POČÍTAČOVÉ A KOMUNIKAČNÉ SIETE

Semestrálne zadanie: Komunikácia s využitím UDP protokolu

Finálne odovzdanie

Michaela Hojová
ID: 127162
xhojova@stuba.sk

Cvičenie
štvrtok 10:00
Ing. Ladislav Zemko

Slovenská technická univerzita
Fakulta informatiky a informačných technológií

Obsah

Michaela Hojová, 127162

1. Úvod

Cieľom tohto dokumentu je opísať návrh vlastného protokolu postavenom nad UDP v transportnej vrstve sieťového modelu TCP/IP, vďaka ktorému je umožnená komunikácia dvoch účastníkov v lokálnej Ethernet sieti. Zároveň zahŕňa opis implementácie jednotlivých mechanizmov zabezpečujúcich danú funkcionalitu.

2. Návrh protokolu z kontrolného bodu

2.1. Mnou navrhnutý protokol

2.1.1. Štruktúra hlavičky

<table>
<tr><td colspan="8">1B</td><td colspan="8">1B</td><td colspan="8">1B</td><td colspan="8">1B</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="16">Sequence Number</td><td colspan="16">Acknowledgment Number</td></tr>
<tr><td colspan="8">Flags</td><td colspan="16">Length of data</td><td colspan="8">Checksum</td></tr>
<tr><td colspan="8">Checksum (cont.)</td><td colspan="16">Window</td><td colspan="8"></td></tr>
</table>

Tabuľka č. 1

- Sequence Number – poradové číslo paketu
- Acknowledgment Number – poradové číslo paketu potvrdenia
- Flags
  - S – žiadosť o spojenie
  - A – potvrdenie spojenia
  - N – negatívne potvrdenie spojenia
  - H – kontrola heartbeatu, teda spojenia
  - F – fragmentácia (uskutoční/neuskutoční sa)
  - D – typ predmetu odoslania (message/file)
  - T – ukončenie spojenia
  - R – rezervovaný na budúce použitie

- Length of data – veľkosť odosielaných dát
- Checksum – kontrolný súčet nad poslanými dátami
- Window - parameter, ktorý určuje maximálny počet nepotvrdených paketov, ktoré môžu byť súčasne odoslané

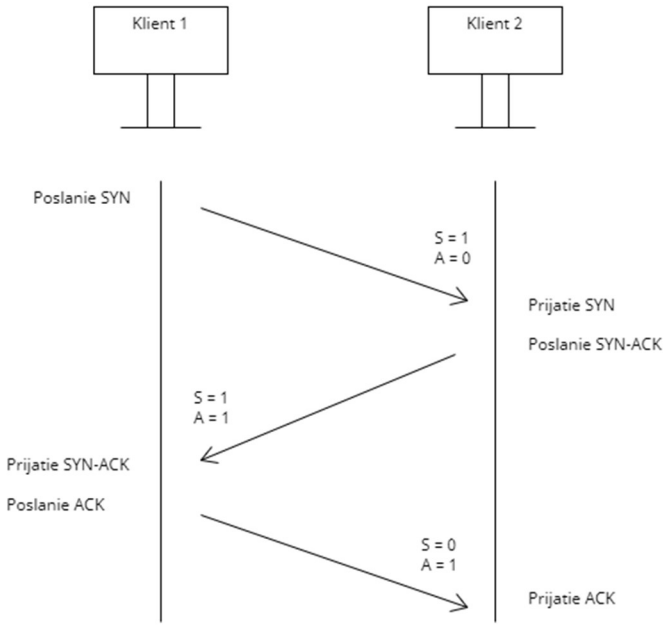
2.1.2. Funkcionalita spojenia

2.1.2.1. Začiatok spojenia

Začiatok spojenia sa realizuje prostredníctvom 3-way hanshaku podobne ako v TCP. Ktorýkoľvek z Klientov môže inicializovať spojenie, kedy potom následne začína daný handshake. Prebieha nasledovne:

1. SYN: Klient inicializujúci spojenie pošle paket, v podstate prázdnu hlavičku bez dát, v ktorej bude Flag S nastavený na 1, čím žiada o nadviazanie spojenia
2. SYN-ACK: Akonáhle druhý Klient spracuje SYN správu, odpovie paketom s hlavičkou, kde budú Flags S a A nastavené na 1, čím potvrdzuje požiadavku na spojenie
3. ACK: Keď prvý Klient spracuje SYN-ACK, posiela potvrdenie ACK, teda paket s Flagom A v hlavičke nastaveným na 1, čím sa spojenie nadviaže

Proces je znázornený nasledujúcim diagramom:

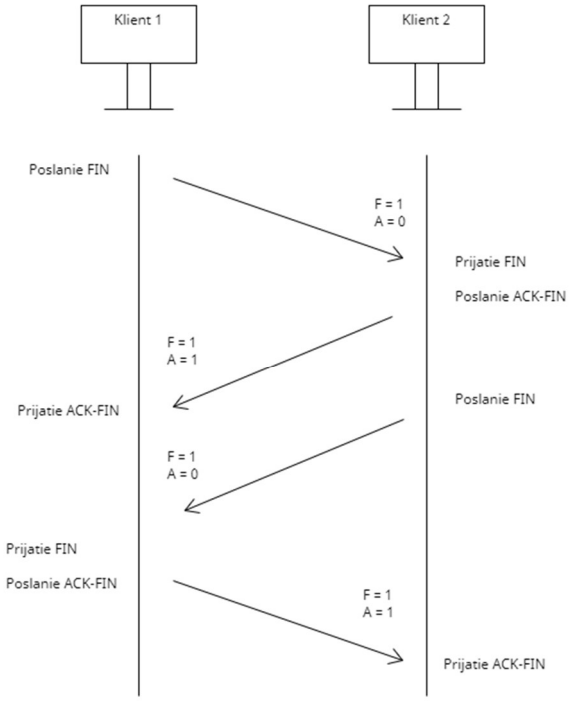


2.1.2.2. Ukončenie spojenia

Ukončenie spojenia sa bude realizovať pomocou 4-way hanshake, ktorý bude prebiehať nasledovne:

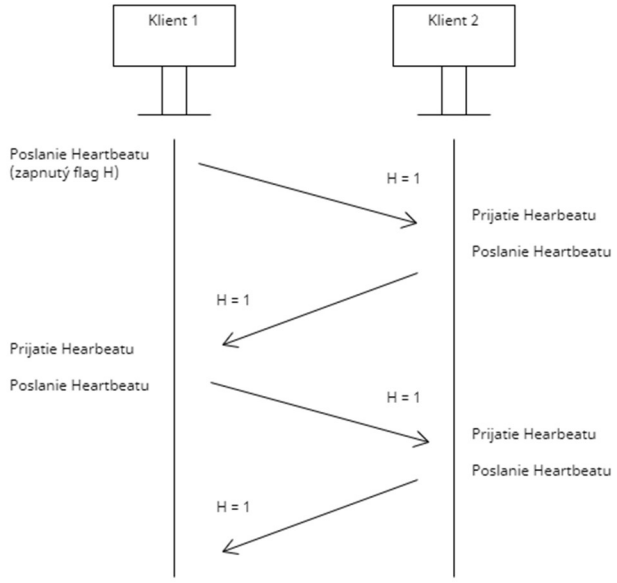
1. FIN: Jeden z Klientov pošle požiadavku na ukončenie spojenia, teda paket s nastaveným Flagom T v hlavičke
2. ACK-FIN: Druhý Klient dostane požiadavku na ukončenie a potvrdí ju odoslaním paketu s nastavenými Flagmi T a A v hlavičke. Odvtedy už prvý Klient nemôže posielať správy
3. FIN: Druhý Klient následne posiela požiadavku na ukončenie spojenia aj z jeho strany, teda paket so zapnutým T Flagom v hlavičke.
4. ACK-FIN: Prvý Klient odpovie potvrdením v podobe paketu s nastavenými Flagmi T a A, čím sa spojenie celkovo ukončí.

Proces je znázornený nasledujúcim diagramom:



2.1.2.3. Udržanie spojenia

Pre udržanie aktívneho spojenia a zistenie, či je druhá strana stále prítomná, sa budú periodicky, každých 5 sekúnd, odosielať Heartbeat pakety z oboch zariadení. Tieto pakety budú obsahovať aktívny Flag H a nebudú v nich žiadne dáta. Ak jedna strana nedostane odpoveď na niekoľko (3) Heartbeat paketov po sebe, spojenie sa bude považovať za prerušené a bude sa vyžadovať opätovné nadviazanie spojenia alebo ukončené, o čom bude zobrazená v rámci daného uzla informácia.



2.1.3. Posielanie správy a súboru

Navrhnutý protokol bude podporovať odosielanie správ a súborov. Pre správu fragmentácie bude slúžiť Flag F. Veľkosť fragmentu, bude možné nastaviť dynamicky, pričom proces fragmentácie bude závisieť od toho, či je správa alebo súbor väčší ako zadaná veľkosť fragmentu. Používateľ má možnosť si zvoliť, či chce poslať správu alebo súbor.

1. Odosielanie správ

Textové správy, menšie ako nastavená veľkosť fragmentu sa odošlú ako jeden paket, v ktorého hlavičke bude Flag F vypnutý ako znak nefragmentovania. Správy bude možné odosielať po voľbe používateľom.

2. Odosielanie súborov

Pri odosielaní súborov alebo správ presahujúcich veľkosť jedného fragmentu sa použije fragmentácia, teda rozdelenie súborov na fragmenty, ktoré budú odoslané postupne nasledujúcim spôsobom:

- Nastavenie fragmentácie:
  Vyberie sa súbor na odoslanie a skontroluje sa jeho veľkosť. Tá sa porovná s maximálnou povolenou veľkosťou fragmentu zadanou od používateľa. Ak je potrebné fragmentovať nastaví sa Flag F. Následne sa vypočíta počet fragmentov ich veľkosť (veľkosť posledného fragmentu môže byť iná).

- Odoslanie informatívnej správy
  V tomto kroku sa odošle prvotná správa s informáciami o súbore. Bude obsahovať: Názov súboru, veľkosť súboru, počet fragmentov, veľkosť fragmentov a cestu k súboru. (Pri správe počet a veľkosť fragmentov)

- Fragmentácia a odoslanie paketov
  Súbor sa rozdelí na fragmenty podľa už prebehnutého výpočtu. Rozdelenie bude pravdepodobne prebiehať postupným čítaním súboru a „ukrajovaním“ podľa nastavenej veľkosti fragmentu. Po vytvorení fragmentu sa fragment spracuje, zabalí do paketu, ten sa uloží do pamäte alebo fronty a pošle. Potom sa bude pokračovať načítaním ďalšej časti dát, a tak ďalej. Pri poslednom fragmente sa načíta už len toľko dát, koľko ostane.
  Každý paket, bude mať takto nastavenú hlavičku:
  - Sequence number – poradové číslo paketu daného súboru
  - Flag F = 1, D =1 (ide o file)
  - Length of data – veľkosť fragmentu
  - Checksum – kontrolný súčet dát

- Záznam o odoslaní
  Pri odosielaní súboru sa na odosielajúcom uzle vypíše: názov súboru, celková veľkosť súboru, počet odoslaných fragmentov a ich veľkosť

3. Prijímanie správ a súborov

Pri správach bez fragmentovania prijímajúci uzol spracuje prijatú správu a vypíše ju na obrazovku.
Pri správach alebo súboroch s potrebou fragmentovania bude prijímajúci uzol prijímať jednotlivé pakety a zobrazovať informácie o prijatí. Tie budú obsahovať: poradové číslo fragmentu a informáciu o bezchybnom prijatí (pomocou kontroly integrity – Checksum).
Prijímané fragmenty sa budú ukladať, pričom ak bude v procese nejaký chýbať, bude sa žiadať o jeho opätovné poslanie.
Po prijatí všetkých fragmentov sa súbor zostaví na základe sekvenčných čísel a po úspešnom zostavení sa na prijímajúcom uzle zobrazia informácie: Správa o úspešnom prijatí súboru, celková veľkosť prijatého súboru, čas trvania prenosu, absolútna cesta uloženia.

2.1.4. Kontrola poškodenia a strát dát

Na zaistenie spoľahlivého prenosu dát bude v protokole použitá metóda Selective Repeat, jeho fungovanie je opísané nasledovne:

- Odoslanie fragmentov
  Fragmenty sa budú odosielať postupne podľa posuvného okna Window, teda pošle sa niekoľko fragmentov naraz bez nutnosti potvrdenia.

- Prijímanie fragmentov
  Prijímateľ bude kontrolovať každý prijatý paket na základe sequence number a uloží ho na správne miesto v rámci celku
  Ak dostane fragmenty v rámci okna Window mimo poradia, teda niektorý chýba, fragmenty sa dočasne uložia a čaká sa na doručenie chýbajúcich fragmentov. Ak je fragment stratený alebo poškodený, odošle sa negatívne potvrdenie pomocou Flagu N s Ack_n - číslom daného paketu

- Potvrdzovanie paketov
  Prijímateľ odosiela pozitívne potvrdenia Ack_n s číslom fragmentu, pre každý fragment, ktorý bol prijatý bez chyby.

- Opätovné odosielanie poškodených paketov
  Keď odosielateľ dostane negatívne potvrdenie pre konkrétny fragment, posiela ho opätovne.

- Kontrola integrity dát
  Na kontrolu integrity prenášaných dát je v hlavičke protokolu vyhradené pole pre kontrolný súčet (Checksum). Tento kontrolný súčet sa vypočíta z dát, ktoré sa majú odoslať, a vloží sa do hlavičky správy. Po prijatí daného paketu na prijímacej strane sa kontrolný súčet vypočíta z prijatých dát rovnakým spôsobom, ako bol vypočítaný na odosielajúcej strane.

Postup výpočtu kontrolného súčtu:

- Prevod dát na bajty: Všetky dáta, ktoré sa majú skontrolovať, sa prevedú na bajty.
- Súčet bajtov: Každý bajt sa postupne pripočíta do súčtu.
-Zápis kontrolného súčtu: Vypočítaný súčet sa zapíše do príslušného poľa hlavičky pred odoslaním paketu.

Po prijatí paketu prijímateľ znovu vypočíta kontrolný súčet z prijatých dát. Ak sa tento kontrolný súčet nezhoduje s hodnotou v hlavičke, znamená to, že došlo k poškodeniu alebo strate dát. V takom prípade prijímateľ vyšle negatívne potvrdenie, čím požiada odosielateľa o opätovné odoslanie chybného fragmentu.

- Simulácia poškodenia dát
  Simulácia poškodenia bude pravdepodobne prebiehať tak, že pred odoslaním sa z dát vypočíta kontrolný súčet Checksum, ktorý sa ešte pred jeho vložením do hlavičky nejak modifikuje na nesprávny. Po prijatí dát prijímateľom sa opäť vypočíta tento kontrolný súčet, ktorý sa však nebude zhodovať s modifikovaným Checksumom v hlavičke. To bude znak toho, že dáta nie sú v poriadku a bude sa postupovať už podľa vyššie opísaného riešenia.

2.2. Špecifikácia implementačného prostredia

Projekt bude implementovaný v programovacom jazyku Python, ktorý je vhodný pre prácu so sieťovými protokolmi. Pre implementáciu budú použité , resp. zatiaľ sú použité, nasledovné knižnice:

- socket – knižnica pre prácu s UDP protokolom, umožňuje jednoduché vytváranie socketov, odosielanie/prijímanie správ
- treading – knižnica na vytváranie vlákien, čo umožní napr. súčasné prijímanie a odosielanie správ
- struct – knižnica, ktorá umožňuje prácu s binárnymi dátami, teda umožní napr. prácu s hlavičkou (jej prenos)

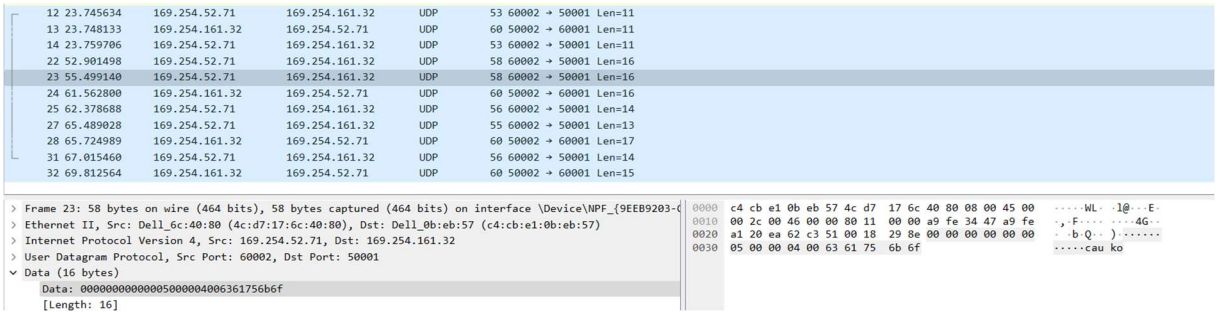
2.3. Funkčnosť odovzdaného programu

Odovzdaný program spĺňa funkčnosť základných podmienok, teda schopnosť nadviazania spojenia medzi zariadeniami, pomocou 3-way handshake a posielanie správ medzi Klientmi. Po spustení programu je potrebné zadať IP adresy a porty a následne majú obaja Klienti možnosť iniciovať nadviazanie spojenia. Nadviazanie prebieha procesom, ktorý je opísaný v dokumentácii vyššie. Po nadviazaní spojenia majú oba uzly možnosť zvoliť, predmet odoslania, pričom zatiaľ je funkčné len posielanie správ. Po zvolení M môžu Klienti posielať a prijímať ľubovoľný počet správ. Informácie o ich odoslaní a prijatí sú vypisované v termináli. Ak je napísaná správa Exit, posielanie správ je ukončené a klient dostane opäť na výber.

Nasledujúce prílohy zobrazujú zachytený prenos vo Wiresharku:

Inicializácia Handshaku:

Zachytená správa:



3. Zmeny v realizácii návrhu z kontrolného bodu

3.1. Štruktúra hlavičky

Z hlavičky bolo odstránené pole Window, nakoľko sa preň počas implementácie nenašlo využitie. Window sa využíva pri metóde Selective Repeat, kde značí koľko paketov sa môže odoslať naraz bez potvrdenia spojenia a ohraničuje pakety pre potvrdeie doručenia. Na to však bola dostatočná lokálna premenná v kóde.

1B								1B								1B								1B							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Sequence Number																Acknowledgment Number															
Flags								Length of data																Checksum							
Checksum (cont.)																															

Tabuľka č.2

V rámci Flagov sa našlo využitie pre posledný Flag R, ktorý signalizuje, že ide o informačnú správu, ktorá sa posiela pred každým súborom, a rovnako pred fragmentovanou textovou správou.
R – signalizácia informačnej správy
Implementovaná hlavička má nakoniec 9 B a zabezpečuje potrebnú funkcionalitu.

Využitie jednotlivých polí hlavičky pri rôznych mechanizmoch prenosu znázorňuje tabuľka č. 1:

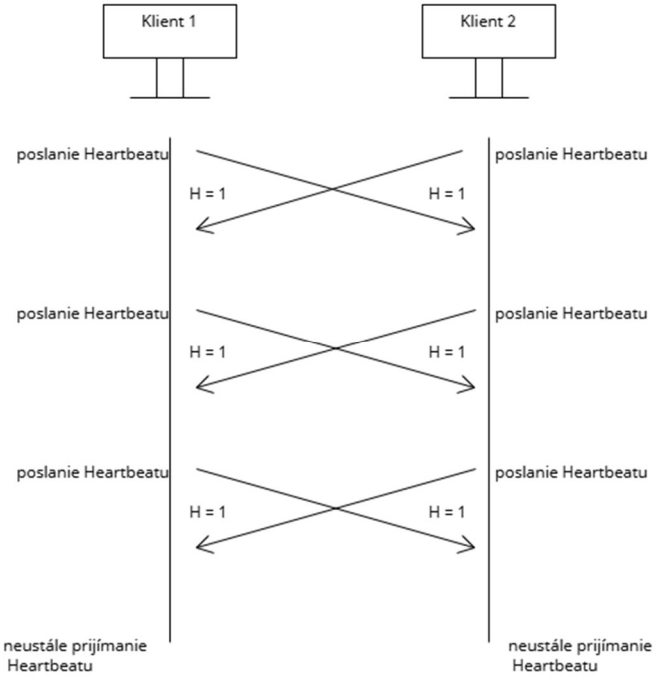
	Seq	Ack	Flags								Length of data	Checksum	Dáta
			S	A	N	H	F	D	T	R			
Handshake													
Žiadosť	0	0	1	0	0	0	0	0	0	0	0	0	
Akceptácia žiadosti	0	0	1	1	0	0	0	0	0	0			
Akceptácia	0	0	0	1	0	0	0	0	0	0	0	0	
Termination													
Žiadosť	0	0	0	0	0	0	0	0	1	0	0	0	
Akceptácia	0	0	0	1	0	0	0	0	1	0	0	0	
Heartbeat													
Posielanie Heartbeatu	0	0	0	0	0	1	0	0	0	0	0	0	
Pozastav Heartbeat	0	0	0	0	0	1	0	0	1	0	0	0	
Informačná správa													
Súbor bez fragmentácie	0	0	0	0	0	0	0	1	0	1	veľkosť dát	0	názov súboru
Súbor s fragmentáciou	veľkosť fragmentu	veľkosť posledného fragmentu	0	0	0	0	1	1	0	1	prvé seq number	počet fragmentov	názov súboru
Správa s fragmentáciou	veľkosť fragmentu	veľkosť posledného fragmentu	0	0	0	0	1	0	0	1	prvé seq number	počet fragmentov	"message"
Odoslanie													
Súbor bez fragmentácie	sekvenčné číslo	0	0	0	0	0	0	1	0	0	veľkosť dát	checksum z dát	dáta
Súbor s fragmentáciou	sekvenčné číslo	0	0	0	0	0	1	1	0	0	veľkosť dát	checksum z dát	dáta
Správa bez fragmentácie	sekvenčné číslo	0	0	0	0	0	0	0	0	0	veľkosť dát	checksum z dát	dáta
Správa s fragmentáciou	sekvenčné číslo	0	0	0	0	0	1	0	0	0	veľkosť dát	checksum z dát	dáta
Potvrdenie													
Pozitívna (bez fragmentovania)	0	ack číslo potvrdenia	0	1	0	0	0	0	0	0	0	0	
Negatívna (bez fragmentovania)	0	ack číslo potvrdenia	0	0	1	0	0	0	0	0	0	0	
Pozitívna (fragmentovanie)	0	ack číslo potvrdenia	0	1	0	0	1	0	0	0	0	0	
Negatívna (fragmentovanie)	0	ack číslo potvrdenia	0	0	1	0	1	0	0	0	0	0	

Tabuľka č.3

3.2. Funkcionalita spojenia

3.2.1. Udržanie spojenia

Pre udržanie spojenia sa periodicky, každých 5 sekúnd odosielajú z oboch zariadení pakety, so zapnutým Flagom H. Zariadenia však na tieto pakety neodpovedajú, iba si uchovávajú informáciu o počte neprijatých Heartbeat paketov od druhého zariadenia (počet sa zvýši ak v priebehu 5 sekúnd nepríjme žiaden heartbeat paket a vynuluje sa ak nejaký príjme). Ak tento počet presiahne číslo 3, znamená to, že druhé zariadenie nebolo v priebehu 15 sekúnd aktívne a teda spojenie sa prehlási za prerušené (ukončené).



Pozastavenie Heartbeatu

V priebehu implementácie bolo nutné zaviesť pozastavenie Heartbeatu v priebehu odosielania správ alebo súborov. V prípade, že si používateľ zvolí možnosť odoslania, zo zariadenia sa odošle paket, ktorý má zapnuté flagy H a T. Tento paket signalizuje druhému zariadeniu, že má pozastaviť fungovanie, teda cyklus funkcie zabezpečujúcej Heartbeat. V zariadení, z ktorého je inicializované toto pozastavenie, sa samozrejme taktiež táto funkcia a teda celkové fungovanie heartbeatu pozastaví.

Ak by počas posielania danej správy alebo súboru došlo k prerušeniu spojenia alebo k nejakej chybe, zariadenie sa pokúša opätovne túto správu, súbor alebo paket odoslať 3 krát s 5 sekundovým odstupom času. To imituje Heartbeat správy. Ak do 15 sekúnd nedostane odpoveď, spojenie sa považuje za prerušené resp. ukončené.

Akonáhle používateľ správu alebo súbor odošle a dostane sa opäť do menu s možnosťou voľby, pozastavenie Heartbeatu sa ukončí tým, že sa odošle paket s aktívnym flagom H, teda paket zabezpečujúci udržanie spojenia. Ten sa v druhom zariadení príjme, čím sa aj v ňom opäť aktivuje odosielanie Heartbeat paketov.

3.2.2. Posielanie správ a súborov

Pri posielaní správ a súborov oproti kontrolnému bodu nevznikli výrazné zmeny, no bolo by vhodné presne a konzistentnejšie opísať spôsob odosielania v implementácii.

Pred samotným odosielaní si dokáže používateľ zvoliť veľkosť fragmentu, od ktorej potom závisí, či sa správy alebo súbory budú alebo nebudú fragmentovať. Následne si zvolí súbor, vložením absolútnej cesty k nemu alebo napíše správu. Potom sa na obrazovku vypíšu základné informácie o súbore/správe (názov, veľkosť, počet fragmentov, veľkosť fragmentov)

Odosielanie správ

a) bez fragmentácie – správe, ktorá sa nefragmentuje nepredchádza informačná správa a daná správa je odoslaná ako jeden paket

b) s fragmentáciou – správe, ktorá je fragmentovaná predchádza informačná správa, ktorá prijímateľa informuje o potrebných počtoch a veľkostiach týkajúcich sa správy, tie sú zobrazené v tabuľke č. 3. Po informačnej správe sa odošlú jednotlivé fragmenty správy.

Odoslanie súborov

a) bez fragmentácie – súboru, ktorý sa nefragmentuje predchádza informačná správa a súbor je odoslaný ako jeden paket

b) s fragmentáciou - súboru, ktorý sa fragmentuje predchádza informačná správa súbor sa rozfragmentuje a jednotlivé pakety sa postupne odošlú

V procese odosielania sa vypisujú informácie o odoslaní jednotlivých paketov alebo informácie, ktoré hovoria o chybe.

Informácie o nastavení hlavičiek jednotlivých informačných správ, rovnako ako typov odosielaných dát sú zobrazené v tabuľke č. 3

3.2.3. Prijímanie správ a súborov

Pri prijímaní nastala pri implementácii zmena v tom, že sa najskôr príjmu informačné správy, a podľa nich sa prijímacie zariadenie pripraví na prijatie. Vtedy sa zároveň vypíše informácia o budúcom prijatí.
Následne sa prijímajú jednotlivé dáta a zobrazujú sa informácie o ich prijatí. Ich ktorých integrita sa kontroluje pomocou kontrolného súčtu na základe čoho sa odosielajú ack správy. Po prijatí sa súbor uloží na miesto, ktoré si používateľ určil ešte po zapnutí programu. Toto miesto uloženia sa môže v priebehu programu meniť. Správa sa vypíše na obrazovku. Po prijatí sa taktiež vypíšu informácie o prijatom súbore/správe – názov (pri súbore), celková veľkosť prijatého súboru/správy, čas trvania prenosu (pri fragmentácii), absolútna cesta uloženia (pri súbore). Pri nefragmentovanej správe sa vypíše len prijatá správa

3.2.4. Kontrola poškodenia a strát dát

Stop and wait

Pri nefragmentovaných správach a súboroch je použitá metóda S&W, ktorá funguje nasledovne:

- Odoslanie správy/súboru
- Čakanie na ACK danej správy/súboru
- Po prijatí správy/súboru prijímateľ posiela ACK paket
- Pri pozitívnom ACK, považuje prenos za úspešný
- Pri negatívnom ACK sa snaží 3 krát s časovým odstupom 5 sekúnd správu/súbor odoslať znovu

Selective Repeat

Pri fragmentovaných správach/súboroch je použitá metóda SR, ktorá funguje nasledovne:

- Odoslanie fragmentov
  Fragmenty sa odosielajú v posuvnom okne Window, ktoré sa priebežne posúva v závislosti od potvrdenia prijatia najstaršieho poslaného prvku od prijímateľa. Moje posuvné okno je o veľkosti 5. Prvých 5 fragmentov sa pošle naraz. Ďalší fragment sa pošle vždy vtedy, keď príde pozitívne Ack o najstaršom odoslanom fragmente.

- Prijímanie fragmentov
  Prijímateľ kontroluje každý prijatý paket na základe sequence number a uloží ho na správne miesto v rámci poľa prijatých fragmentov. Ak je fragment stratený alebo poškodený, odošle sa negatívne potvrdenie pomocou negatívneho Ack číslom daného paketu. Inak posiela pozitívne Ack.

- Opätovné odosielanie poškodených paketov
  Keď odosielateľ dostane negatívne potvrdenie pre konkrétny fragment, posiela ho opätovne. Toto môže prebehnúť 3 krát v dobe 15 sekúnd. (imitácia Heartbeatu). Ak sa fragment nepodarí poslať, spojenie sa považuje za ukončené.

- Kontrola integrity dát

Na kontrolu integrity prenášaných dát je v hlavičke protokolu vyhradené pole pre kontrolný súčet (Checksum). Pri implementácii nastala zmena v spôsobe výpočtu kontrolného súčtu, najmä z dôvodu nedostatočnej odolnosti pôvodnej verzie voči palindrómom.
Postup výpočtu kontrolného súčtu:
- ak je reťazec, prevedie sa na bajty pomocou kódovania UTF-8.
- Iterácia cez bajty - každý bajt sa násobí mocninou dvojky na základe jeho pozície i, hodnota sa pripočíta do výsledku
- výsledný checksum sa ohraničí bitovou maskou 0xFFFF, aby zostal v 16-bitovom rozsahu

3.2.5. Simulácia poškodenia dát

Poškodenie dát je možné simulovať pri správach aj súboroch. Je zabezpečené tak, že do checksumu správy alebo súboru je vnesená chyba, avšak iba pri prvotnom odoslaní. Po opätovnom odoslaní je už správa/súbor/fragment korektný.

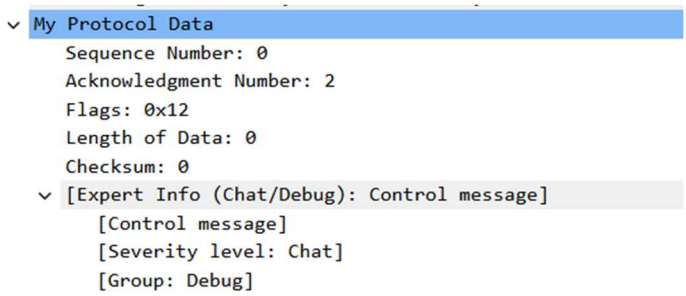
3.2.6. Špecifikácia implementačného prostredia

Projekt je implementovaný v jazyku Python. Pre implementáciu sú použité nasledujúce knižnice:

- socket – knižnica pre prácu s UDP protokolom, umožňuje jednoduché vytváranie socketov, odosielanie/prijímanie správ
- treading – knižnica na vytváranie vlákien, čo umožní napr. súčasné prijímanie a odosielanie správ
- struct – knižnica, ktorá umožňuje prácu s binárnymi dátami, teda umožní napr. prácu s hlavičkou (jej prenos)
- os – knižnica umožňuje získanie názvu súboru z cesty
- time -knižnica umožňuje prácu s časovačmi

3.2.7. Lua skritpt

K programu bolo potrebné vytvoriť Lua skript. Mnou vytvorený Lua skript dokáže vo Wiresharku identifikovať môj protokol podľa konkrétnych portov (50001,50002, 60001, 6002), dekódovať jednotlivé polia protokolu, tie zobrazuje nasledovne:



Zároveň dokážem rozlíšiť (farebne) režijné správy od dátových:

403 22.419050	10.10.35.46	10.10.35.46	My Pro…	41	Control message
404 22.419284	10.10.35.46	10.10.35.46	My Pro…	41	Control message
405 22.439381	10.10.35.46	10.10.35.46	My Pro…	41	Control message
406 22.439735	10.10.35.46	10.10.35.46	My Pro…	41	Control message
407 22.487457	10.10.35.46	10.10.35.46	My Pro…	41	Control message
414 27.440085	10.10.35.46	10.10.35.46	My Pro…	41	Control message
415 27.488173	10.10.35.46	10.10.35.46	My Pro…	41	Control message
416 27.905240	10.10.35.46	10.10.35.46	My Pro…	41	Control message
971 56.196152	10.10.35.46	10.10.35.46	My Pro…	104	Control message
978 57.197733	10.10.35.46	10.10.35.46	My Pro…	1041	Data message
979 57.199419	10.10.35.46	10.10.35.46	My Pro…	1041	Data message
980 57.199445	10.10.35.46	10.10.35.46	My Pro…	41	Control message
981 57.200472	10.10.35.46	10.10.35.46	My Pro…	41	Control message
982 57.210377	10.10.35.46	10.10.35.46	My Pro…	1041	Data message

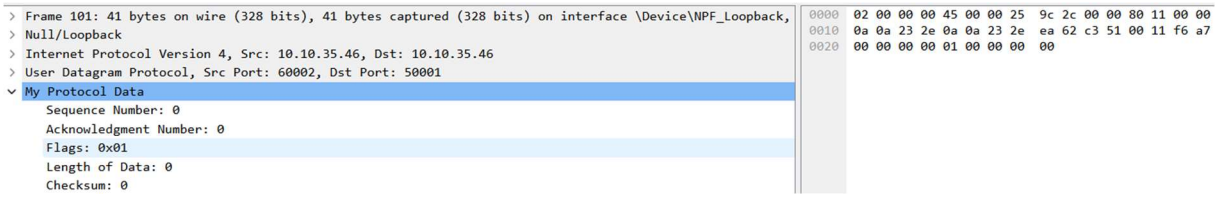
3.2.8. Funkčnosť odovzdaného programu

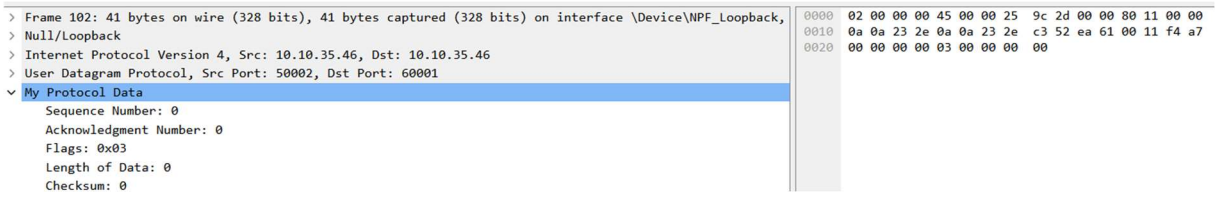
Odovzdaný program by mal spĺňať funkčnosť podmienok odovzdania.

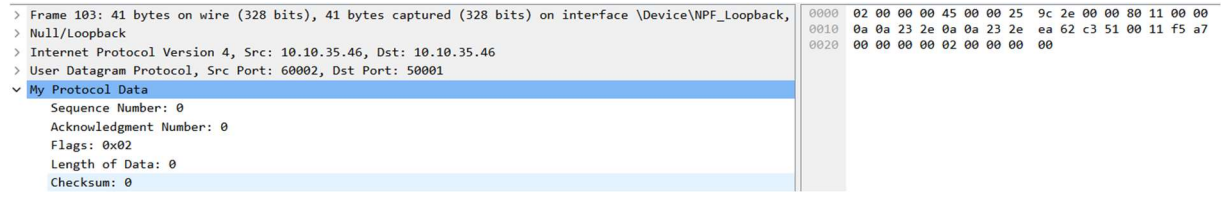
Testovací scenár (pre správu bez fragmentácie):

101	10.242089	10.10.35.46	10.10.35.46	My Pro…	41 Control message
102	10.242324	10.10.35.46	10.10.35.46	My Pro…	41 Control message
103	10.261901	10.10.35.46	10.10.35.46	My Pro…	41 Control message
104	10.262465	10.10.35.46	10.10.35.46	My Pro…	41 Control message
105	10.310203	10.10.35.46	10.10.35.46	My Pro…	41 Control message
106	13.564900	10.10.35.46	10.10.35.46	My Pro…	41 Control message
127	20.526419	10.10.35.46	10.10.35.46	My Pro…	50 Data message
128	20.526550	10.10.35.46	10.10.35.46	My Pro…	41 Control message
185	28.268317	10.10.35.46	10.10.35.46	My Pro…	41 Control message
186	28.316787	10.10.35.46	10.10.35.46	My Pro…	41 Control message
187	29.650491	10.10.35.46	10.10.35.46	My Pro…	41 Control message
728	42.450435	10.10.35.46	10.10.35.46	My Pro…	56 Data message
729	42.453016	10.10.35.46	10.10.35.46	My Pro…	41 Control message
748	47.451743	10.10.35.46	10.10.35.46	My Pro…	56 Data message
749	47.451995	10.10.35.46	10.10.35.46	My Pro…	41 Control message
1330	66.332932	10.10.35.46	10.10.35.46	My Pro…	41 Control message
1331	67.332615	10.10.35.46	10.10.35.46	My Pro…	41 Control message
1332	67.544414	10.10.35.46	10.10.35.46	My Pro…	41 Control message
1333	67.544465	10.10.35.46	10.10.35.46	My Pro…	41 Control message
1334	67.544510	10.10.35.46	10.10.35.46	My Pro…	41 Control message
1335	67.544541	10.10.35.46	10.10.35.46	My Pro…	41 Control message
1336	67.544616	10.10.35.46	10.10.35.46	My Pro…	41 Control message

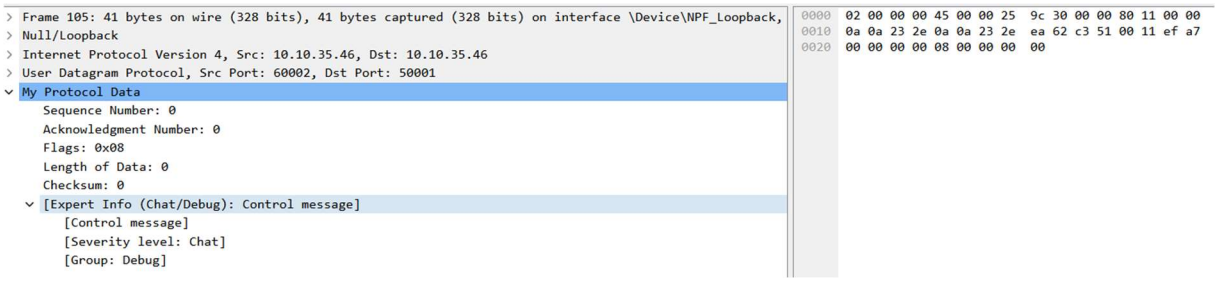
- otvorenie spojenia
  - 3-way handshake Syn, Ack-Syn, Ack



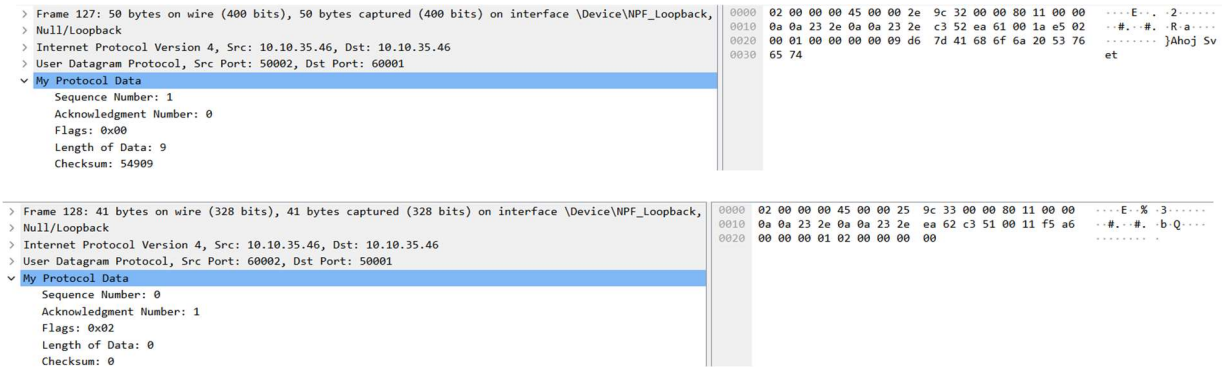




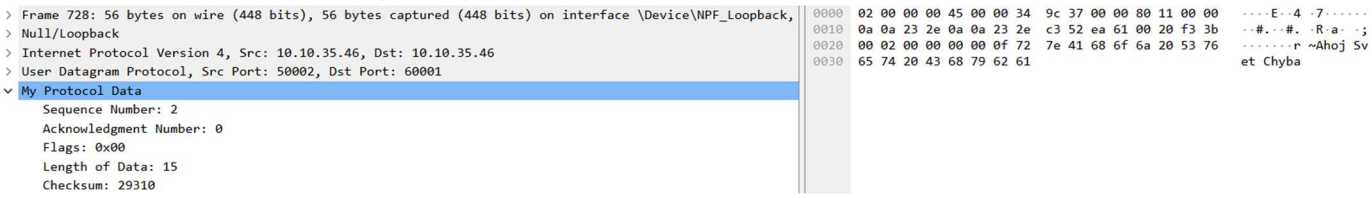
- KeepAlive -> heartbeat 2x + ukončovací heartbeat



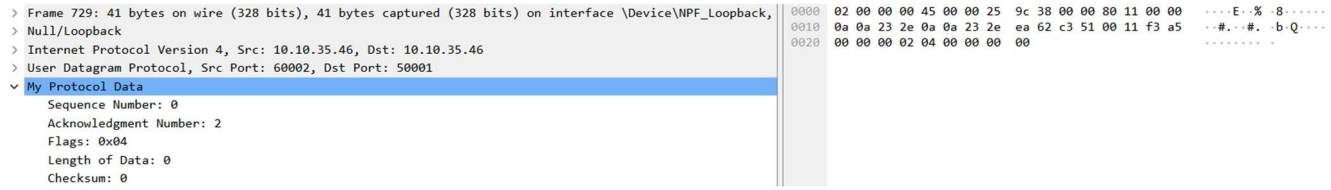
- posielanie bez chyby prenosu -> správa + pozitívne ACK



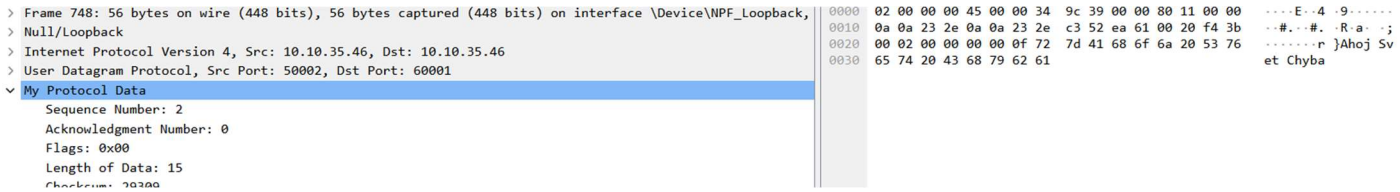
- posielanie s chybou prenosu a následnou opravou
  - Chybná správa



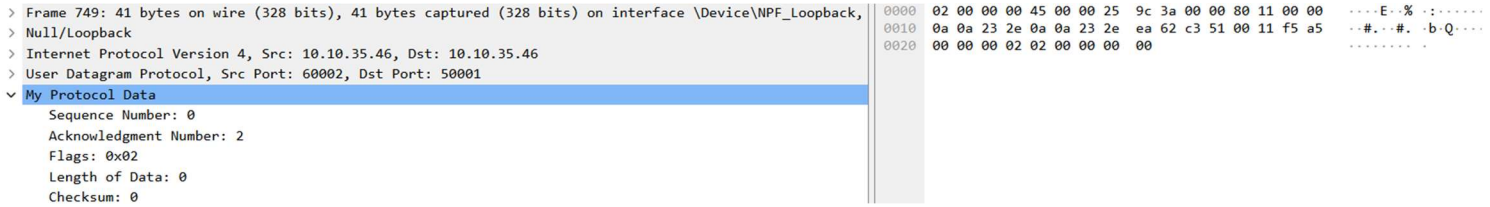
  - Negatívne ACK



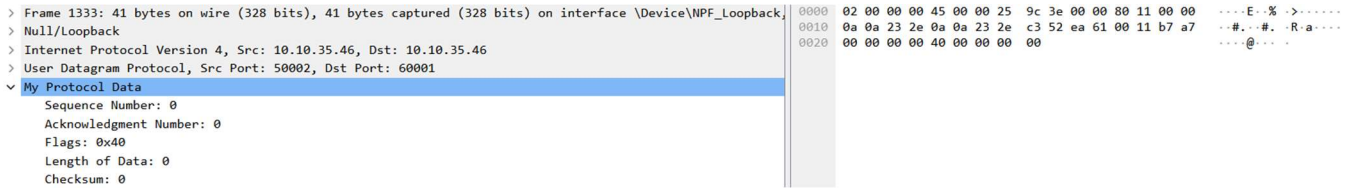
- Opravená správa



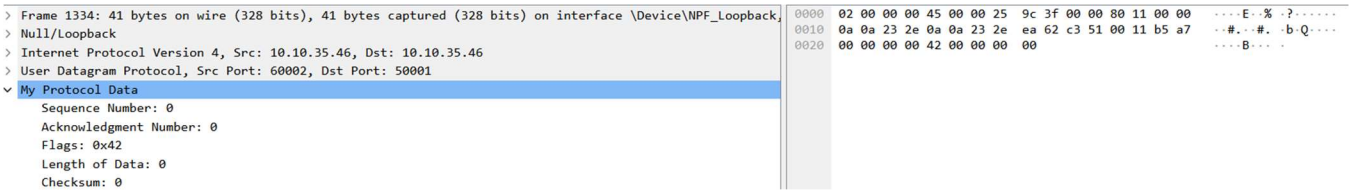
- Pozitívne ACK



- zatvorenie spojenia -> Fin + Ack-Fin (z oboch zariadení)
  - Fin

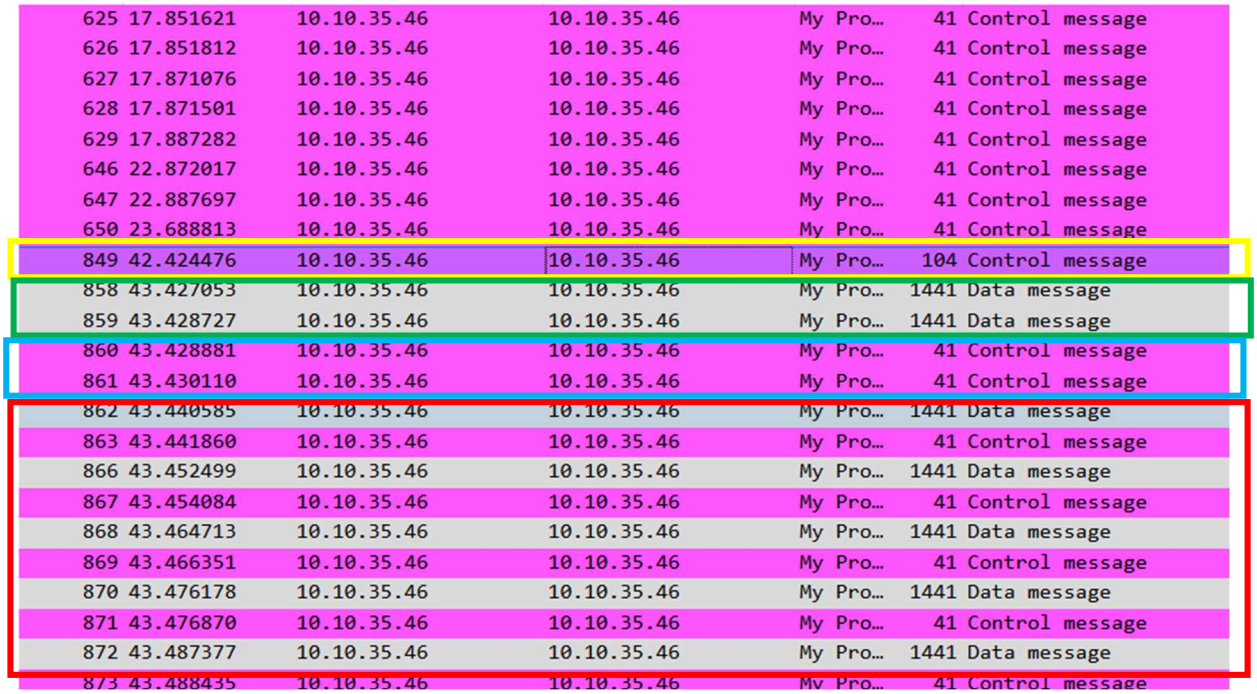


  - Ack-Fin



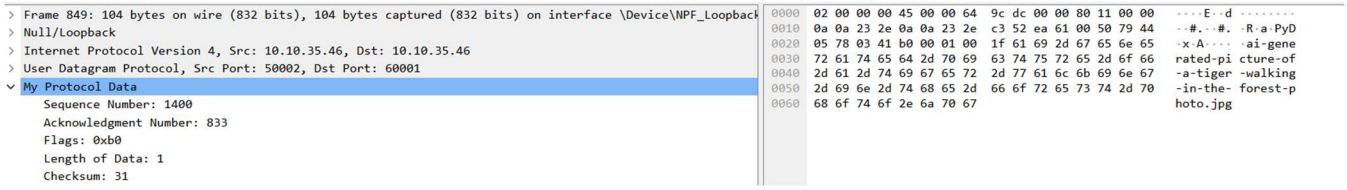
Testovací scenár (pre súbor s fragmentáciou)

- Otvorenie spojenia
  - 3-way handshake – rovnako ako v predchádzajúcom testovacom scenári
- KeepAlive
  - rovnako ako v predchádzajúcom testovacom scenári
- Posielanie bez chyby prenosu



625	17.851621	10.10.35.46	10.10.35.46	My Pro…	41 Control message
626	17.851812	10.10.35.46	10.10.35.46	My Pro…	41 Control message
627	17.871076	10.10.35.46	10.10.35.46	My Pro…	41 Control message
628	17.871501	10.10.35.46	10.10.35.46	My Pro…	41 Control message
629	17.887282	10.10.35.46	10.10.35.46	My Pro…	41 Control message
646	22.872017	10.10.35.46	10.10.35.46	My Pro…	41 Control message
647	22.887697	10.10.35.46	10.10.35.46	My Pro…	41 Control message
650	23.688813	10.10.35.46	10.10.35.46	My Pro…	41 Control message
849	42.424476	10.10.35.46	10.10.35.46	My Pro…	104 Control message
858	43.427053	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
859	43.428727	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
860	43.428881	10.10.35.46	10.10.35.46	My Pro…	41 Control message
861	43.430110	10.10.35.46	10.10.35.46	My Pro…	41 Control message
862	43.440585	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
863	43.441860	10.10.35.46	10.10.35.46	My Pro…	41 Control message
866	43.452499	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
867	43.454084	10.10.35.46	10.10.35.46	My Pro…	41 Control message
868	43.464713	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
869	43.466351	10.10.35.46	10.10.35.46	My Pro…	41 Control message
870	43.476178	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
871	43.476870	10.10.35.46	10.10.35.46	My Pro…	41 Control message
872	43.487377	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
873	43.488435	10.10.35.46	10.10.35.46	My Pro…	41 Control message

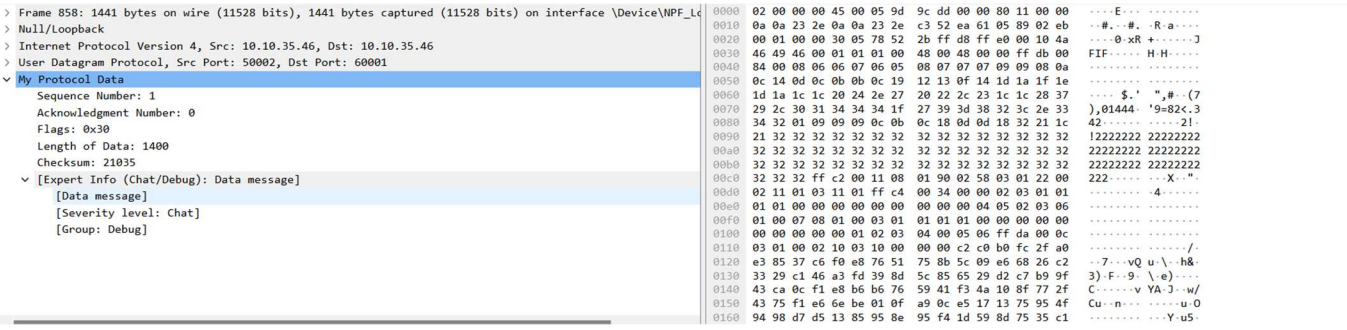
- Informačná správa



```
Frame 849: 104 bytes on wire (832 bits), 104 bytes captured (832 bits) on interface \Device\NPF_Loopback
Null/Loopback
Internet Protocol Version 4, Src: 10.10.35.46, Dst: 10.10.35.46
User Datagram Protocol, Src Port: 50002, Dst Port: 60001
My Protocol Data
    Sequence Number: 1400
    Acknowledgment Number: 833
    Flags: 0xb0
    Length of Data: 1
    Checksum: 31
```

```
0000  02 00 00 00 45 00 00 64  9c dc 00 00 80 11 00 00   ····E··d ········
0010  0a 0a 23 2e 0a 0a 23 2e  c3 52 ea 61 00 50 79 44   ··#.··#. ·R·a·PyD
0020  05 78 03 41 b0 00 01 00  1f 61 69 2d 67 65 6e 65   ·x·A···· ·ai-gene
0030  72 61 74 65 64 2d 70 69  63 74 75 72 65 2d 6f 66   rated-pi cture-of
0040  2d 61 2d 74 69 67 65 72  2d 77 61 6c 6b 69 6e 67   -a-tiger -walking
0050  2d 69 6e 2d 74 68 65 2d  66 6f 72 65 73 74 2d 70   -in-the- forest-p
0060  68 6f 74 6f 2e 6a 70 67                            hoto.jpg
```

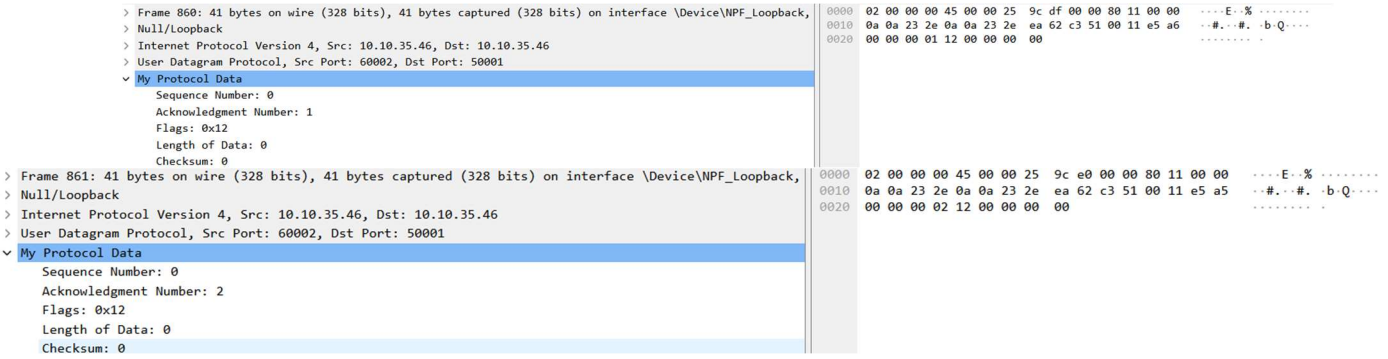
- Fragmenty v rámci okna, ktoré sa posielajú naraz (window ==0)



```
Frame 858: 1441 bytes on wire (11528 bits), 1441 bytes captured (11528 bits) on interface \Device\NPF_Lo
Null/Loopback
Internet Protocol Version 4, Src: 10.10.35.46, Dst: 10.10.35.46
User Datagram Protocol, Src Port: 50002, Dst Port: 60001
My Protocol Data
    Sequence Number: 1
    Acknowledgment Number: 0
    Flags: 0x30
    Length of Data: 1400
    Checksum: 21035
  [Expert Info (Chat/Debug): Data message]
      [Data message]
      [Severity level: Chat]
      [Group: Debug]
```

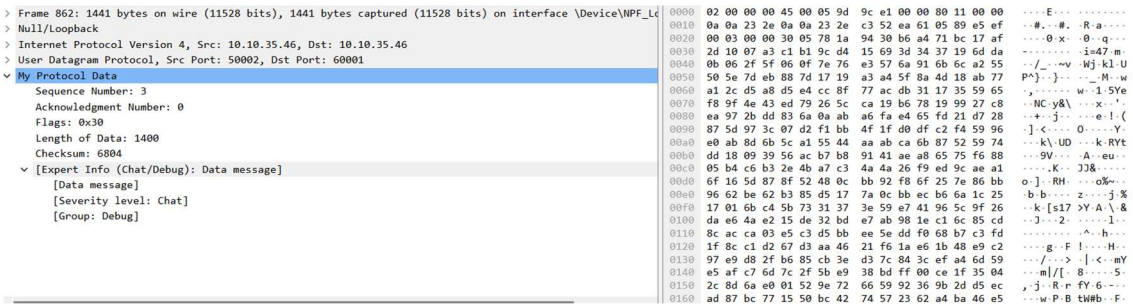
```
0000  02 00 00 00 45 00 05 9d  9c dd 00 00 80 11 00 00   ····E··· ········
0010  0a 0a 23 2e 0a 0a 23 2e  c3 52 ea 61 05 89 02 eb   ··#.··#. ·R·a····
0020  00 01 00 00 30 05 78 52  2b ff d8 ff e0 00 10 4a   ····0·xR +·····J
0030  46 49 46 00 01 01 01 00  48 00 48 00 00 ff db 00   FIF····· H·H·····
0040  84 00 08 06 06 07 06 05  08 07 07 07 09 09 08 0a   ········ ········
0050  0c 14 0d 0c 0b 0b 0c 19  12 13 0f 14 1d 1a 1f 1e   ········ ········
0060  1d 1a 1c 1c 20 24 2e 27  20 22 2c 23 1c 1c 28 37   ···· $.' ",#··(7
0070  29 2c 30 31 34 34 34 1f  27 39 3d 38 32 3c 2e 33   ),01444· '9=82<.3
0080  34 32 01 09 09 09 0c 0b  0c 18 0d 0d 18 32 21 1c   42····· ·····2!·
0090  21 32 32 32 32 32 32 32  32 32 32 32 32 32 32 32   !2222222 22222222
00a0  32 32 32 32 32 32 32 32  32 32 32 32 32 32 32 32   22222222 22222222
00b0  32 32 32 32 32 32 32 32  32 32 32 32 32 32 32 32   22222222 22222222
00c0  32 32 32 ff c2 00 11 08  01 90 02 58 03 01 22 00   222····· ···X··"·
00d0  02 11 01 03 11 01 ff c4  00 34 00 00 02 03 01 01   ········ ·4······
00e0  01 01 00 00 00 00 00 00  00 00 00 04 05 02 03 06   ········ ········
00f0  01 00 07 08 01 00 03 01  01 01 01 00 00 00 00 00   ········ ········
0100  00 00 00 00 00 01 02 03  04 00 05 06 ff da 00 0c   ········ ········
0110  03 01 00 02 10 03 10 00  00 00 c2 c0 b0 fc 2f a0   ········ ····/·
0120  e3 85 37 c6 f0 e8 76 51  75 8b 5c 09 e6 68 26 c2   ··7···vQ u·\··h&·
0130  33 29 c1 46 a3 fd 39 8d  5c 85 65 29 d2 c7 b9 9f   3)·F··9· \·e)····
0140  43 ca 0c f1 e8 b6 b6 76  59 41 f3 4a 10 8f 77 2f   C·····v YA·J··w/
0150  43 75 f1 e6 6e be 01 0f  a9 0c e5 17 13 75 95 4f   Cu··n··· ·····u·O
0160  94 98 d7 d5 13 85 95 8e  95 f4 1d 59 8d 75 35 c1   ········ ···Y·u5·
```

- Pozitívne Ack pre prvotne poslané fragmenty

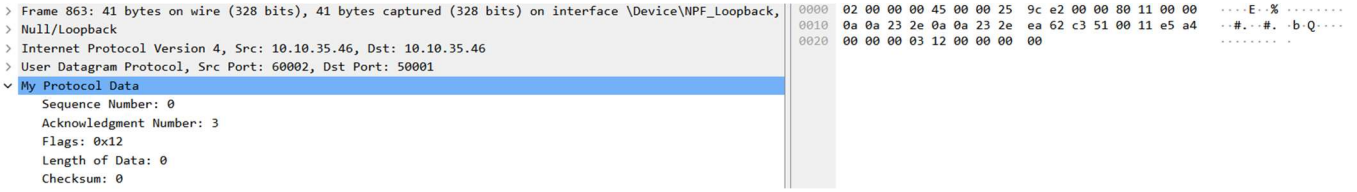


- Postupné posielanie najnovšieho fragmentu, ak je prijaté pozitívne Ack k najstaršiemu fragmentu poslanému v rámci okna (posúvanie window)

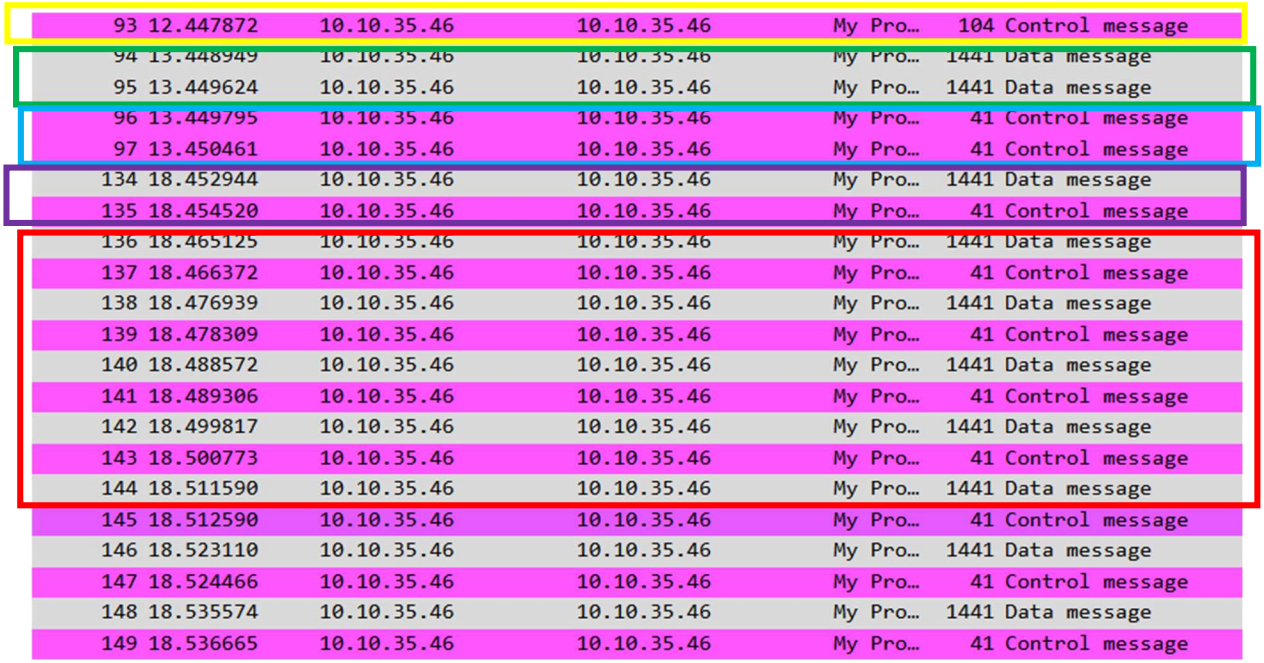
Fragment



Ack



- Posielanie s chybou prenosu



93	12.447872	10.10.35.46	10.10.35.46	My Pro…	104 Control message
94	13.448949	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
95	13.449624	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
96	13.449795	10.10.35.46	10.10.35.46	My Pro…	41 Control message
97	13.450461	10.10.35.46	10.10.35.46	My Pro…	41 Control message
134	18.452944	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
135	18.454520	10.10.35.46	10.10.35.46	My Pro…	41 Control message
136	18.465125	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
137	18.466372	10.10.35.46	10.10.35.46	My Pro…	41 Control message
138	18.476939	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
139	18.478309	10.10.35.46	10.10.35.46	My Pro…	41 Control message
140	18.488572	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
141	18.489306	10.10.35.46	10.10.35.46	My Pro…	41 Control message
142	18.499817	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
143	18.500773	10.10.35.46	10.10.35.46	My Pro…	41 Control message
144	18.511590	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
145	18.512590	10.10.35.46	10.10.35.46	My Pro…	41 Control message
146	18.523110	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
147	18.524466	10.10.35.46	10.10.35.46	My Pro…	41 Control message
148	18.535574	10.10.35.46	10.10.35.46	My Pro…	1441 Data message
149	18.536665	10.10.35.46	10.10.35.46	My Pro…	41 Control message

  - Informačná správa
  - Fragmenty v rámci okna, ktoré sa posielajú naraz (window ==0)
  - Negatívne Ack pre prvý fragment a Pozitívne Ack pre druhý fragment
  - Poslanie chybného fragmentu znovu + pozitívne Ack
  - Postupné posielanie najnovšieho fragmentu, ak je prijaté pozitívne Ack k najstaršiemu fragmentu poslanému v rámci okna (posúvanie window)

- Ukončenie spojenia
  - 4-way handshake - rovnako ako v predchádzajúcom testovacom scenári

4. Záver a zhodnotenie

Navrhnutý a implementovaný protokol nad UDP poskytuje spoľahlivú komunikáciu v lokálnej Ethernet sieti. Zahŕňa mechanizmy, ako sú handshake procesy, fragmentácia správ, spoľahlivý prenos dát pomocou Selective Repeat a Stop and Wait mechanizmov, a kontrola integrity pomocou kontrolného súčtu, ktoré by mali zabezpečiť efektívny a bezchybný prenos dát s pomerne vysokou mierou úspešnosti.

5. Zdroje

Použité zdroje boli poskytnuté v samotnom zadaní projektu:

https://www.youtube.com/watch?v=LnbvhoxHn8M

https://www.youtube.com/watch?v=WfIhQ3o2xow&t=5s

https://wiki.wireshark.org/Lua/Examples