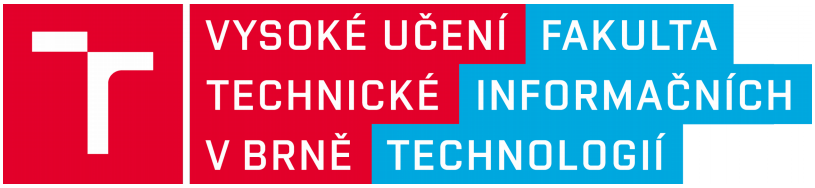

IPK – počítačové komunikace a sítě
2017/2018

Projekt 1: Klient-server pro jednoduchý přenos souborů

Tomáš Pazdiora
login: xpazdi02

12. března 2018

Obsah

1 Zadání

Navrhněte vlastní aplikační protokol, kterými poté spolu budou komunikovat klient a server.

Umístění souborů na straně serveru je v aktuálním adresáři, ve kterém se nachází server. Spuštění klienta předpokládá provedení pouze jedné operace a to uložení nebo přečtení souboru se zadaným jménem. Komunikace mezi serverem a klientem bude schopna se vypořádat se ztrátami paketů (a to buď na aplikační, nebo transportní vrstvě).

1.1 Konvence jména klientské aplikace a jejích povinných vstupních parametrů

./ipk-client -h host -p port [-r|-w] file

- host (IP adresa nebo fully-qualified DNS name) identifikace serveru jakožto koncového bodu komunikace klienta
- port (číslo) cílové číslo portu
- -r značí, že klient bude ze serveru soubor číst
- -w značí, že klient bude na server soubor zapisovat
- file (cesta) určuje cestu k souboru, se kterým se bude manipulovat (buď se soubor odkazovaný cestou uploadne do pracovního adresáře serveru, nebo se obsah souboru stejného jména v pracovním adresáři serveru downloadne na místo specifikované cestou)

např.: *./ipk-client -h eva.fit.vutbr.cz -p 55555 -r myfile.xml*

1.2 Konvence jména serverové aplikace a jejích povinných vstupních parametrů

./ipk-server -p port

- port (číslo) číslo portu, na kterém server naslouchá na připojení od klientů.

např.: *./ipk-server -p 55555*

2 Protokol IPKFTP

IPKFTP je protokol pro přenos souborů mezi počítači s využitím protokolu TCP. Veškerá komunikace probíhá prostřednictvím packetů formátu IPKPacket (jenž je popsán v následující kapitole [3]). Podrobný popis protokolu pomocí stavových automatů je k nalezení v kapitole [4].

Komunikaci zahajuje klient připojením k serveru pomocí TCP. Pro potvrzení připojení, a o to, zda se jedná o kompatibilní server, posílá klient na server IPKPacket se statusem *PING* (dále jen názvy statusů, viz Tabulka 2). Pokud je server kompatibilní, dorazí odpověď *OK*. Pokud se vyskytne například problém s verzí protokolu, server odpoví *ERROR*. V případě připojení na zcela nekompatibilní aplikaci nepřijde odpověď žádná a klient se ukončí s chybou kvůli vypršení timeoutu.

Po úspěšném připojení má klient možnost stahovat soubory ze serveru, nebo je na server nahrávat.

V případě nahrávání souboru zašle klient *OFFER* s názvem a daty souboru. Po úspěšném nahrání server odpoví *OK*. V případě chyby server odpoví *ERROR* a klient se může pokusit *OFFER* zopakovat. V případě nemožnosti zápisu souboru server odpoví *INACCESSIBLE*.

V případě stahování souboru zašle klient *REQUEST* s názvem souboru. Server zašle odpověď *OFFER* s názvem a daty souboru, *ERROR* v případě chyby a *INACCESSIBLE* v případě nemožnosti čtení souboru. V případě nesprávně přijatého souboru (například nevalidní CRC) či chyby se může klient pokusit *REQUEST* zopakovat.

Po dokončení přenosů klient uzavře TCP komunikaci. Protokol IPKFTP uzavření spojení jinak neřeší.

3 Formát IPKPacket

IPKPacket je binární formát pro komunikaci pomocí protokolu IPKFTP. Tento formát není zabezpečený/šifrovaný, ale obsahuje cyklický redundantní součet pro kontrolu chyb dat. Tento formát nemá fixní velikost, ani maximální velikost. Proto je vhodné využít nižší protokol, který je schopný rozdělit tento formát na menší části. V případě protokolu IPKFTP je použit protokol TCP.

offset	velikost	jméno
0h	6 bytů	signatura „IPKFTP“
6h	1 byte	verze protokolu
7h	1 byte	**status ***
8h	8 bytů	celková velikost zprávy
10h	volitelné	jméno souboru (null terminated)
…	volitelné	data souboru
konec - 4h	4 byty	CRC32 zprávy (polynom 0x04C11DB7)

Tabulka 1 : formát IPKPacket

* **status**:

hodnota	název	význam
0	REQUEST	žádost o soubor (vyžaduje jméno souboru)
1	OFFER	nabídka souboru (vyžaduje jméno a data souboru)
2	PING	ping
3	OK	ok
4	ERROR	chyba
5	INACCESSIBLE	nedostupný soubor

Tabulka 2 : položka „status“

4 Serverový a klientský stavový automat

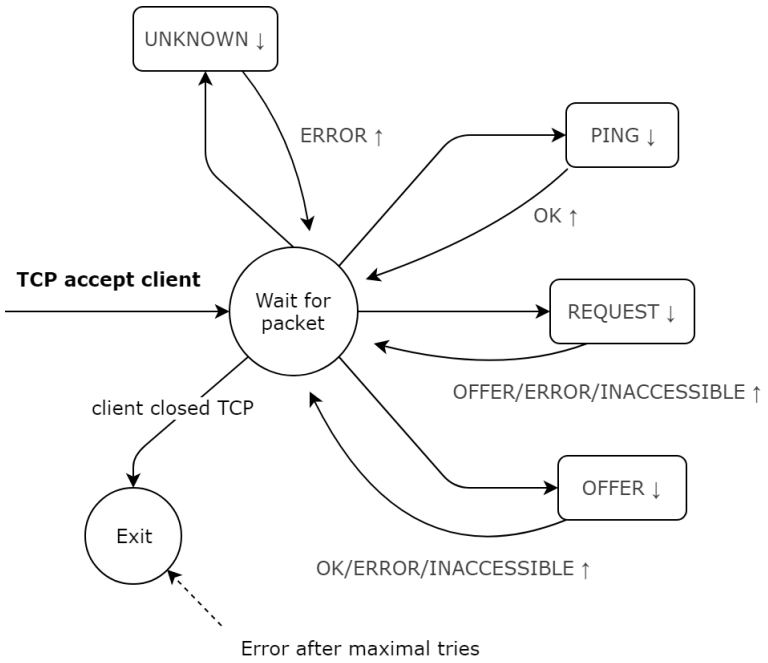

Diagram 1 : stavový automat serveru

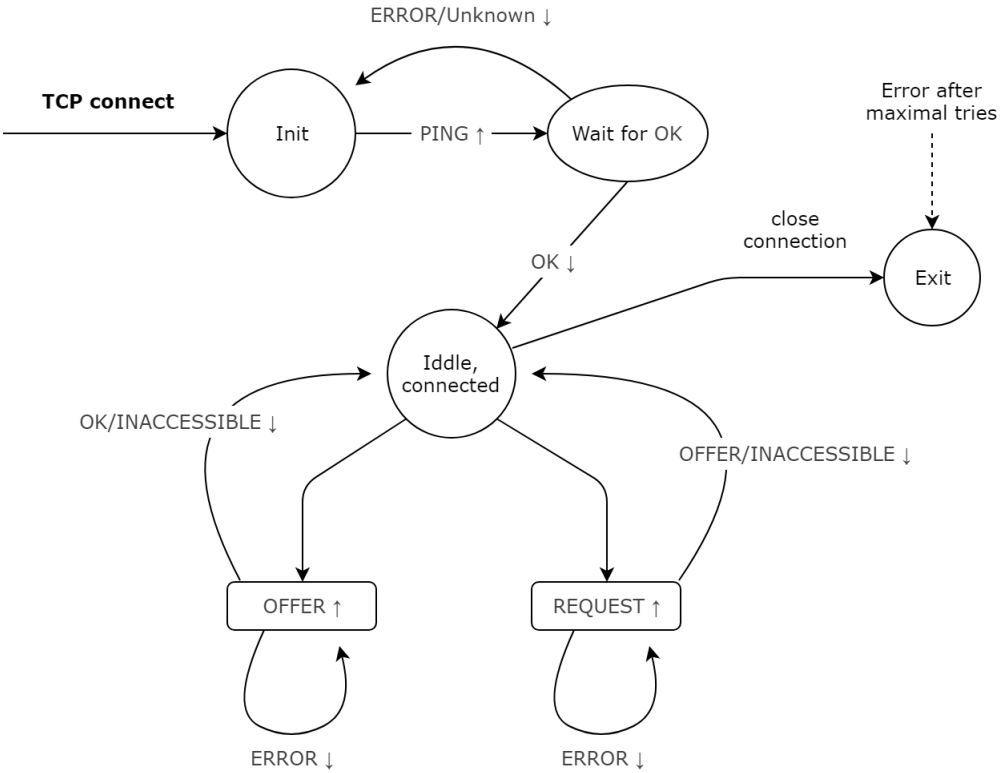

Diagram 2 : stavový automat klienta

5 Struktura programu

Program se skládá z modulů:

client.cpp – modul programu klienta

server.cpp – modul serverového programu

a následujících sdílených modulů:

CRC32.cpp – zajišťující funkci pro aplikaci CRC32 na data

IPKFTP.cpp – realizující protokol IPKFTP

IPKPacket.cpp – realizující binární formát IPKPacket

TCP.cpp – zajišťující multiplatformní TCP komunikaci

6 Reference

[1] Stephan Brumme, *Fast CRC32*, 4. únor 2015, http://create.stephan-brumme.com/crc32/#sarwate

[2] Ed Rosten, *Compile time lookup tables*, 12. květen 2016, https://deathandthepenguinblog.wordpress.com/2016/05/12/compile-time-lookup-tables/

[3] Evan Klitzke, *Blocking I/O, Nonblocking I/O, And Epoll*, 10. leden 2017, https://eklitzke.org/blocking-io-nonblocking-io-and-epoll