Simulation benchmarking of Riverbed modeler and Omnet++

Quantitative and Qualitative Analysis

Sourabh Acharya, Arun Kumar Rajendra Kumar, Niraant Adhatrrao
School of Information Sciences
University of Pittsburgh
Pittsburgh, USA
SOA25@pitt.edu, ARR105@pitt.edu, NNA13@pitt.edu

Abstract— The simulation tools we use offer different capabilities at hand. There has to be a method by which we can benchmark the tools so as to decide which tool must be used for a particular application in network design and implementation after the process of simulation.

The Simulation IDE is where you create and evaluate your simulations (but you can develop, build and run simulations from the command line as well.) The IDE is based on Eclipse, so you can install your own extensions into it alongside the tools provided by OMNeT++.

Models are written in C++. They make use of the simulation kernel, an efficient and feature-rich C++ runtime library. The IDE has your codebase fully indexed, and offers code completion, go to definition, call hierarchy, find references, and other features that make C++ programming a joy.

Model components are described and assembled in a high-level domain-specific language called NED. NED can be edited both graphically and in source mode. The NED source editor also boasts intelligent navigation and code manipulation features similar to the C++ editor.

Protocol headers and other messages are represented in the code with C++ classes, but the simulation framework spares you the tedium of writing the C++ code manually by providing a domain-specific language that the code can be generated from, alongside with serialization and reflection code.

In OMNeT++, model parameters can be assigned or given default value already in NED, but real parameterization takes place in ini files that also carry configuration options for the simulator. Ini files also let you describe multiple configurations (~experiments) and parameter studies (simulation campaigns) to be carried out on your model. Ini file details can be hidden behind a GUI for non-expert users of your model.

Riverbed Modeler Academic edition 17.56 is used to simulate the given network here. There are several commercial versions in river bed like Steel Head, SteelCentral for Network Performance and Management, SteelCentral for Application Performance Management, SteelCentral for Central Control Management, Steel Fusion, Steel Connect. Riverbed Modeler includes a set of conventions and advances with a modern improvement environment. Modeler gives you a chance to test and exhibit innovation plans before generation; increment organize R&D profitability; create exclusive remote conventions and advancements; and assess improvements to guidelines based conventions. By displaying all system sorts and advances, Riverbed Modeler breaks down systems to analyze the effect of various innovation plans on end-to-end conduct.

Keywords—OMNeT++, NED, INI, Eclipse, C++, Riverbed

I. Introduction

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. "Network" is meant in a broader sense that includes wired and wireless communication networks, on-chip networks, queueing networks, and so on. Domain-specific functionality such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, photonic networks, etc., is provided by model frameworks, developed as independent projects. OMNeT++ offers an Eclipse-based IDE, a graphical runtime environment, and a host of other tools. There are extensions for real-time simulation, network emulation, database integration, SystemC integration, and several other functions. Although OMNeT++ is not a network simulator itself, it has gained widespread popularity as a network simulation platform in the scientific community as well as in industrial settings, and building up a large user community. OMNeT++ provides a component architecture for models. Components *(modules)* are programmed in C++, then assembled into larger components and models using a high-level language *(NED)*. Reusability of models comes for free. OMNeT++ has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into your applications.

Riverbed Modeler provides you with a modeling and simulation environment for designing communication protocols and network equipment. Network technology designers that use Modeler gain a better understanding of design trade-offs earlier in the product development process. This reduces the need for time-intensive and expensive hardware prototyping. Riverbed Modeler Academic Edition is a limited-feature version for educational users who want to utilize simulation software for networking classes. Riverbed Modeler Academic Edition incorporates tools for all phases of a study, including model design, simulation, data collection, and data analysis. Modeler Academic Edition replaces *IT Guru Academic Edition*.

II. SOFTWARE DESIGN OF THE TOOLS

Below are the main components of the OMNeT++ tool:
It consists of a NED file, a source code file written in C++, an INI file which is the initialization file to be run for simulation start. All the header files are included in the C++ general library which is included in the OMNET++ package.

Figure 1: Graphical NED editor

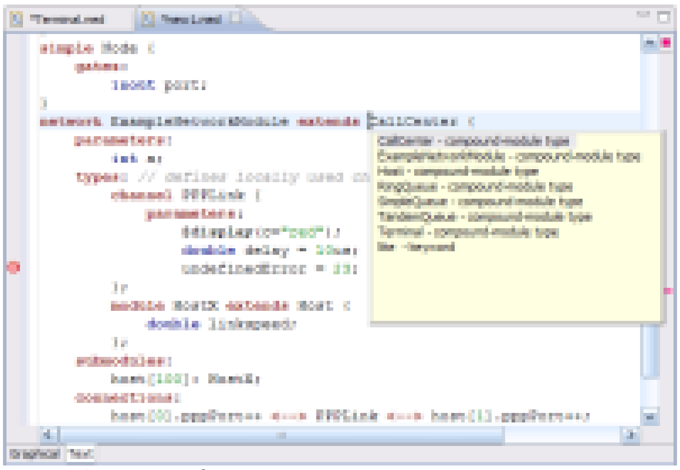

Figure 2: Source NED editor

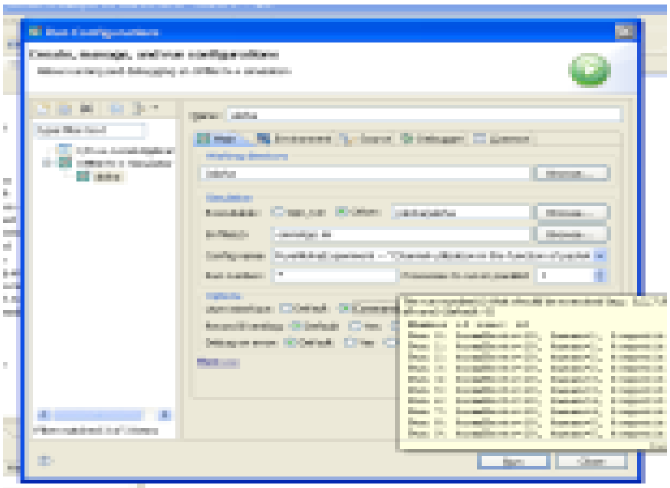

Figure 3: Launching a simulation

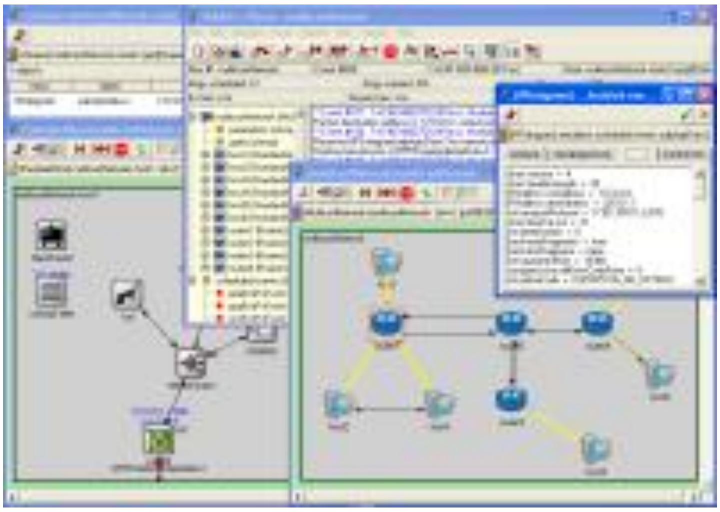

Figure 5: Graphical Runtime Environment

In our project, we implemented a 6-node network with 1 server, 1 switch and 4 clients. Each client sends a packet of 1000 bytes to the server via the switch at different timestamps. The end-to-end delay is measured by taking the difference in timestamps for each client with the server. Figure 6 shows the network architecture of our implementation.

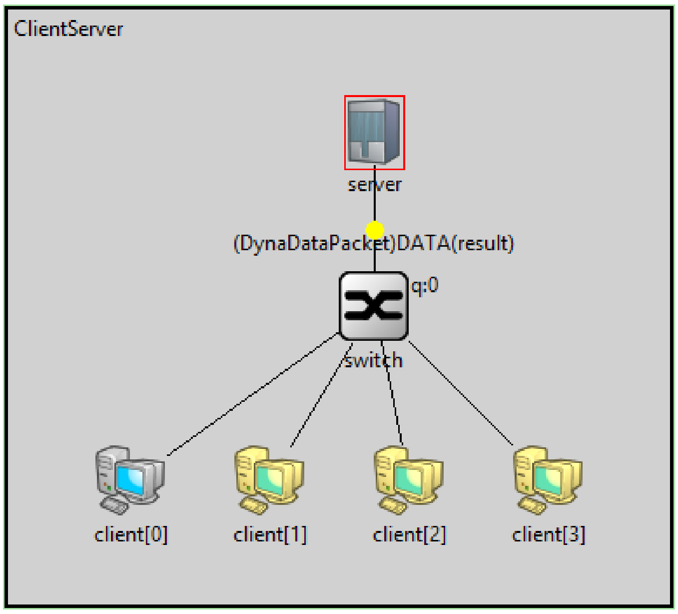

Figure 6: Our implementation of the Client-Server (Dyna) sample network architecture with 1 server, 1 switch and 4 clients.

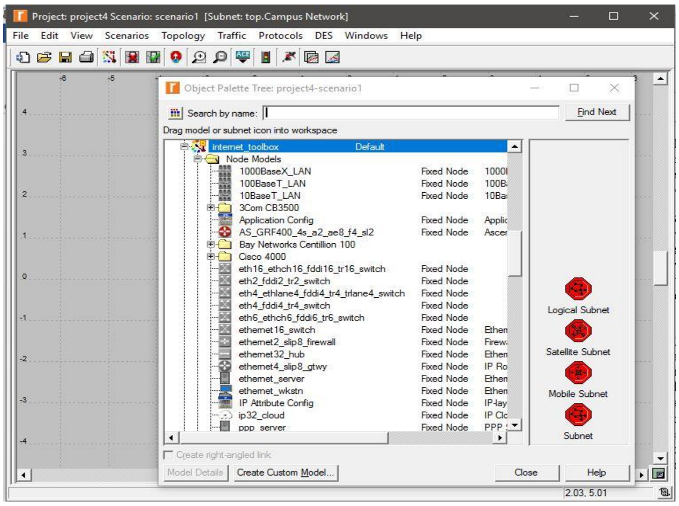

Figure 7. Object Palette Tree

III. RIVERBED MODELER

A. Design Procedure

A six node network is considered with one switch, one server and four client. A link is established between all the nodes as Ethernet/IP. Ethernet/IP is defined as the Ethernet foundation to deal with the association between different computerization gadgets, for example, robots, Programmable logic controller, sensors and other modern machines. It is overseen by the Open Device Net Vendors Association and depends on the Common Industrial Protocol. The step by step procedure is mentioned below,

1) Run the Riverbed Modeler Academic Edition 17.5 as an administrator to save the time spent for validating and verification of database

2) Accept the license agreement that pops out and select File followed by New. Type a name in the new text window and chose a project name, scenario name in the next window

3) Select create empty scenario followed by office or campus in network scale and continue clicking on next till finish button appears.

4) A new window with object palette tree appears as follows

5) Place the required components from the object palette tree. Here we use the following components.

TABLE I. COMPONENTS FROM OBJECT PALETTE

S.No	Components		
	Name	***Quantity***	***Given name***
1)	ethernet_server	1	Ethernet Server
2)	ethernet_wkstn	4	Ethernet Workstation
3)	ethernet16_switch	1	Ethernet Switch
4)	10BaseT	5	Ethernet 10BaseT
5)	Application Config	1	Application Configuration
6)	Profile Config	1	Profile Configuration
7)	Qos Attribute Config	1	IP Layer Attribute Definer

*6)*As per the above specifications the network is implemented and the parameters are yet to be defined.

*7)*Generally various parameters are defined in Application Configuration, Profile Configuration.

8) The FTP packet parameters are considered in this project and the value of 'start time offset' is defined as constant at 5.

*9)*The QOS scheme chosen is FIFO from the Protocol menu

10) The individual DES statistics are chosen as below

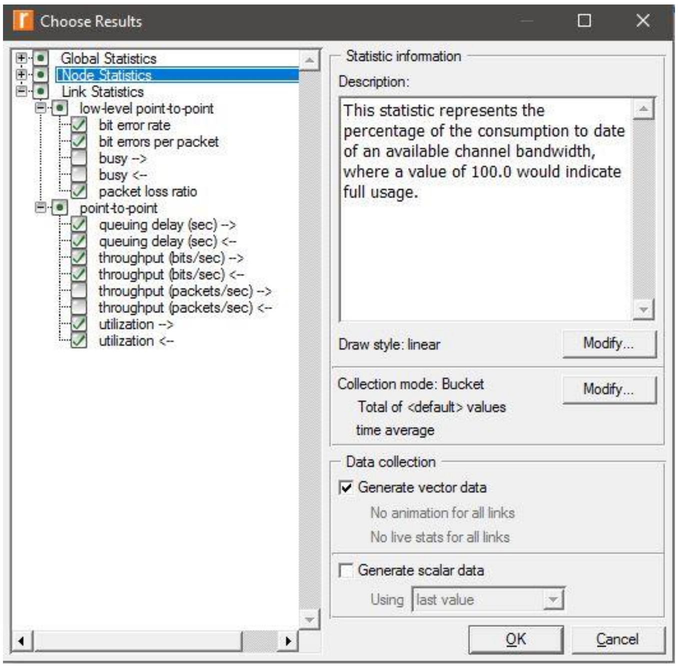

Figure 9. Individual DES characteristics

11) The Configure/Run DES icon is pressed and the duration is chosen as 150 seconds. Then, by clicking Run, simulation is executed. Thereby, the results can be viewed by clicking the DES menu followed by Results and then View Results. If more scenarios are defined, click Compare Results. Moreover, scenarios can be duplicated to run a new simulation without changing the original scenario. It is to be made sure that every time a scenario is made, it must be saved to avoid data loss.

B. Design Implementation

A six node network with one switch, one server and four clients are designed using the components available in Object Palette tree. Attributes are defined for each component. Here, FTP application is simulated. Even Voice and Video application can be simulated.

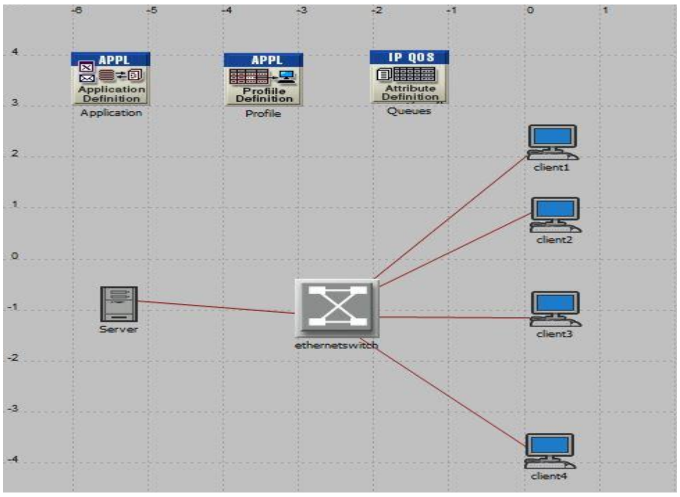

Figure 10. Six node network

The individual DES characteristics can be set in global, node and link statistics.

C. Output

Scenario created is simulated and the total delay, nodal delay, load, traffic received, traffic dropped, traffic sent and end to end delay are found.

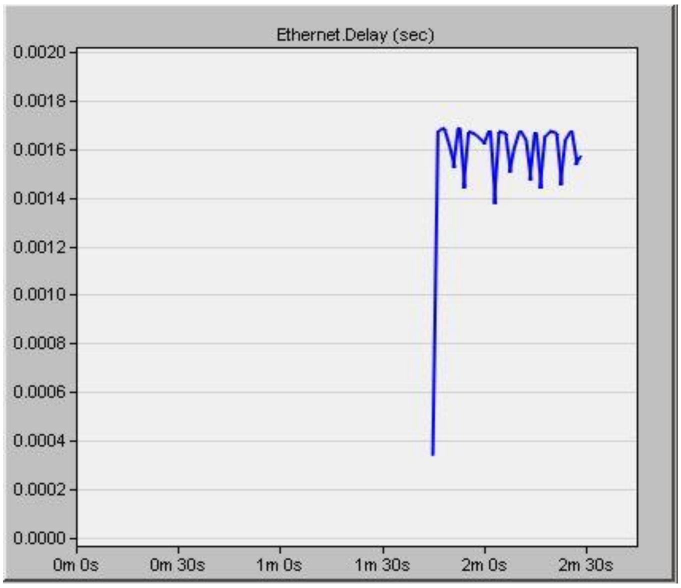

Figure 11. Total Ethernet delay

The end to end delay, throughput can be obtained as a single graph or cascaded to form multiple plots. Riverbed allows the user to graphically analyze various parameters providing a better understanding of the network.

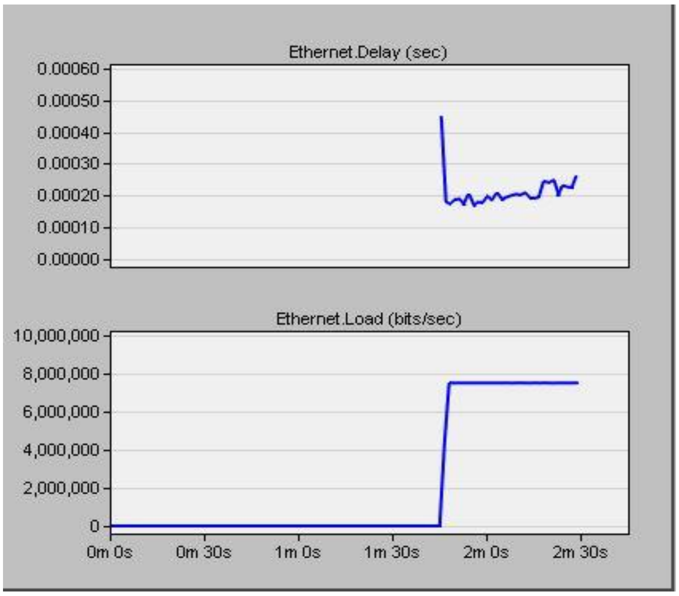

Figure 12. Load and delay in Server

There are four clients in total and here only one client is going to be graphically represented. Hence, we are considering Client 1 and its delay, traffic received and load are obtained as a parallel plot

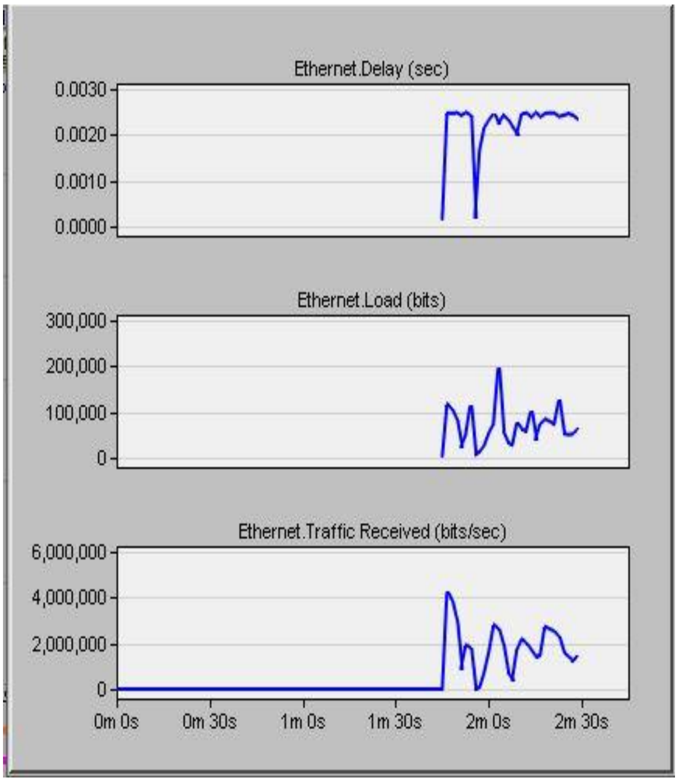

Figure 13. Total Ethernet delay

Throughput is generally known as the rate at which packets are delivered successfully over a communication channel. Throughput can be expressed in bits per second.

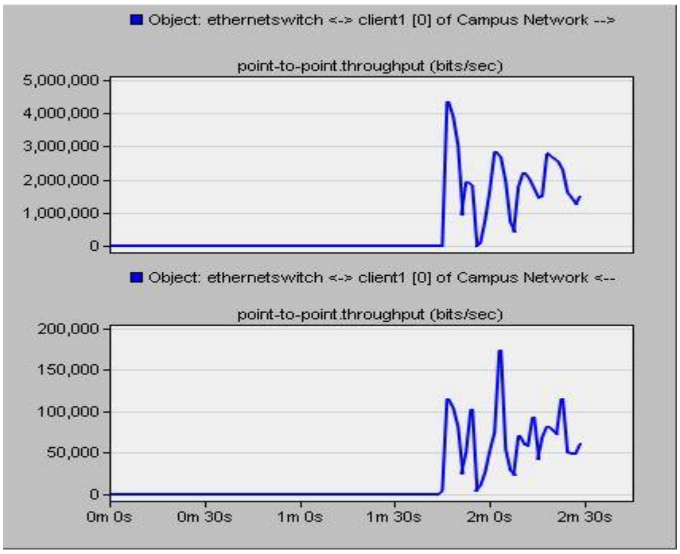

Figure 14. Load and delay in Server

II. QUANTITATIVE ANALYSIS

```
#4    7.958745045657   client[0] --> switch    DYNA_CONN_REQ   id=3   kind=0 length=0 bytes
#6    8.30207837899    switch --> server       DYNA_CONN_REQ   id=3   kind=0 length=0 bytes
#9    8.31207837899    server --> switch       DYNA_CONN_ACK   id=3   kind=1 length=0 bytes
#11   8.655411712323   switch --> client[0]    DYNA_CONN_ACK   id=3   kind=1 length=0 bytes
#12   8.665411712323   client[0] --> switch    DATA(query)     id=10  kind=4 length=0 bytes
#14   8.985603884839   client[1] --> switch    DYNA_CONN_REQ   id=12  kind=0 length=0 bytes
#16   9.008745045656   switch --> server       DATA(query)     id=10  kind=4 length=0 bytes
#19   9.218745045656   server --> switch       DATA(result)    id=10  kind=4 length=0 bytes
```

Figure 15: Hops in network simulated for zero bytes in Omnet++4.6

```
Event#  Time              Src/Dest               Name            Info
#145    21.027140588518   switch --> server      DYNA_DISC_REQ   id=106  kind=2 len
#147    21.037140588518   server --> switch      DYNA_DISC_ACK   id=106  kind=3 len
#149    21.213807255184   switch --> client[3]   DYNA_DISC_ACK   id=106  kind=3 length=1000 bytes
```

Figure 16: Length of bytes changed to 1000 bytes in Omnet++4.6

$$
\begin{aligned}
\text{End-to-end delay} &= \text{Difference in timestamps} \\
&= \text{Timestamp2} - \text{Timestamp1} \\
&= 21.2138 - 21.0371 \\
&= 0.1767 \text{ sec} = 176.7 \text{ msec}
\end{aligned}
$$

In Figure 15 and Figure 16, we can see the event log of each hop taking place in the network. The user is able to enter the number of bytes in the packet to be transmitted over the Client-Server network which was shown initially in Figure 6.

III. QUALITATIVE ANALYSIS

Popularity : We can compare the popularity levels of both the tools. Omnet++ is popular for small enterprises as it is an open-source tool and can be used at entry level of any networking project.

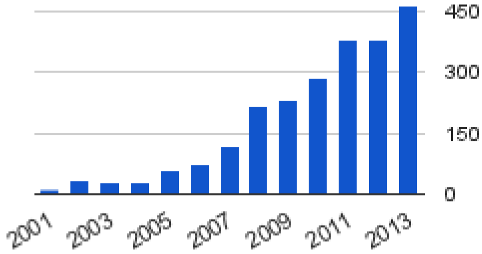

Figure 17: Number of Omnet++ publications 2001-2013 [Google scholar search results]

Figure 18: Popularity levels of Riverbed in IT industry. [5]

Ease of installation: the riverbed modeler can be installed on any device using single installer file available after registering on the riverbed website. This reduces any potential errors which may occur as in the case of OMNET++. OMNET++ is installed using a windows command prompt which requires the user to input a certain sequence of commands and any errors would mean that the whole software needs to be re-installed.

Interface comparison: Riverbed offers users with an inbuilt GUI unlike OMNET++ where a separate add-on called INET needs to be installed which provides GUI capabilities.

Proprietary components: Riverbed offers users with proprietary components from manufacturers like Cisco, Juniper etc. OMNET++ offers users with just generic models which can then be programmed to operate as certain industry components.

Licensing: Riverbed requires users to explicitly register for the academic version after which a rolling 6-month license is issued to each user. This license restricts the installation to academic version capabilities. OMNET++ is a freeware and has a simplified open ware license and does not require any registration.

Updates and Available versions:
Riverbed's academic version was last updated in 2014 and no subsequent changes were made to the package. OMNET++ has version 5.0 which is the latest version available. Although version 5.0 came out in 2016 it has a lot of bugs especially during the installation phase making it quite unstable compared to its predecessor OMNET++ v.4.3.

Module and library constraints**:** The academic version from riverbed limits the use of certain library components or simply doesn't provide them this especially seen for wireless modules and for certain technologies like MPLS. OMnet++ does not put any restrictions on the number of components that can be used, additionally a number of components can be added by installing add-ons.

Documents and tutorial content: Riverbed offers a large amount documentation along with some basic tutorials regarding certain applications.
OMNET++ provides documentation for the main package and each add-on carries its own documentation although the website itself does not provide any tutorials apart from the getting started ones there are other sources from which information on implementation of topologies and technologies can be gathered.
This information is not necessarily valid and depends on the users knowledge of the technology in question.

Project migration: All project files in the academic version are tainted which prevents the user from transferring the project to a fully licensed one which effectively forces the user to recreate all the work in the fully licensed version .
OMNET++ does not impose any such constraints and provides backward compatibility to previous versions though some add-ons versions can only be used with certain versions of OMNET++.

Support: Riverbed provides community support in the form of riverbed splash as well as dedicated technical support. On the other hand OMNET++ does not have any dedicated support and any queries can be hoped to be resolved on peer forums.

IV. CONCLUSIONS

Quantitively, we cannot firmly infer that the end-to-end delay values are different for the same packet size. It is not possible for us to compare the corresponding latency values in this project, as the components used in OMNET++ are generalized, and the Riverbed components are not designed using those same parameters. There is no concrete method by which we are able to measure the byte length of the packet as we are not using a source generator for the implemented client-server network.
Qualitatively, riverbed and OMNET++ both have some flaws and some positives. Riverbed offers a better range of proprietary devices from vendors like Cisco, Juniper etc. But the academic version limits the availability of these components. OMNET++ may offer no proprietary components but has no limitations on the ones that can be used. The academic version of riverbed doesn't support newer LTE modules while an add-on (simuLTE) can be used to simulate LTE components and networks on OMNET++.

ACKNOWLEDGMENT

We would like to thank Dr. David Tipper for proposing the topic of simulation benchmarking in his list of project topics and also for guiding us on Omnet++ tutorials and the INET framework. We also would like to convey our gratitude to the staff of School of Information Sciences, University of Pittsburgh for allowing us to use the networking laboratory.

REFERENCES

[1] Andras Varga https://omnetpp.org ©2001-2015 OpenSim Ltd.

[2] Evizal Abdul Kadir, Apri Siswanto, Abdul Syukur, "Performance analysis of wireless LAN 802.11n standard for e-Learning", 2016.

[3] Narottam Das, Wu Ma, Syed Islam, "Analysis of end-to-end delay characteristics for various packets in IEC 61850 substation communications system", 2015.

[4] Fatin Hamadah, M.A. Rahman, Thein Wan Au, "Impact of IPSec on MANET," 2016.

[5] https://www.riverbed.com/products/steelhead/steelhead-cx.html
©2016, Riverbed Technology