

Zadanie 2

Komunikácia s využitím UDP protokolu

Počítačové a komunikačné siete

Frederik Duvač
3. semester
15.10. 2023

Obsah

| | |
|-------------------------------------------------------------------------|-----------|
| 1. Technologické pozadie | 2 |
| 1.1. Hardware | 2 |
| 1.2. Programovací jazyk | 2 |
| 1.3. Vývojové prostredie | 2 |
| 2. Návrh programu | 3 |
| 2.1. Štruktúra hlavičky | 3 |
| 2.2. CRC | 4 |
| 2.3. Keep Alive | 4 |
| 2.4. Selective Repeat ARQ | 4 |
| 2.4.1. Diagram spracovávania komunikácie | 5 |
| 2.5. Implementácia tried a metód | 6 |
| 2.6. Používateľské rozhranie | 6 |
| 3. Zmeny návrhu vo finálnej verzii | 8 |
| 3.1. Štruktúra hlavičky | 8 |
| 3.2. Keep alive | 9 |
| 3.3. Stop and wait ARQ | 10 |
| 3.3.1. Sekvenčný diagram | 10 |
| 4. Implementácia | 11 |
| 4.1. Triedy | 11 |
| 4.2. Swap inicializácia zo strany servera | 11 |
| 4.3. Simulácia chybových paketov | 12 |
| 4.4. Ukončenie komunikácie | 13 |
| 5. Príklad komunikácie | 13 |
| 5.1. Nadviazanie komunikácie | 13 |
| 5.2. Posielanie dát | 14 |
| 5.3. Packet retransmission | 15 |
| 5.4. Udržiavanie spojenia | 16 |
| 5.5. Swap inicializácia | 16 |
| 5.6. Ukončenie komunikácie | 18 |
| 6. Používateľské rozhranie | 18 |
| 7. Záver | 20 |
| 7.1. Zoznam splnených podmienok | 20 |
| 7.1.1. Úloha 1 - Návrh programu a komunikačného protokolu | 20 |
| 7.1.2. Úloha 2 - Príprava | 20 |
| 7.1.3. Úloha 3 - Nastavenie IP a port | 21 |
| 7.1.4. Úloha 4 - Prenos súboru menšieho ako nastavená veľkosť fragmentu | 21 |
| 7.1.5. Úloha 5 - Simulácia chyby pri prenose súboru a správy | 22 |
| 7.1.6. Úloha 6 - Prenos 2MB súboru | 23 |
| 7.1.7. Úloha 7 - Udržiavanie spojenia | 23 |
| 7.1.8. Úloha 8 - Finálna dokumentácia a kvalita spracovania | 24 |

1. Technologické pozadie

1.1. Hardware

Zariadenie, na ktorom bol skompilovaný zdrojový kód a testovaný program, je notebook Lenovo Legion S7, 12. generácia Intel® Core™ i5-12500H, 16 GB RAM.

1.2. Programovací jazyk

Celý projekt je implementovaný v programovacom jazyku Python 3.11. V programe je použitých viacero knižníc ako socket (používaná na prácu s IP adresami, portami a posielaním dát), binascii (používaná funkcia na výpočet crc), struct (zabalenie dát do formátu hlavičky), tqdm (profesionálne progress bary), threading (knižnica uľahčuje prácu s vláknami), os (práca so súbormi), re (kontrola vstupu IP adres pomocou regexu), timeit (meranie času) a keyboard (hotkey bindings).



Obr. 1. Python

1.3. Vývojové prostredie

Sieťový komunikátor bol vyvíjaný v IDE PyCharm v balíku Professional vo verzii 2023.2.3 od spoločnosti JetBrains. Pycharm zabezpečoval programátorský príjemnú interakciu pri zostavovaní, interpretovaní a spúšťaní programu.

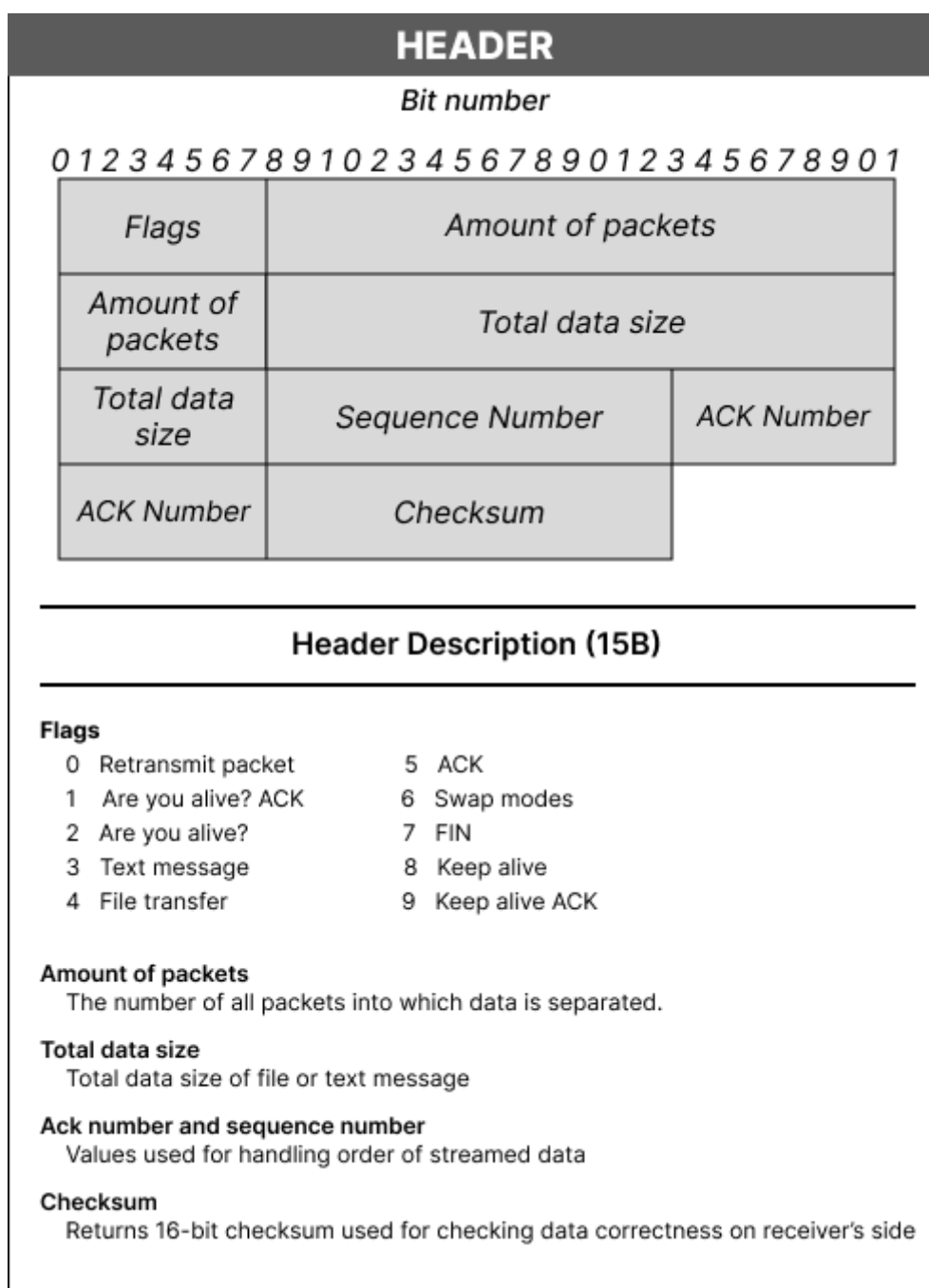


Obr. 2. Pycharm IDE

2. Návrh programu

2.1. Štruktúra hlavičky

Navrhnutá hlavička má veľkosť 15 Bytov a obsahuje, potrebné flagy, informáciu o množstve paketov na prenesenie, celkovú veľkosť dát, sekvenčné a ack čísla a kontrolný checksum. Navrhnuté flagy na označenie paketov a ďalšie informácie sú na obrázku číslo 3.



Obr. 3. Štruktúra navrhutej hlavičky

2.2. CRC

CRC je algoritmus Cyclic Redundancy Check (CRC) používaný na detekciu chýb v dátach. V programe bola implementovaná funkcia `crc_hqx()` z knižnice `binascii`. Princíp spočíva v postupnom prechádzaní jednotlivých bitov dát, aplikovaní XOR operácií s najvyšším bitom aktuálnej hodnoty CRC a využívaní tabuliek hodnôt a koeficientov na aktualizáciu hodnoty CRC. Výsledná hodnota CRC slúži na detekciu chýb v dátach a je reprezentovaná 16 bitmi, čo znamená, že výstup má veľkosť 2 bajty.

2.3. Keep Alive

Udržiavanie spojenia nastáva hneď po úspešnom odoslaní posledného dátového paketu. Keep alive packet sa posiela zo strany Sendera periodicky každých 5 sekúnd a zisťuje sa tým prípadné prerušenie spojenia. Celý tento proces sa spracováva na pozadí na samostatných vláknach. Ak sa počas tohto procesu stane Receiver nedostupným, Sender je informovaný správou o nedostupnosti Receivera. Pre prípad ak by sa Receiver stal dostupným, Sender má možnosť pokúsiť sa 2 krát opätovne obnoviť spojenie, inak sa dostane do úvodného stavu výberu módu.

Počas udržiavania spojenia má Sender možnosť začať posilať znova dáta. Po výbere z možností Sendera sa udržiavanie spojenia vypne a počas prenosu dát sa keep alive packet neposiela. V prípade ak by nastala situácia, že Receiver sa stane nedostupným počas prenosu dát je o tom Sender opäť informovaný.

2.4. Selective Repeat ARQ

Zvolená ARQ metóda bola Selective Repeat. Selective Repeat Automatic Repeat reQuest (ARQ) je protokol na detekciu a opravu chýb v bezdrôtových alebo spojitom prijímaných dátových prenosoch.

Implementovaná základná veľkosť okna (sliding window) bude veľkosti 4. Neskôr môže byť naprogramovaná ako dynamic sliding window. Zmena veľkosti sa bude odvíjať od nehostinnosti prenosového prostredia. V prípade vysokej chybovosti bude veľkosť menšia a opačne. Taktiež v prípade úspešnosti všetkých paketov zo sliding window ACK odpoveď môže byť zredukovaná do jedného ACK.

Fungovanie tohto algoritmu je nasledovné:

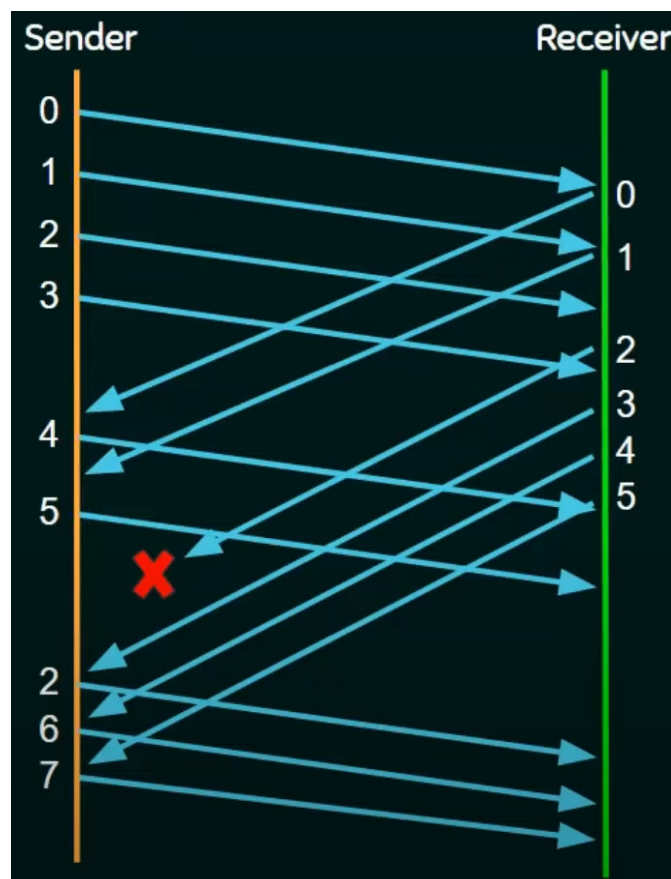
- Odosielané dáta sa rozdelia na menšie bloky - fragmenty a označujú sa ich číslami sekvencie.
- Tieto pakety sú posielané na prijímateľa a zároveň ukladané do okna o veľkosti 4.
- Receiver prijíma pakety a potvrdzuje ich pomocou potvrdzujúcich správ - ACK.
- Potvrdenie obsahuje informáciu o najvyššej úspešne prijatej sekvencii.

- Okno o veľkosti 4 na strane Sendera sleduje potvrdenia prijatých paketov.
- Ak je potvrdený nejaký paket, posúva sa okno dopredu.
- Ak príde k strate paketu, odosielateľ opakuje iba stratený paket namiesto celej skupiny.
- Proces odosielania a potvrdenia pokračuje, až kým všetky pôvodné dáta nie sú úspešne prenesené a potvrdené.

Prostredníctvom Selective Repeat sa minimalizuje vplyv chýb a zabezpečuje sa efektívny prenos dát v priebehu komunikácie.

2.4.1. Diagram spracovávania komunikácie

Na obrázku 2. sa odosielajú 4 pakety. Na nultý a prvý bol zo strany Receivera poslaný ACK paket. Sliding window sa posunie a odoslú sa pakety 4 a 5. Druhý paket mal chybový crc a preto sa odošle paket s príznakom na znovu odoslanie druhého paketu. Tretí, štvrtý a piaty paket sú korektné a preto sa odošle ACK. Korektne prijaté pakety zo sliding window sa neposielajú znova. A takto sa pokračuje ďalej.



Obr. 4. Sekvenčný diagram Selective Repeat ARQ

2.5. Implementácia tried a metód

Módy Receiver a Sender sú implementované každý, ako vlastná trieda s príslušnými atribútmi a metódami. Sender obsahuje funkciu `send()`, ktorá rieši výber funkcionality medzi odoslaním správy alebo odoslaním súboru, výmenou módov alebo ukončení komunikácie. Ďalsiou dôležitou funkciou je `send_data()`, ktorá spracováva hlavnú logiku fragmentovania a odosielania dát receiverovi. Ďalšie funkcie `keep_alive()` a `check_aliveness()` sú na udržiavanie a nadviazanie spojenia, `send_swap_request()` a `send_fin_request()` slúžia na inicializáciu spomenutých udalostí.

Receiver obsahuje hlavnú funkciu `receive()`, ktorá prijíma dáta od odosielateľa a následne ich spracováva.

Všetky potrebné funkcie programu, týkajúce sa spracovávania chýb, konzolových vstupov, vedľajších výpočtov a overovaní a podobných funkcionality sa nachádzajú v samostatnom súbore `utils.py`.

Hlavička je reprezentovaná ako trieda `DataHeader` s funkciou `pack_data()`, ktorá je zodpovedná za zabalenie potrebných dát do hlavičky a výpočet crc.

Hlavná funkcionality inicializovania módov a potrebná úvodná logika sa nachádza v súbore `main.py`.

2.6. Používateľské rozhranie

Používateľské rozhranie bolo implementované so vstupom a výstupom do konzoly. Program na úvod privíta a zobrazí IP adresu zariadenia. Ďalej sa spýta používateľa na mód (Sender, Receiver), ktorý bude uzol reprezentovať. V prípade módu Receiver, je potrebné zadať port, na ktorom bude program počúvať. Na druhej strane v prípade módu Sender, je potrebné nastaviť IP adresu a port, s ktorým bude prebiehať komunikácia. Receiver počúva stále na príslušnej adrese na dáta zo strany odosielateľa.

Odosielateľ po nastavení adresy, má na výber z možností poslať textovú správu alebo poslať súbor. Taktiež môže poslať požiadavku na výmenu módov alebo ukončenia komunikácie. Pri posielaní správy alebo súboru je potrebné nastaviť maximálnu veľkosť jedného fragmentu a napísať správu. V prípade posielania súboru je potrebné zadať názov aj s príponou súboru. Nežadáva sa cesta k súboru! Súbor si program najde v projektovej štruktúre automaticky.

```
Welcome 🖥️ LenovoLegionS7 you are '10.15.39.210' 🌐

⚙️ Choose operation mode: 🐦 RECEIVER → 1 | 📧 SENDER → 2 >> 2

MODE 2 → SENDER 📧
Connected to 🖥️ '10.15.39.210:42069' 🔊 Got your request! We are connected! ❤️

💡 What you wanna do?
  1 Send text message 📧
  2 Send file 📁
  3 Swap modes 🔄
  4 End communication ❤️
>> 1
🔪 Input fragment size (max 1461 B) >> 10
📧 Input message for '10.15.39.210:42069' 📧 >> Ahoj, ako sa mas?

🔔 INFO:
  🔪 Size of the data: 17B
  📦 Data was parsed into 2 packets

📖 LOG:
  🖥️ '10.15.39.210:42069' 🔊 Got your message! Thank you! 🔥
  ✅ Packet 1 was successfully sent 📧
  🖥️ '10.15.39.210:42069' 🔊 Got your message! Thank you! 🔥
  ✅ Packet 2 was successfully sent 📧

📄 Summary:
  📦 Sent packets: 2
  🔄 Retransmitted packets: 0
  🔪 Size of the data: 17B

💡 What you wanna do?
  1 Send text message 📧
  2 Send file 📁
  3 Swap modes 🔄
  4 End communication ❤️
>> |
```

Obr. 5. Príklad možnej interakcie s programom zo strany Sender


```
Welcome 🖥️ LenovoLegionS7 you are '10.15.39.210' 🌐

⚙️ Choose operation mode: 🐦 RECEIVER → 1 | 📧 SENDER → 2 >> 1

MODE 1 → RECEIVER 🐦
Connected 🖥️ '10.15.39.210:64246' 🔊 Are you alive? 🛎️
Message from 🖥️ '10.15.39.210:64246' is: Ahoj, ako
Message from 🖥️ '10.15.39.210:64246' is: sa mas?

📄 Received message is: Ahoj, ako sa mas?
🔪 Message size: 17B

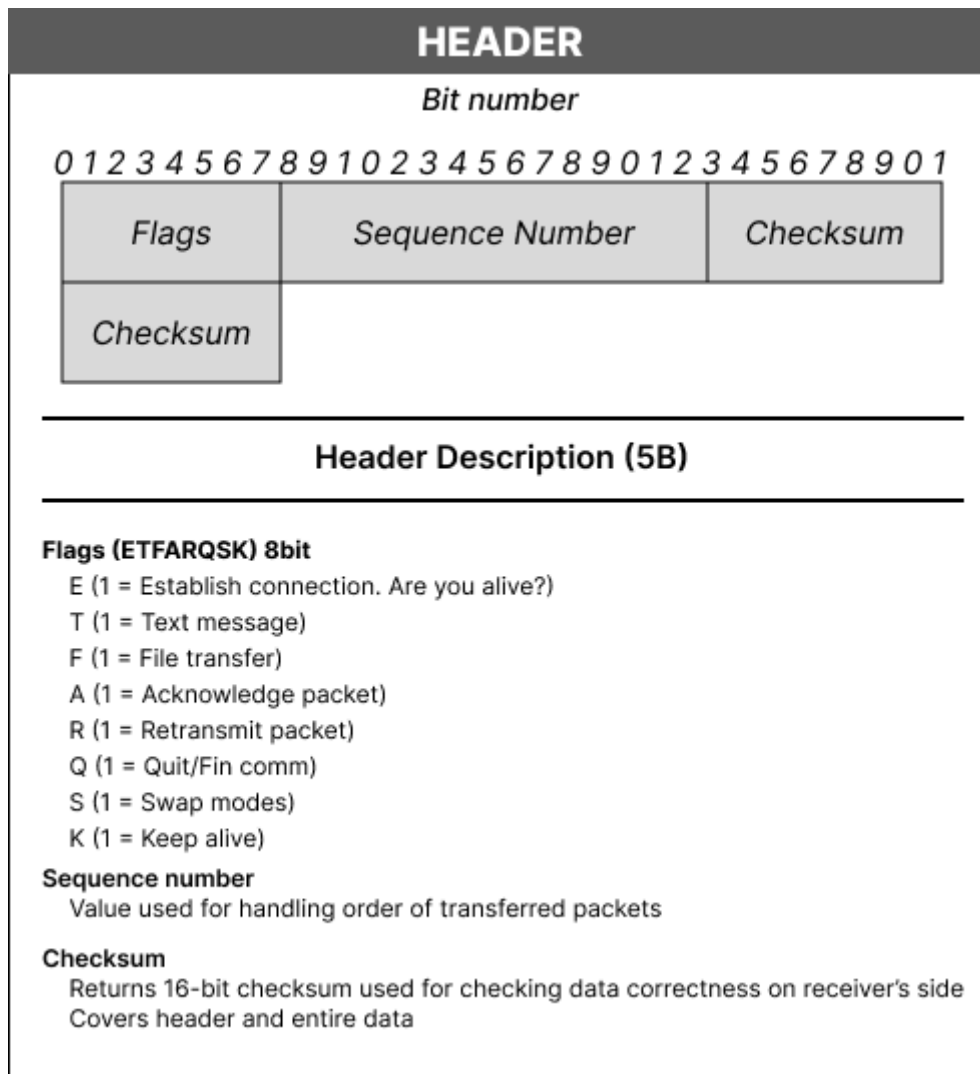
🐦 Waiting for data on '10.15.39.210:42069' 🌐
```

Obr. 6. Príklad výstupu CLI zo strany Receiver

3. Zmeny návrhu vo finálnej verzií



3.1. Štruktúra hlavičky

Za účelom zjednodušenia a zefektívnenia komunikácie s použitím nových flagov počas prenosu paketov bola hlavička výrazne prepracovaná. Nová hlavička má veľkosť 5 Bytov a obsahuje zmenený systém označovania paketov pomocou flagou (1 Byte), kedy sa jednotlivé flagy môžu kombinovať spolu pomocou zapínania príslušných bitov. Ďalej hlavička obsahuje len sekvenčné číslo paketu a kontrolný checksum (viď obr. 7.).



Obr. 7. Štruktúra hlavičky

3.2. Keep alive

Keep alive systém zostal takmer nezmenený. Program automaticky udržiava spojenie každých 5 sekúnd po nadviazaní komunikácie. V prípade odosielania dát sa posielanie keep alive paketov pozastaví a opäť sa obnoví po odoslaní posledných dát. Ak sa počas posielania keep alive paketu stane Receiver nedostupným, Sender je informovaný správou o nedostupnosti Receiveru. Program sa automaticky bude 30 sekúnd pokúšať nadviazať spojenie s Receiverom. Ak sa spojenie obnoví pokračuje sa v komunikácii inak sa program ukončí so správou  CONNECTION WAS LOST .

V prípade ak by nastala situácia, že Receiver sa stane nedostupným počas prenosu dát je o tom Sender opäť informovaný, program sa pokúsi opäť znovu nadviazať spojenie. V prípade úspešnosti funguje program tak, že odosiela data tam kde sa prerušilo posielanie inak sa vypíše oznam o strate spojenia.

3.3. Stop and wait ARQ

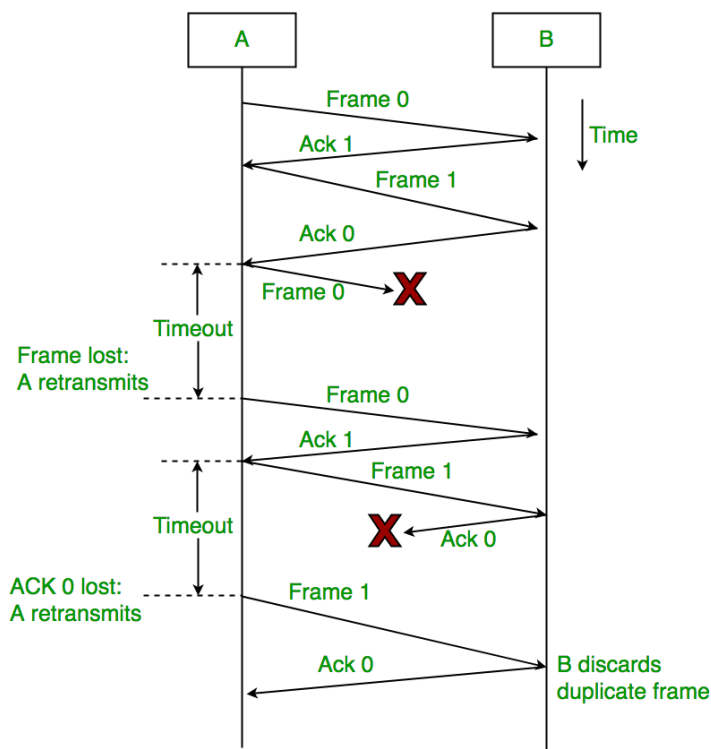
Stop-and-Wait ARQ je jednoduchý a často používaný protokol pre spoľahlivý prenos údajov cez komunikačný kanál. Je bežne používaný v situáciách, kde oneskorenie prenosu je malé a pravdepodobnosť chýb je relatívne nízka, napríklad v lokálnych počítačových sieťach (LAN) alebo priamočiarom bod-ku-bodu spojení.

Fungovanie tohoto algoritmu je nasledovné:

1. Odosielateľ odošle rámec obsahujúci blok údajov príjemcovi. Po odoslaní rámca odosielateľ čaká na potvrdenie (ACK) od príjemcu.
2. Príjemca prijme rámec a skontroluje prípadné chyby. Ak je rámec bez chýb, príjemca odošle späť ACK odosielateľovi. Ak sú prítomné chyby, príjemca rámec zahodí a vyžiada si ho poslať znova.
3. Po prijatí príznaku na znovu odoslanie paketu odosielateľ odošle paket znova a opäť čaká na ACK.
4. Tieto kroky sa opakujú až pokiaľ nie je poslaný paket s príznakom ukončenia komunikácie. To však záleží už od prispôsobenia implementácie.

Stop-and-Wait poskytuje jednoduchú formu riadenia toku. Odosielateľ neodošle nový rámec, kým neobdrží ACK pre predchádzajúci rámec.

3.3.1. Sekvenčný diagram



Obr. 8. Sekvenčný diagram Stop and Wait ARQ

(zdroj: <https://www.geeksforgeeks.org/stop-and-wait-arq/>)

4. Implementácia

4.1. Triedy

Program ostal po návrhu stále v objektovo orientovanej podobe, čiže základné informácie o implementácii tried a metód sa dozvieme z návrhu kapitola [Implementácia tried a metód](#). Väčšina metód týchto tried bola upravená, pribudli však aj nové ako napríklad funkcia `try_reestablish_connection()` v triede `Sender`, ktorá sa stará o znovu nadviazania komunikácie v prípade straty spojenia.

4.2. Swap inicializácia zo strany servera

V triede `Receiver` bola implementovaná funkcionálna možnosť inicializácie výmeny módov. Pomocou knižnice `keyboard` bol pridaný hotkey 'CTRL+S', ktorý volá funkciu `swap_initialization()`. Inicializácia výmeny módov je možná aj počas posielania dát, čo znamená, že `Receiver` počká na doposlanie všetkých dát a do posledného paketu s flagom F - file alebo T - text a flagom A - acknowledge pridá aj flag S - SWAP. `Sender` následne odpovie na túto inicializáciu zo strany `Receiver` flagmi SA a módy sa vymenia. Rovnakým princípom dokáže `Receiver` inicializovať SWAP aj počas posielania KEEP ALIVE paketov.

```
def __init__(self, address):
    ...
    keyboard.add_hotkey('ctrl+s', self.swap_initialization)
    self.SWAP_INITIALISATION = False
    ...

...
# Keep alive
self.receiver.sendto(
    DataHeader('KA' if not self.SWAP_INITIALISATION else 'KAS',
"KA-ACK".encode(), 0).pack_data(), sender_address)
...

def swap_initialization(self):
    if not self.SWAP_INITIALISATION:
        print("\n🔄 Swap Initialization")
        self.SWAP_INITIALISATION = True
```

Snippet kódu 1. Ukážka inicializácie výmeny zo strany Receiver

4.3. Simulácia chybových paketov

Simulácia chybovosti sa nastavuje na strane Sendera, kedy do checksumu hlavičky je vnesená úmyselná chyba. Používateľ zadá percentuálnu úspešnosť v rozsahu od 0 do 80% na dosiahnutie chybového paketu. Povedzme, že sme nastavili chybovosť na 50%, následne si program vygeneruje náhodné číslo od 0 do 100 a ak je vygenerované číslo menšie ako 50 paket bude poškodený.

```
# Sender Class
self.sender.sendto(
    DataHeader('F' if seq_number < amount_of_packets - 1 else
'FQ', fractional_data, seq_number)
    .pack_data(random.randrange(0, 100) < error_sim),
    self.address
)
...
elif contain_flags(flags, 'R'):
    failed_counter += 1
    if not show_progress_bar:
        print(f"\t💻 {format_address(receiver_address)} 📶
{d.decode()}")
        print(f"\t🔄 Packet {seq_number} with size
{packet_length}B will be retransmitted 🚫")
    ...

# Receiver Class
new_crc = get_crc_value(flags, seq_number, data)
if new_crc == crc:
    ...
else:
    failed_counter += 1
    if not self.show_progress_bar:
        print(f"\rPacket {seq_number} from 💻
{format_address(sender_address)} was rejected 🚫🛑")
        print(f"\t📦 Packet data size: {len(sender_message)}B
-", bin(flags), seq_number, crc, data)
        self.receiver.sendto(
            DataHeader('R', f"Got corrupted packet {seq_number}! It
needs to be retransmitted! 🚫".encode(),
                seq_number).pack_data(),
            sender_address
        )
    ...
```

```
# DataHeader Class
def pack_data(self, error_sim=False):
    header = struct.pack("!BH", int.from_bytes(self.flags,
byteorder='big'), self.seq_number)
    crc = binascii.crc_hqx(header + self.data, 0)
    if error_sim:
        crc = self.simulate_crc_error(crc, 2)
    header += struct.pack("!H", crc)

    return header + self.data

@staticmethod
def simulate_crc_error(crc, num_bits_to_flip):
    bitmask = (1 << num_bits_to_flip)
    return crc ^ bitmask
```

Snippet kódu 2. Ukážka simulácie chybovosti paketu

Do checksumu sa vnesie chyba pomocou jednoduchšej zmeny posledných 2 bitov. Receiver po rozbalení paketu vypočíta novú crc hodnotu a porovná ju, s tou ktorú dostal v hlavičke paketu. V prípade, že sa nezhodujú pošle paket s flagom R - retransmission. Sender po prijatí paketu s flagom R a nie A odošle predchádzajúci paket znova.

4.4. Ukončenie komunikácie

Komunikácia sa môže ukončiť viacerými spôsobmi. Prvý spôsob ukončenia komunikácie je, ak už nechce Sender posilať nič vyberie v menu task 4. pre ukončenie komunikácie a následne sa odošle paket s flagom Q - quit communication. Receiver odpovie paketom s flagom A a dostane sa do ponuky výberu módu. Sender prijme paket s flagom A a vypne sa program. Ďalším spôsobom ukončenia komunikácie je, ak sa po prerušení spojenia nepodarí do 30 sekúnd Sender pripojiť, program sa vypne s vypísaním upozornenia.

5. Príklad komunikácie

5.1. Nadviazanie komunikácie

Sender odošle paket s flagom E a dostane odpoveď s flagmi A a E.

| No. | Time | Source | Src Port | Destination | Dst Port | Protocol | Length | Info |
|-----|------------|-----------------|----------|-----------------|----------|----------|--------|----------------------|
| 210 | 244.888635 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 61 | 49717 → 42069 Len=19 |
| 211 | 244.891203 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 102 | 42069 → 49717 Len=60 |
| 212 | 256.133771 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 49 | 49717 → 42069 Len=7 |
| 215 | 256.137655 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 76 | 42069 → 49717 Len=34 |

| | | | | | | |
|-------------------------------------------------------------------------------------------------------------|--|--|--|------|-------------------------|-------------------------|
| Frame 210: 61 bytes on wire (488 bits), 61 bytes captured (488 bits) on interface \Device\NPF_{AD3...} | | | | 0000 | e4 a8 df cc 1a c8 00 e0 | 4c 70 8a 19 08 00 45 00 |
| Ethernet II, Src: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19), Dst: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8) | | | | 0010 | 00 2f d7 88 00 00 80 11 | 00 00 a9 fe 4c be a9 fe |
| Internet Protocol Version 4, Src: 169.254.76.190, Dst: 169.254.110.238 | | | | 0020 | 6e ee c2 35 a4 55 00 1b | 0f d6 80 00 00 2b bc 41 |
| User Datagram Protocol, Src Port: 49717, Dst Port: 42069 | | | | 0030 | 72 65 20 79 6f 75 20 61 | 6c 69 76 65 3f |
| UDPv4 Protocol Data | | | | | | |
| Flags: 0x80 [E] | | | | | | |
| Sequence Number: 0 | | | | | | |
| CRC: 11196 | | | | | | |
| Data: 41726520796f7520616c6976653f | | | | | | |

| No. | Time | Source | Src Port | Destination | Dst Port | Protocol | Length | Info |
|-----|------------|-----------------|----------|-----------------|----------|----------|--------|----------------------|
| 210 | 244.888635 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 61 | 49717 → 42069 Len=19 |
| 211 | 244.891203 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 102 | 42069 → 49717 Len=60 |
| 212 | 256.133771 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 49 | 49717 → 42069 Len=7 |
| 215 | 256.137655 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 76 | 42069 → 49717 Len=34 |
| 216 | 256.137943 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 49 | 49717 → 42069 Len=7 |

| | | | | | | |
|-------------------------------------------------------------------------------------------------------------|--|--|--|------|-------------------------|-------------------------|
| Frame 211: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface \Device\NPF_{AD3...} | | | | 0000 | 00 e0 4c 70 8a 19 e4 a8 | df cc 1a c8 08 00 45 00 |
| Ethernet II, Src: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8), Dst: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19) | | | | 0010 | 00 58 22 92 00 00 80 11 | 08 5a a9 fe 6e ee a9 fe |
| Internet Protocol Version 4, Src: 169.254.110.238, Dst: 169.254.76.190 | | | | 0020 | 4c be a4 55 c2 35 00 44 | e4 a0 90 00 00 48 1d 47 |
| User Datagram Protocol, Src Port: 42069, Dst Port: 49717 | | | | 0030 | 6f 74 20 79 6f 75 72 20 | 72 65 71 75 65 73 74 21 |
| UDPv4 Protocol Data | | | | 0040 | 20 57 65 20 61 72 65 20 | 63 6f 6e 6e 65 63 74 65 |
| Flags: 0x90 [E][A] | | | | 0050 | 64 21 20 f0 9f ab b1 f0 | 9f 8f bc e2 80 8d f0 9f |
| Sequence Number: 0 | | | | 0060 | ab b2 f0 9f 8f bc | |
| CRC: 18461 | | | | | | |
| Data: 476f7420796f75722072657175657374212057652061726520636f6e6e656374656 | | | | | | |

Obr. 9. Nadviazanie komunikácie

5.2. Posielanie dát

Dátové pakety sa posielajú s flagom F - pre file a flagom T - pre text.

| No. | Time | Source | Src Port | Destination | Dst Port | Protocol | Length | Info |
|-----|------------|-----------------|----------|-----------------|----------|----------|--------|----------------------|
| 210 | 244.888635 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 61 | 49717 → 42069 Len=19 |
| 211 | 244.891203 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 102 | 42069 → 49717 Len=60 |
| 212 | 256.133771 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 49 | 49717 → 42069 Len=7 |
| 215 | 256.137655 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 76 | 42069 → 49717 Len=34 |
| 216 | 256.137943 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 49 | 49717 → 42069 Len=7 |

| | | | | | | |
|-------------------------------------------------------------------------------------------------------------|--|--|--|------|-------------------------|--|
| Frame 212: 49 bytes on wire (392 bits), 49 bytes captured (392 bits) on interface \Device\NPF_{AD3...} | | | | 0000 | e4 a8 df cc 1a c8 00 e0 | |
| Ethernet II, Src: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19), Dst: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8) | | | | 0010 | 00 23 d7 89 00 00 80 11 | |
| Internet Protocol Version 4, Src: 169.254.76.190, Dst: 169.254.110.238 | | | | 0020 | 6e ee c2 35 a4 55 00 0f | |
| User Datagram Protocol, Src Port: 49717, Dst Port: 42069 | | | | 0030 | 6f | |
| UDPv4 Protocol Data | | | | | | |
| Flags: 0x40 [T] | | | | | | |
| Sequence Number: 0 | | | | | | |
| CRC: 3746 | | | | | | |
| Data: 686f | | | | | | |

| No. | Time | Source | Src Port | Destination | Dst Port | Protocol | Length | Info |
|-----|------------|-----------------|----------|-----------------|----------|----------|--------|----------------------|
| 210 | 244.888635 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 61 | 49717 → 42069 Len=19 |
| 211 | 244.891203 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 102 | 42069 → 49717 Len=60 |
| 212 | 256.133771 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 49 | 49717 → 42069 Len=7 |
| 215 | 256.137655 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 76 | 42069 → 49717 Len=34 |
| 216 | 256.137943 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 49 | 49717 → 42069 Len=7 |
| 217 | 256.144097 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 76 | 42069 → 49717 Len=34 |

| | | | | | | |
|-------------------------------------------------------------------------------------------------------------|--|--|--|------|-------------------------|---|
| Frame 215: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface \Device\NPF_{AD3...} | | | | 0000 | 00 e0 4c 70 8a 19 e4 a8 | d |
| Ethernet II, Src: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8), Dst: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19) | | | | 0010 | 00 3e 22 93 00 00 80 11 | 0 |
| Internet Protocol Version 4, Src: 169.254.110.238, Dst: 169.254.76.190 | | | | 0020 | 4c be a4 55 c2 35 00 2a | 8 |
| User Datagram Protocol, Src Port: 42069, Dst Port: 49717 | | | | 0030 | 6f 74 20 70 61 63 6b 65 | 7 |
| UDPv4 Protocol Data | | | | 0040 | 6e 6b 20 79 6f 75 21 20 | f |
| Flags: 0x50 [T][A] | | | | | | |
| Sequence Number: 0 | | | | | | |
| CRC: 11903 | | | | | | |
| Data: 476f74207061636b657420302e205468616e6b20796f752120f09f918c | | | | | | |

Obr. 10. Posielanie dát

Ukončenie posielania dát. Do posledného dátového paketu sa pridá flag Q.

| Time | Source IP | Destination IP | Source Port | Destination Port | Protocol | Length | Info |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-----------------------|-------------|------------------|----------|---------------|--------|
| 228.256.158178 | 169.254.76.190 | 49717.169.254.110.238 | 42069 | UDP | 48 | 49717 → 42069 | Len=6 |
| 229.256.159825 | 169.254.110.238 | 42069.169.254.76.190 | 49717 | UDP | 81 | 42069 → 49717 | Len=39 |
| 230.261.171220 | 169.254.76.190 | 49717.169.254.110.238 | 42069 | UDP | 57 | 49717 → 42069 | Len=15 |
| 233.261.175620 | 169.254.110.238 | 42069.169.254.76.190 | 49717 | UDP | 60 | 42069 → 49717 | Len=11 |
| <p>Frame 228: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface \Device\NPF_{AD3...}</p> <p>Ethernet II, Src: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19), Dst: CompalInform_cc:1a:c8 (e4:a8:df:00:10:00:22:d7:90:00:00:80:11)</p> <p>Internet Protocol Version 4, Src: 169.254.76.190, Dst: 169.254.110.238</p> <p>User Datagram Protocol, Src Port: 49717, Dst Port: 42069</p> <p>UDPKS2 Protocol Data</p> <p>Flags: 0x42 [T][Q]</p> <p>Sequence Number: 7</p> <p>CRC: 56799</p> <p>Data: 3f</p> | | | | | | | |
| 228.256.158178 | 169.254.76.190 | 49717.169.254.110.238 | 42069 | UDP | 48 | 49717 → 42069 | Len=6 |
| 229.256.159825 | 169.254.110.238 | 42069.169.254.76.190 | 49717 | UDP | 81 | 42069 → 49717 | Len=39 |
| 230.261.171220 | 169.254.76.190 | 49717.169.254.110.238 | 42069 | UDP | 57 | 49717 → 42069 | Len=15 |
| 233.261.175620 | 169.254.110.238 | 42069.169.254.76.190 | 49717 | UDP | 60 | 42069 → 49717 | Len=11 |
| <p>Frame 229: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface \Device\NPF_{AD3...}</p> <p>Ethernet II, Src: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8), Dst: RealtekSemic_70:8a:19 (00:e0:4c:00:10:00:43:22:9a:00:00:80:11)</p> <p>Internet Protocol Version 4, Src: 169.254.110.238, Dst: 169.254.76.190</p> <p>User Datagram Protocol, Src Port: 42069, Dst Port: 49717</p> <p>UDPKS2 Protocol Data</p> <p>Flags: 0x50 [T][A]</p> <p>Sequence Number: 7</p> <p>CRC: 22049</p> <p>Data: 476f74206c617374207061636b657420372e205468616e6b20796f752120f09f918c</p> | | | | | | | |

Obr. 11. Ukončenie posielania dát

5.3. Packet retransmission

Pokiaľ Sender odošle dátový paket a dostane paket s flagom R. Paket sa odošle znova.

| | | | | | | | | | |
|---|--------------------------------------------------------------------------------------------------------------|------------|-----------------|-------|-----------------|-----------|------|---------------|----------|
| | 361 | 294.521062 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 UDP | 1514 | 49717 → 42069 | Len=1472 |
| | 362 | 294.524725 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 UDP | 106 | 42069 → 49717 | Len=64 |
| | 363 | 294.525110 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 UDP | 1514 | 49717 → 42069 | Len=1472 |
| > | Frame 361: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{A...} | | | | | | 0020 | | |
| > | Ethernet II, Src: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19), Dst: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8) | | | | | | 0030 | | |
| > | Internet Protocol Version 4, Src: 169.254.76.190, Dst: 169.254.110.238 | | | | | | 0040 | | |
| > | User Datagram Protocol, Src Port: 49717, Dst Port: 42069 | | | | | | 0050 | | |
| > | UDPKS2 Protocol Data | | | | | | 0060 | | |
| > | Flags: 0x20 [F] | | | | | | 0070 | | |
| > | Sequence Number: 52 | | | | | | 0080 | | |
| > | CRC: 6855 | | | | | | 0090 | | |
| > | Data [truncated]: 891bf71a501ab3189116b01308121511891038100a10f00fe00fd80fd30fce0fcd0fcc0fcc0fc... | | | | | | 00a0 | | |
| > | | | | | | | 00b0 | | |
| | 361 | 294.521062 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 UDP | 1514 | 49717 → 42069 | Len=1472 |
| | 362 | 294.524725 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 UDP | 106 | 42069 → 49717 | Len=64 |
| | 363 | 294.525110 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 UDP | 1514 | 49717 → 42069 | Len=1472 |
| > | Frame 362: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{A...} | | | | | | 0000 | | |
| > | Ethernet II, Src: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8), Dst: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19) | | | | | | 0010 | | |
| > | Internet Protocol Version 4, Src: 169.254.110.238, Dst: 169.254.76.190 | | | | | | 0020 | | |
| > | User Datagram Protocol, Src Port: 42069, Dst Port: 49717 | | | | | | 0030 | | |
| > | UDPKS2 Protocol Data | | | | | | 0040 | | |
| > | Flags: 0x08 [R] | | | | | | 0050 | | |
| > | Sequence Number: 52 | | | | | | 0060 | | |
| > | CRC: 25160 | | | | | | | | |
| > | Data: 476f7420636f72727570746564207061636b657420353221204974206e6565647320746f20626520726574726: | | | | | | | | |

| | | | | | | | | | |
|-----|------------|-----------------|-------|-----------------|-------|-----|------|---------------|----------|
| 362 | 294.524725 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 106 | 42069 → 49717 | Len=64 |
| 363 | 294.525110 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 1514 | 49717 → 42069 | Len=1472 |
| 364 | 294.528965 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 106 | 42069 → 49717 | Len=64 |
| 365 | 294.529380 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 1514 | 49717 → 42069 | Len=1472 |


```

Frame 363: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF...
Ethernet II, Src: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19), Dst: CompalInform_cc:1a:c8 (e4:a8:df:cc:
Internet Protocol Version 4, Src: 169.254.76.190, Dst: 169.254.110.238
User Datagram Protocol, Src Port: 49717, Dst Port: 42069
UDP/PS2 Protocol Data
  Flags: 0x20 [F]
  Sequence Number: 52
  CRC: 6855
  Data [truncated]: 891bf71a501ab3189116b01308121511891038100a10f00fe00fd80fd30fce0fcd0fcc0fcc0fcc0f
  
```

Obr. 12. Príklad znovuposielania paketu

5.4. Udržiavanie spojenia

Sender posiela pakety s flagom K každých 5 sekúnd a prijíma pakety s flagmi K a A.

| | | | | | | | | | |
|-----|------------|-----------------|-------|-----------------|-------|-----|----|---------------|--------|
| 229 | 256.159825 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 81 | 42069 → 49717 | Len=39 |
| 230 | 261.171220 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 57 | 49717 → 42069 | Len=15 |
| 233 | 261.175620 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 60 | 42069 → 49717 | Len=11 |
| 234 | 266.176094 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 57 | 49717 → 42069 | Len=15 |
| 237 | 266.180636 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 60 | 42069 → 49717 | Len=11 |


```

Frame 230: 57 bytes on wire (456 bits), 57 bytes captured (456 bits) on interface \Device\NPF_{AD3...
Ethernet II, Src: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19), Dst: CompalInform_cc:1a:c8 (e4:a8:df:
Internet Protocol Version 4, Src: 169.254.76.190, Dst: 169.254.110.238
User Datagram Protocol, Src Port: 49717, Dst Port: 42069
UDP/PS2 Protocol Data
  Flags: 0x01 [K]
  Sequence Number: 0
  CRC: 8394
  Data: 4b45455020414c495645
  
```


| | | | | | | | | | |
|-----|------------|-----------------|-------|-----------------|-------|-----|----|---------------|--------|
| 230 | 261.171220 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 57 | 49717 → 42069 | Len=15 |
| 233 | 261.175620 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 60 | 42069 → 49717 | Len=11 |
| 234 | 266.176094 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 57 | 49717 → 42069 | Len=15 |
| 237 | 266.180636 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 60 | 42069 → 49717 | Len=11 |


```

Frame 233: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{AD3...
Ethernet II, Src: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8), Dst: RealtekSemic_70:8a:19 (00:e0:4c:
Internet Protocol Version 4, Src: 169.254.110.238, Dst: 169.254.76.190
User Datagram Protocol, Src Port: 42069, Dst Port: 49717
UDP/PS2 Protocol Data
  Flags: 0x11 [A][K]
  Sequence Number: 0
  CRC: 12083
  Data: 4b412d41434b
  
```

Obr. 13. Udržiavanie spojenia

5.5. Swap inicializácia

Inicializácia zo strany Sender. Ošle sa paket s flagom S a prijíme sa paket s flagom S a A, uzly sa vymenia.

| | | | | | | | | | |
|------|------------|-----------------|-------|-----------------|-------|-----|-----|---------------|--------|
| 3547 | 309.596735 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 | UDP | 51 | 49717 → 42069 | Len=9 |
| 3550 | 309.609621 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 | UDP | 94 | 42069 → 49717 | Len=52 |
| 3562 | 323.170293 | 169.254.110.238 | 51693 | 169.254.76.190 | 42069 | UDP | 61 | 51693 → 42069 | Len=19 |
| 3563 | 323.171027 | 169.254.76.190 | 42069 | 169.254.110.238 | 51693 | UDP | 102 | 42069 → 51693 | Len=60 |


```

Frame 3547: 51 bytes on wire (408 bits), 51 bytes captured (408 bits) on interface \Device\NPF_{AD312...
Ethernet II, Src: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19), Dst: CompalInform_cc:1a:c8 (e4:a8:df:cc:
Internet Protocol Version 4, Src: 169.254.76.190, Dst: 169.254.110.238
User Datagram Protocol, Src Port: 49717, Dst Port: 42069
UDP/PS2 Protocol Data
  Flags: 0x04 [S]
  Sequence Number: 0
  CRC: 36983
  Data: 53574150
  
```

| | | | | | | | | |
|------|------------|-----------------|-------|-----------------|-----------|-----|---------------|--------|
| 3547 | 309.596735 | 169.254.76.190 | 49717 | 169.254.110.238 | 42069 UDP | 51 | 49717 → 42069 | Len=9 |
| 3550 | 309.609621 | 169.254.110.238 | 42069 | 169.254.76.190 | 49717 UDP | 94 | 42069 → 49717 | Len=52 |
| 3562 | 323.170293 | 169.254.110.238 | 51693 | 169.254.76.190 | 42069 UDP | 61 | 51693 → 42069 | Len=19 |
| 3563 | 323.171027 | 169.254.76.190 | 42069 | 169.254.110.238 | 51693 UDP | 102 | 42069 → 51693 | Len=60 |

| | | | | | | | |
|-------------------------------------------------------------------------------------------------------------|--|--|--|--|--|------|-------------------------|
| Frame 3550: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface \Device\NPF_{AD312...} | | | | | | 0000 | 00 e0 4c 70 8a 19 e4 a8 |
| Ethernet II, Src: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8), Dst: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19) | | | | | | 0010 | 00 50 29 04 00 00 80 11 |
| Internet Protocol Version 4, Src: 169.254.110.238, Dst: 169.254.76.190 | | | | | | 0020 | 4c be a4 55 c2 35 00 3c |
| User Datagram Protocol, Src Port: 42069, Dst Port: 49717 | | | | | | 0030 | 6f 74 20 79 6f 75 72 20 |
| UDPKS2 Protocol Data | | | | | | 0040 | 20 53 77 61 70 70 69 6e |
| Flags: 0x14 [A][S] | | | | | | 0050 | 6f 20 73 65 6e 64 65 72 |
| Sequence Number: 0 | | | | | | | |
| CRC: 26502 | | | | | | | |
| Data: 476f7420796f7572207265717565737421205377617070696e67206d6f646520746f2073656e6465722120f09f93; | | | | | | | |

Obr. 14. Inicializácia výmeny zo strany Sendera

Inicializácia zo strany Receiver počas posielania keep alive paketov. Do paketu na odpoveď na keep alive sa pridá flag S.

| | | | | | | | | |
|------|------------|-----------------|-------|-----------------|-----------|----|---------------|--------|
| 3611 | 362.907035 | 169.254.110.238 | 51693 | 169.254.76.190 | 42069 UDP | 60 | 51693 → 42069 | Len=15 |
| 3612 | 362.907300 | 169.254.76.190 | 42069 | 169.254.110.238 | 51693 UDP | 53 | 42069 → 51693 | Len=11 |
| 3613 | 362.909939 | 169.254.110.238 | 51693 | 169.254.76.190 | 42069 UDP | 96 | 51693 → 42069 | Len=54 |
| 3624 | 384.173498 | 169.254.76.190 | 57589 | 169.254.110.238 | 42069 UDP | 61 | 57589 → 42069 | Len=19 |

| | | | | | | | |
|-------------------------------------------------------------------------------------------------------------|--|--|--|--|--|------|-------------------------|
| Frame 3612: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface \Device\NPF_{AD312...} | | | | | | 0000 | e4 a8 df cc 1a c8 00 e0 |
| Ethernet II, Src: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19), Dst: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8) | | | | | | 0010 | 00 27 de 09 00 00 80 11 |
| Internet Protocol Version 4, Src: 169.254.76.190, Dst: 169.254.110.238 | | | | | | 0020 | 6e ee a4 55 c9 ed 00 13 |
| User Datagram Protocol, Src Port: 42069, Dst Port: 51693 | | | | | | 0030 | 41 2d 41 43 4b |
| UDPKS2 Protocol Data | | | | | | | |
| Flags: 0x15 [A][S][K] | | | | | | | |
| Sequence Number: 0 | | | | | | | |
| CRC: 46044 | | | | | | | |
| Data: 4b412d41434b | | | | | | | |

| | | | | | | | | |
|------|------------|-----------------|-------|-----------------|-----------|----|---------------|--------|
| 3612 | 362.907300 | 169.254.76.190 | 42069 | 169.254.110.238 | 51693 UDP | 53 | 42069 → 51693 | Len=11 |
| 3613 | 362.909939 | 169.254.110.238 | 51693 | 169.254.76.190 | 42069 UDP | 96 | 51693 → 42069 | Len=54 |
| 3624 | 384.173498 | 169.254.76.190 | 57589 | 169.254.110.238 | 42069 UDP | 61 | 57589 → 42069 | Len=19 |

| | | | | | | | |
|-------------------------------------------------------------------------------------------------------------|--|--|--|--|--|------|-------------------------|
| Frame 3613: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface \Device\NPF_{AD312...} | | | | | | 0000 | 00 e0 4c 70 8a 19 e4 a8 |
| Ethernet II, Src: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8), Dst: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19) | | | | | | 0010 | 00 52 29 14 00 00 80 11 |
| Internet Protocol Version 4, Src: 169.254.110.238, Dst: 169.254.76.190 | | | | | | 0020 | 4c be c9 ed a4 55 00 31 |
| User Datagram Protocol, Src Port: 51693, Dst Port: 42069 | | | | | | 0030 | 6f 74 20 79 6f 75 72 20 |
| UDPKS2 Protocol Data | | | | | | 0040 | 20 53 77 61 70 70 69 6e |
| Flags: 0x14 [A][S] | | | | | | 0050 | 6f 20 72 65 63 65 69 70 |
| Sequence Number: 0 | | | | | | | |
| CRC: 7441 | | | | | | | |
| Data: 476f7420796f7572207265717565737421205377617070696e67206d6f646520746f207265636569722120f09f93; | | | | | | | |

Obr. 15. Inicializácia výmeny zo strany Receiver

Inicializácia zo strany Receiver počas posielania dátových paketov. Receiver do posledného ACK pridá flag S.

| | | | | | | | | |
|------|------------|-----------------|-------|-----------------|-----------|------|---------------|----------|
| 4975 | 447.469944 | 169.254.76.190 | 57589 | 169.254.110.238 | 42069 UDP | 1504 | 57589 → 42069 | Len=1462 |
| 4976 | 447.474099 | 169.254.110.238 | 42069 | 169.254.76.190 | 57589 UDP | 83 | 42069 → 57589 | Len=41 |
| 4977 | 447.474354 | 169.254.76.190 | 57589 | 169.254.110.238 | 42069 UDP | 96 | 57589 → 42069 | Len=54 |
| 4980 | 454.086267 | 169.254.110.238 | 65480 | 169.254.76.190 | 42069 UDP | 61 | 65480 → 42069 | Len=19 |
| 4983 | 473.219142 | 169.254.110.238 | 65481 | 169.254.76.190 | 42069 UDP | 61 | 65481 → 42069 | Len=19 |

| | | | | | | | |
|-------------------------------------------------------------------------------------------------------------|--|--|--|--|--|------|-------------------------|
| Frame 4976: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface \Device\NPF_{AD312...} | | | | | | 0000 | 00 e0 4c 70 8a 19 e4 a8 |
| Ethernet II, Src: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8), Dst: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19) | | | | | | 0010 | 00 45 2b a9 00 00 80 11 |
| Internet Protocol Version 4, Src: 169.254.110.238, Dst: 169.254.76.190 | | | | | | 0020 | 4c be a4 55 e0 f5 00 31 |
| User Datagram Protocol, Src Port: 42069, Dst Port: 57589 | | | | | | 0030 | 6f 74 20 6c 61 73 74 20 |
| UDPKS2 Protocol Data | | | | | | 0040 | 37 39 2e 20 54 68 61 6e |
| Flags: 0x34 [F][A][S] | | | | | | 0050 | 9f 91 8c |
| Sequence Number: 579 | | | | | | | |
| CRC: 13405 | | | | | | | |
| Data: 476f74206c617374207061636b6574203537392e205468616e6b20796f752120f09f918c | | | | | | | |

| | | | | | | | | | |
|------|------------|-----------------|-------|-----------------|-------|-----|----|---------------|--------|
| 4976 | 447.474099 | 169.254.110.238 | 42069 | 169.254.76.190 | 57589 | UDP | 83 | 42069 → 57589 | Len=41 |
| 4977 | 447.474354 | 169.254.76.190 | 57589 | 169.254.110.238 | 42069 | UDP | 96 | 57589 → 42069 | Len=54 |
| 4980 | 454.086267 | 169.254.110.238 | 65480 | 169.254.76.190 | 42069 | UDP | 61 | 65480 → 42069 | Len=19 |
| 4983 | 473.219142 | 169.254.110.238 | 65481 | 169.254.76.190 | 42069 | UDP | 61 | 65481 → 42069 | Len=19 |

| | | |
|----------------------------------------------------------------------------------------------------------|------|-------------------------|
| ▶ Frame 4977: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface \Device\NPF_{AD312} | 0000 | e4 a8 df cc 1a c8 00 e0 |
| ▶ Ethernet II, Src: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19), Dst: CompalInform_cc:1a:c8 (e4:a8:df:cc: | 0010 | 00 52 e0 a0 00 00 80 11 |
| ▶ Internet Protocol Version 4, Src: 169.254.76.190, Dst: 169.254.110.238 | 0020 | 6e ee e0 f5 a4 55 00 3e |
| ▶ User Datagram Protocol, Src Port: 57589, Dst Port: 42069 | 0030 | 6f 74 20 79 6f 75 72 20 |
| ▼ UDPK52 Protocol Data | 0040 | 20 53 77 61 70 70 69 6e |
| Flags: 0x14 [A][S] | 0050 | 6f 20 72 65 63 65 69 76 |
| Sequence Number: 0 | | |
| CRC: 7441 | | |
| Data: 476f7420796f7572207265717565737421205377617070696e67206d6f646520746f2072656365697665722120f0: | | |

Obr. 16. Inicializácia výmeny zo strany Receiver

5.6. Ukončenie komunikácie

Sender odošle paket iba s flagom Q. Receiver ho prijme a odošle paket s flagmi A a Q.

| | | | | | | | | | |
|------|------------|-----------------|-------|-----------------|-------|-----|----|---------------|--------|
| 4997 | 483.401288 | 169.254.110.238 | 65481 | 169.254.76.190 | 42069 | UDP | 60 | 65481 → 42069 | Len=8 |
| 4998 | 483.401793 | 169.254.76.190 | 42069 | 169.254.110.238 | 65481 | UDP | 88 | 42069 → 65481 | Len=46 |

| | | |
|----------------------------------------------------------------------------------------------------------|------|-------------------------|
| ▶ Frame 4997: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{AD312} | 0000 | 00 e0 4c 70 8a 19 e4 a8 |
| ▶ Ethernet II, Src: CompalInform_cc:1a:c8 (e4:a8:df:cc:1a:c8), Dst: RealtekSemic_70:8a:19 (00:e0:4c:70: | 0010 | 00 24 2b ae 00 00 80 11 |
| ▶ Internet Protocol Version 4, Src: 169.254.110.238, Dst: 169.254.76.190 | 0020 | 4c be ff c9 a4 55 00 10 |
| ▶ User Datagram Protocol, Src Port: 65481, Dst Port: 42069 | 0030 | 49 4e 00 00 00 00 00 00 |
| ▼ UDPK52 Protocol Data | | |
| Flags: 0x02 [Q] | | |
| Sequence Number: 0 | | |
| CRC: 14867 | | |
| Data: 46494e | | |

| | | | | | | | | | |
|------|------------|-----------------|-------|-----------------|-------|-----|----|---------------|--------|
| 4997 | 483.401288 | 169.254.110.238 | 65481 | 169.254.76.190 | 42069 | UDP | 60 | 65481 → 42069 | Len=8 |
| 4998 | 483.401793 | 169.254.76.190 | 42069 | 169.254.110.238 | 65481 | UDP | 88 | 42069 → 65481 | Len=46 |

| | | |
|----------------------------------------------------------------------------------------------------------|------|-------------------------|
| ▶ Frame 4998: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface \Device\NPF_{AD312} | 0000 | e4 a8 df cc 1a c8 00 e0 |
| ▶ Ethernet II, Src: RealtekSemic_70:8a:19 (00:e0:4c:70:8a:19), Dst: CompalInform_cc:1a:c8 (e4:a8:df:cc: | 0010 | 00 4a e0 a4 00 00 80 11 |
| ▶ Internet Protocol Version 4, Src: 169.254.76.190, Dst: 169.254.110.238 | 0020 | 6e ee a4 55 ff c9 00 34 |
| ▶ User Datagram Protocol, Src Port: 42069, Dst Port: 65481 | 0030 | 6f 74 20 79 6f 75 72 20 |
| ▼ UDPK52 Protocol Data | 0040 | 20 54 68 61 6e 6b 20 75 |
| Flags: 0x12 [A][Q] | 0050 | 79 65 21 20 f0 9f 91 80 |
| Sequence Number: 0 | | |
| CRC: 10799 | | |
| Data: 476f7420796f7572207265717565737421205468616e6b20796f7520616e64206279652120f09f918b | | |

Obr. 17. Ukončenie komunikácie

6. Používateľské rozhranie

CLI prešlo od návrhu miernymi zmenami. Receiver dostal update interakcie kde, si dokážeme nastaviť na začiatku či si prajeme vypisovať log prijatých paketov alebo nám stačí iba loading bar. Ďalej bolo potrebné poskytnúť možnosť kam sa má prijatý súbor uložiť. V logu pribudol taktiež podrobnejší výpis informácií o pakete. Novinkou je tiež možnosť inicializácie výmeny módov zo strany Receivera a preto program počúva na 'CTRL+S'.

```

Welcome 🖥️ LenovoLegionS7 you are '192.168.109.181' 🌐

⚙️ Choose operation mode: 🐦 RECEIVER → 1 | 📁 SENDER → 2 >> 1

MODE 1 → RECEIVER 🐦
Choose listening port 🐦 >> 42069
⌚ Do you want to show progress bar instead log? >> no
📁 Input the directory to save the file (ENTER key for default: './received_files') >>
Connected 🖥️ '192.168.109.181:62661' 🔊 Are you alive? 📢
Packet 0 from 🖥️ '192.168.109.181:62661' was rejected 🚫🔴
📦 Packet data size: 21B - 0b1000010 0 44940 b'ahoj ako sa mas?'
Packet 0. from 🖥️ '192.168.109.181:62661' was accepted 📡✅
📦 Packet data size: 21B - 0b1000010 0 44936 b'ahoj ako sa mas?'

📄 Received message is: ahoj ako sa mas?
🔧 Message size: 16B

(10s) Keeping connection alive...

```

Obr. 18. Príklad CLI zo strany Receiver

Na strane Sendera pribudla možnosť simulovania chyby prenosu paketu (vstup od 0-80%). Ďalej pribudla taktiež možnosť zobraziť progress bar a logu pribudli informácie o znovu posielaní paketu a veľkosť daného paketu.

```

Welcome 🖥️ LenovoLegionS7 you are '192.168.109.181' 🌐

⚙️ Choose operation mode: 🐦 RECEIVER → 1 | 📁 SENDER → 2 >> 2

MODE 2 → SENDER 📁
🐦 Receiver's IP address >> 192.168.109.181
Receiver's port 🐦 >> 42069
Wait some time to setup the RECEIVER or press SPACE key to try connect now...
Connected to 🖥️ '192.168.109.181:42069' 🔊 Got your request! We are connected! 🍷

📡 What you wanna do?
1 Send text message 📄
2 Send file 📁
3 Swap modes 🔄
4 End communication 💔

>> 1
📄 Input message for '192.168.109.181:42069' 📄 >> ahoj ako sa mas?
🔧 Input fragment size (max 1467 B) >> max
🎲 Choose percentual error probability >> 50
⌚ Do you want to show progress bar instead log? >> no

```

Obr. 19. Interakcia s CLI na strane Sender

```
INFO:
  Size of the data: 16B
  Data was packaged into 1 packet

LOG:
  '192.168.109.181:42069' Got corrupted packet 0! It needs to be retransmitted!
  Packet 0 with size 21B will be retransmitted
  '192.168.109.181:42069' Got last packet 0. Thank you!
  Packet 0 with size 21B was successfully sent

Summary:
  All sent packets: 2
  Retransmitted packets: 1
  Size of the data: 16B

What you wanna do?
  1 Send text message
  2 Send file
  3 Swap modes
  4 End communication
(235s) Keeping connection alive... >>
```

Obr. 20. Výpis informácií na strane Sender

7. Záver

Záverom je potrebné zdôrazniť, že všetky zmeny návrhu boli implementované pre čo najlepší možný dosiahnutý výsledok projektu po konzultáciách s cvičiacim. Selective repeat nebol implementovaný z dôvodu nedostatku času a Stop and wait bude pravdepodobne viac prívetivejší pri doimplementácii. Vzhľadom na splnenie nasledujúceho zoznamu podmienok môžeme povedať, že tento projekt bol úspešne implementovaný a počas jeho implementácie sme si odniesli množstvo prospešných informácií.

7.1. Zoznam splnených podmienok

7.1.1. Úloha 1 - Návrh programu a komunikačného protokolu

Návrh programu bol prednesený na cvičení, potrebné zmeny a návrhy boli konzultované s cvičiacim.

7.1.2. Úloha 2 - Príprava

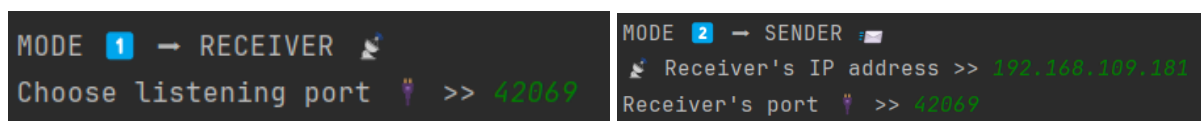
Pre rýchlejšiu identifikáciu protokolu bol spravená .lua script a pridaný medzi pluginy do Wiresharku. Protokol má názov UDPKS2 čiže sa dá pekne vyhľadať alebo môžeme zadať jednoducho filter `udp.port == 'port, na ktorom prebieha komunikácia'`.

| No. | Time | Source | Src Port | Destination | Dst Port | Protocol | Length | Info |
|-----|-----------|-----------------|----------|-----------------|----------|----------|--------|----------------------|
| 22 | 29.938964 | 192.168.109.181 | 54779 | 192.168.109.181 | 42069 | UDP | 51 | 54779 → 42069 Len=19 |
| 23 | 29.939377 | 192.168.109.181 | 42069 | 192.168.109.181 | 54779 | UDP | 92 | 42069 → 54779 Len=60 |
| 24 | 34.940605 | 192.168.109.181 | 54779 | 192.168.109.181 | 42069 | UDP | 47 | 54779 → 42069 Len=15 |
| 25 | 34.940730 | 192.168.109.181 | 42069 | 192.168.109.181 | 54779 | UDP | 43 | 42069 → 54779 Len=11 |
| 26 | 39.941706 | 192.168.109.181 | 54779 | 192.168.109.181 | 42069 | UDP | 47 | 54779 → 42069 Len=15 |
| 27 | 39.942501 | 192.168.109.181 | 42069 | 192.168.109.181 | 54779 | UDP | 43 | 42069 → 54779 Len=11 |

Obr. 21. Custom filter udpks2 Wireshark

7.1.3. Úloha 3 - Nastavenie IP a port

Receiver vyžaduje nastaviť port, na ktorom počúva a Sender musí nastaviť na akú IP adresu a port sa pripája.



Obr. 22. Nastavenie IP adresy a portu

7.1.4. Úloha 4 - Prenos súboru menšieho ako nastavená veľkosť fragmentu

Prenos súboru menšieho ako stanovená veľkosť fragmentu je funkčná. Podľa nasledujúcich 2 obrázkov vidíme, že veľkosť súboru test.txt je 441B čo je menej ako maximálna povolená veľkosť.

```
📁 Input filename >> test.txt
🔧 Input fragment size (max 1467 B) >> max
🎲 Choose percentual error probability >> 0
⌚ Do you want to show progress bar instead log? >> 0

📄 INFO:
  🔧 Size of the data: 441B
  📦 Data was parsed into 2 packets

📖 LOG:
  💻 '192.168.109.181:42069' 🔊 Got packet 0. Thank you! 🔥
  ✅ Packet 0 with size 13B was successfully sent 🚩
  💻 '192.168.109.181:42069' 🔊 Got last packet 1. Thank you! 🔥
  ✅ Packet 1 with size 446B was successfully sent 🚩

📄 Summary:
  📦 All sent packets: 2
  🔄 Retransmitted packets: 0
  🔧 Size of the data: 441B
```

Obr. 23. Prenos menšieho súboru na strane Sender

```
Welcome 🖥️ LenovoLegionS7 you are '192.168.109.181' 🌐

⚙️ Choose operation mode: 🐦 RECEIVER ← 1 | 📡 SENDER ← 2 >> 1

MODE 1 → RECEIVER 🐦
Choose listening port 🗣️ >> 42069
⌚ Do you want to show progress bar instead log? >> no
📁 Input the directory to save the file (ENTER key for default: './received_files') >>
Connected 💻 '192.168.109.181:49287' 🔊 Are you alive? 🔥
Packet 0 from 💻 '192.168.109.181:49287' was accepted ✅✅
  📦 Packet data size: 13B - 0b100000 0 61029 b'test.txt'
Packet 1 from 💻 '192.168.109.181:49287' was accepted ✅✅
  📦 Packet data size: 446B - 0b100010 1 13947 b"\xf0\x9f\x93\x9c Received message is: ajnjqbnlijbflirbrqil"

Data has been successfully streamed to test.txt

📁 File location is 'C:\Users\frede\PycharmProjects\PKS_Zadanie2_Komunikacia-cez_UDP\received_files\test.txt'
🔧 File size: 441B
```

Obr. 24. Prenos menšieho súboru na strane Receiver

7.1.5. Úloha 5 - Simulácia chyby pri prenose súboru a správy
Simuláciu chybovosti paketov je ukázaná v kapitole [Používateľské rozhranie](#).

7.1.6. Úloha 6 - Prenos 2MB súboru

Prenos 2 MB súboru bol úspešný. Informácie o prenesenom súbore 2MB.jpg si môžeme pozrieť na nasledujúcom obrázku.

```
Input filename >> 2MB.jpg
Input fragment size (max 1467 B) >> max
Choose percentual error probability >> 0
Do you want to show progress bar instead log? >> y

INFO:
  Size of the data: 2000379B
  Data was parsed into 1365 packets

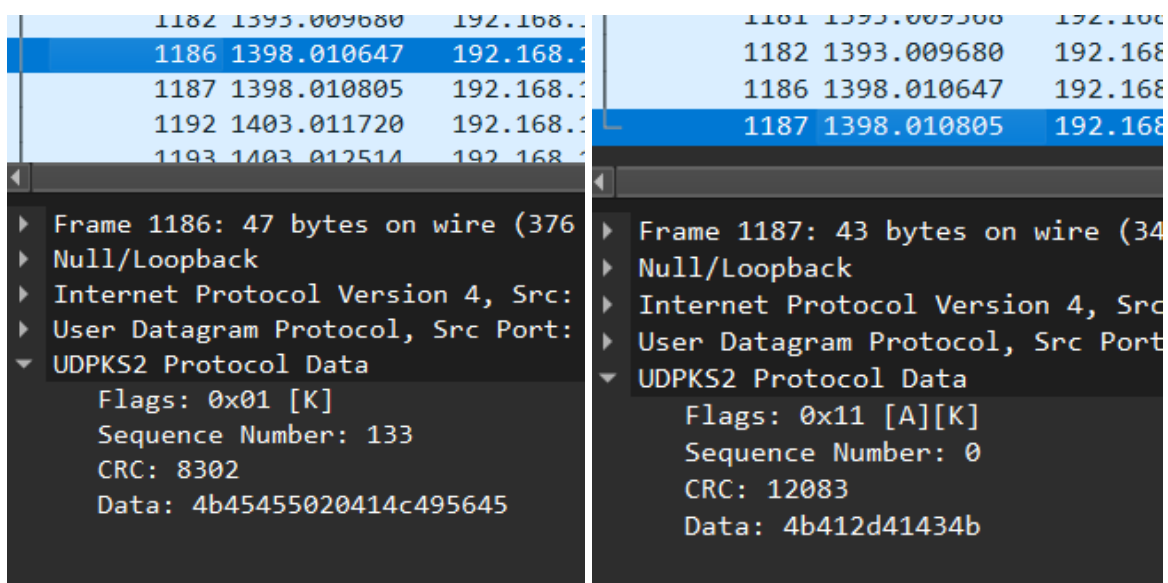
LOG:
Sending file 2MB.jpg: 100%|██████████| 1.36k/1.36k [00:00<00:00, 1.45kB/s]

Summary:
  All sent packets: 1365
  Retransmitted packets: 0
  Size of the data: 2000379B
```

Obr. 25. Prenos súboru väčšieho ako 2 MB

7.1.7. Úloha 7 - Udržiavanie spojenia

Udržiavanie spojenia je bližšie vysvetlené v kapitole [Keep alive](#). Sender pošle paket s flagom K - keep alive a dostane odpoveď s flagmi K a A - acknowledge.



Obr. 26. Sender dostal odpoveď na keep alive packet

7.1.8. Úloha 8 - Finálna dokumentácia a kvalita spracovania

Kvalitu dokumentácie a spracovania projektu je potrebné posúdiť z pohľadu inej osoby. Myslíme si, že z našej strany je na solídnej úrovni ale vždy je priestor sa zlepšovať ;).