

# TFTP

## Trivial File Transfer Protocol

Jan Krňoul

KIV / PSI

# TFTP

- Jednoduchý protokol pro přenos souborů
- 1980 – IEN 133
- 1981 – RFC 783
- 1992 – RFC 1350
- 1998 – RFC 1785, 2090, 2347, 2348, 2349
- Noel Chiappa, Bob Baldwin, Dave Clark

# Přehled

- Přenáší data pomocí UDP
- Každý paket potvrzován samostatně
- Potvrzovací a přenosové schema inspirováno TCP
- Chybové zprávy vychází z PARC's EFTP
- Navržen pro jednoduchou implementaci

# Funkce protokolu

- Podporuje
  - Zápis souboru
  - Čtení souboru
- Nepodporuje
  - Identifikace/autentizace uživatele
  - Mazání souborů
  - Výpisy adresářů, procházení adr. struktury
  - Další pokročilé funkce známé z FTP (např. navázání přerušného přenosu)

# Přenos souboru

- 3 mody
  - Netascii
    - Vychází z modifikovaného ascii pro telnet protokol
  - Octet
    - Binární data, 8bit
  - Mail
    - Formát stejný jako netascii, neměl by být používán
  - Další mody na základě dohody
- 512 byte bloky
- Menší než 512 = ukončení přenosu
- Pořadí hlaviček
  - Lokální médium | Internet | Datagram | TFTP

# TFTP pakety

- 5 typů paketů, každý svůj kód
  1. Read request (RRQ) - požadavek čtení
  2. Write request (WRQ) - požadavek zápisu
  3. Data (DATA) - data
  4. Acknowledgment (ACK) - potvrzení
  5. Error (ERROR) - chyba

# Přenos souboru

- Požadavek na čtení/zápis (WRQ, RRQ)
  - Jako odpověď se očekává potvrzovací paket (při zápisu) nebo první blok dat
  - Při vytváření spojení klient volí zdrojový identifikátor přenosu (transmit identifikator – TID)- náhodně
  - Každý paket má 2 TID – zdrojový a cílový
  - TID odpovídá použitým portům (impl. 69)
  - Server opovídá z jiného portu

- RRQ/WRQ paket

- 2 bytes    string    1 byte    string    1 byte

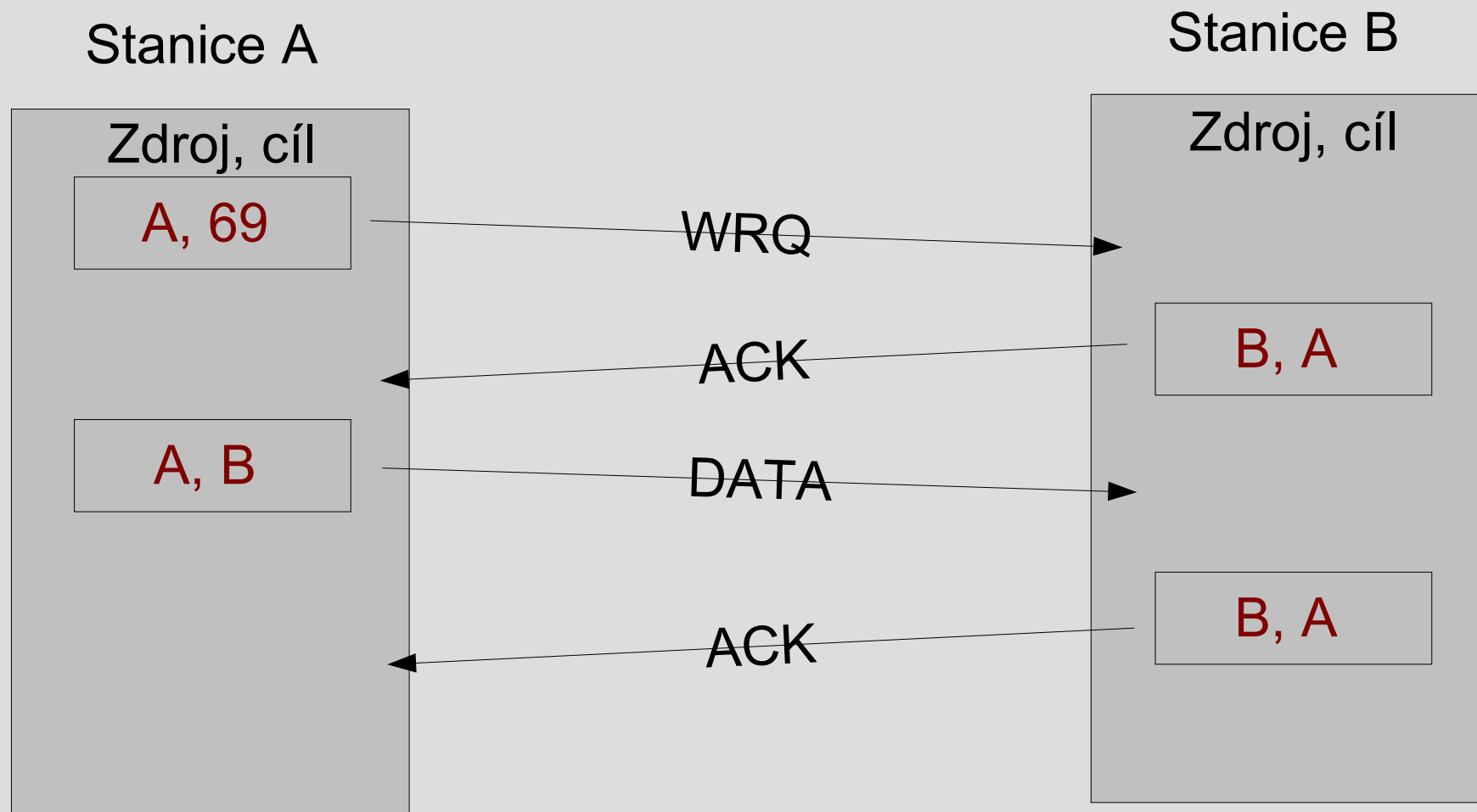
-----  
| Opcode | Filename | 0 | Mode | 0 |  
-----

# RRQ,WRQ paket

- Operační kód- 1 nebo 2
- Filename- netascii sekvence bytů ukončená znakem \0
- Mode- řetězec “netascii”, “octet”, “mail”
  - Opět zakončený \0
  - Case insensitiv
- Stanice která obdrží data v netascii modu musí data překládat do svého nativního formátu
  - Je možné aby jedna stanice vysílala data v jiném formátu než je druhá stanice přijímá



# Vytvoření spojení pro zápis



# Datový paket

- Formát datového paketu

- 2 bytes    2 bytes    n bytes

-----  
| Opcode | Block # | Data |  
-----

- Opcode == 3 pro přenos dat
- Číslo bloku začíná 1
- Datové pole má délku 0 - 512 bytů
  - Pokud == 512 není to poslední datový paket,  
pokud  $0 < \text{délka dat} < 512$  tak se jedná o konec přenosu

# Potvrzování

- Všechny pakety kromě duplikátních ACK a ukončovacích paketů jsou potvrzovány
  - WRQ a DATA pakety jsou potvrzovány ACK nebo ERROR pakety
  - RRQ a ACK pakety jsou potvrzovány DATA nebo ERROR pakety

=> pakety se “potvrzují navzájem”

- Formát ACK paketu

- 2 bytes    2 bytes

-----  
| Opcode | Block # |  
-----

# Chybový paket

- Formát chybového paketu

- 2 bytes    2 bytes    string    1 byte

-----  
| Opcode | ErrorCode | ErrMsg | 0 |  
-----

- Opcode == 5

- Význam kódu chyby

0. Nedefinováno, popis v ErrMsg

1. Soubor nenalezen

2. Narušení přístupu

3. Disk plný

4. Neplatná operace

5. Neznámé TID

6. Soubor již existuje

7. Neexistující uživatel

# Signalizace chyb

- Chyby jsou způsobovány třemi typy událostí
  - nemůže být vyhověno požadavku (soubor nenalezen...)
  - špatně přijatý paket (zpoždění, duplicita...)
  - ztracení přístupu k požadovanému zdroji (plný disk...)
- Jediná chyba při které není přenos ukončen: zdrojový port přijatého paketu je chybný.

# Ukončení přenosu

- Normální ukončení
  - Konec přenosu je singalizován vysláním daového paketu který obsahuje data s délkou 0 – 511 bytů (délka datagramu < 516 B)
- Předčasné ukončení
  - Pokud nemůže být umožněn přístup nebo nastane chyba v průběhu přenosu je vyslán ERROR paket
  - Pro detekci chyb je možné použít vypršení timeout-u

# Multicast

- Rozšíření RFC 2090 z roku 1997
- Jedná se o experimentální rozšíření protokolu TFTP
- Umožňuje více klientům souběžné stahování dat prostřednictvím multicast paketů
- Obvykle pro boot více stanic
- Vyžaduje Options extension (rozšíření protokolu)
  - Ke některým paketům přidává možnost rozšíření a jeho parametru
  - Potvrzuje se paketem OACK (Option ACK)

# Rozšíření read request

- Modifikovaný RRQ:

-----  
| opcode | filename | 0 | mode | 0 | multicast | 0 | 0 |  
-----

- OpCode == 1 (Read request) – beze změny
- Filename – beze změny
- Mode – beze změny
- Multicat – obsahuje string 'multicast' zakončený '\0'
- Poslední nula je hodnota parametru multicast (prázdný řetězec zakončený nulou)



# Odpověď - OACK

- Server odpovídá OACK

-----  
| opcode | multicast | 0 | addr, port, mc | 0 |  
-----

- Opcode == 6
- Multicast je potvrzení multicast vysílání
- Addr je multicast IP adresa ukončená ','
- Port je doporučen 1758
- Mc značí 'master klienta' tedy klienta zodpovědného za potvrzování
- Master client potvrzuje datové pakety (ACK)

# Další klienti

- Server udržuje seznam klientů
- Nejstarší klient = master klient
- Po ukončení přenosu server pošle OACK paket druhému nejstaršímu klientu a ten si vyžádá chybějící (již odvysílaná) data
- Ta jsou vysílána opět jako multicast
- Server má možnost tato data odeslat jako unicast pro omezení zátěže na ostatních klientech

# Zdroje

- RFC 1350
- RFC 2090
- IEN 133
- <http://www.networksorcery.com/enp/protocol/tftp.htm>
- <http://www.earchiv.cz> - e-archiv J. Peterky