



Síťové aplikace a správa sítí

TFTPv2 klient

David Mihola (xmihol00)

Brno, 25. října 2021

Obsah

1	Shrnutí protokolu TFTP	2
1.1	Diagram čtení souboru	3
1.2	Diagram zápisu souboru	3
1.3	Diagram čtení souboru se specifikací podmínek přenosu	3
1.4	Diagram zápisu souboru se specifikací podmínek přenosu	4
2	Implementace TFTPv2 klienta	4
2.1	Časový limit – parametr -t	4
2.2	Velikost přenášeného bloku dat --parametr -s	5
2.3	Velikost přenášeného souboru	5
2.4	Multicast	5
3	Testování	5
3.1	Test IPv4, IPv6 a správy paměti	5
3.1.1	Výpis programu	5
3.2	Test časové prodlevy a opožděného příchodu datagramu	6
3.2.1	Výpis programu	6
3.2.2	Analýza síťového provozu	7
3.3	Test obdržení nesprávného TID	7
3.3.1	Výpis programu	7
3.3.2	Analýza síťového provozu	8
3.4	Test sjednání časového limitu pro znovu odeslání datagramu	9
3.4.1	Výpis programu	9
3.4.2	Analýza síťového provozu	9
3.5	Test menší velikosti přenášeného bloku dat	10
3.5.1	Výpis programu	10
3.5.2	Analýza síťového provozu	10
3.6	Test kódování do netascii (zahrnuje znak na konci bloku dat)	10
3.6.1	Výpis programu	10
3.6.2	Kontrola obsahu souboru programem diff	11
3.6.3	Analýza síťového provozu	11
3.7	Test IPv4 multicast	11
3.7.1	Výpis programu	11
3.7.2	Analýza síťového provozu	12

1 Shrnutí protokolu TFTP

Tento dokument popisuje Trivial File Transfer Protocol (TFTP) [1] i s jeho zpětně kompatibilním rozšířením umožňující specifikaci podmínek přenosu, viz RFC 2347 [2]. Rozšiřující podmínky zahrnují mimo jiné možnost specifikovat velikost přenášeného bloku dat, viz RFC 2348 [3], časový limit pro znovu odeslání datagramu, viz RFC 2349 [4], možnost získat velikost přenášeného souboru před začátkem přenosu jeho obsahu, viz RFC 2349 [4], a možnost provádět přenos současně na více zařízení s využitím multicast [5], viz RFC 2090 [6].

Protokol slouží pouze pro přenos souborů mezi klientem a serverem. Klienti mohou ze serveru číst (stahovat) a zapisovat (nahrávat) soubory. Protokol není moc rozšířený, zejména kvůli chybějícímu zabezpečení a autentizace. Tato skutečnost prakticky vylučuje jeho použití ve veřejném internetu. Na zabezpečených lokálních sítích je jeho využití nejčastěji spojeno se zaváděním software na jednoduchá zařízení bez pevného disku. Pro tento účel je vhodný zejména díky své jednoduchosti.

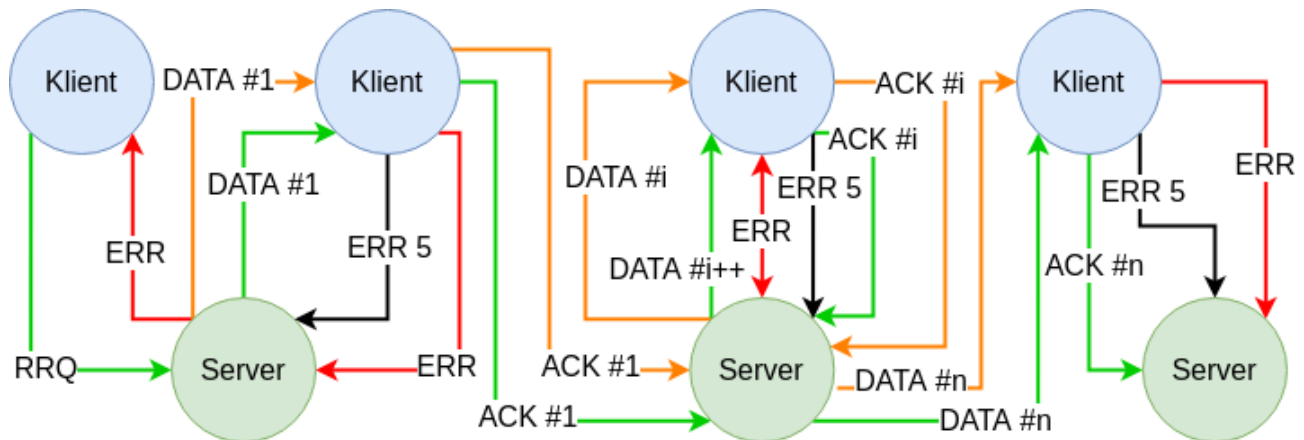
TFTP je protokol aplikační vrstvy a jako transportní vrstvu využívá Internet User Datagram Protokol (UDP) [7]. Protokol je bezztrátový, bezztrátovost zajišťuje na aplikační vrstvě podobně jako TCP [8] na vrstvě transportní, a to pomocí zpráv potvrzující doručení datagramů a jejich případného znovu odeslání, pokud není potvrzení obdrženo.

TFTP definuje pět typů hlaviček paketů, respektive šest ve své rozšířené verzi. Každá hlavička obsahuje svůj operační kód a další informace, jako např. číslo bloku, druh chyby či kódování dat, viz RFC 1350 [1]. Datové pakety navíc obsahují data o základní velikosti 512 B. Velikost přenesených dat v jednom paketu může ale být sjednána vyžádáním klienta na jinou hodnotu. V obou případech platí, že přenos končí po odeslání paketu s objemem dat menším než specifikovaná velikost a potvrzením jeho přijetí. Při velikosti souboru rovné násobku přenášených dat v jednom paketu, je ukončení přenosu souboru signalizováno datovým paketem, který nenese žádná data.

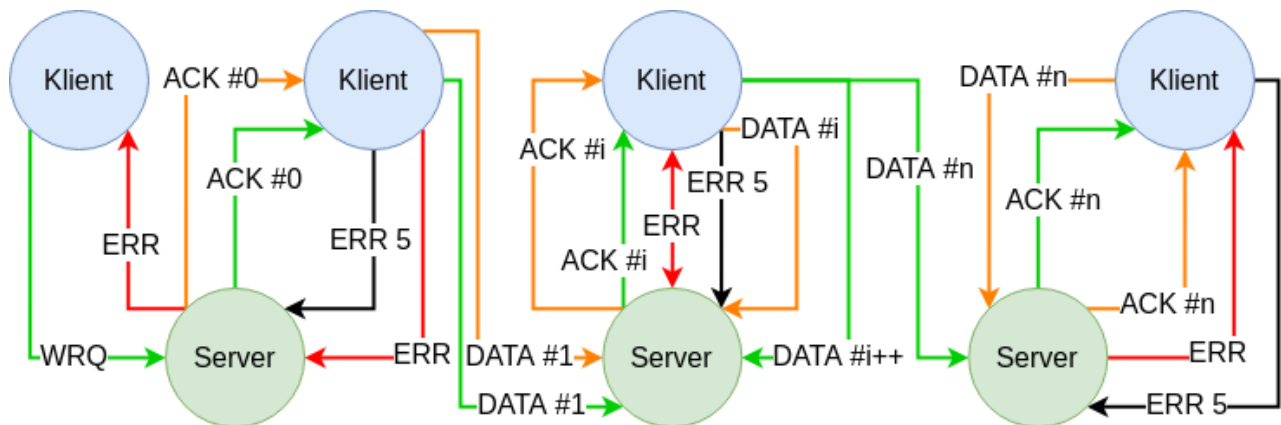
Následující sekce popisují pomocí diagramů komunikaci serveru a klienta. Diagram neznázorňuje situaci, kdy je přenesen pouze jeden datagram s daty. Pro porozumění použitým zkratkám je nutné být obeznámen s RFC 1350 [1]. Barvy šipek interpretujte následovně:

- **zelená** – normální provoz,
- **oranžová** – znovu odeslání datagramu po vypršení časového limitu,
- **černá** – chyba nevedoucí na ukončení přenosu,
- **červená** – chyba vedoucí na ukončení přenosu.

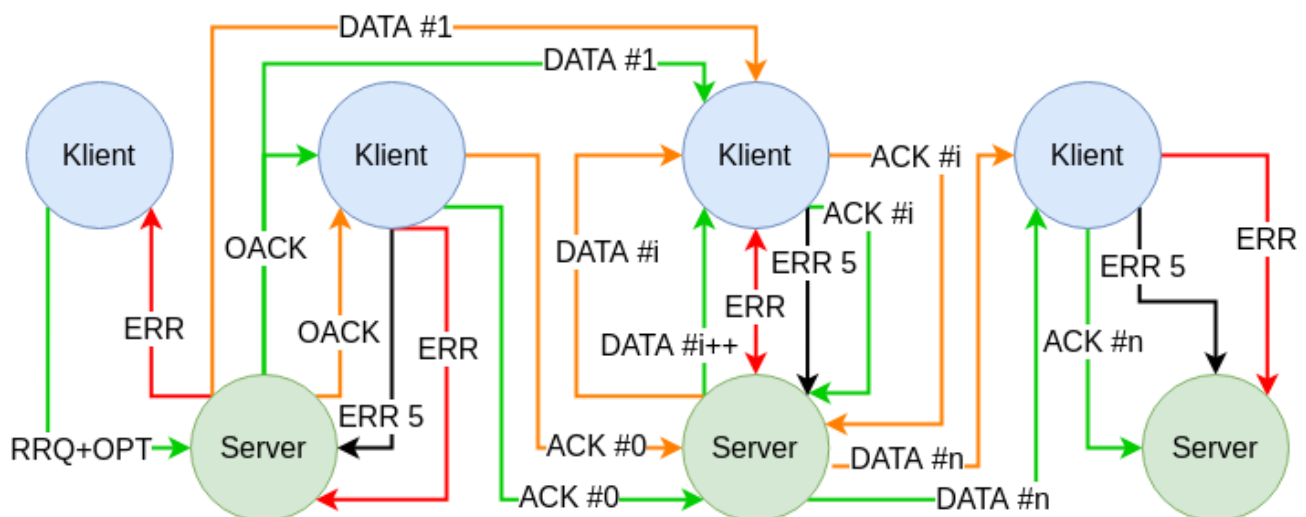
1.1 Diagram čtení souboru



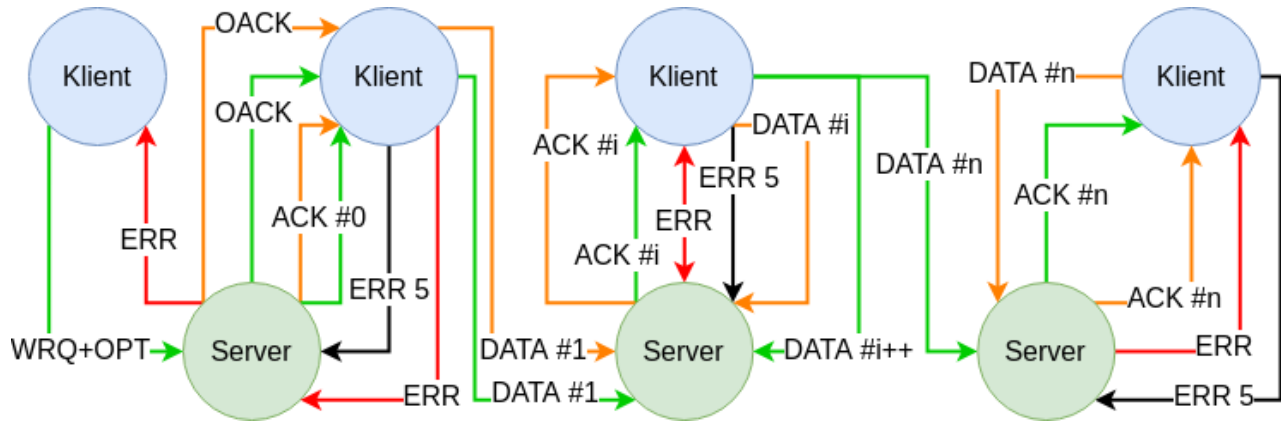
1.2 Diagram zápisu souboru



1.3 Diagram čtení souboru se specifikací podmínek přenosu



1.4 Diagram zápisu souboru se specifikací podmínek přenosu



2 Implementace TFTPv2 klienta

Implementace TFTPv2 klienta je rozdělena do následujících souborů:

- `client.{cpp,h}` – obsahují deklarace a definice funkcí pro přenos souborů. Jedná se o stěžejní soubor projektu, důležitá je funkce `transfer()`, která konfiguruje spojení se serverem, a zejména funkce `read()` a `write()`, které provádí čtení souboru ze serveru, respektive jeho zápis na server.
- `in_out.{cpp,h}` – obsahují deklarace a definice funkcí pro zpracování vstupu uživatele: `parse_line()`; výpis standardního výstupu programu: `print_summary()` a `help_msg()`; konverzi souborů do kódování *netascii*: `fread_to_netascii()` a `fwrite_from_netascii()`; aj.
- `main.{cpp,h}` – obsahují hlavní smyčku programu.
- `TFTP.{cpp,h}` – obsahují definice a deklarace funkcí pro tvorbu a zpracování TFTP paketů. Pro tvorbu paketů jsou použity funkce `RQ_header()`, `ACK_header()` a `ERR_packet()`. Pro získávání informací z přijatých paketů pak funkce `parse_OACK()` a `err_code_value()`.

Podrobnější informace o použitých funkcích lze nalézt přímo ve zmíněných souborech.

2.1 Časový limit – parametr -t

V případě, že uživatel časový limit nespecifikuje, je nastaven na 1s. Pokud jej uživatel specifikuje a server nepodporuje sjednávání podmínek přenosu nebo tuto podmínku neimplementuje, zůstává časový limit na specifikované hodnotě uživatelem. Tímto způsobem může uživatel u serverů nepodporujících sjednávání podmínek přenosu ovlivnit ukončení přenosu v případech extrémně pomalého připojení, viz následující text. Pokud server na podmínku odpoví s jinou hodnotou, než specifikoval klient, pak je použita tato hodnota, i když to dle RFC 2349 [4] není možné. V obou případech je o této skutečnosti uživatel informován výpisem. Ve všech případech je odesláno maximálně 5 stejných datagramů, pokud na žádný z nich není přijata relevantní odpověď, je usouzeno, že došlo k fatální chybě a přenos je přerušen.

2.2 Velikost přenášeného bloku dat – parametr -s

Pokud server nepodporuje sjednávání podmínek přenosu nebo tuto podmínku neimplementuje, je velikost přenášeného bloku dat nastavena na základní velikost (512 B) a uživatel je o této skutečnosti informován výpisem. Pokud server odpoví s menší hodnotou, než specifikoval klient, je použita tato hodnota. Pokud server odpoví s větší hodnotou, než specifikoval klient, což není dle RFC 2348 [3] možné, je odeslána chyba a přenos je ukončen.

2.3 Velikost přenášeného souboru

Tuto podmínku nespecifikuje uživatel. Klient ji zasílá u všech žádostí o přenos, které probíhají binární formou. U kódování ascii není možné před začátkem přenosu jednoduše zjistit velikost přenášeného souboru, proto zde podmínka není specifikována.

2.4 Multicast

Podpora multicast je implementována pouze velmi zjednodušeně. V případě specifikování přenosu přes multicast, je po odpovědi serveru s adresou a portem uzavřen aktuálně používaný soket a otevřen nový, který je nastaven přijímat datagramy na získané adrese. Vše ostatní zůstává nezměněno. Testována byla pouze IPv4 verze, implementována je i IPv6.

3 Testování

Výsledný program byl důkladně testován za použití referenční implementace TFTP klienta [9] a TFTP serveru [10]. Dále byly pro testování použity nástroje Valgrind [11], Wireshark [12] aj. Pro simulaci neobvyklých situací byly použity soubory a skripty v adresáři `tests`. Výsledky vybraných testů jsou zobrazeny v následujících sekcích. V sekci 3.2 a 3.3 si lze navíc všimnout, že implementace řešení tzv. Sorcerer's Apprentice Syndrome, který identifikuje a opravuje RFC 1350 [1]. Datagramy jsou odeslány znovu jen v případě vypršení časového limitu.

3.1 Test IPv4, IPv6 a správy paměti

3.1.1 Výpis programu

```
==42878== Memcheck, a memory error detector
==42878== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==42878== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==42878== Command: ./mytftpclient
==42878==
> -W -d file1 -a 192.168.1.147,69
Writing file 'file1' ...

Writing file 'file1' succeeded.
Transfer summary:
  - 34 ms elapsed during the transfer,
  - 737 B of data were sent in 2 datagrams of a maximum data size 512 B.
```

```
> -W -d file2 -a 2001:db8:1:1:204:76ff:fe47:c,69
Writing file 'file2' ...

Writing file 'file2' succeeded.
Transfer summary:
  - 2 ms elapsed during the transfer,
  - 873 B of data were sent in 2 datagrams of a maximum data size 512 B.

> -R -d file2 -a 192.168.1.147,69
Reading file 'file2' ...

Reading file 'file2' succeeded.
Transfer summary:
  - 10 ms elapsed during the transfer,
  - 873 B of data were recieved in 2 datagrams of a maximum data size 512 B.

> -R -d file1 -a 2001:db8:1:1:204:76ff:fe47:c,69
Reading file 'file1' ...

Reading file 'file1' succeeded.
Transfer summary:
  - 1 ms elapsed during the transfer,
  - 737 B of data were recieved in 2 datagrams of a maximum data size 512 B.

> q
==42878==
==42878== HEAP SUMMARY:
==42878==      in use at exit: 0 bytes in 0 blocks
==42878==    total heap usage: 27 allocs, 27 frees, 95,530 bytes allocated
==42878==
==42878== All heap blocks were freed -- no leaks are possible
==42878==
==42878== For lists of detected and suppressed errors, rerun with: -s
==42878== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

3.2 Test časové prodlevy a opožděného příchodu datagramu

3.2.1 Výpis programu

```
> -d file -W
Writing file 'file' ...
Warning: Server does not support Option Negotiation.

Writing file 'file' succeeded.
Transfer summary:
  - 2353 ms elapsed during the transfer,
```

- 1536 B of data were sent in 4 datagrams of a maximum data size 512 B, of which 1 datagram carrying data may have been lost of overall data size 512 B.

```
> -R -d file
```

Reading file 'file' ...

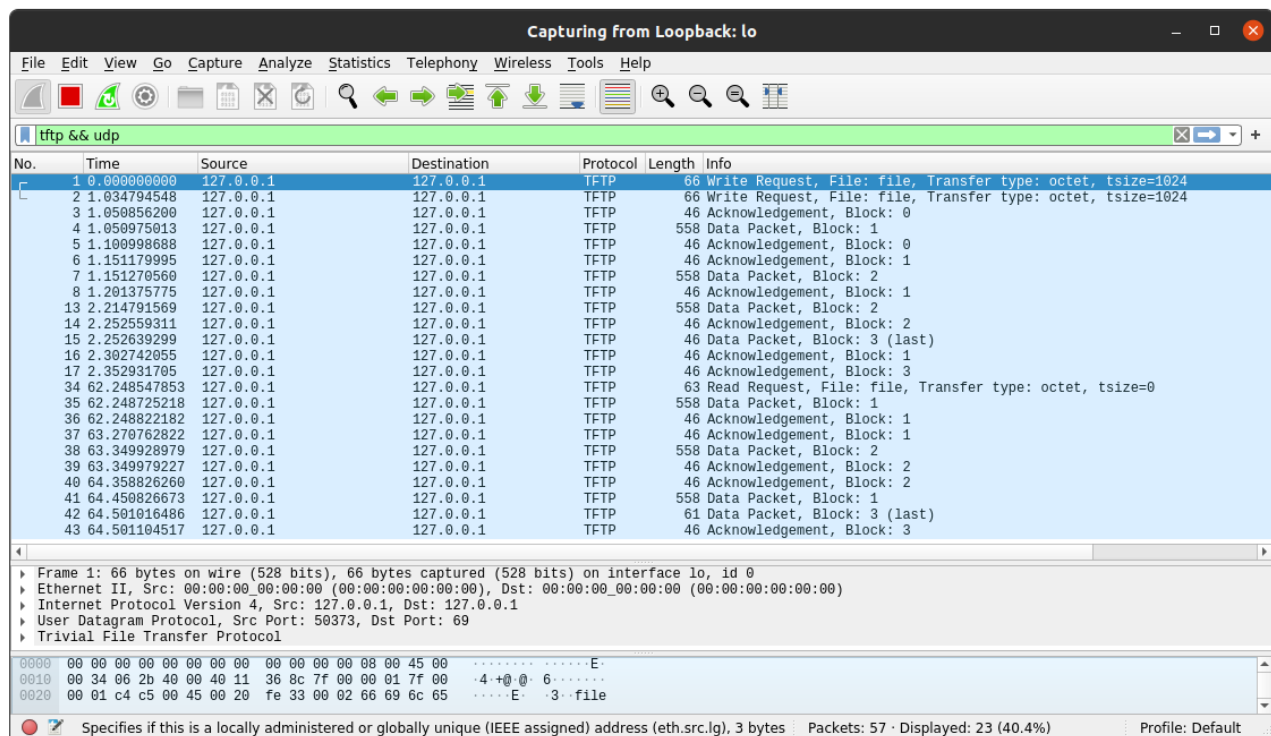
Warning: Server does not support Option Negotiation.

Reading file 'file' succeeded.

Transfer summary:

- 2252 ms elapsed during the transfer,
- 1039 B of data were recieved in 3 datagrams of a maximum data size 512 B, additional 2 datagrams carrying data may have been lost, which compounds up to 1024 B.

3.2.2 Analýza síťového provozu



3.3 Test obdržení nesprávného TID

3.3.1 Výpis programu

```
> -R -d file
```

Reading file 'file' ...

Warning: Server does not support Option Negotiation.

Warning: Recieved TID does not match the establshlished one, server informed and transfer continues.

Warning: Recieved TID does not match the estabhlshed one, server informed

and transfer continues.

Reading file 'file' succeeded.

Transfer summary:

- 151 ms elapsed during the transfer,
- 1039 B of data were recieved in 3 datagrams of a maximum data size 512 B.

> -d file -W

Writing file 'file' ...

Warning: Server does not support Option Negotiation.

Warning: Recieved TID does not match the estabhlshed one, server informed and transfer continues.

Warning: Recieved TID does not match the estabhlshed one, server informed and transfer continues.

Writing file 'file' succeeded.

Transfer summary:

- 251 ms elapsed during the transfer,
- 1024 B of data were sent in 3 datagrams of a maximum data size 512 B.

3.3.2 Analýza síťového provozu

The screenshot displays a Wireshark packet capture on the interface `*Loopback: lo`. The packet list pane shows the following entries:

No.	Time	Source	Destination	Protocol	Length	Info
19	43.336768799	127.0.0.1	127.0.0.1	TFTP	63	Read Request, File: file, Transfer type: octet, tsize=0
20	43.336867830	127.0.0.1	127.0.0.1	TFTP	558	Data Packet, Block: 1
21	43.336922692	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 1
22	43.386921241	127.0.0.1	127.0.0.1	TFTP	57	Data Packet, Block: 2 (last)
23	43.386973333	127.0.0.1	127.0.0.1	TFTP	61	Error Code, Code: Unknown transfer ID, Message: Incorrect TID.
24	43.437093508	127.0.0.1	127.0.0.1	TFTP	558	Data Packet, Block: 2
25	43.437185172	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 2
26	43.487289525	127.0.0.1	127.0.0.1	TFTP	57	Data Packet, Block: 3 (last)
27	43.487318752	127.0.0.1	127.0.0.1	TFTP	61	Data Packet, Block: 3 (last)
28	43.487415464	127.0.0.1	127.0.0.1	TFTP	61	Error Code, Code: Unknown transfer ID, Message: Incorrect TID.
29	43.487438299	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 3
54	111.799586232	127.0.0.1	127.0.0.1	TFTP	66	Write Request, File: file, Transfer type: octet, tsize=1024
55	111.799774839	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 0
56	111.799877662	127.0.0.1	127.0.0.1	TFTP	558	Data Packet, Block: 1
57	111.849856353	127.0.0.1	127.0.0.1	TFTP	57	Acknowledgement, Block: 0
58	111.849914724	127.0.0.1	127.0.0.1	TFTP	61	Error Code, Code: Unknown transfer ID, Message: Incorrect TID.
59	111.900016349	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 1
60	111.900072306	127.0.0.1	127.0.0.1	TFTP	558	Data Packet, Block: 2
61	111.950207958	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 2
62	111.950288723	127.0.0.1	127.0.0.1	TFTP	46	Data Packet, Block: 3 (last)
63	112.000371661	127.0.0.1	127.0.0.1	TFTP	57	Acknowledgement, Block: 1
64	112.000492349	127.0.0.1	127.0.0.1	TFTP	61	Error Code, Code: Unknown transfer ID, Message: Incorrect TID.
65	112.050497080	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 3

Two detailed packet views are shown at the bottom:

- Wireshark · Packet 55 · Loopback: lo**: Shows the structure of a TFTP Write Request. The User Datagram Protocol (UDP) section indicates Src Port: 5906 and Dst Port: 38969. The Trivial File Transfer Protocol (TFTP) section is also visible.
- Wireshark · Packet 57 · Loopback: lo**: Shows the structure of a TFTP Acknowledgement. The User Datagram Protocol (UDP) section indicates Src Port: 69 and Dst Port: 38969.

3.4 Test sjednání časového limitu pro znovu odeslání datagramu

3.4.1 Výpis programu

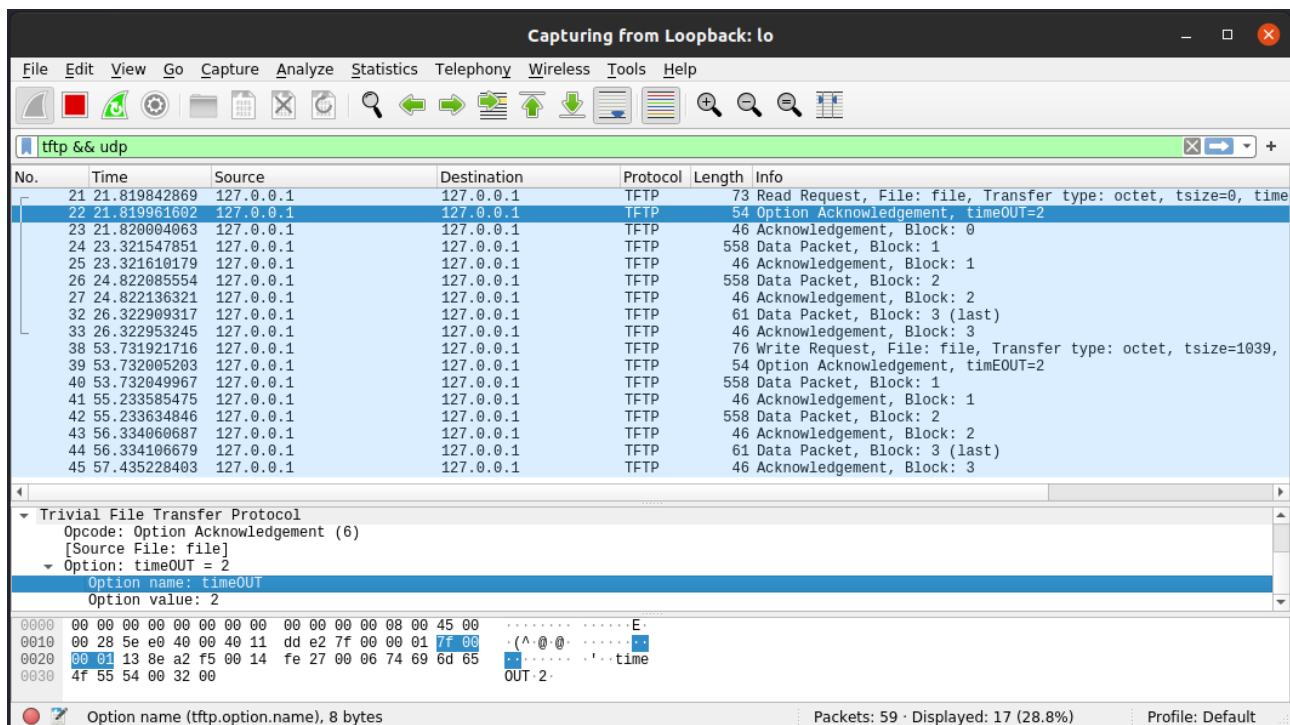
```
> -R -d file -t 2
Reading file 'file' ...
Warning: Server did not recognize transfer size option. Transfer continues.

Reading file 'file' succeeded.
Transfer summary:
  - 4503 ms elapsed during the transfer,
  - 1039 B of data were recieved in 3 datagrams of a maximum data size 512 B.
```

```
> -W -d file -t 4
Writing file 'file' ...
Warning: Server did not accept specified timeout of 4 s. Timeout specified
        by server of 2 s is used instead.
Warning: Server did not recognize transfer size option. Transfer continues.

Writing file 'file' succeeded.
Transfer summary:
  - 3703 ms elapsed during the transfer,
  - 1039 B of data were sent in 3 datagrams of a maximum data size 512 B.
```

3.4.2 Analýza síťového provozu



3.5 Test menší velikosti přenášeného bloku dat

3.5.1 Výpis programu

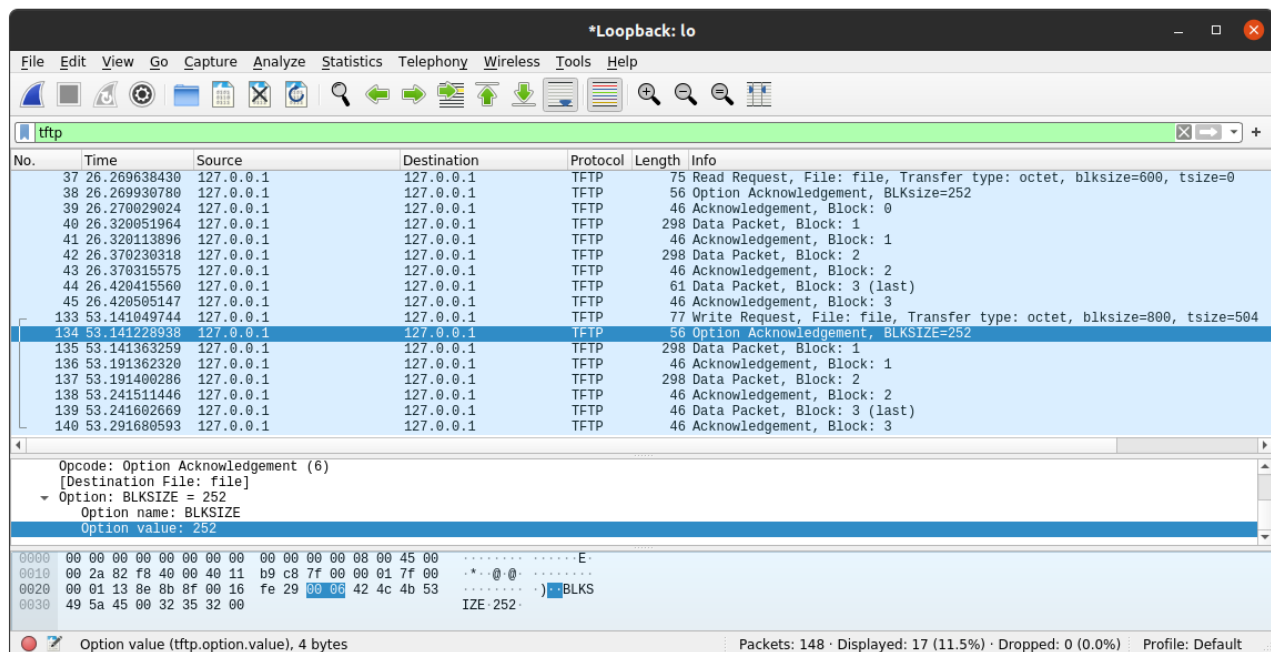
```
> -R -d file -s 600
Reading file 'file' ...
Warning: Server did not recognize transfer size option. Transfer continues.

Reading file 'file' succeeded.
Transfer summary:
  - 151 ms elapsed during the transfer,
  - 519 B of data were recieved in 3 datagrams of a maximum data size 252 B.

> -s 800 -d file -W
Writing file 'file' ...
Warning: Server did not recognize transfer size option. Transfer continues.

Writing file 'file' succeeded.
Transfer summary:
  - 151 ms elapsed during the transfer,
  - 504 B of data were sent in 3 datagrams of a maximum data size 252 B.
```

3.5.2 Analýza síťového provozu



3.6 Test kódovaní do netascii (zahrnuje znak na konci bloku dat)

3.6.1 Výpis programu

```
> -W -d tests/CR_LF_all.txt -s 77 -c ascii
Writing file 'CR_LF_all.txt' ...
```

Writing file 'CR_LF_all.txt' succeeded.

Transfer summary:

- 2 ms elapsed during the transfer,
- 94 B of data were sent in 2 datagrams of a maximum data size 77 B.

```
> -R -d tests/CR_LF_all.txt -s 77 -c netascii
```

Reading file 'CR_LF_all.txt' ...

Reading file 'CR_LF_all.txt' succeeded.

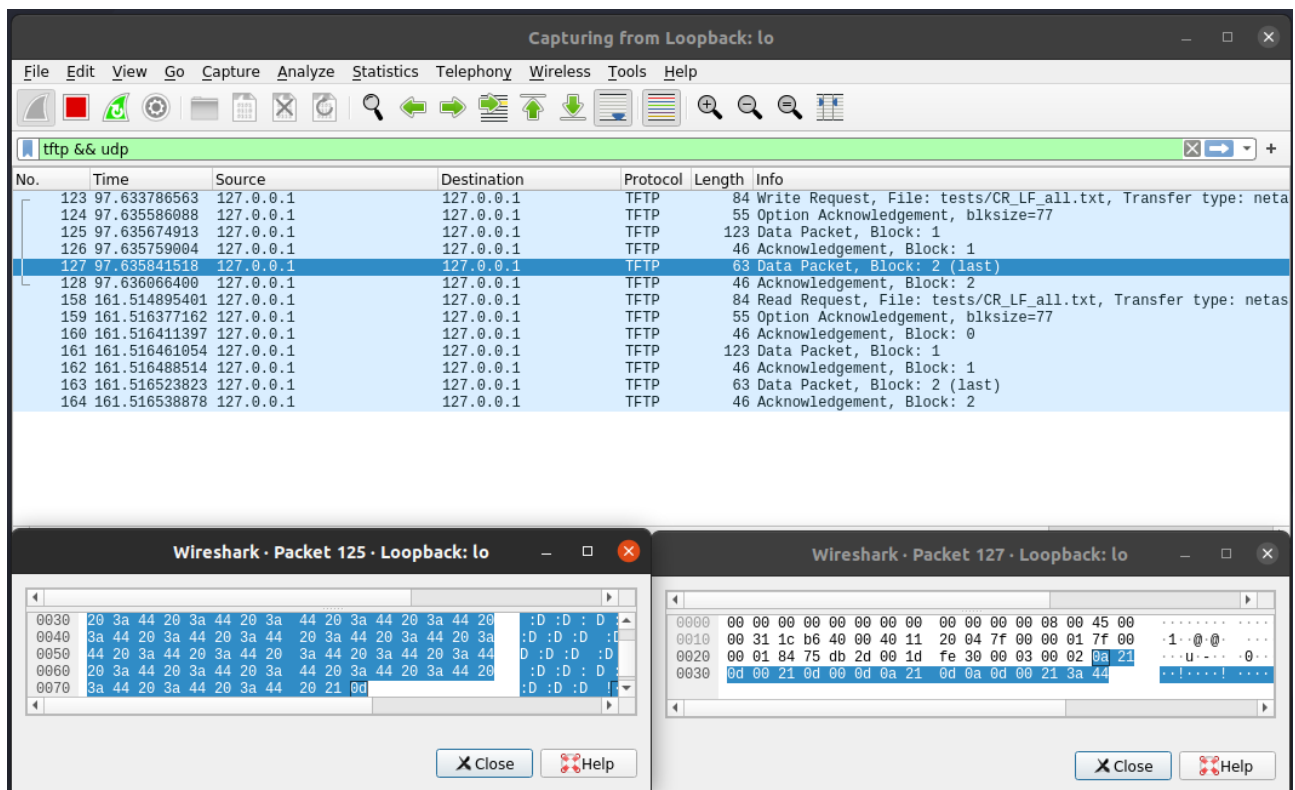
Transfer summary:

- 2 ms elapsed during the transfer,
- 94 B of data were recieved in 2 datagrams of a maximum data size 77 B.

3.6.2 Kontrola obsahu souboru programem diff

Files tests/CR_LF_all.txt and CR_LF_all.txt are identical

3.6.3 Analýza síťového provozu



3.7 Test IPv4 multicast

3.7.1 Výpis programu

```
> -R -d file -m
```

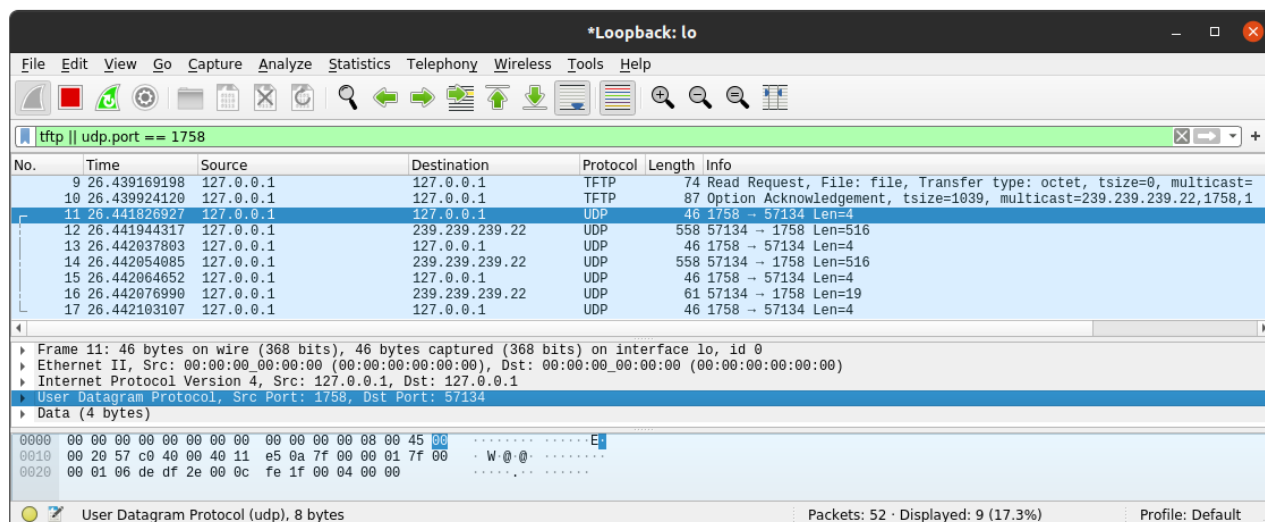
Reading file 'file' ...

Reading file 'file' succeeded.

Transfer summary:

- 1 ms elapsed during the transfer,
- 1039 B of data were recieved in 3 datagrams of a maximum data size 512 B.

3.7.2 Analýza síťového provozu



Reference

- [1] Sollins, K. *THE TFTP PROTOCOL (REVISION 2)* [online], červenec 1992, [cit. 24. 10. 2021]. Dostupné z: <https://www.rfc-editor.org/rfc/rfc1350.txt>
- [2] Malkin, G. *TFTP Option Extension* [online], květen 1998, [cit. 24. 10. 2021]. Dostupné z: <https://www.rfc-editor.org/rfc/rfc2347.txt>
- [3] Malkin G. *TFTP Blocksize Option* [online], květen 1998, [cit. 24. 10. 2021]. Dostupné z: <https://www.rfc-editor.org/rfc/rfc2348.txt>
- [4] Malkin G. *TFTP Timeout Interval and Transfer Size Options* [online], květen 1998, [cit. 24. 10. 2021]. Dostupné z: <https://www.rfc-editor.org/rfc/rfc2349.txt>
- [5] Deering S. *Host Extensions for IP Multicasting* [online], srpen 1989, [cit. 24. 10. 2021]. Dostupné z: <https://www.rfc-editor.org/rfc/rfc1112.txt>
- [6] Emberson A. *TFTP Multicast Option* [online], únor 1997, [cit. 24. 10. 2021]. Dostupné z: <https://www.rfc-editor.org/rfc/rfc2090.txt>
- [7] Postel, J. *User Datagram Protocol* [online], srpen 1980, [cit. 24. 10. 2021]. Dostupné z: <https://www.rfc-editor.org/rfc/rfc768.txt>
- [8] *TRANSMISSION CONTROL PROTOCOL* [online], září 1981, [cit. 24. 10. 2021]. Dostupné z: <https://www.rfc-editor.org/rfc/rfc793.txt>
- [9] die.net. *tftp(1) - Linux man page* [online], [cit. 24. 10. 2021]. Dostupné z: <https://linux.die.net/man/1/tftp>
- [10] die.net. *tftpd(8) - Linux man page* [online], [cit. 24. 10. 2021]. Dostupné z: <https://linux.die.net/man/8/tftpd>
- [11] Valgrind™ Developers. *Valgrind* [online], [cit. 24. 10. 2021]. Dostupné z: <https://valgrind.org>
- [12] *Wireshark* [online], [cit. 24. 10. 2021]. Dostupné z: <https://www.wireshark.org>