

Ime in priimek:

Datum:

Laboratorijska vaja: Izvajanje simulacij v omrežnem simulatorju NS2

1 Namen

Namen vaje je seznaniti učečega z izvajanjem simulacij v omrežnem simulatorju NS2. Posamične simulacije se lotimo po korakih, in sicer v naslednjem vrstnem redu:

- postavitve odpravnikov dogodkov
- postavitve topologije omrežja
- opis usmerjanja omrežja
- postavitve prometa
- vstavljanje napak
- sledenje prometu

2 Potek

Za izvedbo vaje bomo uporabili NS2 simulator (Network Simulator version 2) ter omrežni animator NAM. Pogleдали si bomo različne primere uporabe simulatorja NS2.

- Poženite VMware player in v njem odprite datoteko »Red Hat Linux.vmx«. Počakajte, da se sistem naloži in se vanj prijavite kot `root`. Odprite ukazno okno.
- Preverite, če se nahajate v mapi `/root/`, oz. pojdite tja:

```
#pwd
```

2.1 Osnovna skripta za zagon simulacije

Za začetek naj vam bo v pomoč spodnja koda (`vaja0.tcl`), ki jo boste uporabljali v vseh `tcl` skriptah.

`#postavitev odpravnika dogodkov`

```
set ns [new Simulator]
```

`#odpiranje datoteke za vpis podatkov, ki bodo kasneje prikazani v omrežnem animatorju`

```
set nf [open out.nam w]
```

```
$ns namtrace-all $nf
```

`#zaključna procedura, ki zapre sledilno datoteko in požene omrežni animator`

```
proc finish {} {  
    global ns nf  
    $ns flush-trace  
    close $nf  
    exec nam out.nam &  
    exit 0  
}
```

#določitev časa izvedbe zaključne procedure

```
$ns at 5.0 "finish"
```

#v zadnji vrstici je ukaz za začetek simulacije

```
$ns run
```

2.2 Izvajanje simulacij

Na osnovi podanega seznama ukazov na predavanjih v NS2 izvedite sledeče naloge oz. simulacije.

Naloga 1a:

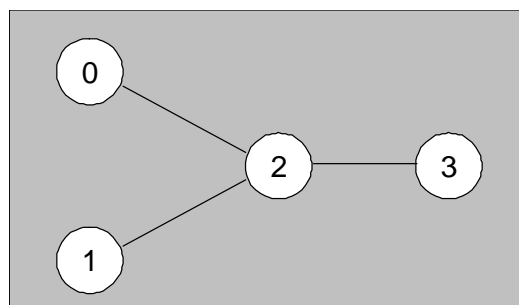
V simulaciji postavite dve vozlišči in dvosmerno linijo med njima. Hitrost povezave je 1 Mbit/s. Zakasnitev na liniji je 10 ms, kar ustreza fizični dolžini linije 3000 km, če EM val potuje s hitrostjo $c = 3 \cdot 10^8$ m/s. Uporabite izhodno čakalno vrsto DropTail. Kodo vpišite pred zaključno (`finish`) proceduro. Shranite datoteko pod imenom `vaja1.tcl` in jo poženite z ukazom `ns vaja1.tcl` v direktoriju, kjer se nahaja datoteka, sicer morate navesti celotno pot do datoteke. Vsi ukazi se izvajajo v terminalnem oknu (angl. terminal window).

Naloga 1b:

Izvedite prenos podatkov od vozlišča `n0` do vozlišča `n1`. Uporabite prejšnjo skripto. Pri NS2 simulatorju se podatki dejansko prenašajo od enega agenta do drugega. Zato je potrebno najprej postaviti oddajnega agenta na vozlišče `n0` in sprejemnega agenta na vozlišče `n1`. V tem primeru bomo delali z UDP agenti. Za tem je potrebno postaviti prometni izvor (CBR generator) na oddajnega agenta in ponor (`Null`) na sprejemnega agenta. Postavite velikost paketov na 500 byteov ter časovni interval za ponovno oddajo paketa na 0,005 s (vsakih 0,005 s generator odda paket). Končno še oba agenta povežite skupaj ter določite začetek in konec oddaje CBR generatorja. Oddaja paketov se prične ob času $t = 0,5$ s in konča ob času $t = 4,5$ s. Oddaja se mora končati pred časom $t = 5$ s. Kliknite na gumb predvajaj omrežnega animatorja in razišči simuliran primer tako, da kliknete z miško na linijo, posamezna vozlišča ter na pakete, ki vas zanimajo.

Naloga 2a:

Postavite 4 vozlišča od `n0` do `n3` ter jih povežite kot kaže slika 1. Lastnosti povezav so iste kot v prejšnji nalogi.



Sl. 1: Topologija

Skripto shranite pod imenom `vaja2.tcl` in jo poženite v terminalnem oknu. V omrežnem simulatorju topologija ni tako lepo postavljena, kot je na sliki 1, zato uporabite sledeče ukaze:

```
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
```

Naloga 2b:

Na osnovi topologije iz naloge 2a postavi UDP agenta ter CBR generator na vozlišča `n0` in `n1`. Pri obeh je velikost paketov 500 byteov in časovni interval za ponovno oddajo paketa 0,005 s. Za tem postavite agent Null na vozlišče `n3`. Pri tem ne pozabite povezati oba oddajna agenta s sprejemnim Null agentom. Na koncu dodajte še ukaze za začetek oddaje prvega generatorja ob času $t = 0,5$ s in konec ob $t = 4,5$ s ter za drugi generator začetek oddaje paketov ob $t = 1,0$ s in konec oddaje ob $t = 4,0$ s. Poženi simulacijo in opazuj dogajanje v omrežnem animatorju. Zaradi večje obremenitve, kot jo povezava med vozliščema `n1` in `n2` zmore, pride do zavrženja paketov na vozlišču `n2`. Na žalost pa ne vidimo kateri paketi so zavrženi – od vozlišča `n0` ali od vozlišča `n1`? Tok paketov vozlišč `n0` in `n1` lahko ločimo z različnimi barvami, če dodamo naslednjo kodo:

#dodajte na začetek `tcl` skripte takoj po ukazu `set ns [new Simulator]`

```
$ns color 1 Blue
$ns color 2 Red
```

#dodaj poleg kode od vsakega od postavljenih oddajnih agentov

```
$udp0 set class_ 1
$udp1 set class_ 2
```

Če želimo opazovati napolnjenost izhodne čakalne vrste vozlišča `n2`, je potrebno pod kodo, kjer smo postavili dvosmerne povezave, dodati še:

```
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

V primeru, da spremenimo disciplino čakalne vrste iz DropTail na SFQ (Stochastic Fair Queueing), bi moralo biti število zavrženih rdečih in modrih paketov enako. To preverite tako, da nadomestite DropTail disciplino z disciplino SFQ in poženete simulacijo ter opazujete dogajanje v omrežnem animatorju.

Naloga 3:

Pri tej nalogi bomo simulacije izvajali na nakoliko večji topologiji, in sicer s sedmimi vozlišči. Da poenostavimo ustvarjanje vozlišč, uporabimo naslednji kos kode:

```
for {set i 0} {$i < 7} {incr i} {
    set n($i) [$ns node]
}
```

Sedaj so vozlišča označena kot polje z elementi od $n(0)$ do $n(6)$, zato je potrebno paziti, da v nadaljevanju pišemo oznake za vozlišča na takšen način. Vozlišča želimo povezati v obroč, zato je potrebno dodati še:

```
for {set i 0} {$i < 7} {incr i} {  
    $ns duplex-link $n($i) $n([expr ($i+1)%7]) 1Mb 10ms DropTail  
}
```

V naslednjem koraku omogočite prenos podatkov od vozlišča $n(0)$ do $n(3)$. Zopet uporabite UDP agent ter CBR generator z istima vrednostma parametrov kot v prejšnjih nalogah. Start generatorja je ob $t = 0,5$ s ter stop ob $t = 4,5$ s. Sedaj lahko že poženetes simulacijo.

Kot dodatek bomo simulacijo dopolnili tako, da se ob času $t = 1,0$ s prekine povezava med vozliščema $n(1)$ in $n(2)$, ter se ob času $t = 2,0$ s zopet vzpostavi. V ta namen dodajte ustrezen ukaz na začetku skripte po postavitvi odpravnika dogodkov (navodila za ustrezne ukaze poiščite na prosojnicah iz predavanj). Prekinitev opazujete na simulaciji; pot paketov se v obdobju prekinitve preusmeri preko vozlišč $n(6)$, $n(5)$ in $n(4)$.

Ime in priimek:

Datum:

Laboratorijska vaja: Simulacija delovanja paketnih stikal v NS2

1 Namen

Ogledali si bomo delovanje paketnih stikal s simulatorjem diskretnih dogodkov v omrežjih – NS2. Simulirali bomo paketa stikala z vhodnim čakanjem IQ (Input Queueing), navidezno izhodnim čakanjem VOQ (Virtual Output Queueing) ter izhodnim OQ čakanjem (Output Queueing). Opis konfiguracije posameznega paketnega stikala se nahaja v Tabeli 1.

| Tip stikala | Velikost medpomnilnika | Število vhodov oz. izhodov | Tip prometnega vira |
|-------------|------------------------|----------------------------|-------------------------|
| IQ | 16 | 4 | poissonov |
| VOQ | 4 | 4 | poissonov |
| OQ | 16 | 4 | konstantni bitni pretok |

Tabela 1. Konfiguracija stikala

2 Potek

2.1 Navodila za izvedbo simulacije

- Poženite VMware player in v njem odprite datoteko »Red Hat Linux.vmx«. Ko se sistem naloži, se vanj prijavite kot `root`. Odprite ukazno okno.
- Preverite, če se nahajate v mapi `/root/`, oz. pojdite tja.
- Razpakirajte arhiv z orodji za vajo in se premaknite v delovni direktorij:

```
#tar -xvf vaja.tar  
#cd Vaja-PaketnaStikala
```

- Poženite skripto `runSim.bash` z ukazom:

```
#!/runSim.bash
```

- Počakajte, da se simulacija konča. Z ukazom `ls` se izpiše seznam vseh datotek v trenutnem imeniku. Iz izpisa morajo biti vidne tudi datoteke s končnico `.tr`.
- Med tem, ko čakate na zaključek simulacije, si oglejte vsebino skripte `runSim.bash` in `statData.bash` ter strukturo sledilnih datotek `.tr`.

2.2 Potek pridobivanja statističnih rezultatov (prepustnost, povprečna in maksimalna zakasnitev paketov)

- po končani simulaciji poženite skripto `statData.bash` z ukazom:

```
#!/statData.bash PAR1 PAR2
```

PAR1 = tip paketnega stikala (IQ, VOQ ali OQ)

PAR2 = obremenitev od 10 do 120 procentov s korakom po 10

- zgornjo skripto poženite za vsako paketno stikalo posebej pri vseh obremenitvah (od 10 do 120)

2.3 Risanje grafov

- dobljene rezultate vpišite v `xls - predloga1`. Na prvi listu vpišite rezultate prepustnosti paketnih stikal, na drugega pa rezultate povprečne in maksimalne zakasnitve paketov. Ob vnosu podatkov v označene tabele se točke avtomatsko izrisujejo v ustrezen graf.

¹ <http://lt.fe.uni-lj.si/gradiva/OTK2/>

Ime in priimek:

Datum:

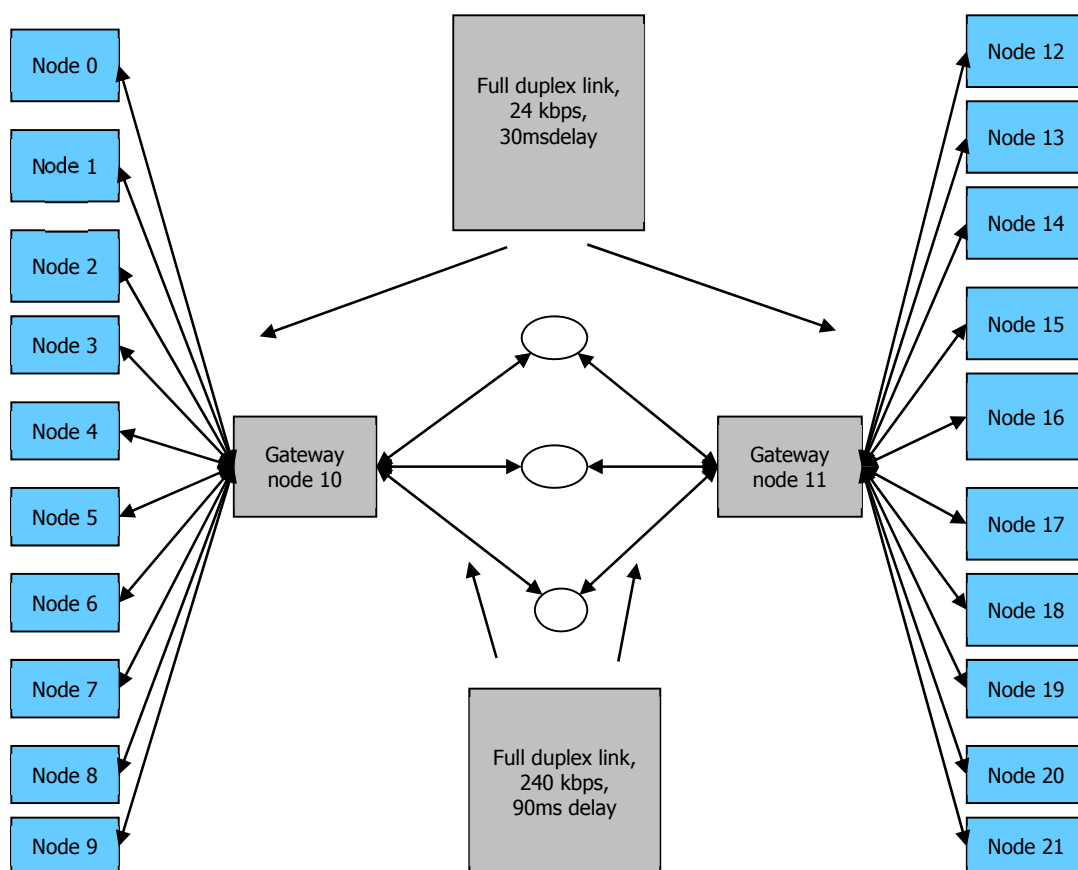
Demonstracija rezultatov simulacije VoIP z uporabo NS2

1 Namen

Namen simulacije je spoznati delovanje omrežja VoIP (angl. *Voice over Internet Protocol*) preko rezultatov, ki vključujejo grafe različnih parametrov (povprečne zakasnitve, izgube paketov, potresavanje zakasnitve – pri različnih obremenitvah).

2 Simulacija omrežja VoIP

Topologija omrežja VoIP je prikazana na sliki 1. Uporabnike predstavljajo končna vozlišča na dveh ločenih straneh. Komunikacija poteka v obe smeri (angl. *full duplex*), in sicer med vozliščem n0 in n12, n1 in n13 in tako naprej.



Sl 1: Topologija omrežja VoIP

2.1 Nastavitve parametrov simulacije

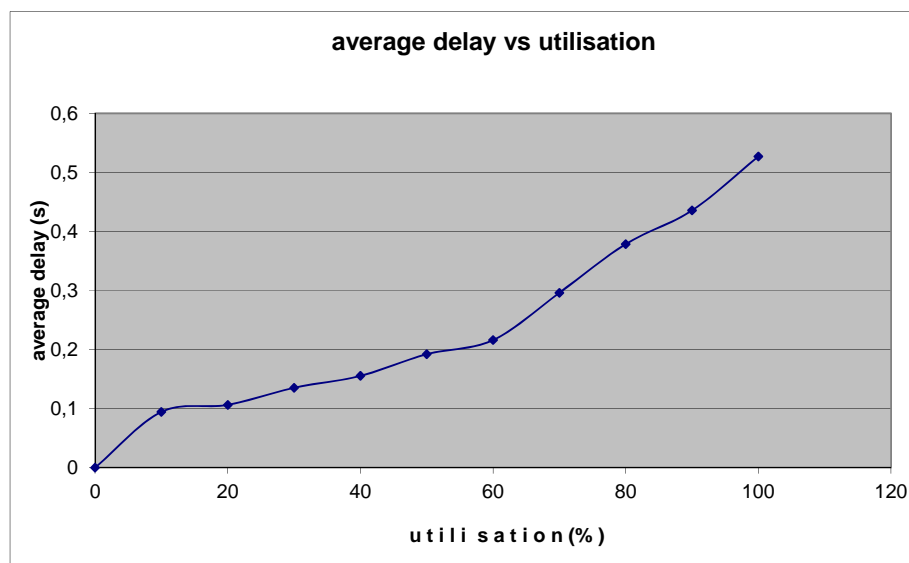
Parametri simulacije in njihove vrednosti:

- velikost celotnega paketa je 60 byteov
 - RTP glava – 12 byteov
 - UDP glava – 8 byteov
 - IPv4 glava – 20 byteov
 - uporabniški podatki 20 byteov
- zakasnitve – celotna zakasnitev v eno smer znaša 150 ms:
 - 2 x 12.5 ms za kodek (odhodni in dohodni)
 - 25 ms za izločanje odboja (angl. *echo cancellation*)
 - 50 ms zakasnitve omrežja
 - 50 ms zakasnitve na prehodih (angl. *gateway*) – 25 ms za vsakega
- hitrost oddaje je 50 paketov na sekundo, torej 24 kbps.
- čas simulacije je 16 sekund. Prvi izvor je pričel oddajati ob času $t = 0$ s (10 % izkoriščenost pasovne širine), drugi ob $t = 1$ s (20 % izkoriščenost pasovne širine) in tako naprej do desetega izvora, ki prične oddajati ob $t = 9$ s (100 % izkoriščenost pasovne širine).

2.2 Rezultati simulacije

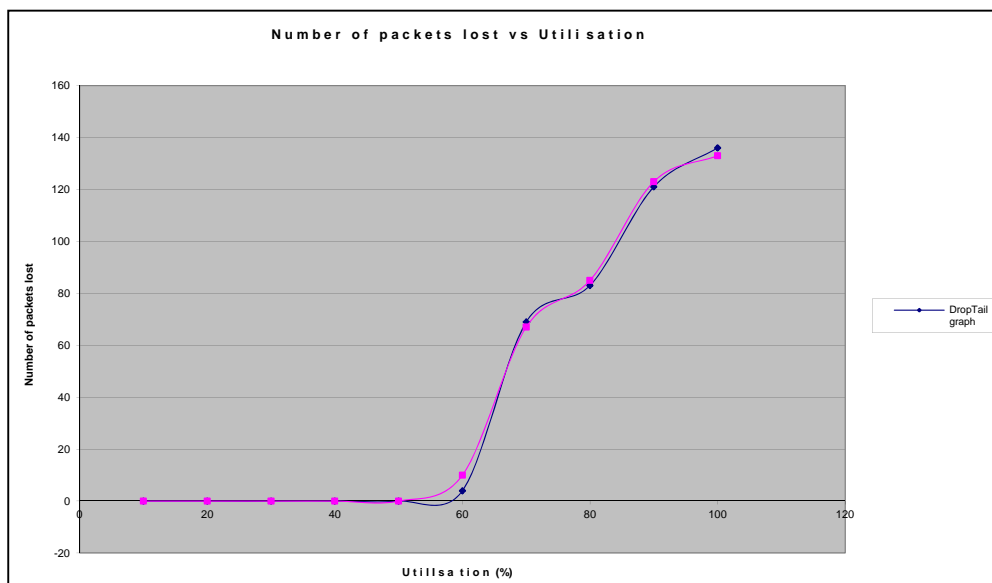
Rezultati simulacije so podani v spodnjih grafih. Komentirajte rezultate!

a.) povprečna zakasnitev



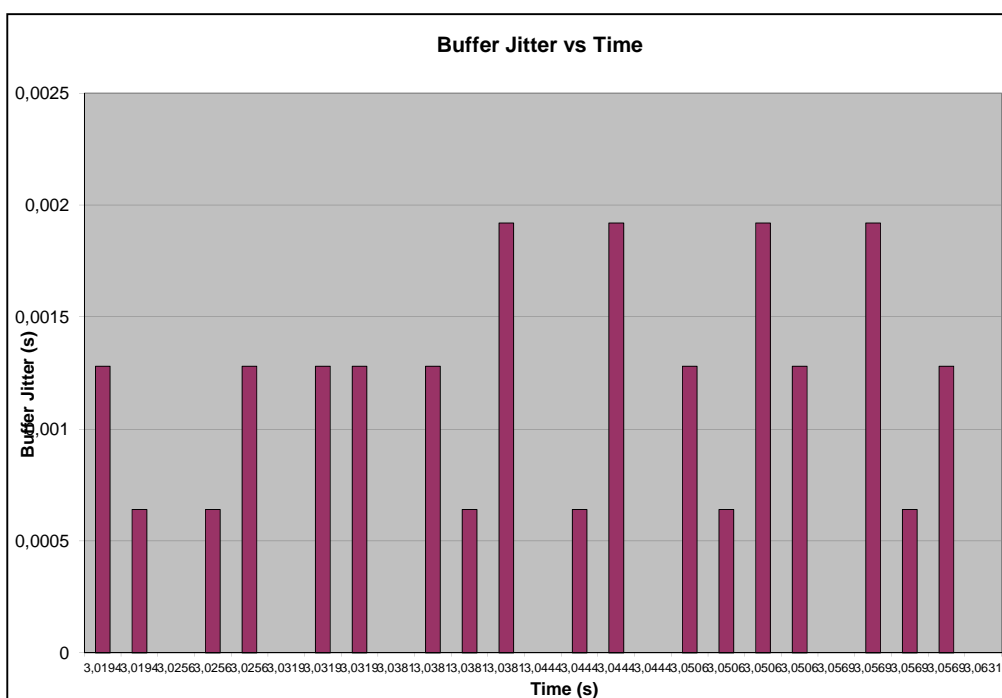
Sl 2: Povprečna zakasnitev v odvisnosti od izkoriščenosti omrežja

b.) število izgubljenih paketov

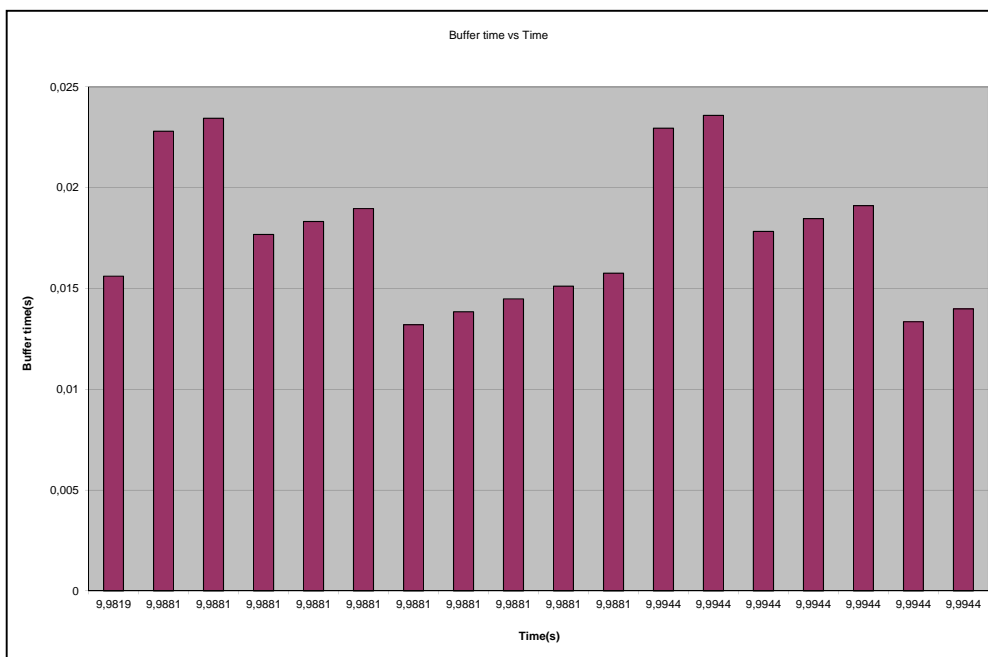


SI 3: Izguba paketov v odvisnosti od izkoriščenosti omrežja

c.) potresavanje v medpomnilniku (angl. *buffer jitter*) pri 40% in 100 % izkoriščenosti pasovne širine

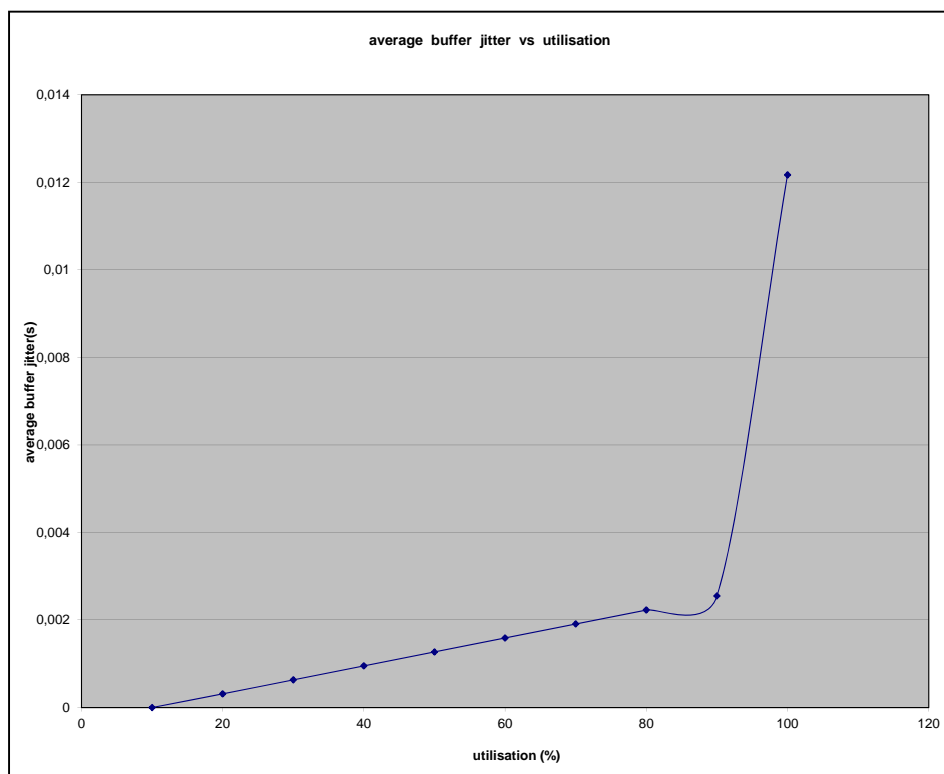


SI 4: Potresavanje v odvisnosti od časa pri 40 % obremenitvi omrežja



SI 5: Potresavanje v odvisnosti od časa pri 100 % obremenitvi omrežja

d.) povprečno potresenje faze (angl. *average buffer jitter*)



SI 6: Povprečno potresavanje v odvisnosti od obremenitve omrežja