

Úloha č.2 – UDP Klient

Situace

Na oběžné dráze planety Mars létá sonda, která sbírá informace a fotografie od robotů pracujících na povrchu planety (viz úloha č.1). Získané informace se poté přenášejí ze sondy do řídicího centra na Zemi.

Úkoly

Vaším úkolem je:

1. načíst ze sondy poslední známou fotografii okolí, aby mohli vědci zjistit, v jakém terénu se roboti nachází
2. provést upload nového firmwaru do sondy

Obecné schema komunikace

Sonda komunikuje pomocí protokolu UDP a přijímá data na portu 4000. Proces běžící v sondě budeme nazývat serverem a proces, který se sondou komunikuje, klientem.

Formát paketu

identifikátor 'spojení'	sekvenční číslo	číslo potvrzení	příznak	data
4B	2B	2B	1B	0-255B

- identifikátor 'spojení' – vygenerovaný serverem (pro umožnění transportu dat více souborů najednou),
- sekvenční číslo – číslo prvního byte v posílaných datech,
- číslo potvrzení – číslo očekávaného byte v přijímaných datech (číslo potvrzení + délka dat),
- příznak – uveden dále,
- data – přenášená data

Identifikátor spojení a sekvenční čísla se přenášejí v reprezentaci network byte order (big endian).
Příklad:

dekadicky	hexadecimálně	pořadí bytů
1234	04D2h	04h D2h
34566	8706h	87h 06h

Identifikátor spojení

Identifikátor spojení je nenulové číslo. Při navazování spojení posílá klient identifikátor spojení nastavený na nulu. Při další komunikaci použije klient identifikátor spojení, který mu vrátí server v prvním paketu.

Příznaky

číslo bitu	7	6	5	4	3	2	1	0
příznak	-	-	-	-	-	SYN	FIN	RST

příznak	význam
SYN	Otevření nového spojení. Posílá klient i server (pouze) na začátku v prvním paketu. V datové části musí být právě 1 byte s kódem příkazu.
FIN	Ukončení spojení. Posílá klient i server, pokud již nemají žádná další data k odeslání. Paket s nastaveným příznakem FIN již nemůže obsahovat žádná data. Ukončení spojení nelze odvolat. Oba směry spojení se uzavírají zvlášť. Sekvenční číslo se po odeslání FIN již nesmí zvětšit.
RST	Zrušení spojení kvůli chybě. Posílá klient i server v případě detekování logické chyby v hodnotách v hlavičce. Např. přijatý paket neobsahuje příznak SYN a ID spojení není evidováno. Nebo je hodnota potvrzovacího čísla menší, než byla v posledním přijatém paketu (klesá). Pozor na přetečení sekvenčních a potvrzovacích čísel. Žádná z komunikujících stran po odeslání paketu s příznakem RST již dále neukončuje spojení standardním způsobem – spojení je přenosem paketu s příznakem RST definitivně ukončeno.

Jednotlivé příznaky (SYN, FIN, RST) nelze spolu kombinovat.

Sekvenční číslo a číslo potvrzení

Sekvenční číslo je pořadové číslo prvního bytu v proudu posílaných dat. Na začátku je toto číslo nastaveno na nulu. Nastavení příznaku SYN nebo FIN toto číslo nemění.

Číslo potvrzení sděluje protistraně pořadové číslo očekávaného bytu v proudu přijímaných dat. Potvrzuje zároveň všechny byty s nižším pořadovým číslem.

Tato čísla nemají znaménko a mohou přetéct. Přetečení nemá na komunikaci vliv.

Data

Délka dat je určena velikostí paketu minus velikost hlavičky. Poslat lze najednou maximálně 255 bytů dat, takže nejmenší velikost datagramu je 9 B (pouze hlavička) a největší $9 + 255 = 264$ B. Data lze poslat pouze v paketu bez nastaveného příznaku FIN a RST. Pokud má paket nastaven příznak SYN, musí být v datové část 1 byte s kódem příkazu.

Data poslaná klientem i serverem se v číslování pomocí sekvenčního čísla, avšak pouze tehdy, pokud není nastaven žádný příznak (zejména SYN).

Popis protokolu

Navázání spojení

Iniciátorem spojení je vždy klient.

Klient pošle první datagram s příznakem SYN a s identifikátorem spojení, sekvenčním číslem a číslem potvrzení nastavenými na nulu. Datová část musí obsahovat právě 1 byte s kódem příkazu:

příkaz	význam
01h	Download fotografie okolí
02h	Upload nového firmwaru

Server odpoví datagramem s nastaveným příznakem SYN, nenulovým identifikátorem spojení a se sekvenčním číslem a číslem potvrzení nastavenými na nulu. Datová část obsahuje 1 byte s kódem příkazu, který bude proveden.

Po přijetí tohoto datagramu klientem je navázáno spojení. V dalším paketu začne klient nebo server odesílat data (podle poslaného příkazu).

Pokud se ztratí paket s příznakem SYN odeslaný klientem, musí klient po 100 ms odeslat nový.

Pokud se ztratí paket s příznakem SYN odeslaný serverem, musí klient po 100 ms odeslat nový SYN paket. V případě downloadu začne zřejmě server odesílat data, klient však nezná číslo spojení a tak tato data nemůže přijmout. Toto serverem polootevřené spojení se po 20. opakování stejného paketu uzavře (bude odeslán paket s příznakem RST a nastaveným identifikátorem spojení). V případě uploadu se toto spojení uzavře po 20. opakování SYN paketu.

Poznámka: Data odeslaná serverem budou zcela určitě doručena klientovi, který o spojení žádal. Zdálo by se tedy, že v případě downloadu nemusí server odesílat paket s příznakem SYN (mohl by odeslat rovnou část dat). Protože však klient může podle této specifikace navázat více spojení najednou, tedy odeslat více inicializačních paketů najednou (třeba s různými příkazy), musí klient počkat na SYN paket s číslem spojení. Potom si také podle kódu příkazu (v datové části v odpovědi od serveru) může správně přiřadit čísla spojení.

Přenos dat

Data lze posílat až po navázání spojení. Posílání dat je neslučitelné s nastavováním libovolných příznaků (SYN, RST, FIN) (v této terminologii nechápeme příkaz v prvním paketu poslaném klientem jako data). V dalším textu nazveme vysílačem tu stranu, která data odesílá a přijímačem protistranu.

Pokud posílá některá ze stran soubor dat delší než 255 bajtů (*což je v této uloze vždy*), používá se okénkové potvrzovací schéma s fixní velikostí okénka $W = 2040$ bajtů a timeoutem $T_{out} = 100$ ms.

Vysílač se snaží mít v komunikačním kanále právě tolik nepotvrzených bajtů odesílaného proudu dat, jako je velikost okénka. Při zahájení komunikace zapíše do kanálu W bajtů a čeká na potvrzení. Přijímač reaguje na všechny příchozí pakety odesláním potvrzovacího paketu (paketu s nastaveným potvrzovacím číslem). Potvrzovací číslo popisuje pořadové číslo očekávaného bajtu v proudu přijímaných dat, takže každé potvrzovací číslo potvrzuje příjem všech dat až do tohoto pořadového čísla (bez něj). Pokud přijme přijímač data mimo pořadí, zapamatuje si je a odešle potvrzovací číslo nastavené za poslední přijatý byte v proudu dat bez mezer. Jakmile jsou mezery zaplněny nově přijatými daty, posune přijímač potvrzovací číslo tak, aby opět ukazovalo za poslední přijatý byte v celistvém proudu přijímaných dat. Data mimo pořadí jsou tedy nakonec využita, přijímač je však zprvu nepotvrzuje. Potvrzuje vždy pouze přijatý datový proud bez mezer.

Jakmile vysílací strana přijme paket s takovým potvrzením, které snižuje počet nepotvrzených dat v odesílaném proudu, odešle další data, tak, aby bylo v kanále opět W nepotvrzených bajtů (říkáme, že posune okénko).

Vysílač si pamatuje čas odeslání posledního paketu (označme T). Pokud server nepřijme do času $T + T_{out}$ žádné nové potvrzení, které snižuje počet nepotvrzených dat, odešle vysílač W bajtů od nevyššího přijatého potvrzovacího čísla (říkáme, že odešle celé okénko). Nastaví také T na novou hodnotu.

Vysílač posílá vždy maximální možné množství dat v jednom paketu (255 bajtů). Vyjímkou je poslední datový paket, který může obsahovat méně dat.

Pokud vysílač přijme 3x po sobě stejné potvrzovací číslo, odešle ihned do kanálu 1 paket s maximálním možným množstvím dat od pořadového čísla shodného s přijatým potvrzovacím číslem a nastaví T na novou hodnotu.

Ukončení spojení

Pokud vysílač odešle celý soubor a má všechna odeslaná data potvrzená, uzavře spojení nastavením příznaku FIN. Tento příznak nelze kombinovat s odesláním posledních dat. Přijímač vzápětí odešle také paket s příznakem FIN.

Spojení je oboustranně ukončeno v momentě, kdy obě strany, které již odeslaly paket s nastaveným

příznakem FIN, obdrží paket s nastaveným příznakem FIN a potvrzovacím číslem se stejnou hodnotou, jako je sekvenční číslo.

Pokud dojde 20x po sobě k opakovanému odvyšování paketu se stejným sekvenčním číslem, je spojení přerušeno, klient musí vypsát chybu při přenosu. To platí i při uzavírání spojení, kdy je odesílán příznak FIN.

Příkazy

příkaz 01h – fotografie okolí

Jakmile robot odešle první paket s nastaveným příznakem SYN, začne ihned odesílat fotografii okolí. Pokud byl první paket (SYN) ztracen, spojení po marném odeslání 20 stejných paketů expiruje (klient žádá o nové spojení posláním nového SYN paketu).

Poslaná fotografie je ve formátu PNG a posílá se pouze její obsah, nikoliv název souboru nebo nějaká další informace. Každá odeslaná fotografie je jedinečná (fotografie získané v rámci různých spojení se liší).

příkaz 02h – upload nového firmwaru

Jakmile klient přijme od serveru první paket s nastaveným příznakem SYN, začne odesílat obsah souboru s firmwarem. Neposílá se žádná jiná doplňující informace, jako např. jméno souboru nebo jeho délka.

Přijetí chybného paketu

Po přijetí chybného paketu odešle příjemce (klient i server) paket s příznakem RST. Chybný paket je paket, který:

- neobsahuje platné ID spojení a nemá nastaven příznak SYN
- nemá potvrzovací číslo v intervalu $\langle seq - velikost okénka, seq \rangle$ kde seq je sekvenční číslo příjemce
- má nastaven více příznaků najednou
- má nastaven příznak SYN a datová část neobsahuje platný příkaz (délka dat není 01h a/nebo příkaz není 01h nebo 02h)
- má nastaven příznak FIN a zároveň obsahuje data

Pozor, sekvenční a potvrzovací čísla mohou přetéct, což nesmí mít na komunikaci vliv.

Příklady komunikace

Příklady komunikace jsou na zvláštní stránce.

Chybovost sítě

Možné chyby na síti:

- ztráta libovolného paketu,
- duplikace libovolného paketu,
- prohození libovolných paketů,

- proměnné zpoždění sítě.

Server tyto chyby emuluje. Může se stát, že server 20x způsobí ztrátu stejného paketu a dojde k odeslání paketu s příznakem RST a rozpadu spojení. Tato chyba není na závadu při odevzdávání úlohy (přenos se při odevzdávání zopakuje).

Požadavky

- program musí 100% fungovat s protistranou na serveru baryk.fit.cvut.cz (viz níže),
- program musí být stabilní a odolný proti neočekávaným vstupům,
- přenesený soubor (download i upload) nesmí být poškozen,
- program se musí umět vyrovnat s faktem, že UDP pakety se občas ztratí, zduplikují či prohodí,
- program by měl vypisovat alespoň základní informace o svém stavu (např. posílané a přijímané příkazy či odpovědi),
- IP adresa nebo DNS jméno serveru se zadává jako parametr v příkazové řádce při spuštění programu (tzn. není zadržovaná ve zdrojovém kódu), syntaxe viz níže,
- zdrojový text musí být komentovaný, v hlavičce všech zdrojových textů musí být uveden autor,
- povolené programovací jazyky jsou následující: Java, C, C++.
- program pište **do jednoho zdrojového souboru**. Tento soubor můžete pojmenovat jakkoliv, avšak s příponou .c, .cpp nebo .java. Tyto soubory **musí být přeložitelné řádkovým překladačem** (gcc/g++/javac). Odevzdává se pouze tento jeden zdrojový kód, neimplementujte tedy nic do dalších souborů mimo něj.
- pokud použijete jazyk Java, napište na začátek zdrojového textu **package robot** a třídu s metodou *main* pojmenujte **Robot**. Můžete použít připravenou šablonu, která je k dispozici ke stažení mezi příklady programů.
- program musí přijímat parametry z příkazové řádky s následující syntaxí:

Download fotografie (příkaz 01h): `./robot <server>`

Výsledná fotografie bude uložena v souboru *foto.png*.

Upload firmwaru (příkaz 02h): `./robot <server> <firmware.bin>`

<server> je jméno nebo IP adresa serveru a *<firmware>* je soubor s novým firmwarem pro robota.

Příklad volání: `java robot.Robot 81.25.17.115 /data/firmware.bin`

- každý student má **povinnost průběžně uploadovat aktuální stav svého zdrojového kódu** přes WWW rozhraní serveru Baryk [<http://baryk.fit.cvut.cz/cgi-bin/robotudp>]. Předpokládá se, že vždy po ukončení práce na programu nahrajete svůj kód na server Baryk [<http://baryk.fit.cvut.cz/cgi-bin/robotudp>]. Také se předpokládá, že pokud k danému dni není záznam o uploadu Vašeho programu, tak jste na programu tento den nepracoval(a). Jinak řečeno, uploadováním svého programu vykazujete činnost (stejně jako v komerčním prostředí) a zároveň se chráníte před obviněním z nepoctivého jednání (přečtěte si obecné požadavky na úlohy na hlavní stránce předmětu).

Testování

Testovací server běží na baryk.fit.cvut.cz na portu 4000. Proti tomuto serveru můžete zkusit funkčnost Vašeho řešení klienta. Výsledek testování bude zobrazen na webu serveru Baryk [<http://baryk.fit.cvut.cz/cgi-bin/robotudp>].

Pro testování uploadu firmwaru do robota využijte poslední verzi firmwaru, která je k dispozici o pár řádek níže.

K dispozici je i staticky přeložený UDP server, který běží na serveru Baryk (pouze pro Linux).
Poznámky k jeho použití:

- Ověřte si občas, není-li k dispozici novější verze.
- Server spusťte např. takto: `./kareludp-server 1000 - foto/AA2E0000.png firmware/firmware-karel-1.6.0.bin`

kde 1000 je UID uživatele, pod kterým server poběží, foto/AA2E0000.png je cesta k fotografii a firmware/firmware-karel-1.6.0.bin je akceptovaný firmware.

Ke stažení

soubor	odkaz	sha1
firmware v1.6.0	firmware-karel-1.6.0.bin [http://baryk.fit.cvut.cz/firmware-karel-1.6.0.bin]	ea0fc5e48008b827f845b427f3ce8ff574db2da8
kareludp v1.7.0 - server a klient	kareludp-1.7.0.tar.bz2 [http://baryk.fit.cvut.cz/kareludp-1.7.0.tar.bz2]	f3fc2e4e115e976f04005c9a9dfe576aeb15d641

Termín a způsob odevzdání

Termín odevzdání určí Váš cvičící dle možností daných harmonogramem akademického roku.

Za každý týden prodlení je penalizace -0,5 bodu (jako přelom týdne se počítá konec cvičení).

Odevzdat lze pouze takový program, který:

- je přeložitelný na serveru baryk.fit.cvut.cz
- dodržuje syntaxi příkazové řádky (je tedy možno jej otestovat na serveru baryk.fit.cvut.cz)
- komunikuje se serverem baryk.fit.cvut.cz a je schopný úspěšně stáhnout fotografii (pro odevzdání stačí předvést úspěšné stažení souboru s fotografií bez ohledu na následky u serveru či klienta)

Postup odevzdání:

1. otestujte svoje řešení na serveru Baryk [http://baryk.fit.cvut.cz/cgi-bin/robotudp]
2. připravte si zdrojové texty k nahlédnutí
3. připravte si internetový prohlížeč na server Baryk [http://baryk.fit.cvut.cz/cgi-bin/robotudp] a připravte se na překlad zdrojového kódu
4. spusťte terminál, nastavte adresář s Vaší úlohou jako aktuální a zadejte příkaz `ls -l`
5. zavolejte cvičícího

Bodování

Za úlohu můžete dostat max. 8 bodů. Penalizace:

- program neumí stáhnout fotografii: *nelze odevzdat !*
- program neumí provést upload firmwaru: -4
- nebylo korektně uzavřeno spojení: -1
- chybně implementovaný okénkový protokol: dle povahy -1 až -4

- nestabilita: -3
- malá odolnost vůči nestandardním vstupům: -3
- chyby v programu, které se neprojeví přímo: dle povahy -1 až -5
- nečistý zdrojový kód: -1
- zdrojový kód nebyl průběžně uploadován na server Baryk: -5

Literatura

Demo aplet pro lepší pochopení okénkového komunikačního schématu [http://www2.rad.com/networks/2004/sliding_window/demo.html]

Doporučená literatura

- Sliding window demo [http://www2.rad.com/networks/2004/sliding_window/]
- Reliable Message Transport Protocol [<http://www.cs.ucsb.edu/~almeroth/classes/S00.276/hw1/hw1.html>]

Odkazy

- Wikipedia – BSD sockets [http://en.wikipedia.org/wiki/Berkeley_sockets]
- Rozhraní Sockets – Java [<http://java.sun.com/docs/books/tutorial/networking/TOC.html>]
- Rozhraní Sockets klient TCP a UDP – Java [<http://www.eli.sdsu.edu/courses/spring96/cs596/notes/andrew/javanet.html>]
- Transportní rozhraní – BSD sockets [<http://www.earchiv.cz/a93/a315c110.php3>]
- **Unix sockets FAQ – vynikající** [<http://www.developerweb.net/forum/forumdisplay.php?f=70>]