



Síťové aplikace a správa sítí

TFTPV2 klient

David Mihola (xmihol00)

Brno, 24. 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	5
2.1	Časový limit – parametr -t	5
2.2	Velikost přenášeného bloku dat – parametr -s	5
3	Testování	6
3.1	Test IPv4, IPv6 a správy paměti	6
3.1.1	Výpis programu	6
3.2	Test obdržení nesprávného TID	7
3.2.1	Výpis programu	7
3.2.2	Analýza síťového provozu	8
3.3	Test sjednání časového limitu pro znovu odeslání datagramu	8
3.3.1	Výpis programu	8
3.3.2	Analýza síťového provozu	9
3.4	Test menší velikosti přenášeného bloku dat	9
3.4.1	Výpis programu	9
3.4.2	Analýza síťového provozu	10
3.5	Test kódování do netascii i na konci bloku dat	10
3.5.1	Výpis programu	10
3.5.2	Kontrola obsahu souboru programem diff	10
3.5.3	Analýza síťového provozu	11

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.

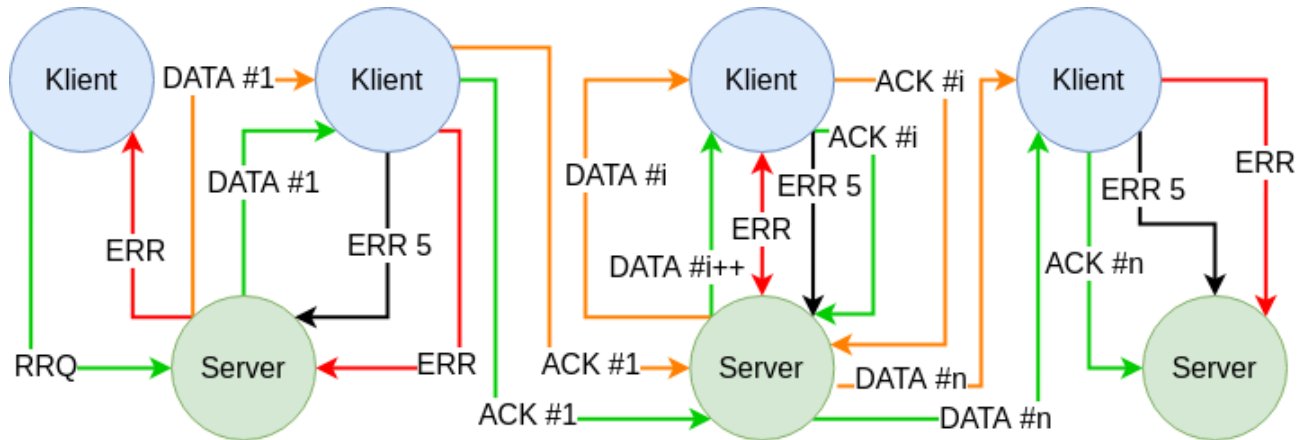
TFTP je protokol aplikační vrstvy a jako transportní vrstvu využívá Internet User Datagram Protokol (UDP) [7]. Protokol je bezztrátový, bezztrátovost zajišťuje na aplikační vrstvě podobně jako TCP [8] na vrstvě transportní, a to pomocí zpráv potvrzující doručení datagramů a jejich případného znovu odeslání,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

TFTP definuje pět typů hlaviček paketů, respektive šest ve své rozšířené verzi. Každá hlavička obsahuje svůj operační kód a případně další informace, 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. Samotná data lze zasílat binárně nebo s využitím kódování netascii TODO.

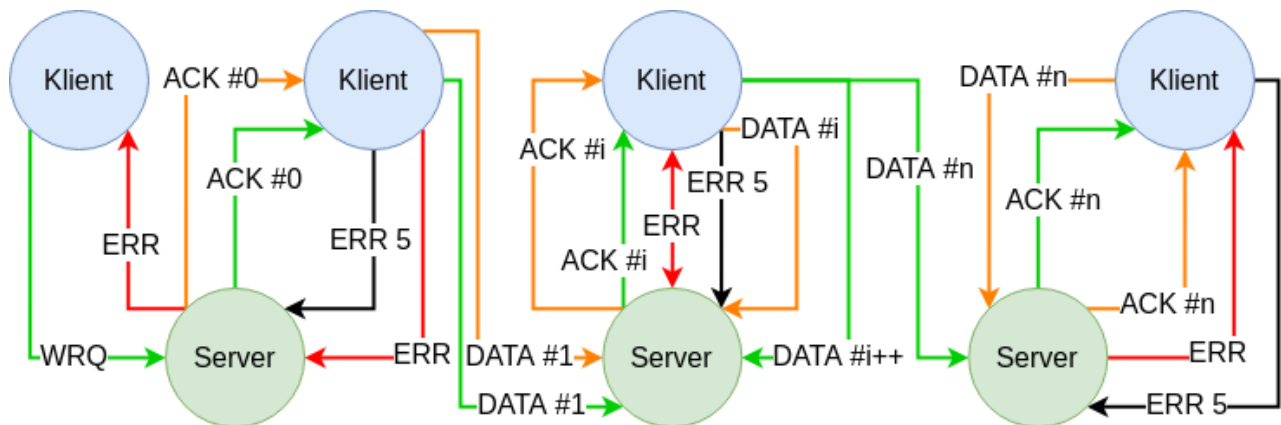
Následující sekce popisují pomocí diagramů komunikaci serveru a klienta. Diagram neznázorňuje situaci, kdy je přenesen pouze jeden datagram s daty. Pro porozumění použitým zkratkám je nutné být obeznámen s RFC 1350 [1]. Barvy šipek interpretujte následovně:

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

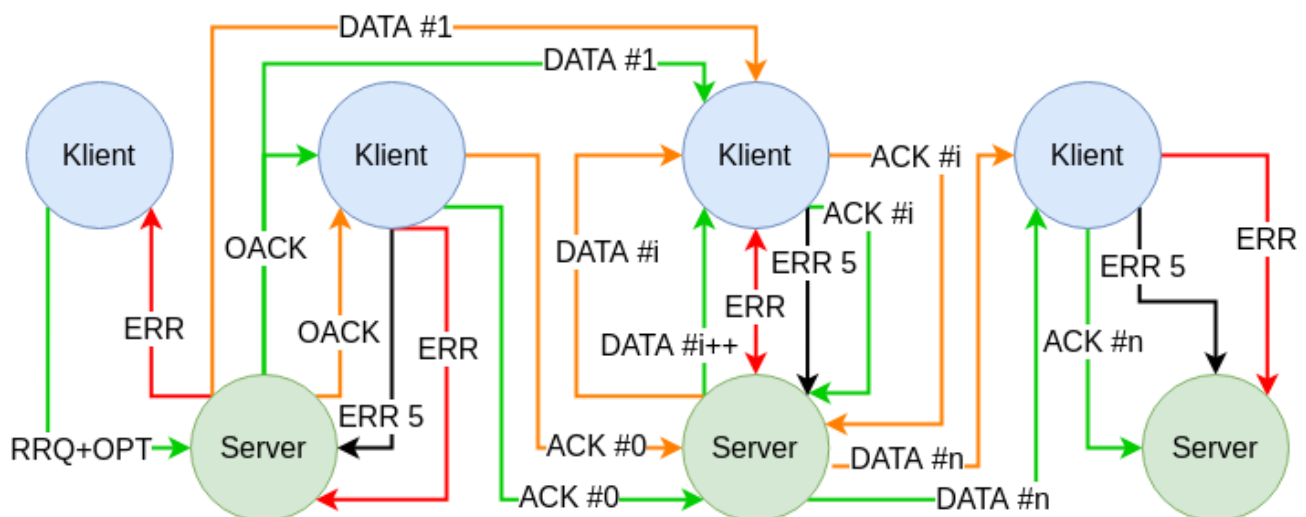
1.1 Diagram čtení souboru



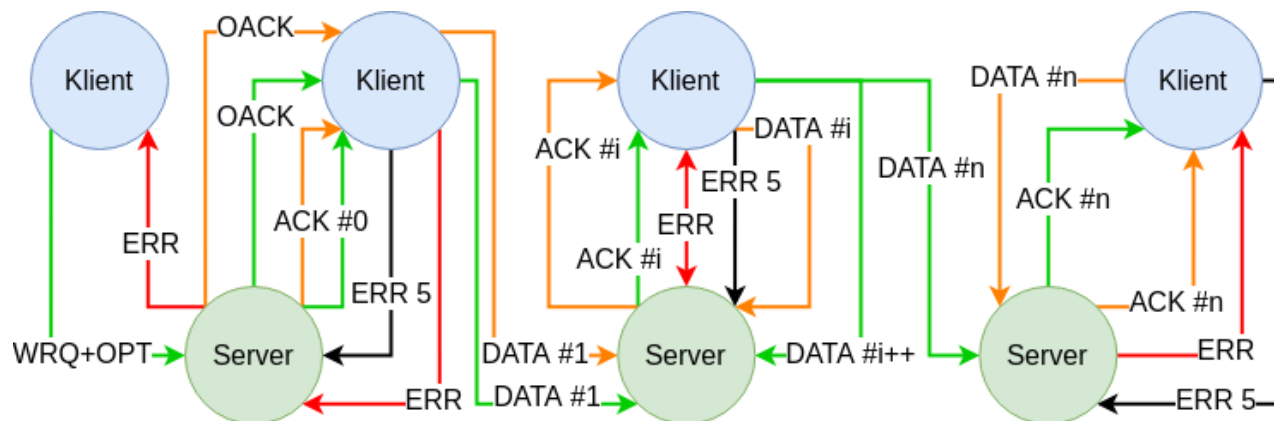
1.2 Diagram zápisu souboru



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.

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.

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 obdržení nesprávného TID

3.2.1 Výpis programu

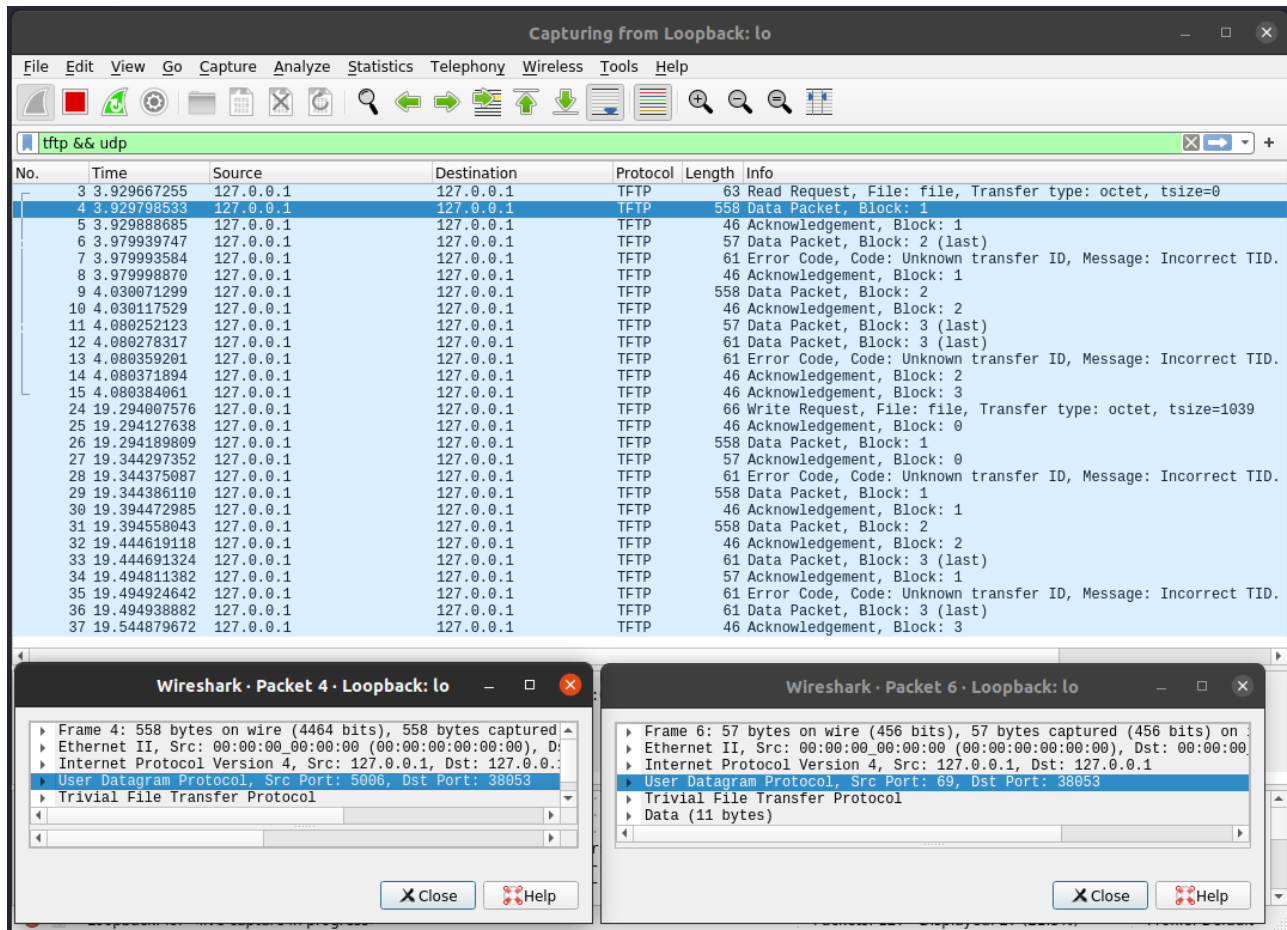
```
> -R -d file
Reading file 'file' ...
Warning: Server does not support Option Negotiation.
Warning: Recieved TID does not match the estabhlshed one,
        server informed and transfer continues.
Warning: Recieved TID does not match the estabhlshed one,
        server informed and transfer continues.

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,
    additional 2 datagrams carrying data may have been lost,
    which compounds up to 1024 B.

> -W -d file
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,
  - 1566 B of data were sent in 5 datagrams of a maximum data size 512 B,
    of which 2 datagrams carrying data may have been lost of overall data
    size 527 B.
```


3.2.2 Analýza síťového provozu



3.3 Test sjednání časového limitu pro znovu odeslání datagramu

3.3.1 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.

3.3.2 Analýza síťového provozu

No.	Time	Source	Destination	Protocol	Length	Info
21	21.819842869	127.0.0.1	127.0.0.1	TFTP	73	Read Request, File: file, Transfer type: octet, tsize=0, time
22	21.819861692	127.0.0.1	127.0.0.1	TFTP	54	Option Acknowledgement, timeOUT=2
23	21.820004063	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 0
24	23.321547851	127.0.0.1	127.0.0.1	TFTP	558	Data Packet, Block: 1
25	23.321610179	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 1
26	24.822085554	127.0.0.1	127.0.0.1	TFTP	558	Data Packet, Block: 2
27	24.822136321	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 2
32	26.322909317	127.0.0.1	127.0.0.1	TFTP	61	Data Packet, Block: 3 (last)
33	26.322953245	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 3
38	53.731921716	127.0.0.1	127.0.0.1	TFTP	76	Write Request, File: file, Transfer type: octet, tsize=1039,
39	53.732005203	127.0.0.1	127.0.0.1	TFTP	54	Option Acknowledgement, timeOUT=2
40	53.732049967	127.0.0.1	127.0.0.1	TFTP	558	Data Packet, Block: 1
41	55.233585475	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 1
42	55.233634846	127.0.0.1	127.0.0.1	TFTP	558	Data Packet, Block: 2
43	56.334060687	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 2
44	56.334106679	127.0.0.1	127.0.0.1	TFTP	61	Data Packet, Block: 3 (last)
45	57.435228403	127.0.0.1	127.0.0.1	TFTP	46	Acknowledgement, Block: 3

Trivial File Transfer Protocol	
Opcode:	Option Acknowledgement (6)
[Source File:	file]
Option:	timeOUT = 2
Option name:	timeOUT
Option value:	2

Offset	Hex	ASCII
0000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00E..
0010	00 28 5e e0 40 00 40 11 dd e2 7f 00 00 01 7f 00	..(^@.
0020	00 00 13 8e a2 f5 00 14 fe 27 00 06 74 69 6d 65time
0030	4f 55 54 00 32 00	OUT-2..

3.4 Test menší velikosti přenášeného bloku dat

3.4.1 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:

- 150 ms elapsed during the transfer,
- 519 B of data were recieved in 3 datagrams of a maximum data size 252 B.

```
> -W -d file -s 800
```

Writing file 'file' ...

Warning: Server did not recognize transfer size option. Transfer continues.

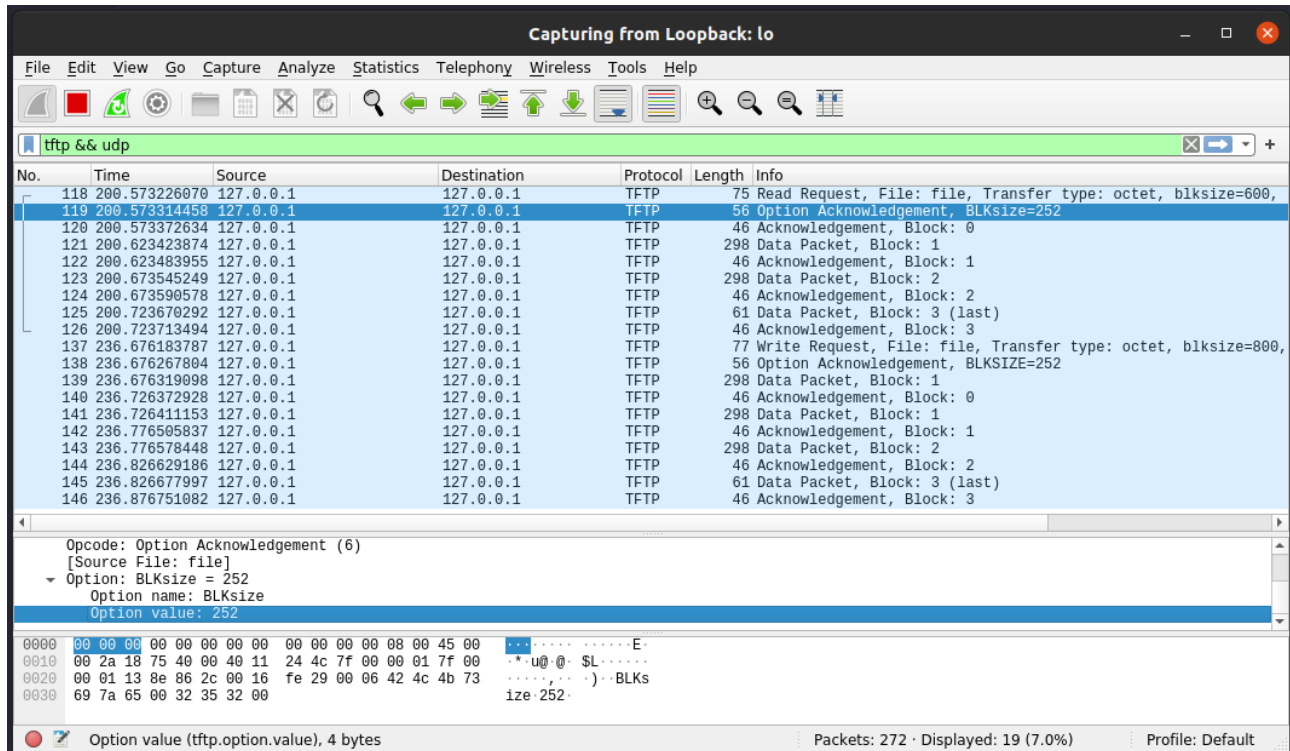
Writing file 'file' succeeded.

Transfer summary:

- 200 ms elapsed during the transfer,
- 771 B of data were sent in 4 datagrams of a maximum data size 252 B, of which 1 datagram carrying data may have been lost of overall data

size 252 B.

3.4.2 Analýza síťového provozu



3.5 Test kódování do netascii i na konci bloku dat

3.5.1 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.

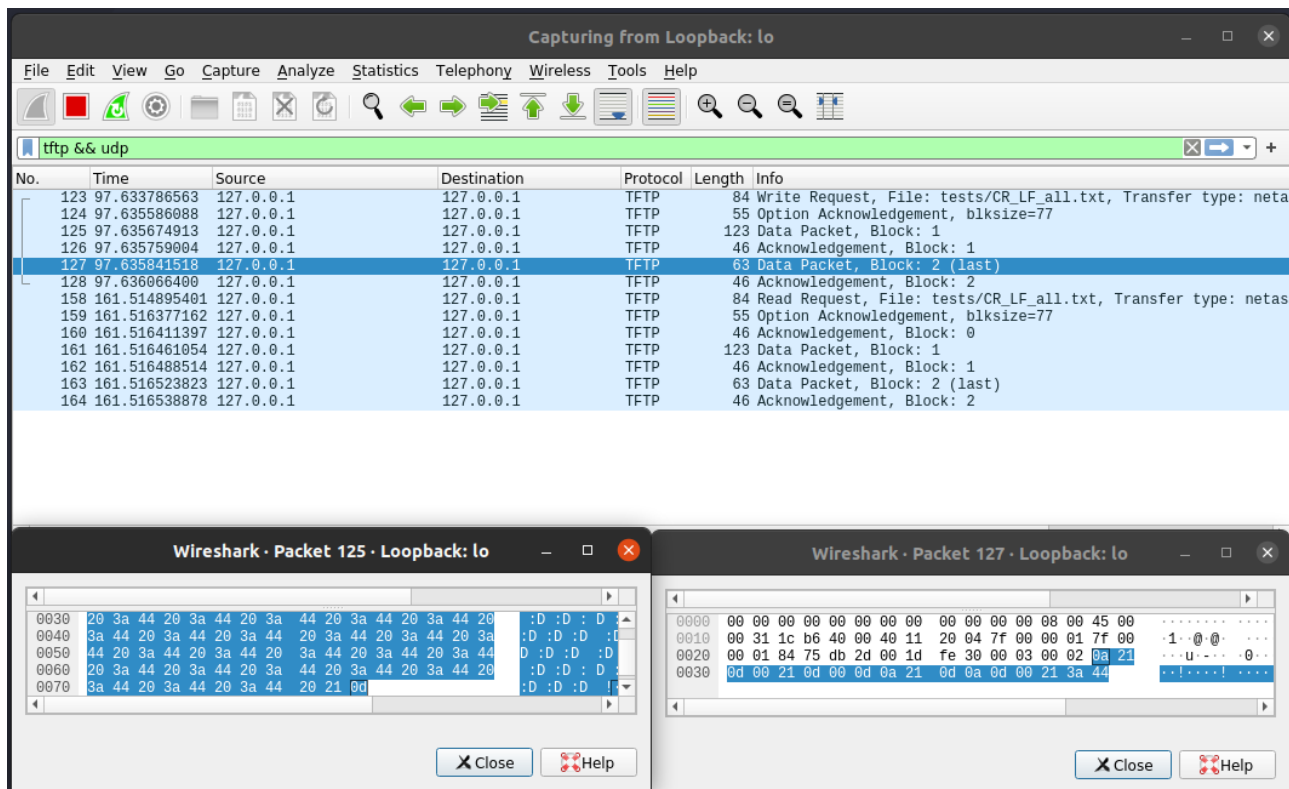
Transfer summary:

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

3.5.2 Kontrola obsahu souboru programem diff

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

3.5.3 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>