Trace Monitoring in NS-2

Deepak Kushwaha
Dept. of Computer Science
National Institute of Technology
Karnataka, Surathkal-575025
dip.kush2008@gmail.com

Vishal Kumar
Dept. of Computer Science
National Institute of Technology
Karnataka, Surathkal-575025
vishalimp304@gmail.com

***Abstract*—Network Simulator-2 is a open Source discrete event simulator to simulate Internet Protocol (IP) Networks and Topologies. NS development was supported by DARPA (Defense Advanced Research Projects Agency) through the VINT (Virtual Internetwork Testbed) project at LBL, Xerox PARC, UCB, and USC/ISI** [1]**. NS-2 is based on two languages: an object based simulator, written in C++, and an object oriented version of Tcl (Tool Command Language) which is OTcl. Tcl is very hefty and dynamic language. The ns-2 spans a very huge number of applications of varied kind of protocols of different network types consisting of many network elements and traffic prototype. The Network Topology and various attributes are defined in the TCL script. After simulation of the script Trace module of NS-2 captures the various outcomes of the network. The contents of the trace file are not very lucid. Ones need some comprehend the terms before examining the Trace File.**

So to facilitate the process of capturing useful information from these trace files we are proposing Trace Analyzer. Visual Trace Analysers [2]**. It is a software tool which will extract the data and present in a format which could be directly used for performance study. Trace Layers consists of three layers, first interface which takes the trace file as input using interactive GUI. Second which preprocess the file and extracts the data and store them in particular data structures and third layer is presenting the extracted data in well groomed manner so that it would make sense to the developer and subsidizes the time required to further study.**

Index Terms*—*NS-2,OTcl, NAM

I. INTRODUCTION

In network research Area and ever incrementing technological advancements led to augmented and complex network topologies. It is very difficult to deploy and test these systems that contain multiple network computers, data links and routers to certify a particular network protocol or any specific network algorithm. The Network Simulator saves a lot of resources in achieving these tasks. The Network Simulators allow network engineers to test New Algorithm and Networks in a recurring fashion. The demand was to recreate a thing like VHDL and Logisim (widely used in microprocessor chip design) [3] that would address the scale and heterogeneity of the Internet protocols. NS-2 is a multi-protocol simulator that supports more extensive study of network protocols. NS-2 provides various protocols TCP, UDP, HTTP and myriad of wireless routing protocols like AODV, DSR etc. It also allows various traffic models like FTP, CBR etc. and various other visualization tools.

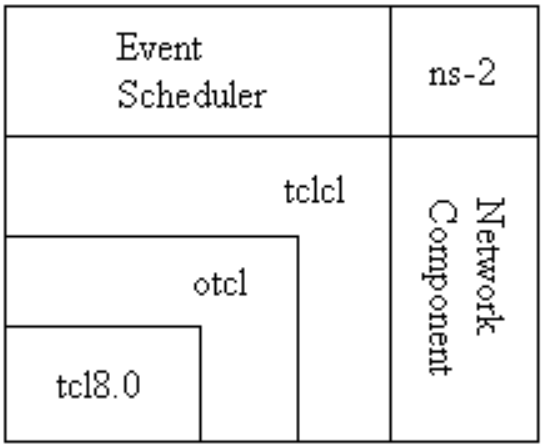

Figure 1: Architecture of NS-2

The Network Simulator consists of following components
As shown in the Figure 1:-

- Network Components
- Event Scheduler.
- OTcl object oriented version of Tcl
- Tcl - a script language

Once the network simulation Scripts are compiled it would produce the trace file which contains all statistical data about which could be used to compute how many packets dropped, received, delay in the network ,bandwidth and other QoS parameters. This work done will ease the effort of analyizing the performance of simulation done in NS-2. Section 2(turn in into roman symbol) give give idea about the working of NS-2. Section III will give a light trace and monitoring support provided by NS-2. Section IV describes about the NS-2 trace Files produced by the simulation of wired nodes. Section V gives a brief idea about the working of the NS-2 Trace Analyzer we developed. Section VI contains the related work in the fields of Trace Analyzers and future work we intend to do in the tool we developed.

II. WORKING OF NS-2

Network Simulator comes with a large number of other modules which make NS-2 most preferable software tool to do

analysis in networks. NS-2 without these packages will become deficient in delivering the right quality of service. The most popular animation tool used for visualization is NAM (Network Animation Stimulator). It produces network animation NAM trace file for network animation purposes. The other file which is generated is trace file which can be used which can be coded in different ways to use graph plotting software like GNU Plot or xgraph as per the convenience. The whole process of simulation in NS-2 and generation of Trace File and Nam file is explained in Figure 2.

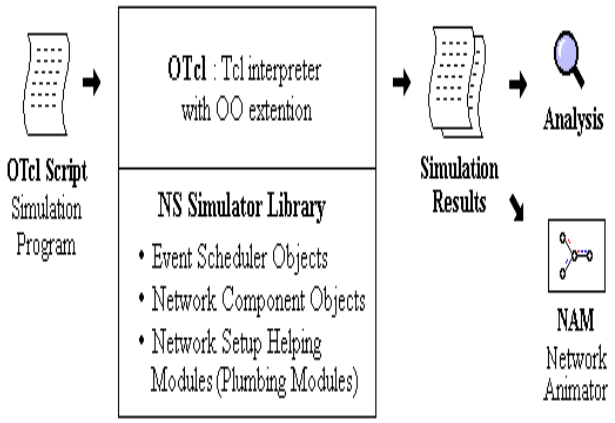

Figure 2: Simulation process of NS-2

III. Trace and Monitoring Support in NS2

There are two different ways of monitoring abilities supported by simulator. The first one is traces which record information about each discrete packet arrives, leaves, dropped at a link or queue. Traced objects are configured as nodes. The second type of objects is called monitors which keeps count of quantities like packet and byte arrivals, departures etc. [4]. Packet.h needs to be included in header to support traces.

Various functions:

- flush-trace {}: The buffers of all trace objects in simulation are flushed.
- create-trace {type file src dst}: Creates a trace object of given type in the given src and dest nodes.
- trace-queue { s1 s2 file }: Trace a specific queue between the given nodes s1 and s2
- trace-callback{ ns cmd }: Arranges to call command when a line is to be traced. The procedure treats cmd as a string and traces it for every line.
- monitor-queue { s1 s2 } This function calls the init-monitor function on the link between nodes n1 and n2.

IV. NS-2 Trace Files for Wired Nodes

The Trace data is in ASCII code and contains 12 fields [9]. All Fields with example trace file is shown in Figure 3.

1. Type Identifier
   a. '+' packets get enqueued
   b. '-' packets get dequeued
   c. 'r' packets received
   d. 'd' packets dropped
   e. 'c' packets collision
2. Timing of the event
3. Source ID
4. Destination ID
5. Packet Type
6. Packets Size
7. 7 digit flags (- disable)
8. Flow ID
9. Source Address
10. Destination Address (format of above two address are "IP.p", where "IP" is the address and "p" is the port)
11. Sequence Number
12. Packet Unique ID

event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

```
r : receive  (at to_node)                    src_addr : node.port (3.0)
+ : enqueue  (at queue)                      dst_addr : node.port (0.0)
- : dequeue  (at queue)
d : drop     (at queue)
```

```
r 1.3556 3 2 ack 40 ------- 1 3.0 0.0 15 201
+ 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
- 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
r 1.35576 0 2 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
d 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
- 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
```

V. Development of NS-2 Trace Analyzer

NS-2 trace examiner is developed using fully platform independent language JAVA which is open source. This tool can be used in any machine with JVM installed. This makes increases the tool usability.

A. Architecture

Conceptually the architecture is such that it takes the input trace file generated from the tcl script as input. The tool's implementation is divided into three stages:

1) Obtain the data from the trace file and store them in ArrayList Data structures. This is called extracting phase.

2) After the extracting phase there is preprocessing phase in which quantities like delay and count are measured.

3) After the preprocessing is done input is taken from the user to do some apply some filter technique to retrieve specific data for presentation.

We will elaborate this phase by explaining the various features of the application:-Report Generation – The report consists of various important information like size of file, Number of Lines, Start Time, End Time and Types of packets. A further report is taken which is per-flow monitoring which gives

detailed results on two nodes like jitter, bandwidth and throughput.

B. Performance Parameters

Performance and Behavior of a network depends on QoS attributes. QoS is an idea that transmission rates, error rates and other major quantities can be computed and enhanced. The QoS parameters on which best service are guaranteed and it is used in the simulation are:

1) Average Jitter: Jitter is the Deviation in the time between packets arriving, caused by network congestion. A packet delay varies with the position of router in queues [5] along the path from source to destination. Jitter gravely affects the video/audio streaming quality as these services require Real Time Communication (RTC). For calculating the average jitter we have used a formula:

$$J_i = D_{i+1} - D_i$$

where J_i is Jitter of i^{th} packet and D_{i+1} and D_i are delays of $(i+1)^{th}$ and i^{th} packet respectively. Then average is taken.

2) Average End-to-End Delay: Due to different queuing and routing paths in packet switch network a data packet may take varying time. End-to-end delay of each packet summed up and average is computed.

End-to-end delay D = T(+) – T(r)

Where T(+) time at which packet is queued and T(r) is time at which it is received.

3) Throughput: Throughput is how fast we can actually send data in a network. Number of packets send per unit time is not equal to number of packets received per unit time because of delay and packet loss so throughput of a network is:

$$T = P(r) \div P(s)$$

Where P(r) is no. of packets of received per/sec and P(s) is no. of packets sent per/sec.

4) Packet Loss: Packet Loss = No. of Packets sent – No. of Packets Received.

C. Interface and Layout.

The interface of NS2 Trace Analyser is developed using JAVA. It has a interactive interface manages to generate report, per flow information, filtering data according to a keyword. Figure 5 shows the Launch Window which has a File Chooser button to choose file and load. The read button reads the lines from the files and stores the information which is like preprocessing and displays some basic information about the file like Number of line and size. Per Flow Traffic window (Figure 6) gives the details related to particular source and destination. The filter window is used to filter the lines according to particular keyword provided by the user. It is displayed in Figure 7

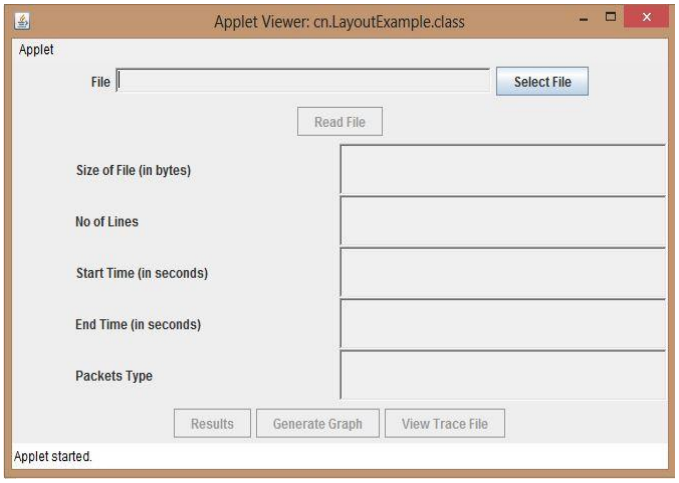

Figure 4: Launch Screen for taking file

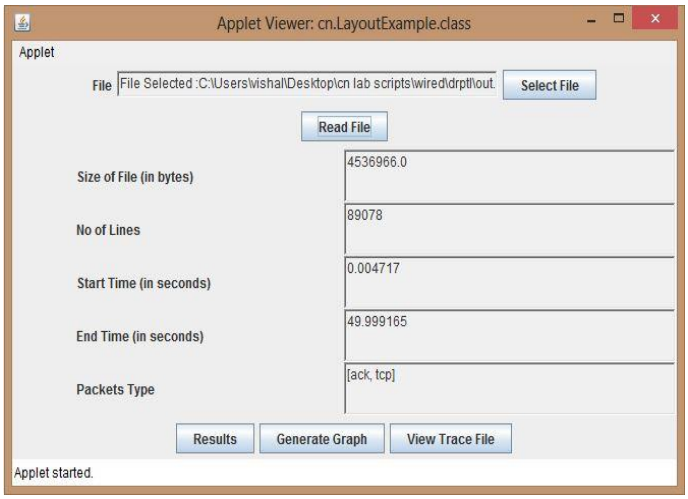

Figure 5: Read Contents of File

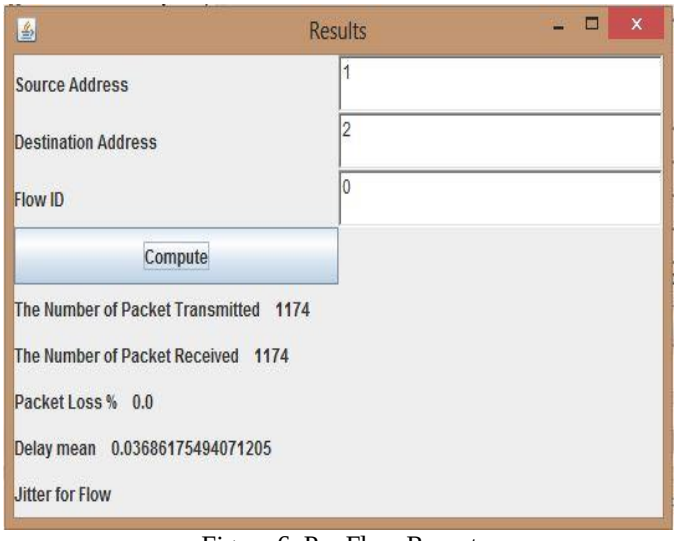

Figure 6: Per Flow Report

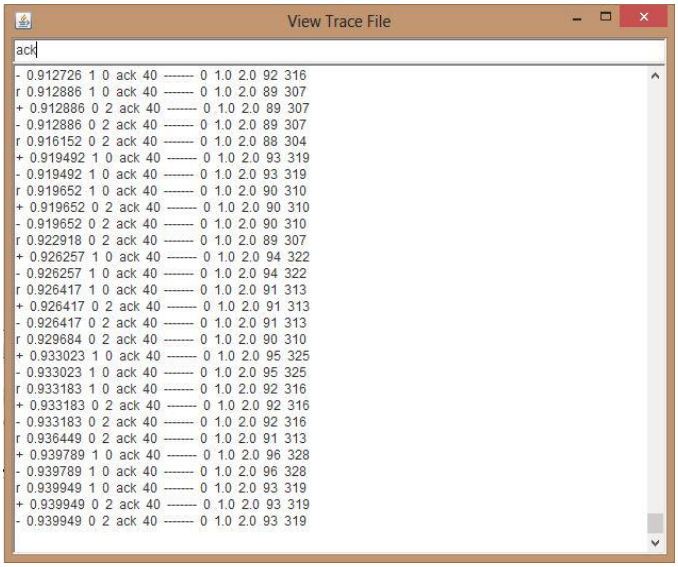

Figure 7: Filter Window

VI. RELATED WORK

NS-2 Visual Trace Analyzer: This document presents the NS2 Visual Trace Analyzer[7], an application capable of analyzing NS-2 normal wired and wireless trace files along with some other formats. This tool is capable of analyzing the trace files and making graphics and statistics about the behavior of the flows in simulation. Simple, standalone, user-friendly, very fast, intuitive, this application will certainly help to get results.

JTrana: This Tool [8] was developed by Hengheng Qian (M.S., now working in CICC China) and Weiwei Fang (Ph.D., now working in BJTU China). jTrana can visualize packet statistics, energy information of the whole network. This can even calculate per flow information like throughput, delay, jitter etc.

This is the first version of this software we developed. We would like to continue our work by adding better User Interface and extending this software to work on wireless trace files. We also want to add feature of producing graphs for various performance parameters like jitter per packet, end-to-end delay per packet.

VII. CONCLUSION

This tool was developed with a purpose by making network performance analysis easy. With the help of few clicks you can have every information associated with a network. This would help to work faster and one could give more attention towards developing better algorithms avoid worrying about analysis report.

REFERENCES

[1] Lloys Armstrong Jr. "Virtual Inter Network TestBed (VINT) methods and system"

[2] Fernando Rocha "NS2 Visual Trace Analyzer Manual"

[3] VINT Project OverView http://www.isi.edu/nsnam/vint/project_overview.html

[4] Notes and Documentation http://www.isi.edu/nsnam/ns/doc/

[5] Amandeep Singh Thethi,R.S. Sawhney "Performance Evaluation of QoS parameters in UMTS Network Using Qualnet"

[6] Michael Karl "A Comparison of the architecture of network simulators NS-2 and TOSSIMM". Young, The Technical Writer's Handbook. Mill Valley, CA:University Science,1989.

[7] Aliff Umair Salleh, Zulkifli Ishak , Norashidah Md. Din, Md Zaini Jamaludin "Trace Analyzer for NS-2"

[8] ns2trans https://sites.google.com/site/ns2trana/

[9] NS by Example http://nile.wpi.edu/NS