

Modeliranje računalniških omrežij
Študijsko leto 2016/2017

**Poenostavljen IP protokol v orodju
OMNeT++**

Končno poročilo prve seminarske naloge

Aljaž Markežič, Žan Valter Dragan
Vpisna št. 63140157, 63140045

Ljubljana, 26. november 2016

Kazalo

1	Uvod in motivacija	2
2	Opis internetnega protokola IP	2
3	Uporabljene knjižnice	2
4	Rešitev	3
5	Simulacije in rezultati	4
5.1	Brez izgubna konfiguracija	4
5.2	Delno izgubna konfiguracija	5
5.3	Izgubna konfiguracija	9
6	Zaključek	10
7	Literatura	11

1 Uvod in motivacija

Za prvo seminarsko nalogo sva si izbrala poenostavljeno implementacijo IP protokola. Želiva preizkusiti delovanje protokola v pravokotnem omrežju in le-to stestirati. S tem si bova pridobila osnovno znanje v postavljanju omrežji v OMNeT++ in njihovem testiranju.

2 Opis internetnega protokola IP

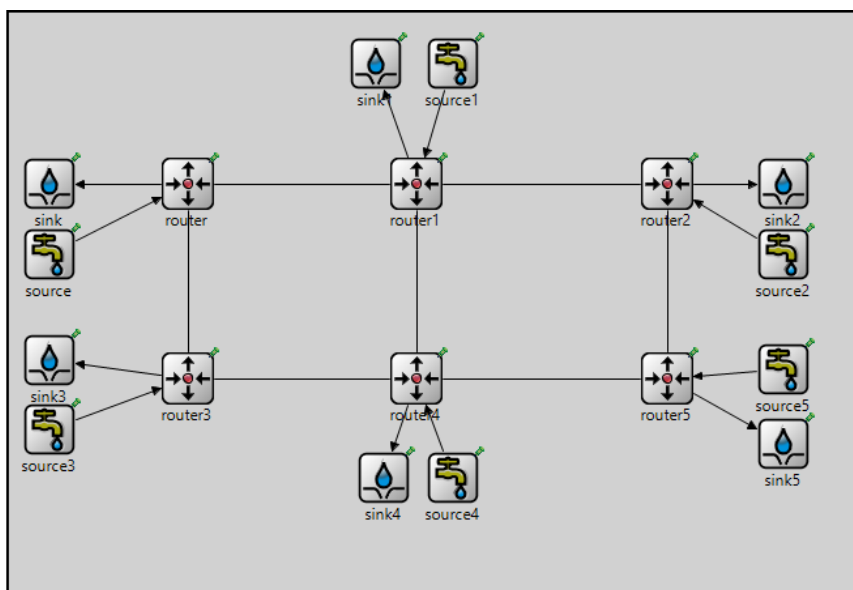
Internetni Protokol oziroma IP je osnovni komunikacijski protokol za prenašanje datagramov preko omrežja. Njegova usmeritvena lastnost omogoča medmrežje in s tem tudi vzpostavi internet. Njegova naloga je dostava paketov od izvora do ponora zgolj s podatkom o IP naslovom, kateri se nahaja v glavi paketa. V ta namen ima IP definirano strukturo paketa, ki obdaja podatke, ter natančno določene metode naslovljevanja za zapis izvora in ponora. Trenutno poznamo dve verziji IP protokola, in sicer IPv4 ter IPv6.

3 Uporabljene knjižnice

Za potrebe te seminarske naloge nisva uporabila nobene knjižnice.

4 Rešitev

Za izvedbo seminarja sva izdelala omrežje v katerem so po 6 izvorov, ponorov in usmerjevalnikov. Naprave so bile razporejene v pravokotnik, kjer so bili štirje usmerjevalniki v oljiščih, dva pa na polovici daljših stranic. Vsak izmed usmerjevalnikov je imel še po en izvor ter en ponor.



Slika 1: Topologija omrežja

V začetku simulacije si usmerjevalniki pošljejo sistemske pakete samim sebi, kateri so ukaz za izvajanje broadcast zahteve. Le-ta je poizvedba po IP naslovih drugih naprav znotraj omrežja. S ponavljajočim se broadcast signalom se usmerjevalne tabele usmerjevalnikov napolnijo z IP naslovi.

Nato se prične pošiljanje dejanskih sporočil. Izvor dobi sistemski klic, ki naznanja čas za izdajo zahteve svojemu usmerjevalniku, da je potrebno poslati novo sporočilo. Usmerjevalnik nato naključno zgenerira IP naslov ter pošlje paket na ustrezna vrata. Naslednji usmerjevalnik nato preveri, ali je morebiti on ciljna naprava. Če je, paket posreduje na svoj ponor, sicer zopet na prava vrata.

V primeru, da so procesi usmerjevalnika zasedeni, se novi paketi postavljajo v vrsto, katera deluje po principu FIFO (First In First Out - Prvi Notri Prvi Ven). Če se vrsta ravno tako polna, se paket zavrže.

5 Simulacije in rezultati

Simulirala sva delovanje omrežja pod različnimi pogoji in obremenitvami. Za potrebe meritev sva uporabila tri različne signale (število izgubljenih paketov, dolžina vrste, poraba procesov). Iz vrednosti, katere so posredovali ti signali, sva izdelala grafe z vgrajenim orodjem.

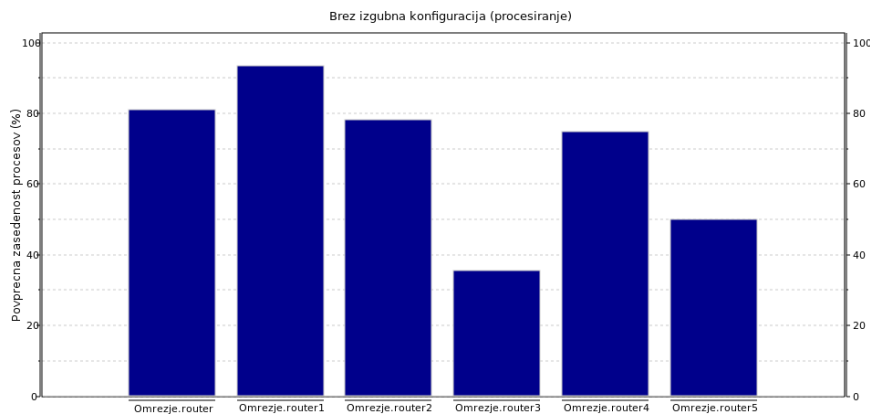
Za prvi dve konfiguraciji sva uporabila prednastavljeno dolžino vrste (60).

Pri simulaciji sva se osredotočila na tri skupine:

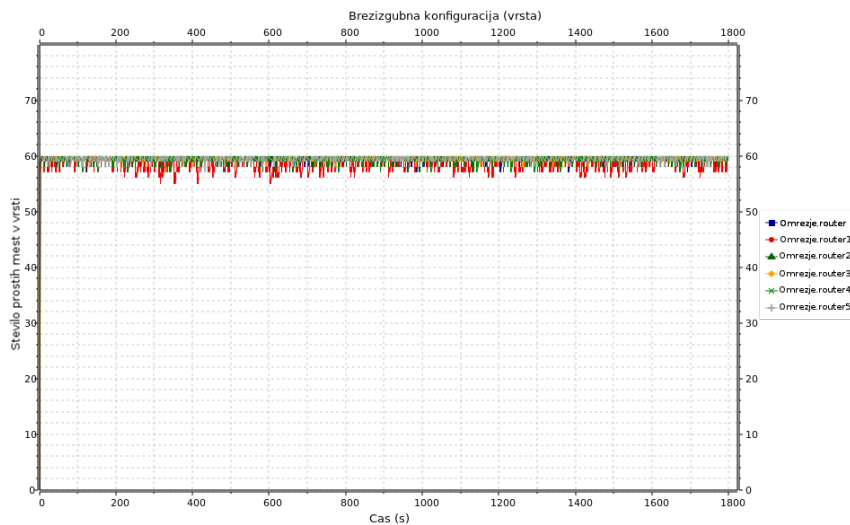
- brez izgubna konfiguracija
- delno izgubna konfiguracija
- izgubna konfiguracija

5.1 Brez izgubna konfiguracija

Pri tej simulaciji sva želela doseči, da se noben izmed paketov ne izgubi. Torej se vrste niso smele nikoli napolniti. Dolžina časa izvajanja je bila postavljena na 1800s, *interArrivalTime* pa na 10s. Parametra *processingTime* pri tej konfiguraciji nisva spremenila - ostal je 1s.



Slika 2: Zasedenost procesov

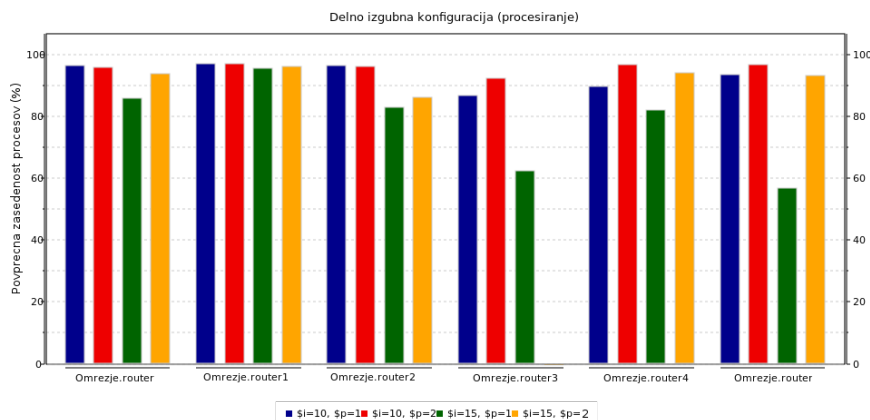


Slika 3: Zasedenost vrste

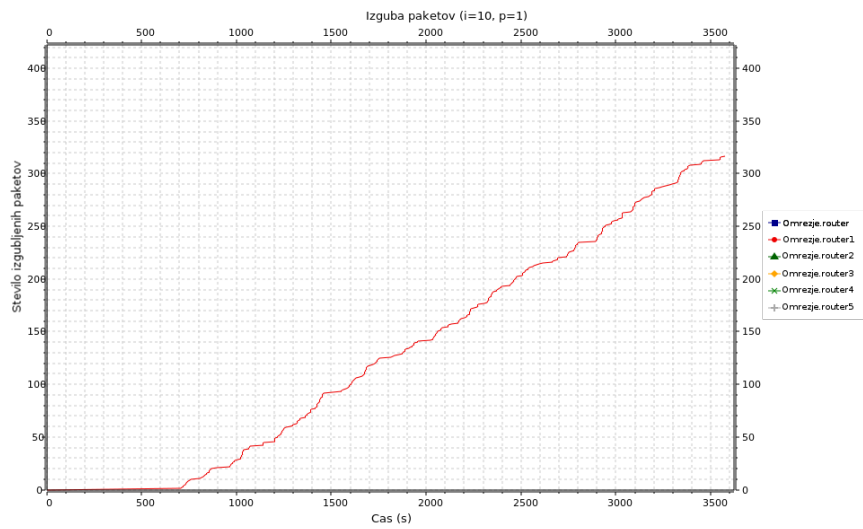
Čeprav sva šla za tem, da se procesi sproščajo, so se vendarle polnile tudi vrste, a nikoli več kot 5 paketov naenkrat. Iz zgornjih rezultatov je možno razbrati, da je *Router1* najšibkejši člen tega omrežja, kar sva nekako pričakovala.

5.2 Delno izgubna konfiguracija

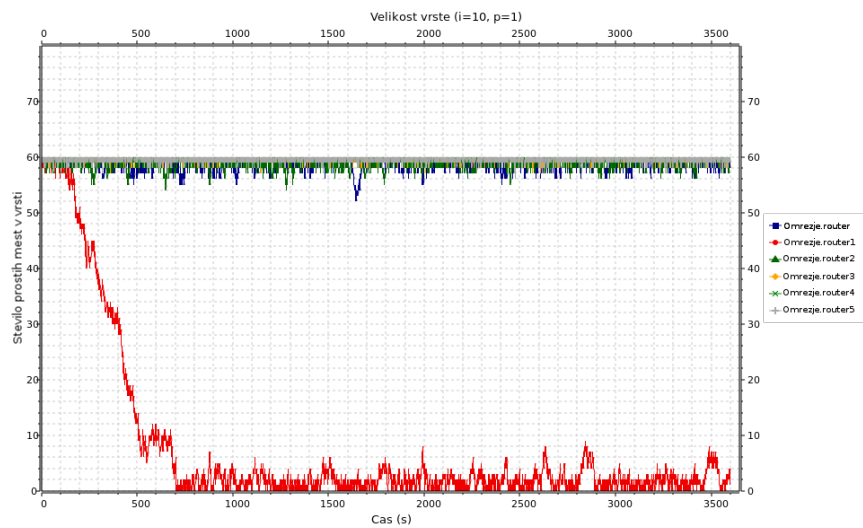
Nekoliko bolj zamudno je bilo iskanje prave mere delno izgubne konfiguracije. S kombinacijo parametrov *interArrivalTime* ter *processingTime* sva želela demonstrirati obremenitev omrežja in raziskati, kje so šibke točke. Da bi bili podatki še bolj izraziti, sva podaljšala tudi čas simulacije na 3600s.



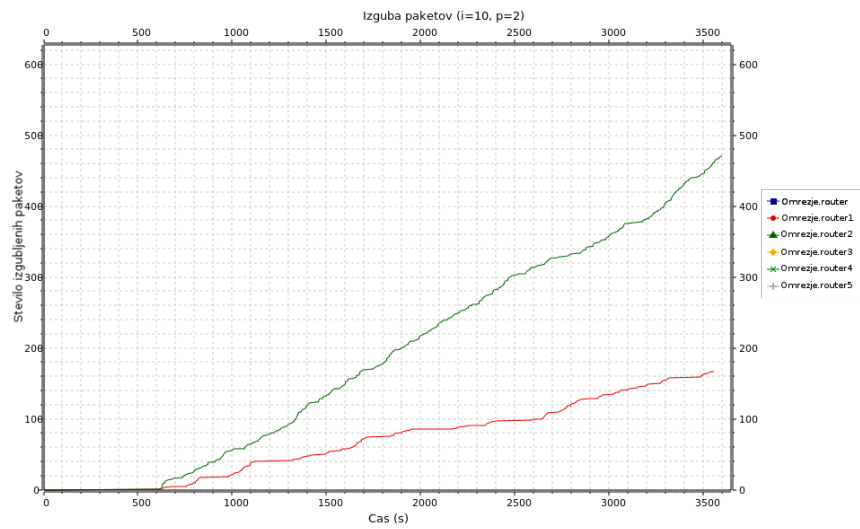
Slika 4: Zasedenost procesov



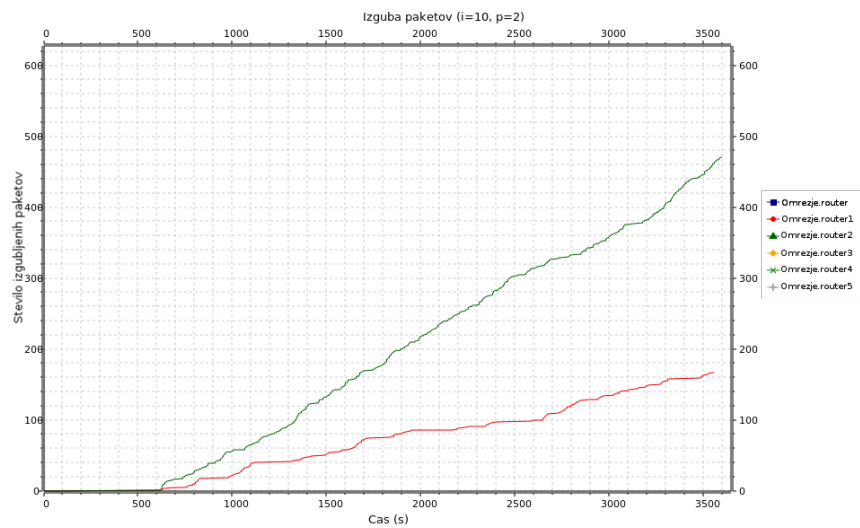
Slika 5: Izguba paketov ($i=10$, $p=1$)



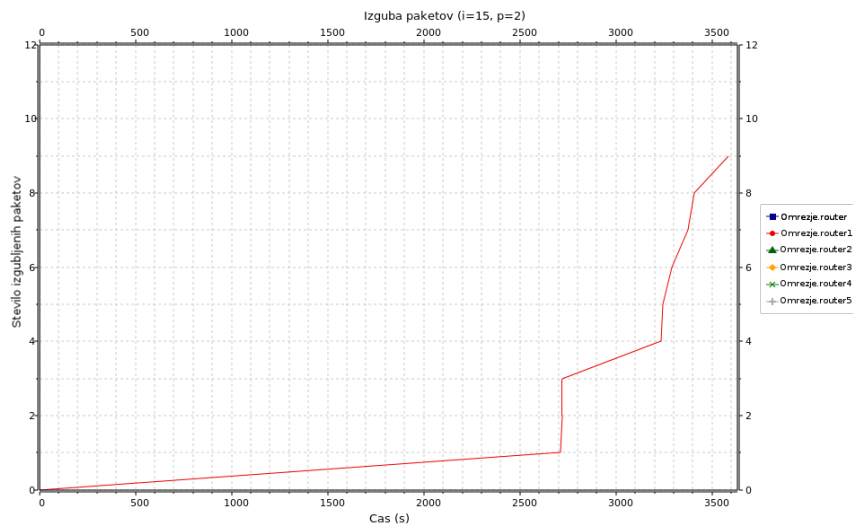
Slika 6: Čakalna vrsta ($i=10$, $p=1$)



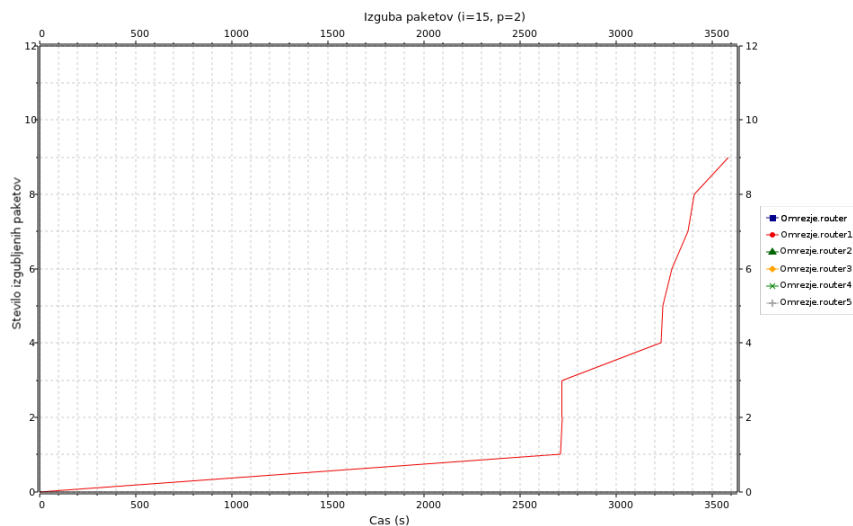
Slika 7: Izguba paketov (i= 10, p=2)



Slika 8: Čakalna vrsta (i= 10, p=2)



Slika 9: Izguba paketov (i= 15, p=2)



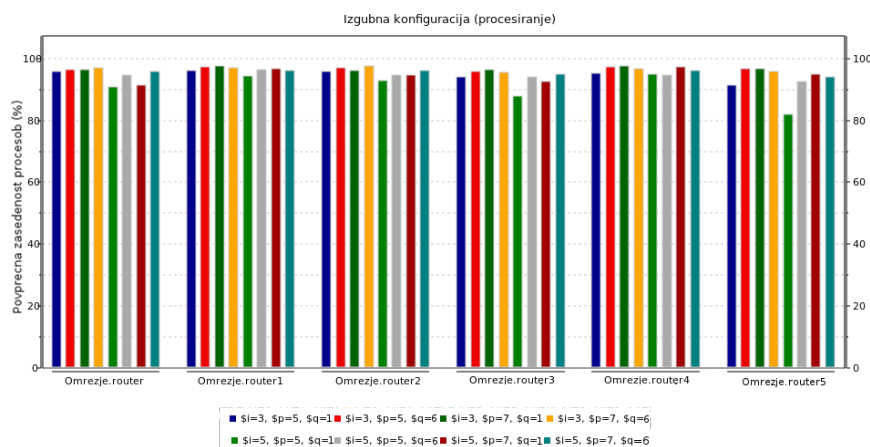
Slika 10: Čakalna vrsta (i= 15, p=2)

Izvedla sva tudi simulacijo z (i=15, p=1), a tam ni prišlo do nobene izgube paketov. Ravno tako je bila najbolj polna vrsta usmerjevalnika Router1, kjer je bilo v eni točki 6 paketov.

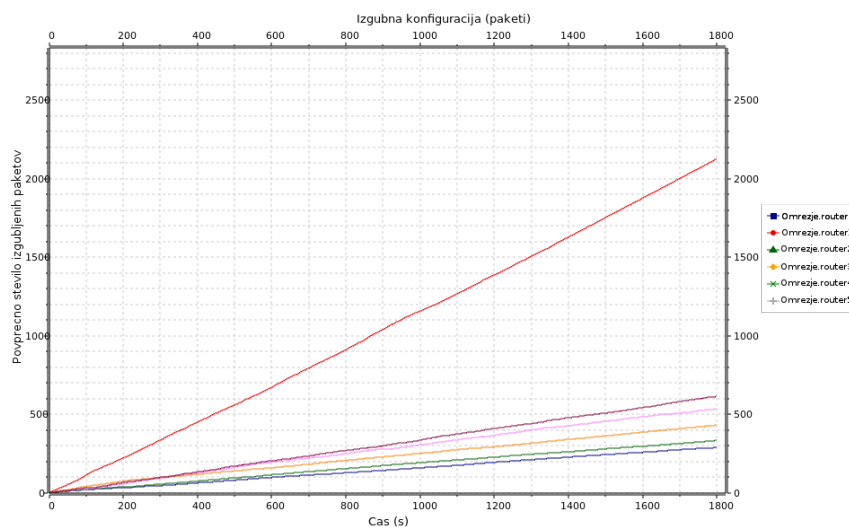
Presenetljivo se je za šibko točko izkazal tudi Router2, kar je bilo v nasprotju z najini pričakovanji. V konfiguraciji z (i=10, p=2) je imel celo večjo izgubo kot Router1. Ti rezultati bi lahko izhajali iz načina določanja naslovnika paketov, kateri je psevdo naključni. Sicer bi najverjetneje vendarle prišlo do največje obremenitve na usmerjevalnikoma Router1 ter Router4.

5.3 Izgubna konfiguracija

V tej simulaciji sva želela, da se izguba paketov prične čimprej, zato sva dolžino vrste zmanjšala na 1 do 6. Ker se je izguba dokaj hitro pričela poznati, sva zopet nastavila čas simulacije na 1800s.



Slika 11: Zasedenost procesov



Slika 12: Povprečna izguba paketov

Ponovno so rezultati pritrdili najini domnevi o usmerjevalniku Router1 kot najšibkejši člen omrežja. Po drugi strani pa je toliko bolj presenetil Router4, čigar izgube so sedaj zaostajale za usmerjevalnikoma Router5 in Router3.

6 Zaključek

Najin cilj implementacije poenostavljene verzije internetnega protokola je bil dosežen. Ravno tako sva preverila svoje domneve glede šibkega člena tovrstne topologije omrežja, kjer so se rezultati nekoliko razlikovali od pričakovanih. Projekt bi se dalo nadgraditi še z ACK paketi, kateri sedaj niso implementirani. Za to bi morali imeti usmerjevalniki interni spomin paketov, ter spremljati pretečeni čas od oddaje le-tega. Če v določenem času ne bi prejeli ACK odgovora, bi paket ponovno poslali. Pri izdelavi projekta in simulaciji sva si pomagala z dokumentacijo na uradni strani orodja OMNeT++.[2] Za opis IP protokola sva uporabila Wikipedijo.[1]

7 Literatura

- [1] Web source. Internet protokol - wikipedija. https://en.wikipedia.org/wiki/Internet_Protocol.
- [2] Web source. Orodje omnet++. <https://omnetpp.org/documentation>.