosiela názov súboru a až následne celý obsah.

Diagramy

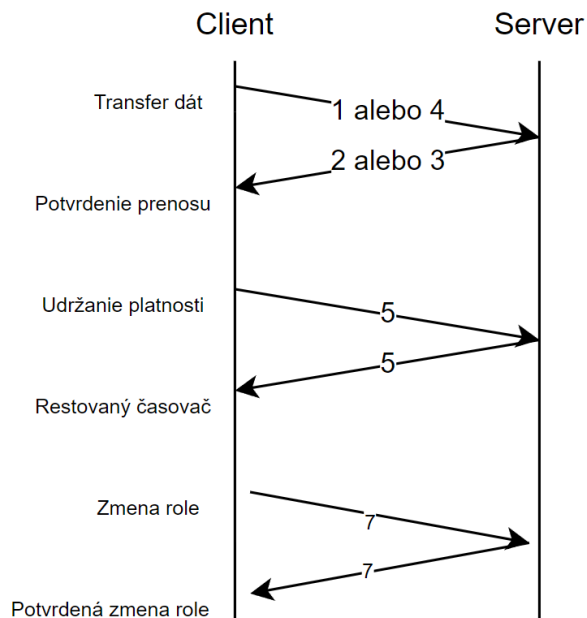
Nadviazanie spojenia

Špeciálne druhy správ 0 a 6 zabezpečujú nadviazanie a ukončovanie spojenia. Klient odošle požiadavku na server, ktorú následne server potvrdí a tým pádom je spojenie nadviazané. V prípade straty ktoréhokoľvek fragmentu sa klient opätovne pokúsi ukončiť spojenie, prípadne nadviazať.



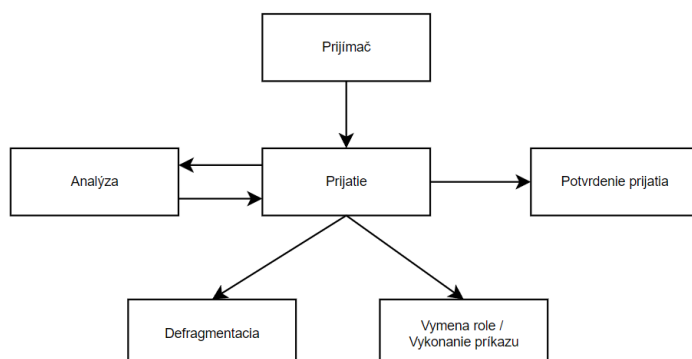
Prenos správ a dát

Ostatné typy sú určené na ostatné operácie medzi prijímačom a vysielačom, hlavnou z nich je prenos dát (1) a správ (4) a ich potvrdzovanie (2,3). Ďalší typ zabezpečuje udržanie spojenia „keep alive“ (5), odosielaný v pravidelných intervaloch pri neaktivite klienta. Posledným typom je špeciálna požiadavka na zmenu role klienta/servera. Server vždy potvrdí prijatie. V prípade straty potvrdzovacieho fragmentu klient odošle rovnaký fragment znova. Server naň znova podobne odpovie, nakoľko ešte neuplynuli hodnoty časovačov pre vykonanie príkazu a časovač bude resetovaný.



Fungovanie prijímača

Prijímač zabezpečuje potvrdzovanie prijatých správ, defragmentáciu fragmentovaných súborov a v neposlednom rade vykonávanie príkazov.



Fungovanie vysielača

Program vysielať využíva multithreading, aby bolo možné súčasne komunikovať so serverom a zároveň prijímať a odosielať dáta. Používateľ má možnosť plniť „buffer“, ktorý následne bude komunikačný thread odosielať a spracovávať.

