

PLATFORM FOR EVALUATION AND ANALYSIS OF CONTROL PLANE PERFORMANCE IN SOFTWARE-DEFINED NETWORKS

ПЛАТФОРМА ЗА ИЗСЛЕДВАНЕ И ОЦЕНКА ПРОИЗВОДИТЕЛНОСТТА НА УПРАВЛЯВАЩАТА РАВНИНА В СОФТУЕРНО ДЕФИНИРАНИТЕ МРЕЖИ

Diyana Kyuchukova, Georgi Hristov

Department of “Telecommunications”, University of Ruse “Angel Kanchev”, 8 “Studentska”
Str, 7017, Ruse, Bulgaria, phone.: +359 82 888 663, e-mail: {dkyuchukova, ghristov}@uni-ruse.bg

Дияна Кючукова, Георги Христов

Катедра „Телекомуникации“, Русенски университет „Ангел Кънчев“, ул. „Студентска“
№8, 7017, Русе, България, тел.: +359 82 888 663, e-mail: {dkyuchukova, ghristov}@uni-ruse.bg

Keywords: SDN, control plane, controller placement problem, mininet

Abstract – Finding the best controller location in the control plane of Software-Defined networks is an important task that is related to the whole network performance. In order to find methods for improving network performance it is necessary to perform thorough analysis of the metric or criteria that affect the controller placement. The analysis can be performed as a survey but in order to achieve better results conducting simulation or practical researches are required. Based on results of such researches correlation between the criteria that affect network performance can be detected. Different simulation and emulation programs can be used for the implementation of such researches. In this paper the authors provide a platform capable for conducting emulations and further analysis and estimation of SDN performance.

Резюме – Определянето на местоположение за контролерите в управляващата равнина на SDN мрежите е важна задача, която засяга производителността на мрежата. За да се намерят методи за подобряване на работоспособността на мрежа е необходимо да се извърши задълбочен анализ на факторите, които влияят на избора къде да бъде поставен контролера. Този анализ може да се направи на база проучване, но за да се постигнат по-добри резултати е необходимо да се пристъпи към провеждане на симулационни и / или практически изследвания, чрез които да се открие корелационната зависимост на факторите, които влияят върху производителността и да се търси метод за нейното подобряване. За провеждането на такива изследвания биха били полезни различни софтуерни симулационни и емуляционни продукти. В настоящата статия се предлага платформа за изследване и оценка производителността на Софтуерно дефинираните мрежи.

1. ВЪВЕДЕНИЕ

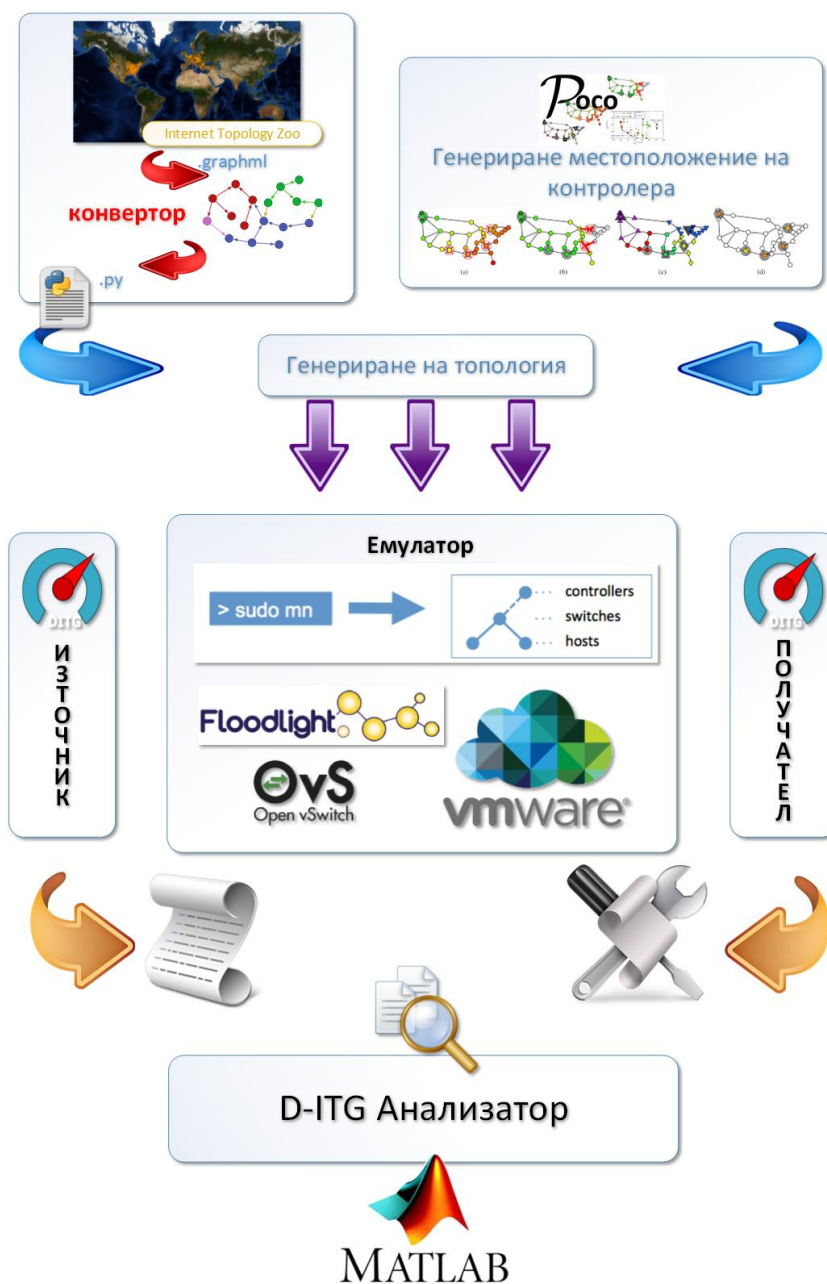
В настоящата статия е предложена платформа за изследване и оценка производителността на управляващата равнина на Софтуерно дефинираните мрежи. Както става ясно от редица проучвания [1,2,5] определянето на местоположение на контролера е важна задача свързана с производителността на мрежата. Съществуват редица критерии като мрежово време закъснение, толеранс към отказ от работоспособност на компонент от мрежата и баланс на натоварването, които пряко засягат производителността. Тези критерии се вземат под внимание, когато се определя местоположението на контролерите. Тук е важно да се отбележи, че в повечето сценарии те взаимно се конкурират и тогава решението на задачата е нееднозначно. В тези случаи е необходимо да се намери компромисно решение, но за да се стигне до там трябва първо да се намерят корелационните връзки между споменатите критерии. Тази зависимост между критериите може да бъде открита чрез провеждане симулационни изследвания при различни местоположения на контролера. Предложената платформа за оценка на производителността на управляващата равнина представлява набор от инструменти, които са необходими за генериране на топология, за определяне местоположението на контролера в управляващата равнина, изпълнение (емулиране) на генерираната топология и провеждане на изследвания.

2. ПЛАТФОРМА ЗА ИЗСЛЕДВАНЕ ПРОИЗВОДИТЕЛНОСТТА НА СОФТУЕРНО ДЕФИНИРАНИТЕ МРЕЖИ

Първият етап при провеждане на изследване е подготовка на средата, която ще се изследва. Ясно е, че е почти невъзможно тестването на нов продукт (протокол, технология и други) в реална среда. Преди да се започне с внедряването на иновации в реално работещи среди е необходимо те да бъдат изследвани в прототипни среди. Прототипната среда може да бъде хардуерно копие на оригиналната среда, но в контекста на Софтуерно дефинираните мрежи това би било нерентабилно. За да се реализира такъв тим мрежов дизайн е много удобно използването на емуляционни програми. За нуждите на Софтуерно дефинираните мрежи се появява Mininet емулятора. Той е изграден върху Linux базирана среда, която дава възможност една система да бъде разделена на множество малки подсистеми (Linux containers), свързани помежду си чрез виртуални връзки, чиито скорости и време закъснения могат да бъдат програмно дефинирани. Емуляторът може да стартира всякакви мрежови топологии, дори такива дефинирани от потребителя. Последните се генерират програмно, чрез използването на Python скриптове.

За целите на изследванията могат се използват реално действащи мрежови топологии от базата данни *Internet Topology Zoo* [4], която е събрала над 250 топологии на мрежи от цял свят. Форматът, който базата използва за описание на топологиите е неразпознаваем в Mininet средата. Затова е необходимо разглежданата платформа за оценка на производителността на предлага метод

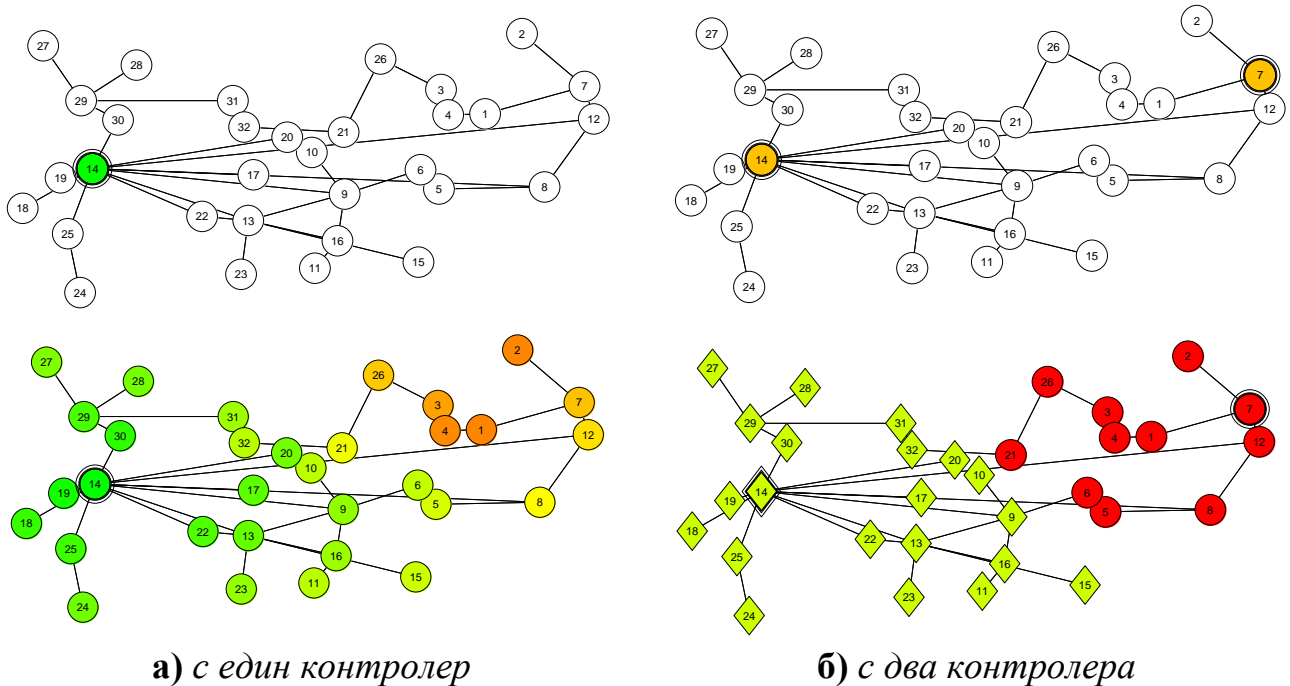
за конвертирането им в Python скриптове. Както се вижда от блоковата диаграма на платформата (фиг.1) това се извършва от конвертор.



Фиг.1 Блокова диаграма на платформа за изследване и оценка производителността на SDN

Интересна особеност на генерираната топология е възможността за задаване параметри на връзките между комутаторите. Тези параметри могат да бъдат скорост, закъснение, големина на опашката и загуби. Закъснението например се изчислява на база разстоянието между комутаторите. Този параметър се генерира от конвертора, който подготвя Python скрипта на желаната топология. Той извлича информация за географските ширина и дължина на два съседни комутатора и така определя разстоянието между тях.

Друг компонент от предложената платформа за изследване на производителността е генераторът на местоположение на контролера. Платформата използва Matlab базиран инструмент, наречен РОСО (*Pareto Optimal COntroller placement*), който предоставя Парето оптимални решения по отношение на различни критерии за ефективност. На фиг. 2 са показани част от възможностите на платформата.

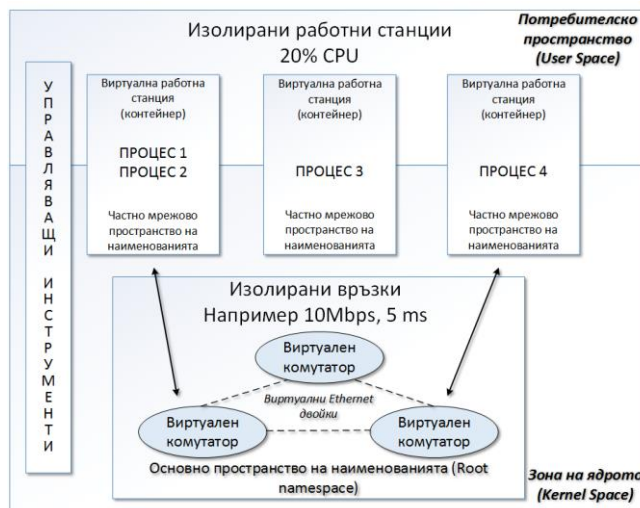


а) с един контролер
б) с два контролера
Фиг.2 Примерни решения за местоположение на контролера от РОСО платформата

Както се вижда от фиг.2 платформата може да генерира местоположение на един или повече контролери, спрямо различни критерии при това може да представя решенията спрямо различните параметри за ефективност с цветен код.

Генерираната мрежова топология се емулира в Mininet среда. Mininet е мрежов емулатор, който позволява създаването на мрежа от виртуални работни станции, комутатори, контролери и връзките между тях. Работните станции в Mininet поддържат стандартен Linux-базиран мрежов софтуер, а комутаторите работят с OpenFlow протокола, който осъществява трафика във вертикална посока между тях и обслужващия ги контролер. Тези комутатори могат да изпълняват много гъвкава потребителска маршрутизация.

Предложеният софтуерен продукт за емулиране мрежови топологии използва олекотени функции за виртуализация в ядрото на Linux операционната система; нарича се още ОС-базирана виртуализация. Това позволява на едно устройство да изпълнява функциите на множество аналогични устройства, чрез споделяне на ресурсите на хардуерната платформа. Виртуализираната система позволява поддръжката на множество виртуални контейнери за една физическа платформа както е показано на фиг.3



Фиг.3 Контейнерна емуляция в Mininet, базирана на различни пространства на наименованията

Контейнерната емуляция в Mininet се основава на различни пространства на наименованията (фиг.3). Всяка работна станция е виртуална потребителска машина, която предоставя свое собствено мрежово пространство на наименованията и участва в мрежата, чрез виртуални връзки предоставени през виртуален комутатор.

Платформата използва виртуален OpenVSwitch комутатор [7], които се изпълняват директно върху ядрото на операционната система. Всъщност той е необходим за да може да се емулира интегрирана контролна равнина. В противен случай, ако се използва стандартен OpenFlow комутатор, който е вграден в Mininet средата, интегрираната контролна равнина може да не функционира коректно. Виртуалните комутатори работят в OpenFlow режим. Това означава, че за да могат да функционират като устройства и да могат коректно да комутират данни, те трябва да получат инструкции от управляващия ги елемент. Този елемент е SDN контролерът. В предложената платформа се използва Floodlight контролер [8], който е Java базиран и може да се инсталира на всеки сървър с общо предназначение. До момента има разработени над 30 контролера, всеки от които би работил добре в разглежданата среда. Някои автори отбелязват, че типа на контролера също влияе върху производителността на SDN. Това е предпоставка за изследване и анализ, които могат да се извършат с разглежданата платформа.

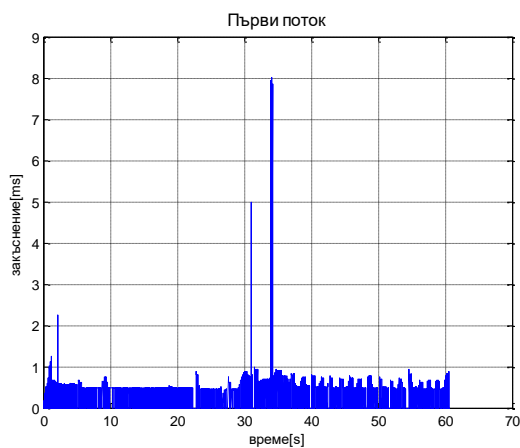
3. АНАЛИЗ НА ПРОИЗВОДИТЕЛНОСТТА НА SDN МРЕЖИ ПОСРЕДСТВОМ ПРЕДЛОЖЕНАТА ПЛАТФОРМА

Предложената платформа за изследване производителността на SDN мрежите използва D-ITG трафик генератор [3]. Той може да генерира IP трафик, който точно отговаря на натовареността на текущите Интернет приложения. Производителността може да бъде оценена по няколко параметъра: пропускателна способност, закъснение, вариация на закъснението

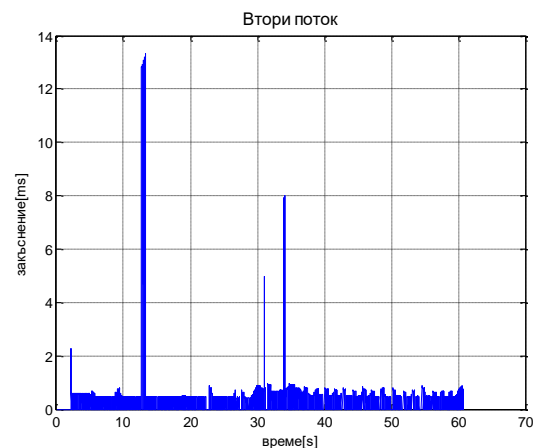
и загуба на пакети по време на предаването. Генераторът се състои от няколко модула, като основните са: *ITGSend* (източник), *ITGRecv* (получател) и *ITGDec* (декодер). Първият модул е отговорен за генерирането на трафик. Той може да генерира един поток или множество потоци. За да се демонстрира работоспособността на платформата е проведен експеримент, при който един източник (в случая комутатор с идентификационен номер 26 от топологията на Еволинк) генерира три потока към един получател с адрес 10.0.0.3, като следва:

```
-a 10.0.0.3 -rp 1003 -c 10000 -t 60000
-a 10.0.0.3 -rp 1004 -c 100000 -t 60000
-a 10.0.0.3 -rp 1005 -c 150000 -t 60000
```

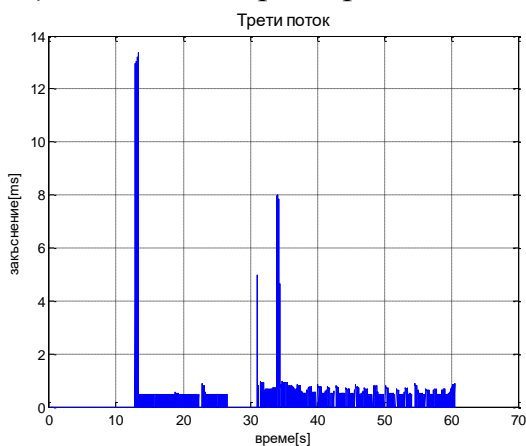
Използват се различни транспортни номера за да се разграничат отделните потоци. Трите потока се генерират с различна скорост на предаване на информация - първия предава 10×10^3 пакета за секунда, втория 100×10^3 пакет/секунда и третият 150×10^3 пакет/секунда. Големината на пакетите също може да бъде дефинирана. По подразбиране за TCP поток един пакет е с големина 512 байта. За да може потребителят да получи адресираната до него информация, той трябва да бъде подготвен за нея. С други думи тук в действие влиза вторият модул на D-ITG генератора, а именно D-ITGRecv.



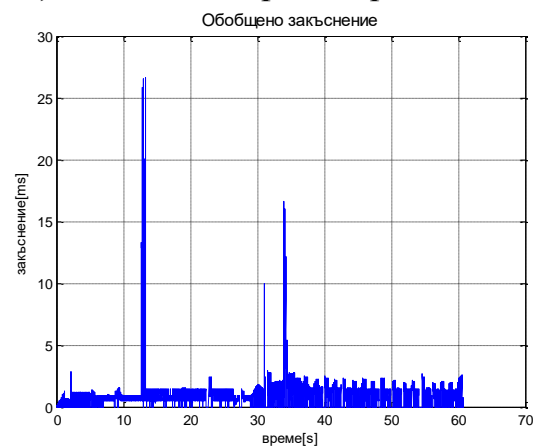
а) закъснение при първия поток



б) закъснение при втория поток



в) закъснение при третия поток

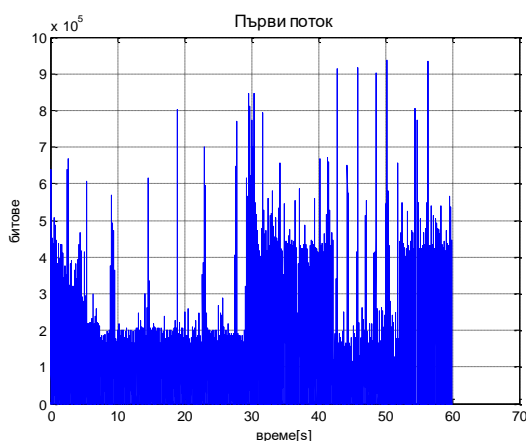


г) обобщено време закъснение

Фиг. 4 Време закъснение при отделните потоци и обобщено време закъснение

По време на симулацията източника и получателя създават файл, в които се съхранява информация за генерираните потоци в мрежата. Тази информация ще бъде полезна при анализа на производителността. От този файл, D-ITG декодера извлича информация за параметрите, по които се оценява производителността. Тези параметри в разгледаният сценарий са следните: време закъснение (фиг.4), пропускателна способност (фиг.5), вариация на закъснението и загуби на пакети.

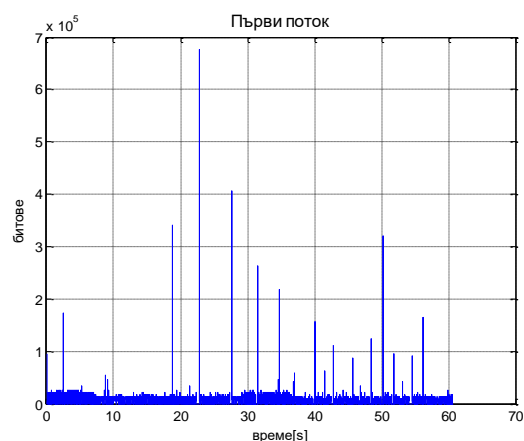
Както се вижда от фиг.4 и при трите потока се забелязват по няколко върхови стойности на време закъснението. При първият поток върховата стойност на време закъснението е най-малка, тя е около 8 милисекунди. Вторият и третият поток имат приблизително равни стойности на пиковото време закъснение - малко по-малко от 14 милисекунди. На фигура 4.г е показано и обобщеното време закъснение в мрежата, което е получено сумарно от време закъсненията на отделните потоци. Средното време закъснение и при трите сценария е по-малко от 4 милисекунди.



```
-----
Flow number: 1
From 10.0.0.28:44706
To 10.0.0.3:1003
-----
```

```
-----
Total time           = 59.960952 s
Total packets        = 598938
Minimum delay        = 0.000000 s
Maximum delay        = 0.000000 s
Average delay        = 0.000000 s
Average jitter       = 0.000000 s
Delay standard deviation = 0.000000 s
Bytes received       = 306656256
Average bitrate      = 40914.127714 Kbit/s
Average packet rate  = 9988.800712 pkt/s
Packets dropped      = 0 (0.00%)
Average loss-burst size = 0.000000 pkt
-----
```

а)



```
-----
Flow number: 1
From 10.0.0.28:44706
To 10.0.0.3:1003
-----
```

```
-----
Total time           = 60.644229 s
Total packets        = 62730
Minimum delay        = 0.053149 s
Maximum delay        = 8.015104 s
Average delay        = 0.539471 s
Average jitter       = 0.003727 s
Delay standard deviation = 0.193957 s
Bytes received       = 32117760
Average bitrate      = 4236.876027 Kbit/s
Average packet rate  = 1034.393561 pkt/s
Packets dropped      = 536169 (89.53%)
Average loss-burst size = 26.118911 pkt
-----
```

б)

фиг. 5 Пропускателна способност на първия поток при а) източника и б) получателя

На фиг. 5 е показана пропускателната способност на първият поток от страна на източника (фиг.5 а) и от страна на получателя (фиг.5 б). При първият поток се генерират 10000 пакета за една секунда. Ако средната големина на един пакет е 512 байта (4096 бита), то скоростта на предаване е приблизително

40 Mb/s. Ясно е, че това е скорост четири пъти по-голяма от капацитета на връзките между комутаторите - по подразбиране Mininet емулятора конфигурира 10 мегабитови връзки между комутаторите. Това е и предпоставка за голямото количество загуби, които могат да се отбележат в приемната страна (фиг.5 б).

Matlab [6] е последният компонент от предложената платформа, който предоставя инструментариума за анализ и оценка на резултатите от емуляцията.

4. ЗАКЛЮЧЕНИЕ

Предложената платформа дава възможност да се оцени производителността на SDN мрежите при различни сценарии. Могат да се провеждат експерименти на топологии при различно местоположение на контролера. По този начин ще се оцени влиянието на местоположението на контролера върху работоспособността на мрежата. От друга страна могат да се провеждат експерименти при различни сценарии на отказ на елемент, а дори и на контролер. Както става ясно и от приложеният пример, платформата за изследване и оценка производителността дава адекватна оценка на параметрите: време закъснение, пропускателна способност, вариация на време закъснението и загуби. Платформата е един добър инструмент за бъдещи изследвания.

БЛАГОДАРНОСТИ:

В настоящия документ са представени резултати, които са получени при изпълнението на дейностите по проект ФНИ 2016-ФЕЕА-04 „Изследване влиянието на местоположението на контролерите в управляващата равнина върху производителността на софтуерно-дефинираните мрежи“, финансиран от Фонд „Научни изследвания“ на РУ „Ангел Кънчев“.

5. ИЗПОЛЗВАНА ЛИТЕРАТУРА

- [1] B. Heller, R. Sherwood, and N. McKeown, “*The controller placement problem*” in Proceedings of the first workshop on Hot topics in software defined networks, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 7–12.
- [2] D. Hock, S. Gebert, M. Hartmann, T. Zinner, P. Tran-Gia, “POCO-Framework for Pareto-Optimal Resilient Controller Placement in SDN-based Core Networks”, *Network Operations and Management Symposium (NOMS), 2014 IEEE, May 5-9, 2014, Krakow, Poland*
- [3] S. Avallone, S. Guadagno, D. Emma, A. Pescapè, and G. Ventre, “*DITG distributed Internet traffic generator*”, in Proceedings of the First International Conference on the Quantitative Evaluation of Systems. QEST 2004., 2004, pp. 316–317.
- [4] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, “*The Internet Topology Zoo*”, *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.
- [5] Y. Hu, W. Wang, X. Gong, X. Que, S. Cheng, “On the placement of controllers in software-defined networks”, *The Journal of China University of Posts and Telecommunications*, vol.18, pp 92-97,171, Oct., 2012
- [6] “Matlab,” May 2013. [Online]. Available: <http://www.mathworks.de/products/matlab>
- [7] “Open vSwitch,” May 2013. [Online]. Available: <http://openvswitch.org>
- [8] “Project floodlight,” May 2013. [Online]. Available: <http://www.projectfloodlight.org/floodlight>