Comparision between two TCP agents TCP Vega and TCP NewReno in a SANET

A. Anas
2^{nd} *year Software Engeneering Student*
ENSIAS
Rabat, Morocco
anas.abouali@ieee.org

***Abstract*—This paper presents sequence of various critical steps of installation and simulation for an experiment aimed to compaire two major TCP agents, TCP Vega and TCP NewReno using Network Simulator2 (NS2). This experiment highlight the efficiency of both TCP agents in a Static ad hoc network by compairing there packet loss ratio by sending FTP packets.**

***Index Terms*—NS2, Tracegraph, Access Point, Base Station, Packet Loss.**

I. Introduction

The Wireless Local Area Network (WLAN) is a network enabling ad-hoc wireless networks is an infrastructure-free environment, generally called ANETs (Ad hoc NETWORKs), consists of a large relatively dense population of wireless units whose only means of communication is the use of wireless interfaces without the aid of a pre-existing infrastructure or centralized administration. A characteristic The special feature of the ad hoc network is the absence of any fixed installation. This allows him to be quick and easy to deploy. For this, the ad hoc network is used for tactical applications such as rescue, military or explorations. This document is a comparison between the two diffrent TCP agent TCP Vega and TCP NewReno in a Static ad hoc network, by simulating a two source, two destination, 3 gatways, in a FTP packet sending senario.

II. NS2 Simulation tool

A. NS2 Installation

This installation is done on Linux Ubuntu 10.04 and the network simulator used here is NS-2.29 i.e version 2.29. This paper includes the specific installation steps required to Simulate FTP application in NS 2.29 platform. It also includes installation process required for GnuPlot utility. This utility is further used to plot the data extracted from the simulation.
Critical NS2 installation steps on Ubuntu 10.04 are given below:
1. Download the ns-allinone-2.29.tar.gz
2. Place it in /home/simulator etc and extract it $ cd /home/simulator $ tar -xvf ns-allinone-2.29.tar.gz
3. Download install some packages from repository $ Sudo apt-get install build-essential autoconf automake libxmu-dev
4. Install the ns-2.29 $cd ns-allinone-2.29 $./install
5. You might face problem with the installation of otcl-1.13, the problem can be due to the gcc- 4.4.1 / g++-4.4.1 compilers.
Do try this: $ Sudo apt-get install g++-4.3 $CC=gcc-4.3 CXX=g++-4.3. /install
6. Edit some paths in bashrc file $gedit /.bashrc
7. Put the following lines in bashrc file.
You might change /home/simulator as it depends on where do you extract ns-allinone-2.29.tar.gz.

```
#LD_LIBRARY_PATH
OTCL_LIB=/home/simulator/ns-allinone-2.29/otcl-1.11
NS2_LIB=/home/simulator/ns-allinone-2.29/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib TCL_LIBRARY
TCL_LIB=/home/simulator/ns-allinone-2.29/tcl8.4.14/lirary
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
#PATH
XGRAPH=/home/simulator/ns-allinone-
2.29/bin:/home/simulator/ns-allinone-
2.34/tcl8.4.18/unix:/home/simulator/ns-allinone-
2.29/tk8.4.14/unix
NS=/home/simulator/ns-allinone-2.29/ns-2.29/
NAM=/home/simulator/ns-allinone-2.29/nam-1.11/
export PATH=$PATH:$XGRAPH:$NS:$NAM
```

8. Validate it (it will take longer time than installation)

```
$ cd ns-2.29
$./validate
```

9. Try to run it by typing $ ns Then % symbol appearing on the screen indicates successful installation of ns-2.29.

B. Simulation

The simulation is done using ns2 tool and the nam tool to visualise the 6 cells architecture, as it's shown in the figure below:

1) NewReno: Here we've simulateda the same architecture with both sources using the TCP New Reno agent. here is the following resolut as it's shown in the figure 2. The Y axe represent the packet lost percent, and the X axe represent the time of simulation. As it's shown in the graph there's a big jump of the packet lost in the small intervall after that it settles down at around 34%.

2) Vega: Here we've simulateda the same architecture with both sources using the TCP Vega agent. here is the following

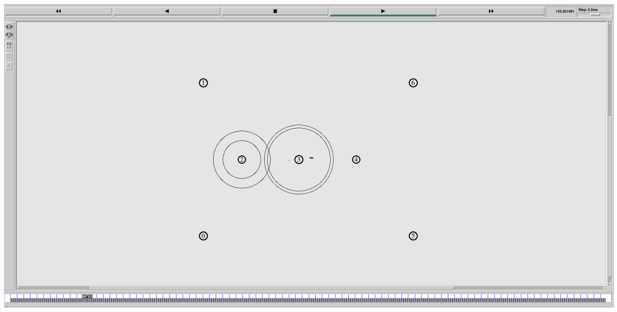

Fig. 1. 2 sources, 3 gateways, 2 destinations.

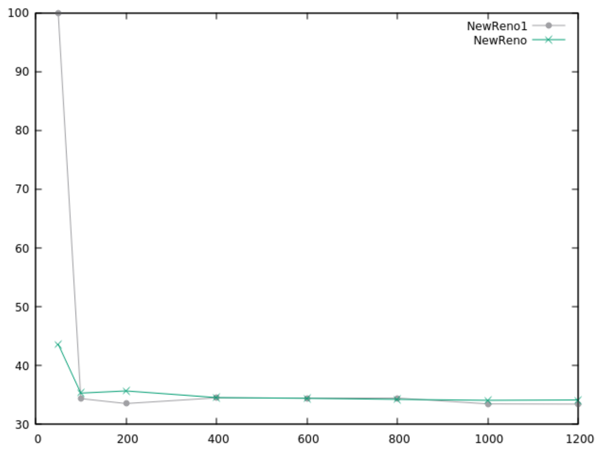

Fig. 2. TCP New Reno

resolut as it's shown in the figure 3. The Y axe represent the packet lost percent, and the X axe represent the time of simulation. As it's shown in the graph as well there's a big jump of the packet lost in the small intervall after that it settles down at around 33%.

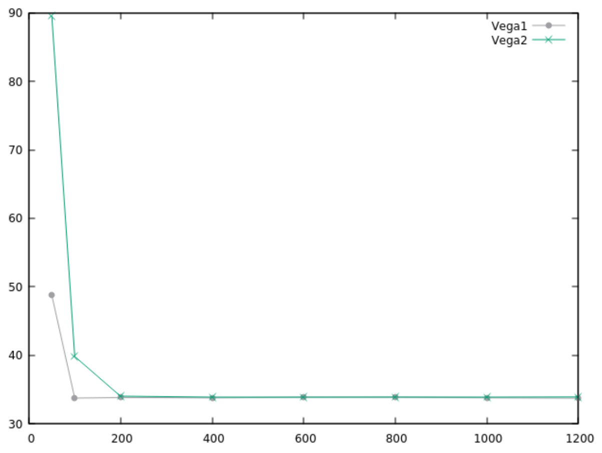

Fig. 3. TCP Vega

3) New Reno vs Vega: Here we've simulateda the same architecture with one of the sources using the TCP Vega agent, and the other TCP New Reno agent. here is the following resolut as it's shown in the figure 4. The Y axe represent the packet lost percent, and the X axe represent the time of simulation. As it's shown in the graph as well there's a small variation between the two agent that is approximately less than 1% , with the TCP Vega having the highest packet lost rate.

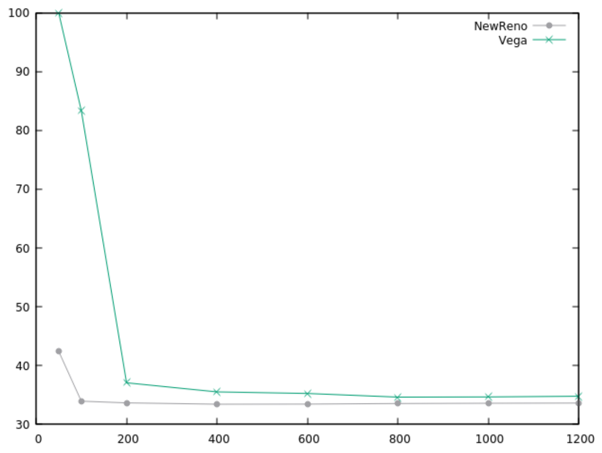

Fig. 4. TCP Vega vs New Reno

C. Conclusion

As it's shown in this simulation, there is only a small variation between the two TCP agents, and the TCP New Reno being slightly better with a smaller packet lost rate.

REFERENCES

[1] https://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/ns-allinone-2.35.tar.gz/download
[2] https://www.ieee.org org Conference-template-A4