

Zadanie 2: Komunikácia s využitím UDP protokolu (EN)

Zadanie úlohy

Navrhните a implementujte program s použitím vlastného protokolu nad protokolom UDP (User Datagram Protocol) transportnej vrstvy sieťového modelu TCP/IP. Program 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 max. 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 znovuvyžiadanie chybných fragmentov, vrátane pozitívneho aj negatívneho potvrdenia. Po zapnutí programu, komunikátor automaticky odosiela paket pre udržanie spojenia každých 5s pokiaľ používateľ neukončí spojenie ručne. Odporúčame riešiť cez vlastne definované signalizačné správy a samostatné vlákno.

Program musí spĺňať nasledujúce požiadavky (minimálne):

1. Program musí byť implementovaný v jazykoch C/C++ alebo Python s využitím knižníc na prácu s UDP socket, skompilovateľný a spustiteľný v učebniach. Odporúčame použiť python modul socket, C/C++ knižnice `sys/socket.h` pre linux/BSD a `winsock2.h` pre Windows. Iné knižnice a funkcie na prácu so socketmi musia byť schválené cvičiacim. V programe môžu byť použité aj knižnice na prácu s IP adresami a portami: *arpa/inet.h* a *netinet/in.h*.
2. Program musí pracovať s dátami optimálne (napr. neukladať IP adresy do 4x int).
3. Pri posielaní súboru musí používateľovi umožniť určiť cieľovú IP a port.
4. Používateľ (stačí na strane vysieláča) musí mať možnosť zvoliť si max. veľkosť fragmentu a meniť ju dynamicky počas behu programu pred poslaním správy/súboru (neplatí pre pakety na udržanie spojenia).
5. Obe komunikujúce strany musia byť schopné zobrazovať:
 - a) názov a absolútnu cestu k súboru na danom uzle,
 - b) veľkosť a počet fragmentov vrátane celkovej veľkosti správy/súboru.
6. Možnosť simulovať chybu prenosu odoslaním minimálne 1 chybného fragmentu pri prenose správy a súboru (do dátovej časti fragmentu alebo do checksum je cielene vnesená chyba, to znamená, že

prijímajúca strana deteguje chybu pri prenose).

7. Prijímajúca strana musí byť schopná oznámiť odosielateľovi správne aj nesprávne doručenie fragmentov. Pri nesprávnom doručení fragmentu vyžiada znovu poslať poškodené dáta.
8. Možnosť odoslať 2MB súbor a v tom prípade ho uložiť na prijímacej strane ako rovnaký súbor, pričom používateľ zadáva iba cestu k adresáru kde má byť uložený.
9. Program musí byť organizovaný tak, aby oba komunikujúce uzly mohli prepínať medzi funkciou vysielajúca a prijímača bez reštartu programu automatizovane (jedna strana pošle správu na prepnutie, dostane ACK z druhej strany a uzly sa automaticky prepnú) - program nemusí (ale môže) byť vysieláč a prijímač súčasne.

Odvzdáva sa:

1. Návrh riešenia
2. Predvedenie riešenia v súlade s prezentovaným návrhom

Pri predvedení riešenia je podmienkou hodnotenia schopnosť doimplementovať jednoduchú funkcionality na cvičení.

Hodnotenie

Celé riešenie max. 20 bodov (min. 10), z toho:

- max. 4 bodov za návrh riešenia;
- max. 1 bod za doplnenú funkčnosť (doimplementáciu) priamo na cvičení v požadovanom termíne podľa harmonogramu cvičení. V prípade, ak študent nesplní úlohu zadanú priamo na cvičeniach, nehodnotí sa výsledné riešenie;
- max. 15 bodov za výsledné riešenie.

Návrh a zdrojový kód implementácie študent odovzdáva v elektronickom tvare do AISu do 4.12.2023 23:59.

Hodnotiacia tabuľka

| Označenie úlohy | Názov úlohy | Max bodov | Min bodov |
|-----------------|--|-----------|-----------|
| 1 | Návrh programu a komunikačného protokolu | 4 | 2 |
| 2 | Príprava | 0.5 | 7.5 |
| 3 | Nastavenie IP a port | 0.5 | |
| 4 | Prenos súboru menšieho ako nastavená veľkosť fragmentu | 2 | |
| 5 | Simulácia chyby pri prenose súboru a správy | 4 | |
| 6 | Prenos 2MB súboru | 2 | |

| Označenie úlohy | Názov úlohy | Max bodov | Min bodov |
|-----------------|---|-----------|-----------|
| 7 | Udržiavanie spojenia | 3 | |
| 8 | Finálna dokumentácia a kvalita spracovania | 3 | |
| 9 | Doplnená funkčnosť (doimplementácia) priamo na cvičení. | 1 | 0.5 |
| | Spolu | 20 | 10 |

Opis úloh v hodnotiacej tabuľke

Študent by mal byť vedieť filtrovať segmenty aplikácie pomocou nástroja Wireshark a vedieť identifikovať typy správ navrhnutého protokolu počas demonštrovania svojej aplikácie na cvičení.

Úloha 1 - Návrh programu a komunikačného protokolu

Hodnotí sa prehľadnosť a zrozumiteľnosť odovzdanej dokumentácie ako aj kvalita navrhovaného riešenia. 5 bodov získa študent, ktorý má v dokumentácii uvedené všetky podstatné informácie o fungovaní jeho programu. Korektne navrhnutú štruktúru hlavičky vlastného protokolu, opis použitej metódy kontrolnej sumy a fungovania ARQ, metódy pre udržanie spojenia, diagram spracovávania komunikácie na oboch uzloch (sekvenčný), popis jednotlivých častí zdrojového kódu (knížnice, triedy, metódy, ...). Podčiarknuté požiadavky sú minimálne.

Úloha 2 - Príprava

Nastaviť a overiť konektivitu medzi 2 uzlami, spustiť na oboch uzloch Wireshark. Spustiť zachytávanie vo Wireshark a nastaviť filter na zobrazenie iba komunikácie vlastného programu. Každý prenos sa kontroluje aj cez Wireshark.

Úloha 3 - Nastavenie IP a port

Body získa študent, ktorého program umožňuje nastaviť na prijímajúcom uzle port, na ktorom počúva a na vysielajúcom uzle IP adresu a port prijímača.

Úloha 4 - Prenos súboru menšieho ako nastavená veľkosť fragmentu

Body získa študent, ktorého program umožňuje úspešne preniesť súbor menší ako je nastavená veľkosť fragmentu podľa pokynov cvičiaceho.

Úloha 5 - Simulácia chyby pri prenose súboru a správy

1 body získa študent, ktorého program umožňuje úspešne preniesť súbor/správu pri simulácii chyby prenosu a má korektne implementovanú detekciu chyby a ARQ metódu. Korektným použitím komplexnejšej ARQ metódy je možné získať ďalší 3b. Príklady ARQ metód su medzi zdrojmi nižšie. Vaším návrhom na zlepšenie sa nebránime. Jednotlivé metódy sú hodnotené nasledovne:

- Stop & wait + 0b (vrátane jedného ACK pre skupinu segmentov)

- Go Back-N + 1b
- Selective Repeat + 2b
- Vylepšená Go Back-N/Selective Repeat + 3b (dynamický window sliding, optimalizácia počtu ACK správ,)

Úloha 6 - Prenos 2MB súboru

Body získa študent, ktorého program umožňuje úspešne preniesť súbor s veľkosťou 2MB pri nastavení veľkosti fragmentu podľa pokynov cvičiaceho a uložiť ho ako rovnaký súbor, zobrazuje absolútnu cestu k súboru a počet fragmentov spolu s ich veľkosťou.

Úloha 7 - Udržiavanie spojenia

2 body získa študent, ktorého program po prenesení súboru udržiava spojenie pomocou vlastných signalizačných správ a zobrazí informáciu, ak bolo spojenie prerušené. Korektným použitím komplexnejšej metódy pre udržanie spojenia je možné získať ďalší 1b.

Úloha 8 - Finálna dokumentácia a kvalita spracovania

Hodnotí sa prehľadnosť a zrozumiteľnosť odovzdanej dokumentácie ako aj kvalita pracovania celového riešenia. 3 body získa študent, ktorý má v dokumentácii uvedené všetky podstatné informácie o fungovaní jeho programu vrátane zmien, ktoré nastali v implementácii oproti návrhu. Dokumentácia tiež musí obsahovať aspoň 1 ukážku testovacieho scenáru (ideálne ako screeny z Wireshark).

Úloha 9 - Doplnená funkčnosť (doimplementácia) priamo na cvičení.

1 bod získa študent, ktorý doimplementuje úlohu v jej plnom rozsahu a predvedie jej funkčnosť bez toho, aby program padal alebo vyhadzoval akékoľvek chybové hlášky súvisiace s touto úlohou.

Literatúra

1. Stop & Wait - [Sliding Window Protocol](#)
2. [Selective repeat](#)
3. [Go Back-N](#)