2. domácí úloha (Simulátor síťového provozu)

z předmětu

MI-SIB

(Síťová bezpečnost)

zimní semestr 2012/13

Vypracovali: Pavel Tvrdík, Jan Žentek

Datum: 12.06.2012

Zadání úkolu

1. Použijte traffic generator 'tg' podobně jako na cvičení pro poslání velkého datového toku s náhodnými časy mezi pakety a náhodnými velikostmi paketů (vše nezávislé náhodné veličiny). Časy mezi pakety s exponenciálním rozdělením s průměrným časem 0.02 sekund. Velikost paketů s exponenciálním rozdělením s průměrnou velikostí 576 bytů.

    - (7 bodů) Použijte monitorovací nástroj podobný tcpdump či WireShark k nasledování časů odesílání paketů. Spočtěte časy mezi pakety a ověřte, zda se jejich rozdělení schoduje s rozdělením zadaným generátoru 'tg'.

Příkazy odesílajícího pakety s mezičasy odpovídajícími rozdělení dle zadání:

```
$ sudo tcpdump -i any port 4322 > tcpdump.log
$ echo "on 15 tcp 0.0.0.0.4322 server wait 200"|./tg -f
$ echo "on 0:15 tcp 0.0.0.0.4322 at 5 setup at 6 arrival exponential 0.02
length exponential 576 time 20"|./tg –f
```

Pro odchycení odesílaných paketů jsme použili program tcpdump. Informace o zachycených paketech byly uloženy do logu a následně zpracovány příkazem:

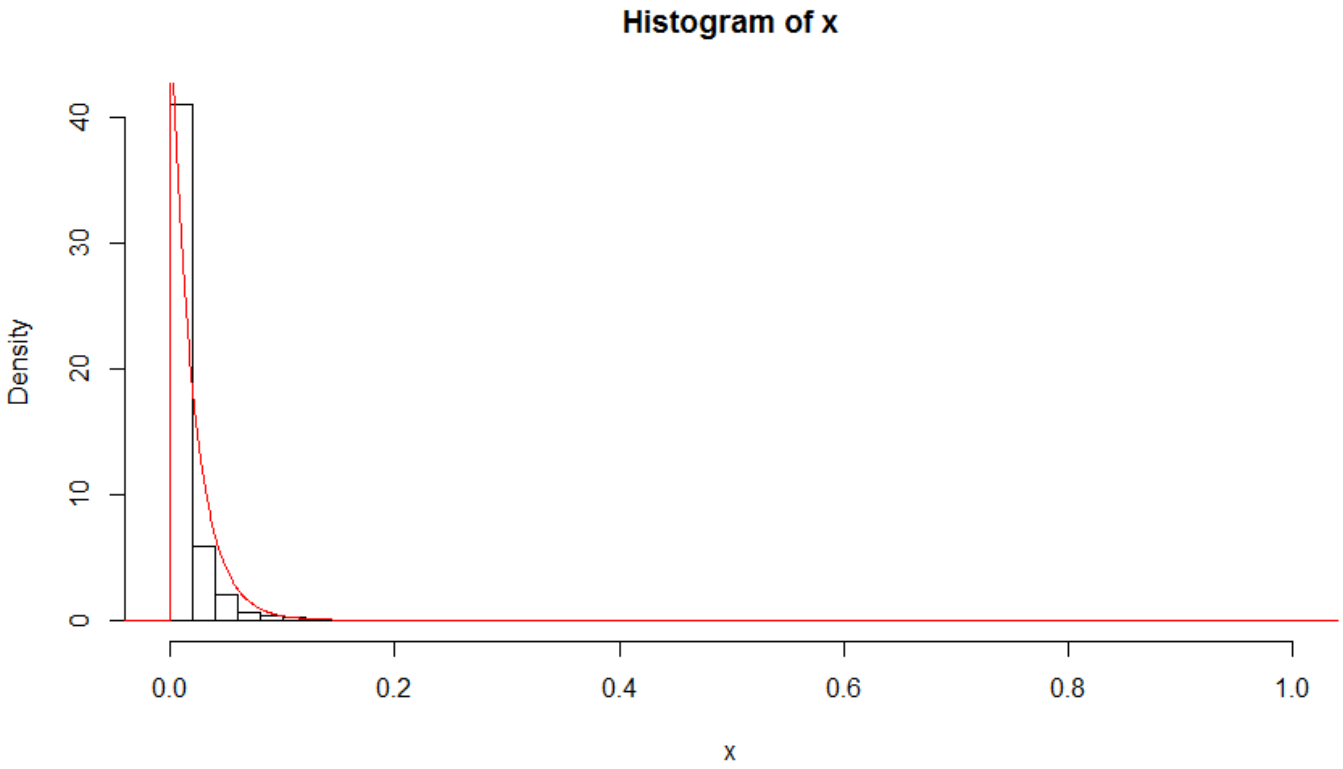
```
$ cat tcpdump.log | awk '{print $1}'|awk -F ':' '{ casy[NR] = $3} END { for
( i=2; i < NR; i++) printf("%.50f\n", (casy[i]-casy[i-1])) }' > mezicasy.txt
```

Výstupem příkazu je seznam časů mezi odesláním jednotlivých paketů. Tento seznam je dále uložen do souboru mezicasy.txt, který je zpracován statistickým programem R.

Zdrojový kód pro ověření korektnosti požadovaného rozdělení mezičasů odesílaných paketů:

```
data = read.table("C:\\Users\\Hanz\\Desktop\\SIB_2DU\\mezicasy.txt", header =
FALSE)
x <- c(data$V1)
x
hist(x, breaks = 200,freq=FALSE, probability=true)
xWidth=max(x)-min(x)
xGrid=seq(min(x)-0.2*xWidth,max(x)+0.2*xWidth,length=length(x))
lines (xGrid,dexp(xGrid, rate=1/0.02), col='red')
rm(data, expon, x, xGrid, xWidth)
```

Po spuštění zdrojového kódu je vykreslen graf, který je zobrazen na obr. 1.



Obr. Histogram rozdělení pravděpodobnosti odchycených mezičasů

Z obr. 1 je patrné, že rozdělení časů odesílaných paketů se shoduje s rozdělením zadaným generátoru 'tg'.

- (7 bodů) Použijte monitorovací nástroj podobný tcpdump či WireShark k nasledování velikostí poslaných paketů. Ověřte, zda se jejich rozdělení schoduje s rozdělením zadaným generátoru 'tg'.

Zdrojový kód skriptu odesílajícího pakety s velikostí odpovídající zadání je shodný s předchozím úkolem. Odchycená data jsou rovněž stejná. Rozdílné je pouze zpracování odchycených informací, které proběhlo pomocí následujícího příkazu[1]:

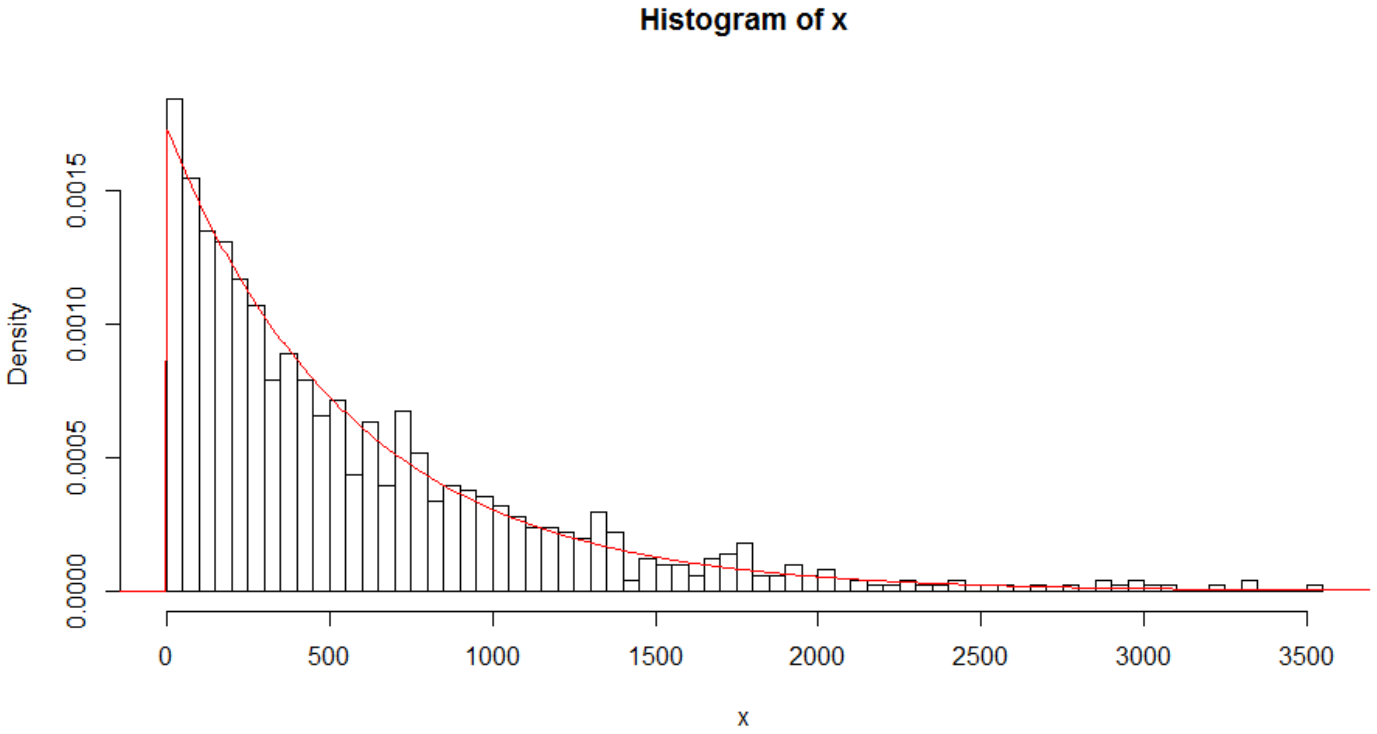
```
$ cat tcpdump.log | awk '{print $21}'|sed ':a;N;$!ba;s/\n\n/\n/g' > velikosti.txt
```

Výstupem příkazu je soubor obsahující velikosti všech odeslaných paketů. Tento soubor je následně zpracován programem R pomocí zdrojového kódu:

1První dva a poslední tři řádky souboru bylo ještě nutné smazat z důvodu odstranění dat nesouvisejících s pozorováním.

```
data = read.table("C:\\Users\\Hanz\\Desktop\\SIB_2DU\\velikosti.txt", header =
FALSE)
x <- c(data$V1)
x
hist(x, breaks = 70,freq=FALSE, probability=true)
xWidth=max(x)-min(x)
xGrid=seq(min(x)-0.2*xWidth,max(x)+0.2*xWidth,length=length(x))
lines (xGrid,dexp(xGrid, rate=1/576), col='red')
rm(data, expon, x, xGrid, xWidth)
```

Výsledek spuštění zdrojového kódu je zobrazen na obr. 2



Obr. Histogram rozdělení pravděpodobnosti velikostí jednotlivých paketů

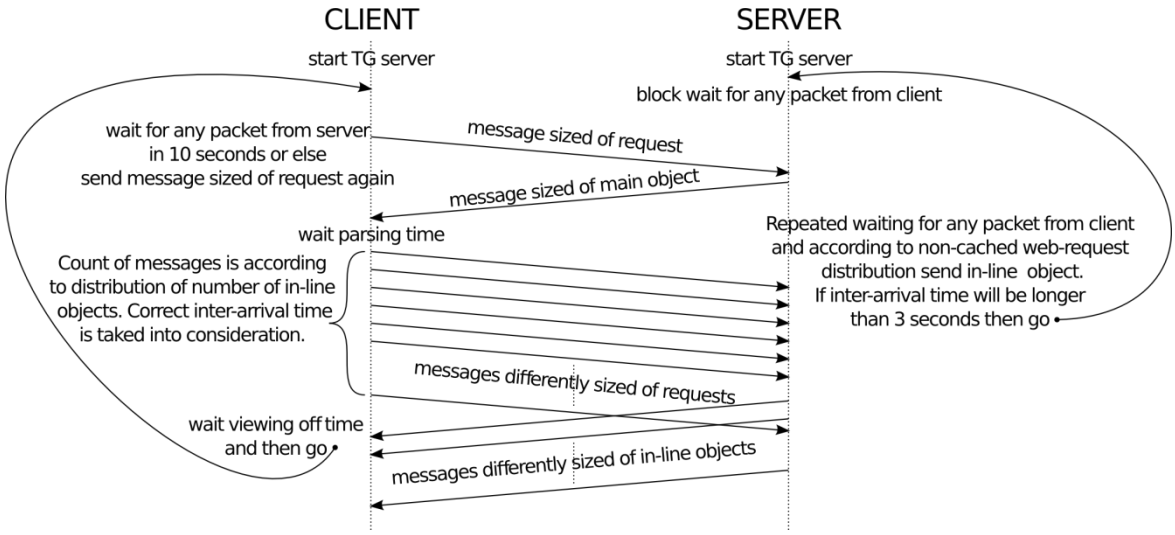
Na obr. 2 je patrné, že rozdělení velikostí paketů není úplně shodné s exponenciálním rozdělení zadaným generátoru 'tg'. Jelikož velikosti paketů nepodléhají chybě měření, tak bychom tento fakt přisuzovali nejspíše samotnému programu „tg“.

2. (16 bodů) Naprogramujte program pro generování webového provozu dle výše uvedeného článku (viz také přednáška 6.11.). Doporučená metoda je použít skriptu bash a nástrojů 'rg', 'microsleep', a 'tg' podobně jako na cvičení. Použít však můžete i jiný skriptovací či programovací jazyk a jiný generátor síťového provozu. V takovém případě však musíte použité nástroje podrobně popsat, včetně odkazů na jejich zdroj.

Pro naprogramování generátoru webového provozu jsme se drželi doporučeného postupu a napsali jej ve skriptovacím jazyce „bash“ s využitím programů „tg“, „rg“ a „microsleep“. Výsledný program je rozčleněn do tří souborů obsahujících serverovou část „server.sh“, klientskou část „client.sh“ a knihovnu „library.sh“, která obsahuje všechny potřebné metody, pro generování náhodných čísel, dle požadovaných rozdělení. Všechny soubory jsou obsaženy v příloze ve složce „src“.

Další část přílohy obsahuje vypočtené hodnoty parametrů pro generování náhodných čísel dle požadovaného rozdělení. K výpočtům byl použit online matematický kalkulátoru „Wolfram alpha“. Použité výrazy pro výpočet jsou obsaženy v souboru s vypočtenými hodnotami.

Pro představu fungování generátoru webového provozu jsme jeho princip shrnuli do obr. 3. Tento obrázek zobrazuje posloupnost jednotlivých zpráv posílaných mezi serverem a klientem během zpracování pomyslné jedné webové stránky.



Obr. Grafické znázornění životního cyklu generátoru webového provozu

Závěr

V první části úkolu se nám podařilo experimentálně ověřit, že exponenciální rozdělení na časy odeslání paketů i jejich velikost pracuje v TG správně.

V druhé části úkolu se nám podařilo naimplementovat fungující generátor webového provozu. Jeho nejslabší stránkou je vlastnost aplikace 'tg' – počáteční synchronizace více spuštěných instancí, kterou se nám nepodařilo obejít. Zkoušením jsme zjistili, že nejkratší fungující synchronizační čas jsou 2 vteřiny, přičemž bohužel 'tg' „občas“ kvůli počáteční synchronizaci skončí bez jakéhokoli odeslání dat a s návratovou hodnotou 255 nebo 1. Pak je nutné 'tg' pustit znovu. Tato vlastnost je nevýhodná zejména pro odezvu na žádost o HTML stránku (Main Object), odezvu mezi doručením HTML stránky a zažádáním o vložené objekty (In-Line Objects) a odezvu mezi přijmutím žádostí o vložené objekty a posláním objektů. Naopak se této vlastnosti dobře využije pro dodržení statistického rozdělení mezičasů odesílaných žádostí na vložené objekty.
Kdyby se program 'tg' nahradil jiným generátorem se startupem blížící se nule, pak by měl výsledný webový generátor odpovídat modelu popsanému v článku: A Behavioral Model of Web Traffic.