

**Slovenská technická univerzita**  
**Fakulta informatiky a informačných technológií**

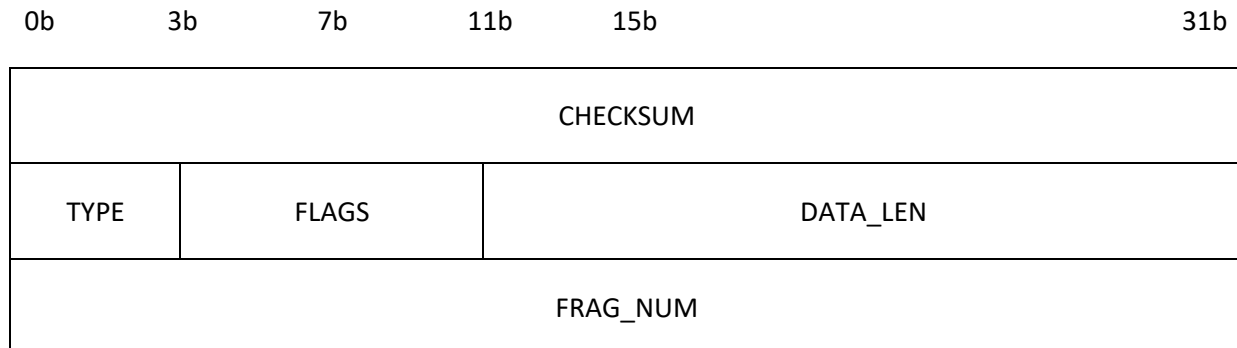
# **Dokumentácia k zadaniu 2**

## **Komunikácia s využitím UDP protokolu**

Meno: Dominik Mifkovič

Obdobie: ZS 2022/2023

## Hlavička vlastného protokolu

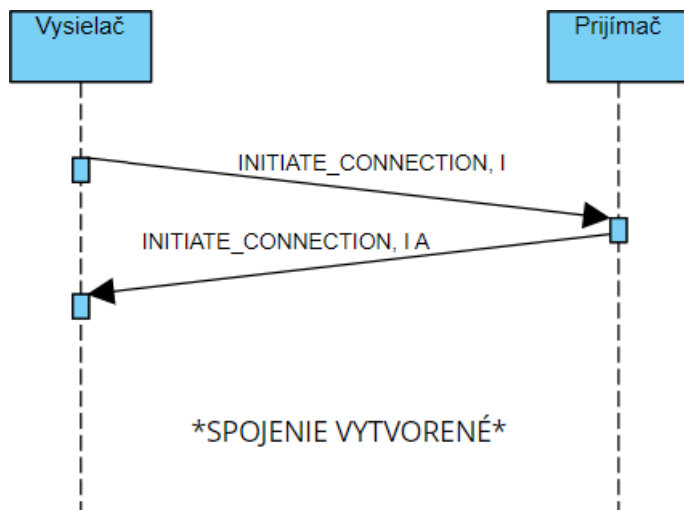


- CHECKSUM – 4B (je potrebných 32 bitov pretože checksum bude vypočítaný pomocou crc32 funkcie)
- TYPE – 4b
  - 0001 – packet bez dát
  - 0010 – packet pre posielanie textu
  - 0100 – packet pre posielanie súborov
- FLAGS – 1B
  - I – Initiate connection
  - A – Acknowledge
  - E – End connection
  - K – Keep alive
  - N – New data stream
  - R – Request again (odstránené, ale stále má políčko FLAGS 1B kvôli tomu aby nebolo nutné prerábať logiku programu)
  - C – Data stream complete
  - S – Swap mode (receiver/sender)
- DATA\_LEN – 20b, dĺžka pola pre dáta
- FRAG\_NUM – 4B, poradie daného fragmentu

Hlavička protokolu bola pozmenená tak, že bola odobratá časť FRAG\_TOTAL, ktorá bola redundantná a zbytočná. Netreba posilať v každom packete totálny počet fragmentov. Taktiež bol odstránený flag R. Keďže používam systém Stop and wait, tak jednoducho ignorujem poškodený packet na strane prijímača a vysielateľ situáciu vyhodnotí ako keby packet nedorazil a pošle ho znovu.

## Nadviazanie spojenia

Nadviazanie prvého spojenia prebehne v podobe 2-way handshake-u. Najskôr predom nastavený vysielateľ odošle packet bez dát čiže políčko TYPE bude mať hodnotu 0. Tu sa nastaví flag I (initiate connection) a packet sa odošle prijímateľovi. Po úspešnom prijatí prijímateľ odošle späť packet toho istého typu ale s flagmi I a A (Initialize, Acknowledge). Keď vysielateľ tento packet prijme, spojenie bolo nadviazané.



Prijímač

```
1. - Client
2. - Server
2
This server ip: 127.0.0.1
Enter server port:
42
Server is listening at port 42...
('127.0.0.1', 57769) connected.
```

Vysielač:

```
1. - Client
2. - Server
1
Server ip:
127.0.0.1
Server port:
42
Connection request sent to 127.0.0.1 at port 42. Awaiting confirmation...
Connection successful

1. - Send message
2. - Send file
3. - Swap modes
```

Ukážka na loopbacku:

127.0.0.1	127.0.0.1	UDP	45 50286 → 42 Len=13
127.0.0.1	127.0.0.1	UDP	45 42 → 50286 Len=13

```
00000010 00000000 00000000 00000000 01000101 00000000 00000000 00101001
01111110 10111001 00000000 00000000 10000000 00010001 00000000 00000000
01111111 00000000 00000000 00000001 01111111 00000000 00000000 00000001
11101110 10111101 00000000 00101010 00000000 00010101 00000110 11011011
11110100 11011011 11011111 00100010 00001000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00110000
```

Flag je nastavený na 1000000 (I) čo znamená signalizáciu že vysielateľ chce zahájiť spojenie.

```
00000010 00000000 00000000 00000000 01000101 00000000 00000000 00101001
01111110 10111010 00000000 00000000 10000000 00010001 00000000 00000000
01111111 00000000 00000000 00000001 01111111 00000000 00000000 00000001
00000000 00101010 11101110 10111101 00000000 00010101 00110010 11011100
11110100 11011011 11011111 00100001 00001100 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
```

Flag je nastavený na 11000000 (IA) čo znamená signalizáciu že prijímač spojenie prijal a spojenie sa začalo.

Takýmto spôsobom sa posielajú packety vo všetkých ostatných interakciách. Vždy so svojimi príslušnými flagmi.

## Udržiavanie spojenia

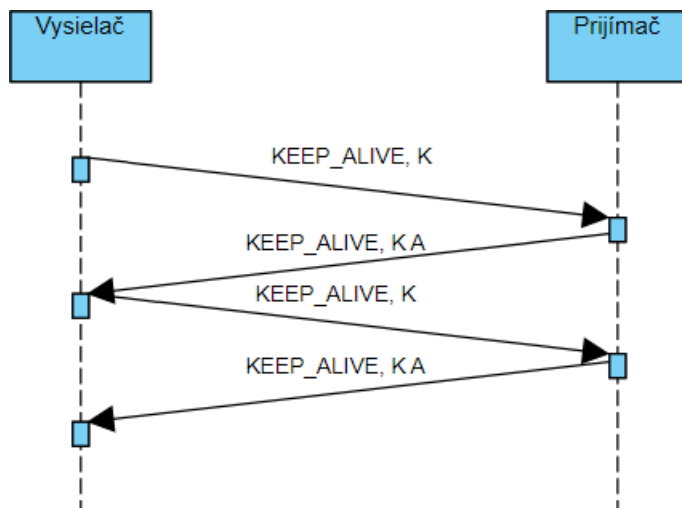
Udržiavanie spojenia bude prebiehať jednoducho. Vysielateľ najskôr odošle packet typu 0 s flagom K (Keep alive). Po úspešnom prijatí prijímateľ odošle späť packet toho istého typu ale s flagmi K a A (Keep alive, Acknowledge). Tento proces sa opakuje.

Vysielateľ:

```
Keep alive request sent to 127.0.0.1 at port 42. Awaiting confirmation...
Keep alive request successful.
```

Prijímač:

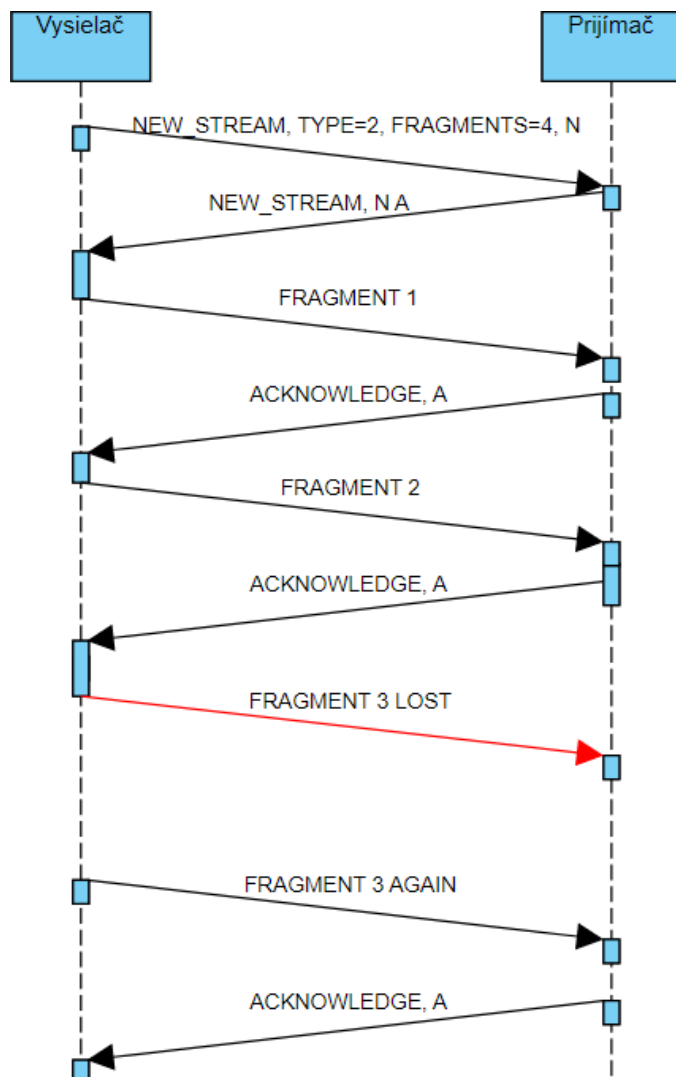
```
Keep alive request arrived from ('127.0.0.1', 53515). Sending confirmation...
```

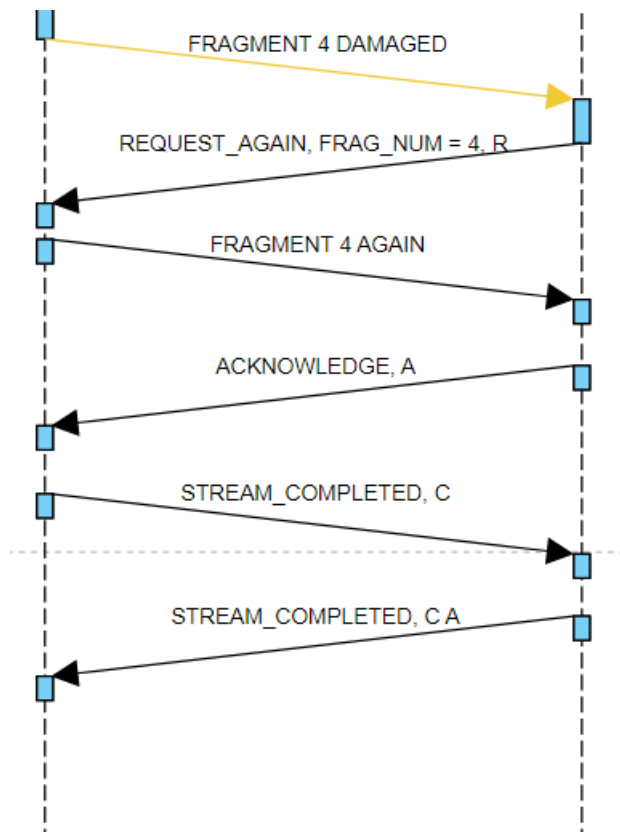


## Odosielanie dát

Pri posielaní dát najskôr vysielač oznámi, že chce začať nový dátový tok. To spraví tak, že pošle prijímateľovi packet s potrebným typom (posielanie textu – 2, posielanie súboru – 7), a flagom N (New data stream). Prijímateľ odošle späť packet s flagmi N a A. Ak vysielač úspešne prijme tento packet, prenos sa začne. Ak je veľkosť dát väčšia ako predvolená veľkosť fragmentu, dáta sa rozdelia do fragmentov a posielajú sa postupne. Fragment s dátami nemá žiadny flag. Obsahuje len typ, fragment dát, svoje poradie a celkový počet fragmentov daného toku dát. Odosielanie bude fungovať systémom Stop & Wait. Vysielač odošle fragment a čaká až prijímateľ pošle späť potvrdenie o prijatí fragmentu. Ak prijímateľ nepošle potvrdenie do nejakého časového intervalu, vysielač pošle fragment znova. Ak fragment dorazí poškodený, t.j. hodnota checksum nebude sedieť, prijímateľ pošle vysielaču požiadavku na znovu-poslanie daného fragmentu. To spraví tak, že pošle packet s flagom R (Request again) a nastaveným poradovým číslom fragmentu. Vysielač následne pošle fragment znova. Ak však ku chybe nedôjde, vysielač pošle packet s flagom A. Toto sa opakuje kým sa nepošlú všetky fragmenty. Ak sa posiela súbor, pošlú sa dodatočné fragmenty, ktoré indikujú typ a meno súboru. Po dokončení odosielania vysielač odošle posledný packet typu 0 s flagom C (Data

stream complete). Prijímateľ v zápäťí odošle packet toho istého typu s flagmi C a A. Prenos sa ukončí a spojenie sa udržiava.





```

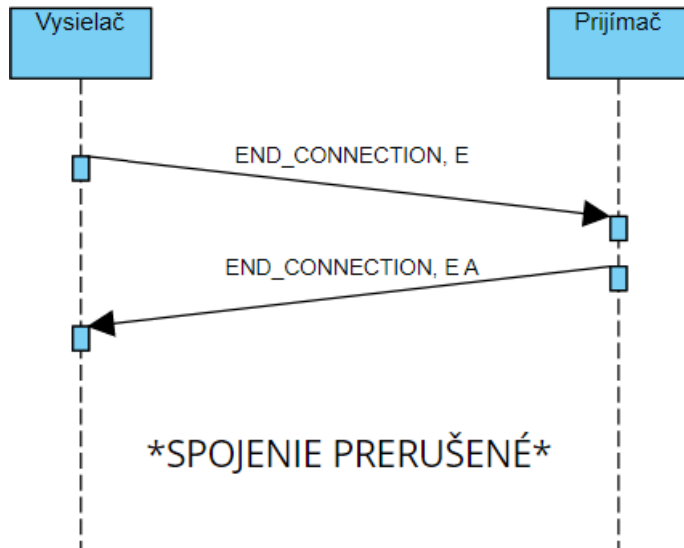
Enter path to the file:
test.txt
Specify fragment size in bytes (min 1 / max 1400):
100
Sending file C:\Users\domin\Desktop\test.txt
Sending packet 1
Sending packet 2
Sending packet 2 again
Sending packet 3
Sending packet 4
Sending packet 4 again
Sending packet 5
Sending packet 6
Sending packet 7
Sending packet 8
Sending packet 9
Sending packet 10
Sending packet 11
Sending packet 12
Sending packet 13
Sending packet 14
Sending packet 15
Sending packet 16
Sending packet 17
Sending packet 18
Sending packet 19
1. - Send message

Keep alive request arrived from ('127.0.0.1', 53515)
Keep alive request arrived from ('127.0.0.1', 53515)
Keep alive request arrived from ('127.0.0.1', 53515)
Keep alive request arrived from ('127.0.0.1', 53515)
Keep alive request arrived from ('127.0.0.1', 53515)
Keep alive request arrived from ('127.0.0.1', 53515)
File incoming! Fragment size: 100
Received Packet: 1 from ('127.0.0.1', 53515)
Received Packet: 2 from ('127.0.0.1', 53515)
Received Packet: 3 from ('127.0.0.1', 53515)
Received Packet: 4 from ('127.0.0.1', 53515)
Received Packet: 5 from ('127.0.0.1', 53515)
Received Packet: 6 from ('127.0.0.1', 53515)
Received Packet: 7 from ('127.0.0.1', 53515)
Received Packet: 8 from ('127.0.0.1', 53515)
Received Packet: 9 from ('127.0.0.1', 53515)
Received Packet: 10 from ('127.0.0.1', 53515)
Received Packet: 11 from ('127.0.0.1', 53515)
Received Packet: 12 from ('127.0.0.1', 53515)
Received Packet: 13 from ('127.0.0.1', 53515)
Received Packet: 14 from ('127.0.0.1', 53515)
Received Packet: 15 from ('127.0.0.1', 53515)
Received Packet: 16 from ('127.0.0.1', 53515)
Received Packet: 17 from ('127.0.0.1', 53515)
Received Packet: 18 from ('127.0.0.1', 53515)
Received Packet: 19 from ('127.0.0.1', 53515)
File saved at C:\Users\domin\Desktop\test.txt
  
```

Ukážka posielania súboru

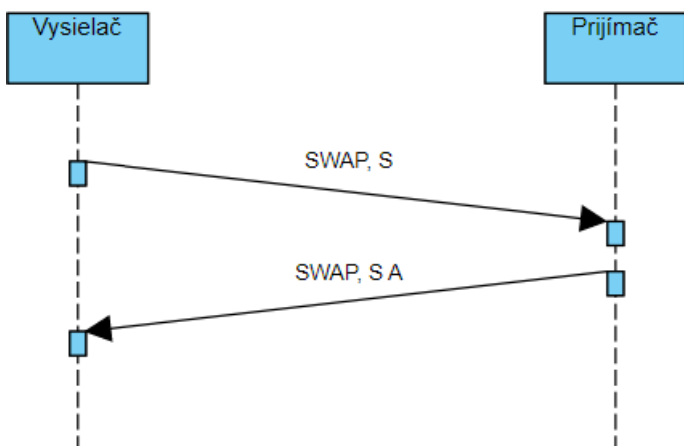
## Ukončenie spojenia

Ukončiť spojenie môže hociktorý uzol. Avšak spojenie je možné ukončiť iba vtedy, keď sa neodosielajú dáta. Ukončenie sa inicializuje tým, že jeden uzol pošle druhému packet typu 0 s flagom E (End connection). Po prijatí prijímajúci uzol odošle späť packet rovnakého typu s flagmi E a A. Potom sa spojenie ukončí a packety KEEP ALIVE sa už nebudú posilať, následne sa program ukončí.



## Prepínanie typu uzlov

Pri prvom spustení programu sa typ nastaví manuálne. Prepínanie medzi prijímačom a vysielačom prebehne poslaním packetu typu 0 s flagom S (Swap mode). Tento packet môže posilať aj prijímač aj vysielač. Po prijatí, prijímajúci uzol odošle packet rovnakého typu s flagmi S a A a typy sa prehodia.







Vysielač -> Prijímač

```
3
Request to swap sent successfully.
Keep alive request sent to 127.0.0.1 at port 42. Awaiting confirmation...
This server ip: 127.0.0.1
42
Server is listening at port 42...
```

Prijímač -> Vysielač

```
Swap request arrived from ('127.0.0.1', 57769). Sending confirmation...
Connection request sent to 127.0.0.1 at port 42. Awaiting confirmation...
Connection successful

1. - Send message
2. - Send file
3. - Swap modes
```

## Fragmentácia

Vysielač bude mať možnosť na začiatku zvoliť veľkosť fragmentu. Maximálnu veľkosť by som chcel nastaviť na 1400B aby sa program mohol rozširovať keď to bude nutné (napr. ak bude treba doplniť informácie do hlavičky) a teda aby sa nepresiahla veľkosť 1500B a tým nenastala fragmentácia na linkovej vrstve. Ak odosielané dáta majú menšiu veľkosť ako veľkosť fragmentu, tak sa vytvorí iba jeden fragment. Ak však veľkosť presahujú, tak sa dáta rozdelia podľa predvolenej dĺžky fragmentu, na každom packete fragmentu sa nastaví poradové číslo, totálny počet fragmentov a posielajú sa postupne. Po dokončení toku dát sa prijaté dáta poskladajú do celku podľa poradia FRAG\_NUM.

100000100110000010001110110110111

```

-----
111101101000100000100101110100000
100000100110000010001110110110111
-----
111010011101000101010110000101110
100000100110000010001110110110111
-----
110101110110001110110001100110010
100000100110000010001110110110111
-----
10101010000001100111110100001010
100000100110000010001110110110111
-----
101000011001101111000001011110100
100000100110000010001110110110111
-----
100011111110110100111110100001100
100000100110000010001110110110111
-----
110110001101101100000101110110000
100000100110000010001110110110111
-----
101101010111011100010110000001110
100000100110000010001110110110111
-----
110111000101111001100011011100100
100000100110000010001110110110111
-----
10111100011111011101101101010011

```

**Zostane nám 32 – bitový zvyšok:**

0b10111100011111011101101101010011 = 0xBC7DDB53

**Na zvyšok aplikujeme XOR s 0xFFFFFFFF:**

0b01000011100000100010010010101100 = 0x438224AC

**Prevrátime bity:**

0b00110101001001000100000111000010 = 0x352441C2 = 891568578

**CRC-32 výstup pre 'abc' je: 0x352441C2 v hexadecimálnom tvare a 891568578 v desiatkovom tvare.**

Zdroj príkladu: <https://www.autohotkey.com/boards/viewtopic.php?f=74&t=35671>