SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta informatiky a informačných technológií

ZADANIE 2 – KOMUNIKÁCIA S VYUŽITÍM UDP PROTOKOLU

Dávid Kromka
Cvičenie: Piatok 8:00
9.12.2021

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

Obsah

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

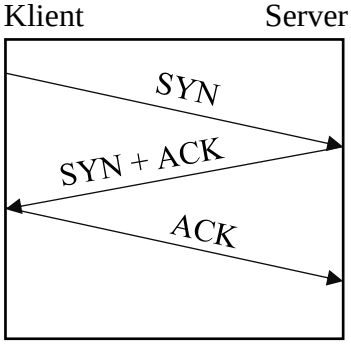
1. Úvod

Cieľom zadania bolo vytvorenie programu na komunikáciu s využitím protokolu UDP a nad ním vytvorenie vlastného protokolu. Program dokáže prenášať textové správy a súbory do veľkosti 2 MB medzi 2 uzlami. Prijímajúca strana je schopná oznámiť odosielateľovi prijatie správy a jej správnosť. Používateľ si môže zvoliť maximálnu veľkosť fragmentu a cestu k uloženiu súboru. Obidva uzly sú schopné vypísať absolútnu cestu k súboru, jeho veľkosť, veľkosť fragmentov ako aj ich počet. Je zabezpečené simulovanie chyby na overenie fungovania potvrdzovania správnosti prijatých správ. Dokumentácia postupne prechádza predstavením implementácie, štruktúry a fungovania programu. Následne je predstavené grafické rozhranie, návrh štruktúry hlavičky s vysvetlením využitia jednotlivých častí, opis použitia metódy kontrolnej sumy s algoritmus na jej vytvorenie a fungovanie ARQ metódy. Súčasťou návrhu sú aj grafické znázornenia významných častí. Ďalšou časťou je opis metódy na udržanie spojenia, testovanie, záver a diagram.

2. Štruktúra programu

Program je implementovaný v jazyku Python verzia 3.9.7 s využitím modulov Socket, Threading, Tkinter a Time. Je organizovaný podľa MVC (model, view, controller) vzoru. Komunikácia prebieha spôsobom peer to peer, medzi sebou komunikujú dva rovnaké programy. Program sa skladá z odosielajúcej časti a prijímajúcej časti, ktorá je spustená v threade.

Odosielajúca časť má dve základné úlohy, prijímať vstup od používateľa a odosielať ho druhému uzlu v lokálnej sieti a porovnávať potvrdzovacie správy od druhého uzla s odoslanou správou. Potvrdzovacia správa môže hovoriť o správnom prijatí rámca alebo nesprávnom, kedy je daná správa odoslaná znova. Po otvorení programu je potrebné zadať IP adresu a port cieľového uzla a IP adresu a port, na ktorom má súčasný uzol počúvať. Taktiež používateľ zadáva veľkosť fragmentu. Používateľ má k dispozícii grafické rozhranie vytvorené modulom Tkinter. Modul socket zabezpečuje nadviazanie spojenia a prenos dát na transportnej úrovni a nižšej. Prvá sa spúšťa prijímajúca časť a následne je možné spustiť odosielajúcu časť, kedy sa nadviaže spojenie prostredníctvom 3-way handshake s použitím flagov SYN a ACK.



10 2.54717487	192.168.0.179	224.0.0.251	MDNS	123 Standard query 0x0000 A	
11 2.761855972	192.168.0.189	192.168.0.186	UDP	60 53545 → 65432 Len=11	
12 2.772284454	192.168.0.186	192.168.0.189	UDP	53 44814 → 65432 Len=11	
13 2.783243728	192.168.0.189	192.168.0.186	UDP	60 53545 → 65432 Len=11	

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

Po inicializácii je možné začať odosielať správy a súbory, ktoré sú prekonvertované na typ bytes. Ukladajú sa do poradia, v akom boli zadané a pripravia sa na odoslanie.

Správy zo vstupu musia byť menšie alebo rovné maximálnej veľkosti fragmentu, v opačnom prípade sú rozdelené do fragmentov a odoslané po častiach, naspäť sa zložia na prijímajúcom uzle. Maximálna veľkosť fragmentu je taká, aby sa fragment na nižších vrstvách už nerozdeľoval na fragmenty. Maximálna veľkosť prenesená v Ethernet rámci je 1518 B, hlavička a päta tvorí 18 B, ostáva 1500 B. Odčítaním hlavičky IP protokolu (20 B) a hlavičky UDP protokolu (8 B), ostáva veľkosť 1472 B. Veľkosť hlavičky nášho protokolu je 11 B, z čoho vyplýva, že maximálna veľkosť fragmentu je 1461 B.

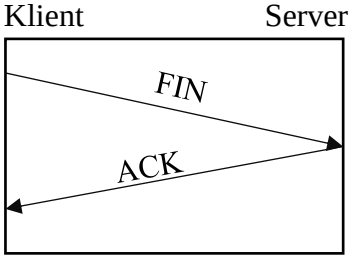
Jednotlivým správam je pridelená hlavička, ktorá nesie informácie o dátach a jednoznačne ich identifikuje. Bližšie je opísaná v časti 4. Štruktúra hlavičky. Nad každou dátovou časťou sa pred odoslaním vykoná crc funkcia, ktorej hodnota je uložená v hlavičke a je bližšie opísaná v časti 5. Metóda kontrolnej sumy.

Každému rámcu sa priradí číslo rámca postupne od 0 do 224 -1 . Po prekročení najvyššej hodnoty sa pokračuje od 0. V prípade že ide o správu alebo súbor rozdelený na fragmenty, všetky fragmenty budú mať nastavený flag FRA a posledný bude mať nastavený flag END. Pri odosielaní súboru bude v prvom odoslanom rámci pred rámcami s dátami súboru v dátovej časti jeho názov a typ.

Na strane servera sa budú prijaté rámce kontrolovať crc funkciou, spájať fragmenty podľa čísla rámcov a vypisovať. Pri prijatí súboru je možné zvoliť miesto uloženia. Server odosiela potvrdzujúce správy o prijatí a správnosti prijatých dát s ACK a NACK flagom, kde číslo rámca je zhodné s číslom rámca, na ktorú sa odpoveď viaže. Klient odosiela opäť tie rámce, na ktoré prišla odpoveď s NACK flagom alebo odpoveď neprišla. Je použitá metóda arq stop and wait, takže ďalší rámec bude odoslaný až po potvrdení správneho prijatia predchádzajúceho rámca.

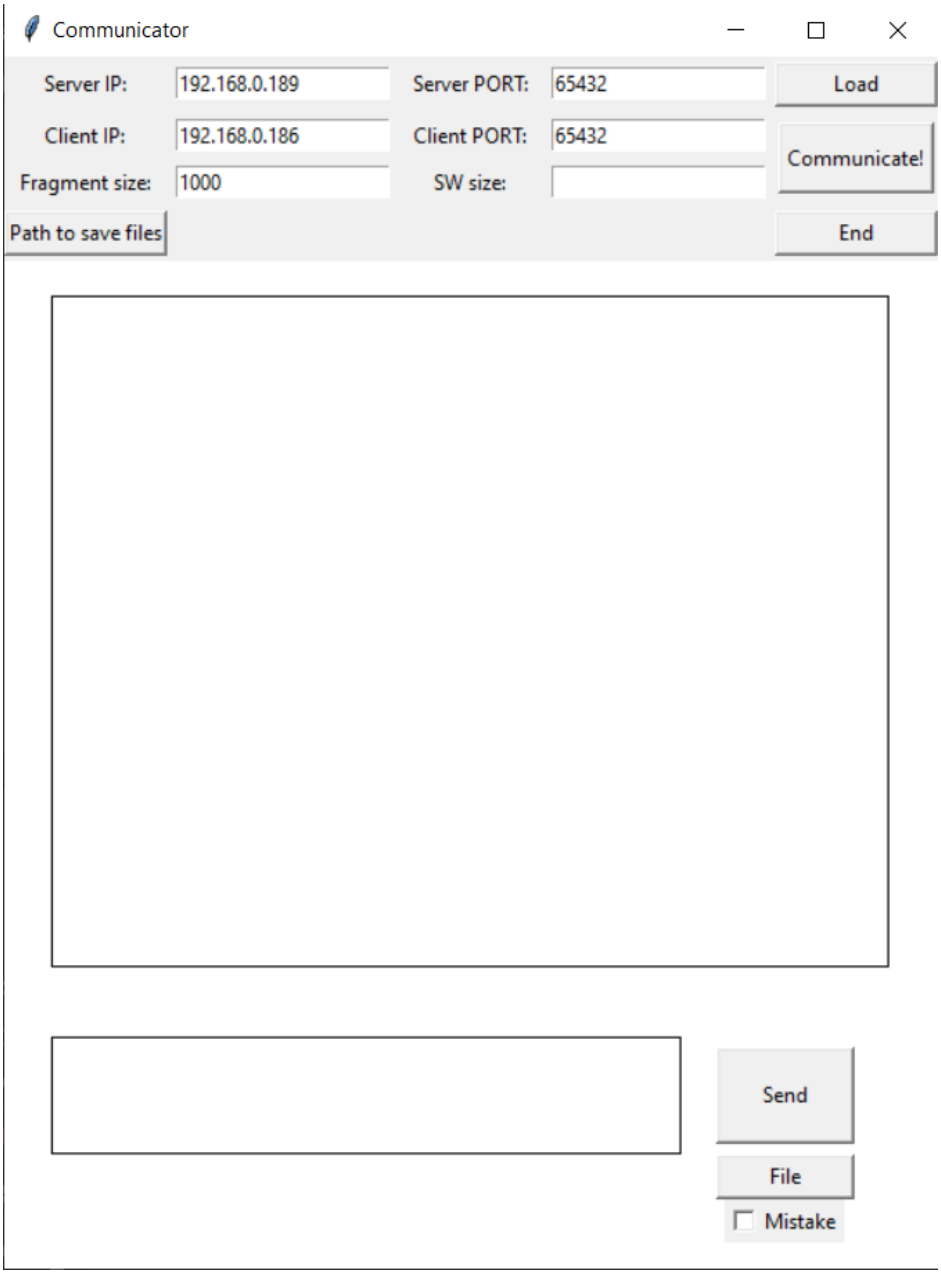
Ak práve komunikácia neprebieha, spustí sa metóda na udržanie spojenia popísaná v časti 6. Metóda pre udržanie spojenia.

Ukončenie spojenia môžu inicializovať obidve strany pomocou flagu FIN od inicializujúcej strany, ktorá očakáva prijatie potvrdzujúcej správy s flagom ACK a až tak je spojenie ukončené.



Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

3. Grafické rozhranie a ovládanie



Používateľ má k dispozícii grafické rozhranie s jednoduchým ovládaním:

1. Po zapnutí programu používateľ zadá IP adresu a port serveru, na ktorom sa budú prijímať správy, IP adresu a port klienta, to je uzol, na ktorý sa budú odosielať správy, veľkosť fragmentu a cestu, kde sa budú ukladať prijaté súbory. Zadanie cesty na ukladanie súborov nie je povinné a cesta sa dá určiť priamo pri prijímaní súboru. Tieto vstupy sú potvrdená tlačidlom „Load“, ktoré zabezpečí spustenie počúvania prijímacej časti programu.

2. Stlačením tlačidla „Communicate!“ sa začne komunikácia a vykoná 3-way handshake.

3. Ako vstupy na správy slúži okno v dolnej časti pri tlačidle „Send“, ktoré slúži na odoslanie správy. Tlačidlo „File“ slúži na výber a odoslanie súboru. Prijaté a odoslané správy sa zobrazia vo veľkom hlavnom okne.

4. V prípade označenia checkboxu „Mistake“, bude každá druhá správa odoslaná s chybou v dátovej časti a bude musieť byť odoslaná znova.

5. Tlačidlo „End“ slúži na ukončenie komunikácie.

4. Štruktúra hlavičky

Hlavička vlastného protokolu má veľkosť 11 B a pozostáva zo 4 častí. Prvé 3 bajty sú požité na číslovanie rámcov seq. Ďalšie 4 bajty má pole na kontrolnú sumu z funkcie crc, ktorá je popísaná v časti 4. Metóda kontrolnej sumy. Pole pre flags má veľkosť 1 B a 3 B má pole veľkosť odosielanej správy. Spolu tvorí hlavičku každej správy 11 B.

Rámec môže mať flagy SYN, ACK, NACK, FRA, END, FIN, UP, TYP. Flag SYN je použitý na inicializáciu komunikácie. Flag ACK je použitý na úspešné potvrdenie prijatia rámca, flag NACK na neúspešné potvrdenie prijatia rámca. Flag FRA majú všetky rámce, ktoré nesú dáta fragmentu a budú poskladané z viacerých rámcov, posledný z nich má aj flag END. Flag UP sa používa pri správach na udržanie spojenia, ktoré je bližšie opísané v časti 6. Metóda pre udržanie spojenia. Ak je flag TYP inicializovaný, znamená to že typ posielanej správy je súbor a ak nie je inicializovaný, ide o textovú správu. Flag FIN znamená správu o ukončení spojenia.

00000001 - SYN
00000010 - ACK
00000100 - NACK
00001000 – FRA
00010000 - END
00100000 - UP
01000000 - TYP
10000000 – FIN

Grafické znázornenie štruktúry hlavičky s veľkosťou jednotlivých častí v bitoch:

1 24 56 64 88

Číslo rámca	Kontrolná suma	Flagy	Veľkosť

Za hlavičkou pokračuje dátová časť, jej maximálna veľkosť je 1461 bajtov.

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

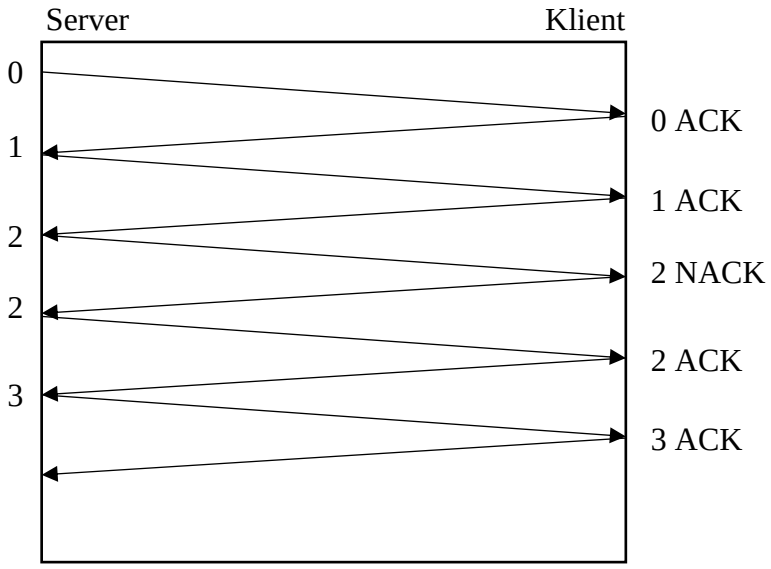
5. Metóda kontrolnej sumy

Metóda kontrolnej sumy je vykonávaná nad dátami pred ich odoslaním klientom a je zapísaná v hlavičke v poli checksum. Po prijatí na serveri sa znova vykoná metóda kontrolnej sumy a porovná sa výsledok s hodnotou v poli checksum v hlavičke. Ak sa hodnoty zhodujú, odošle sa rámec s flagom ACK a ak sa nezhodujú, s flagom NACK – dáta sú poškodené alebo zmenené. Pri správach bez dátovej časti je hodnota check v hlavičke 0.

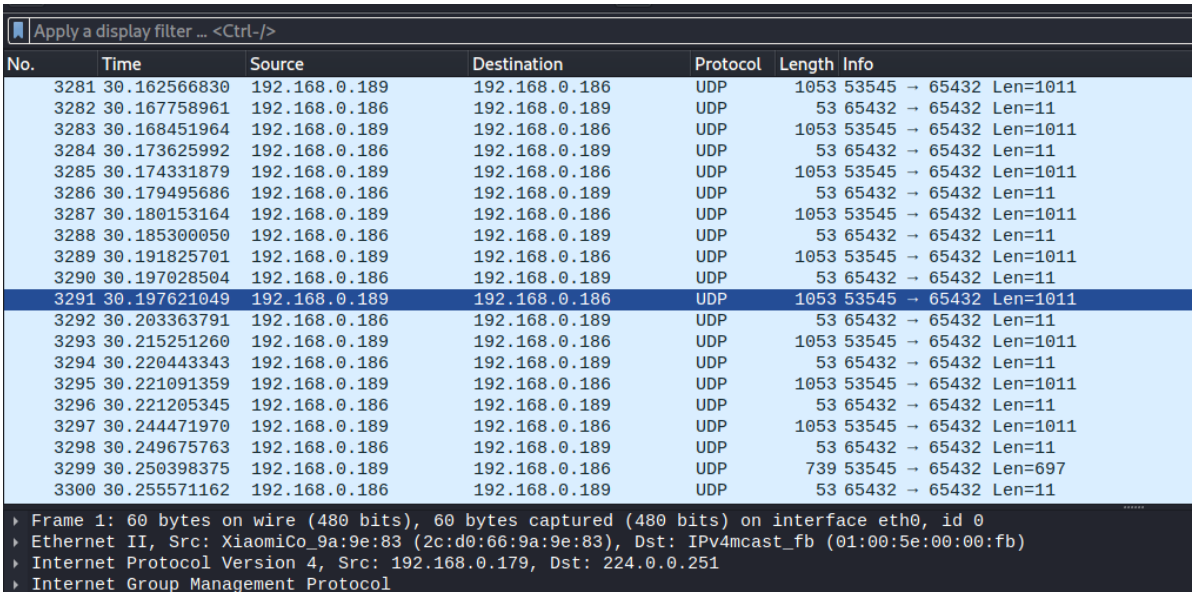
V projekte je implementovaná metóda kontrolnej sumy crc32 z knižnice Zlib. Argumentom funkcie je správa alebo súbor vo formáte bytes a výsledkom je kontrolný súčet, ktorý je vo formáte unsigned 32-bit integer. Táto hodnota je výsledkom po matematických operáciach delenia polynomom cez xor vykonaných nad vstupom funkcie, teda odosielanými dátami.

6. Fungovanie ARQ

Na kontrolu a riešenie chýb pri prenose dát je v projekte využitá metóda Stop and wait ARQ. Pri odosielaní správ je do hlavičky vložený výsledok kontrolnej sumy vytvorený z dátovej časti rámca. Pri rámcoch bez dátovej časti je na tomto mieste 0. Po prijatí správy prijímajúcim uzlom je crc funkcia vykonaná znova a výsledok porovnaný s hodnotou v hlavičke. Ak sa zhoduje, je odoslaná potvrdzujúca správa s flagom ACK a poradovým číslom rovnakým ako poradové číslo potvrdenej správy a odosielajúca časť môže odoslať ďalšiu správu v poradí. Ak pri odosielaní nastane chyba a kontrolné sumy sa nezhodujú, je odoslaná správa s flagom NACK. V takomto prípade je správa odoslaná znova a ďalšia správa môže byť odoslaná až po prijatí potvrdenia správneho prijatia predtým poškodenej správy.



Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00



Na obrázku z wiresharku je možné vidieť potvrdzovanie rámcov a tiež veľkosti hlavičky a fragmentov.

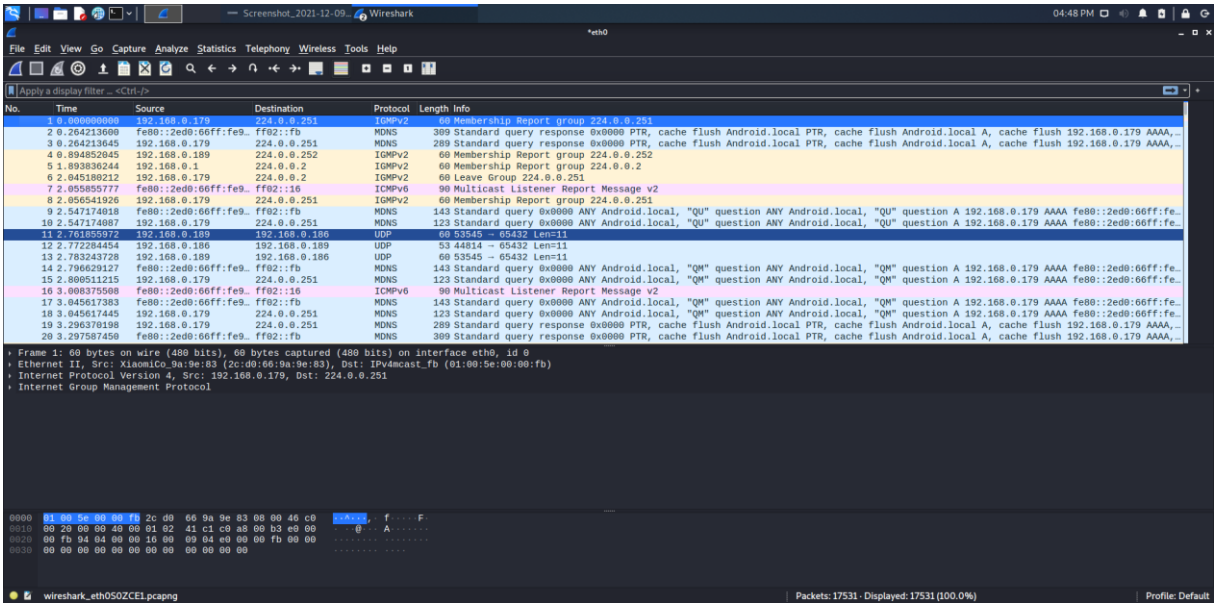
7. Metóda pre udržanie spojenia

Pre udržanie spojenia medzi obidvomi komunikujúcimi uzlami sa používa správa bez dátovej časti, zložená z hlavičky s flagom UP, ktorý je bližšie opísaný v časti 4. Štruktúra hlavičky. Takáto správa na udržanie spojenia sa odosiela každých 10 sekúnd, ak neprebieha žiadna iná komunikácie medzi uzlami. Na správu sa očakáva odpoveď správou bez dátovej časti s flagom ACK. V prípade, že odpovede neprichádzajú na po sebe idúce správy na udržanie komunikácie, spojenie je ukončené a signalizačné správy sa neodosielajú. V takomto prípade je potrebné znova inicializovať komunikáciu.

8. Testovanie

Testovanie bolo uskutočnené pri komunikácii medzi zariadením s operačným systémom Windows 10 Pro a virtuálnym strojom s operačným systémom Kali Linux bežiacom na danom zariadení. Obidva zariadenia majú svoju vlastnú IP adresu, sú schopné medzi sebou nadviazať spojenie, udržiavať ho, prenášať textové správy, súbory a cieľový uzol potvrdzuje správne alebo nesprávne prijatie správy, kedy je daná správa poslaná opätovne.

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00



9. Záver

Program je implementovaný zároveň ako odosielajúci aj prijímajúci uzol, používateľovi je umožnené zadať IP adresu a port cieľového zariadenia a IP adresu a port na ktorom prijímajúca časť počúva. Posielať sa dajú textové správy a súbory s veľkosťou 2MB s možnosťou výberu veľkosti fragmentu. Pri posielaní súborov je na cieľovom uzle možné vybrať miesto uloženia súboru. Program funguje bez nutnosti reštartovať ho a spĺňa požiadavky plynúce zo zadania. Pri prenášaní súborov väčších ako 2 MB je čas prenosu dlhší, viac ako 15 sekúnd. Dlhší čas je pravdepodobne spôsobený ARQ metódou stop and wait.

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Janeba, piatok 8:00

10. Diagram