**Komunikácia s využitím UDP protokolu**

*Počítačové a komunikačné siete*

[xfridrichj@stuba.sk](mailto:xfridrichj@stuba.sk)

Obsah

[Zadanie 2](#_Toc180368088)

[Návrh 2](#_Toc180368089)

[Druhy správ a Flagy 2](#_Toc180368090)

[Hlavička 2](#_Toc180368091)

[Vývojový diagram 4](#_Toc180368092)

[Použité metódy 4](#_Toc180368093)

[Záver 5](#_Toc180368094)

## Zadanie

Navrhnúť a implementovať P2P aplikáciu využívajúcu vlastný protokol postavený nad UDP (User Datagram Protocol) v transportnej vrstve sieťového modelu TCP/IP. Aplikácia bude umožňovať komunikáciu dvoch účastníkov v lokálnej Ethernet sieti, vrátane výmeny textu a prenosu ľubovoľných súborov medzi počítačmi (uzlami). Oba uzly budú fungovať súčasne ako prijímač aj odosielateľ.

Zadanie pozostáva z dvoch častí: teoretickej a praktickej. V teoretickej časti navrhnete vlastný protokol nad UDP. V praktickej časti následne implementujete a predvediete jeho funkčnosť pri komunikácii medzi dvoma uzlami, pričom zdôrazníte úpravy vykonané počas implementácie v porovnaní s pôvodným návrhom.

## Návrh

### **Druhy správ a Flagy**

|  |  |
| --- | --- |
| **Pozícia bitu** | **Význam Flagu** |
| 0000 0001 | ACK signál |
| 0000 0010 | SYN signál |
| 0000 0100 | FIN signál |
| 0000 1000 | Signál opakovaného poslania |
| 0001 0000 | Bežná správa |
| 0010 0000 | Súborová správa |

### **Hlavička**

Navrhnutá hlavička má 8B:

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

Automaticky generovaný popis

**Message type + Flags –** výber z uvedených vyššie - kombinovaná veľkosť 1B

**Payload size –** veľkosť odoslaného fragmentu v bitoch (0-65 536) - 2B

**Fragment offset –** indikácia štartovného bitu fragmentovaných dát (0-65 536) - 2B

**Total Fragments –** celkový počet odosielaných fragmentov (0-255) – 1B

**Checksum –** zvyšok po delení 65 536 – 2B

**Data –** samotná správa

## Vývojový diagram

**Obrázok, na ktorom je diagram, text, rad, písmo

Automaticky generovaný popis**

## Použité metódy

*Trieda Header:*

**build\_packet(self)** – poskladá jednotlivé zložky a samotnú správu do jedného packetu

**parse\_packet(packet)**– rozoberie packet na jednotlivé zložky a vráti ich oddelené

*Trieda Peer:*

Táto trieda slúži ako základ pre objekty Klient a Server.

**\_\_init\_\_(self, ip, port)**- inicializácia hodnôt

**send\_message(self, message, receiver=None)** – slúži na odoslanie SYN, SYN-ACK, ACK správ

**receive\_packet(self)** – pomocou threadovania príjme správu, rozoberie ju na zložky a printne ju užívateľovi

**send\_packet(self)** – odošle poskladaný packet používateľom zvolenému Peer-ovi

**chatting(self)** – prebieha tu threadovanie a používa sa na posielanie správ P2P

**quit(self)** – metóda na ukončenie threadovania

*Trieda Client (Peer):*

**three\_way\_hs\_c(self, server\_ip, server\_port)** – klientova verzia *3 way handshake*-u, odošle SYN signál, očakáva SYN-ACK signál a odošle ACK signál

*Trieda Server (Peer):*

**three\_way\_hs\_s(self)** – serverová verzia *3 way handshake*-u, hneď po spustení čaká na SYN signál, odošle SYN-ACK klientovi a zapamätá si klientove IP a port, čaká na ACK signál

## Záver