



A Novel Tool for Automating the Trace File Analysis in Vehicular Ad Hoc Networks

Sanaa Achchab, Souad El Houssaini, Amal Tmiri, Souad Ajjaj
and Mohammed-Alamine El Houssaini

EasyChair preprints are intended for rapid
dissemination of research results and are
integrated with the rest of EasyChair.

November 4, 2020

A novel tool for automating the trace file analysis in vehicular ad hoc networks

Sanaa Achchab^{1*}, Souad El Houssaini¹, Amal Tmiri¹, Souad Ajjaj²,

Mohammed-Alamine El Houssaini³

¹ Department of Computer Science, Faculty of Sciences, Chouaib Doukkali University,
El Jadida, Morocco

Sanaa Achchab sanaachab204@gmail.com

Souad El Houssaini souad_elhoussaini@yahoo.fr

Amal Tmiri b_tmiri@yahoo.fr

² ENSAM, Hassan II University, Casablanca, Morocco

Souad Ajjaj ajjaj.souad@gmail.com

³ ESEF, Chouaib Doukkali University, El Jadida, Morocco

Mohammed-Alamine El Houssaini elhoussaini.m@ucd.ac.ma

Abstract

vehicular ad hoc networks (VANETs) are an emerging area of research that has gained considerable attention in recent years due to their role in the design of intelligent transportation systems. NS-3 (Network Simulator 3) is one of the most widely used discrete event simulators for the simulation of VANET networks. Trace files, which capture events occurring in these networks, can be used to study performance. This article presents a new tool to automate the analysis of trace files in VANETs networks using the concept of a multi-agents system, this tool calculates performance parameters, such as throughput, packet delivery rate, loss of packets, packets sent and packets received. In addition, it facilitates and speeds up the entire task of analyzing a large number of network simulations while offering the possibility of storing each trace file as well as each measurement produced.

Keywords: VANET, trace file, multi-agents system, NS-3

1 Introduction

The vehicular networks also called VANETS (vehicular ad hoc network) constitute a subclass of MANETS and represent one of the most promising components of Intelligent Transport Systems (STI). These systems aim to integrate new information and

communication technologies in the field of transport in order to improve the safety and comfort of road users. A VANET is an ad-hoc type network. It can opportunistically use communications with infrastructures, thus allowing access to other networks and also to the Internet. The various infrastructures used within the framework of the VANETs are called "Roadside Units" (RSU). These units can be traffic lights, parking meters, or any other terminal placed at the edge of the road.

Network simulation [1][2] in telecommunications and computer networks is a technique where the behavior of a real network is modeled using software.

Network simulation tools [1][2] are used to model real networks with great ease. You can find many network simulators (open source or commercial), which offer different functionalities [3], such as NS-3 [4], NS-2 [5], NetSim, OMNeT, OPNET, QualNet, J-Sim [6] and REAL.

A very popular network simulation tool is NS-3. The main objective of this NS-3 project is to give network researchers an open simulation environment to test their network protocols. Trace files (NS-3 Trace Formats) are considered one of the most valuable information that a network simulator like NS-3 has to offer, they describe all the events that have taken place during a simulation and on the Based on these events, users retrieve information about the behavior of the network. There are different types of files, such as trace file .tr and file. Pcap.

In this article, we have developed a new tool for automating the analysis of the NS-3 trace file .tr in VANETs networks using multi-agent systems that automatically calculates VANET network performance parameters such as throughput, packet delivery rate, packet loss, packets sent and packets received, it can also draw a graph. This tool calls Ns-3 Analyzer using Multi-Agents System (NsAMAS).

The rest of this article is organized as follows: Section 2 provides similar work developed. Section 3 presents the concept of agent and multi-agents system. Section 4 introduces NsAMAS, presents its concepts and explains its architecture. Section 5 we describe the most important and unique features of NsAMAS. Section 6 explains the uses and actual results. Then the study ends in section seven a conclusion and Future Work.

2 RELATED WORK

There are several analysis tools have been developed to generate statistics on network simulation performance [7][8]. For example, JTrana, Trace Graph, and NS-2 Trace Analyzer. They use NS-2 trace files.

JTrana [9] is a Java-based NS-2 trace analyzer used to study wireless network simulation using a graphical user interface. Its input is an NS-2 wireless trace, and its output consists of many graphs or data sets of the network.

Trace Graph [10][11] is another NS-2 trace file analysis tool. This tool offers many analysis options. It is implemented in MATLAB 6.0 [12] and can be compiled to work

without MATLAB. This tool also allows the user to extract useful statistics from a specific NS-2 trace file through a graphical user interface.

NS-2 Trace Analyzer [13][14] is a command line tool written in C / C ++ and is designed for use on all OS and Cygwin platforms. Similar to the previous tools, NS-2 Trace Analyzer could be used to receive popular network data based on the output of the trace file in the simulation process. This tool does not offer the possibility of creating graphs concerning the statistics that the user retrieves on a simulation.

All the tools mentioned above are designed for the analysis of the NS-2 trace file. In addition, they do not allow the user to locally store each analyzed trace file in a database, in order to be able to reuse it without having to reopen it. To retrieve information about another simulation, the user must download the trace file again. This is a fairly slow task and can be acceptable for small trace files, but when it comes to simulations that produce large trace files, this process can take a long time. In our work we will try to find a solution that will help us to solve these problems.

3 MULTI-AGENTS SYSTEM (MAS)

The topic of multi-agents system (MAS), if not recent, is currently a very active area of research. It is a discipline which is interested in collective behaviors produced by the interactions of several autonomous and flexible entities called agents, these interactions revolve around cooperation, competition or coexistence between these agents. We are going to present the key concepts needed by agents and multi-agents system (MAS) to design a practical architecture for analyzing the NS-3 trace file.

3.1 Agent

The concept of agent is a whole concept born decades ago. In the literature, many works have focused on agents and multi-agents system. Definitions differ from author to author and from field of application to field. We can cite the most consensual definitions that we have found [15] as follows:

- An agent is an independent entity, real or abstract, capable of acting on itself and its environment, and which, in a multifactorial universe, can communicate with other agents.
- An agent is an entity which presents the following characteristics: independence, reactivity and anticipation and has a specific social capacity.

3.2 Multi-Agents System

Multi-Agents System (MAS) [16] is a system made up of agents of the same type or not, who cooperate and interact in a coordinated manner in order to achieve a common goal or objective in an environment of which they have the same perception, point of view, perceptions or complementary points of view.

A multi-agents system (or MAS) is called a system composed of the following elements [16]:

- An environment E , that is to say a space generally having a metric.
- A set of objects O . These objects are located, that is to say that, for any object, it is Possible, at a given moment, to associate a position in E . These objects are passive, c that is, they can be perceived, destroyed and modified by agents.
- A set A of agents, which are particular objects, understood to represent the active entities of the system.
- A set of relations R which brings together objects.
- A set of Op operations allow O objects to be interpreted, produced, consumed, transformed and manipulated by agents.
- Operators responsible for verifying the application of these operations and the reaction of the world to this attempt at modification, which is called the laws of the universe.

4 NsAMAS overview

NsAMAS is a new tool for automating the analysis of NS-3 trace file .tr in VANETS networks using multi-agents system. The objective of NsAMAS is to perform a practical and automatic analysis of the trace file .tr produced by a simulation using graphical interfaces.

4.1 Architecture

This software is built on three layers: a presentation layer, a business layer and a database access layer, as shown in Fig.1.

- Presentation Layer

This layer implements the graphical interface through which the user can interact with the software. This layer contains the master agent to read the file, extract the fields and transfer the data to another agent. User requests move from the presentation layer to the actual processing layer. It also provides data in graph, table and report form to study network performance.

- Treatment Layer

This layer processes the data obtained from the source and stores the results in the database, it contains two agents, one for calculating metrics and the other for drawing graphs. The primary processor is responsible for analyzing the trace files and storing the results, which is one of the basic software procedures. It is also responsible for calculating general simulation information's and general node software. The Charts agent is responsible for plotting each type of chart.

- Database Access Layer

This layer is used to store analysis results in the local database or to return requested data. The database access layer interacts with the agent which calculates the performance parameters to store the results, and with the master agent to retrieve the data and store the information for each trace file.tr. The database component of NsAMAS is the MySQL database. The contents of the database include one named table for each trace file entered by the user.

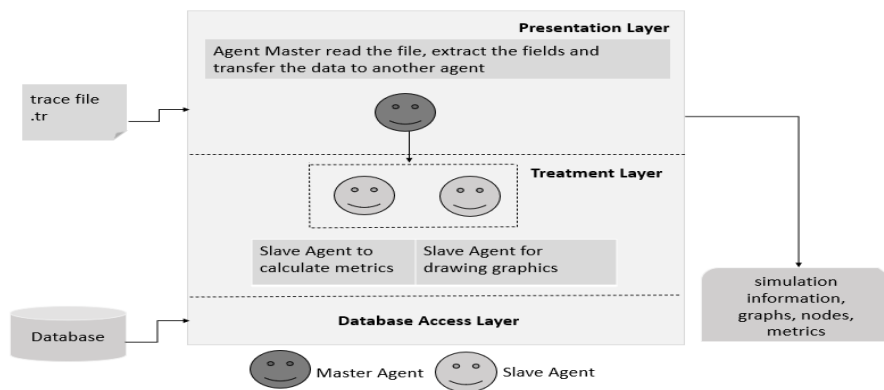


Fig. 1. NsAMAS Architecture

4.2 Issues with the design

One of the main goals of NsAMAS is to facilitate the analysis of trace files. For the moment, NsAMAS is focusing on NS-3 .tr trace files. As we mentioned earlier.

The trace data is in ASCII code and is organized into 12 fields [2]. Each trace line begins with an event (+, -, d, r) descriptor followed by the simulation time (in seconds) of that event, and the from and to node. Then we have the type, the size of the packet (in bytes) and Flags (ignore). The next field is the packet class, which can be used to identify a particular TCP connection. The next two fields are the source address and the destination address. Finally, the sequence number of the protocol packet and the packet identifier.

When a trace file .tr is sent as input, the agent can read the file, extract the 12 fields mentioned above for use in the processing phase, and finally store the information about this file in the database.

5 Trace file analysis

In this section, we describe the most important and unique features of NsAMAS. In addition, we present and discuss the structure and use of this software.

5.1 Simulation execution

The first step is to load the trace file .tr produced which corresponds to the overall configuration of the NS-3 simulation scenario. After this task the file will be read and the information processing will start using an agent. The data on the produced documents will be recorded in the database at the end of the processing and analysis step. The general idea of the system, for analyzing, processing and storing file information, is shown in Fig.2.

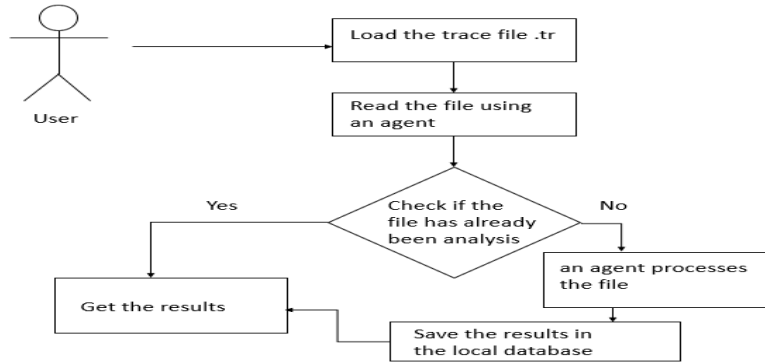


Fig. 2. Simulation execution steps.

5.2 Identification and storage

The first step in analyzing the trace file .tr is to extract 12 fields from the trace file that were provided as input. When a trace file has been selected, an agent first tries to extract the fields, then another agent uses these fields to process and analyze the data. Finally, when the processing and analysis phase of the trace file is completed, the information must be stored in the database. Each trace file presented as an entry has its own table named and information about it is stored there.

5.3 Statistics calculation

Trace file processing procedures are the step at which NsAMAS calculates all general simulation information about the input trace file.tr. These metrics are calculated automatically using an agent and without any user intervention and it calculates other information regarding the metrics for each node.

— General simulation information.

The general simulation data calculated for the input file includes the file path and the general simulation time. for information on packets transferred or dropped during simulation, the recorded information is the number of packets sent, the number of packets received and the number of packets dropped.

— **Node specific information.**

After processing of the .tr trace file, the user of the simulation data can obtain information corresponding to a specific node. This information includes statistics such as TX Packets, RX Packets, Total dropped packet, Data received, Data sent, Throughput, Goodput and Packets Delivery Ratio (PDR).

6 NsAMAS USAGES AND RESULTS

In this section, we present an example of using NsAMAS in which we open a new trace file and we show what information NsAMAS can extract. We present the calculation of general simulation information, general node information and graphics. The table 1 provides the simulation parameters that we used in our experiment.

Table 1. Simulation parameters

Parameters	Values
Version of Simulator	NS-3.27
Routing Protocol	DSR
Number of Vehicles	50
Simulation Time	120 s
Physical and MAC layer	802.11p
Packet Size	1000 bytes
Mobility Model	ManhattanGrid

6.1 Simulation parameters extraction

To measure network performance, using our NsAMAS trace file analyzer, an agent calculates various performance parameters: the general simulation parameters (Figs.3) and the specific parameters to each node (Fig.4). As an example of these parameters: throughput, packet drop rate, good throughput, packet delivery rate (PDR), packet loss, packet loss rate (PLR), normalized routing load (NRL) and concentration activity. These parameters are the most common requested by researchers and at the same time the most useful for evaluating the performance of VANET networks. User can also retrieve the results of previously scanned trace files from a local database as shown in Fig.5.

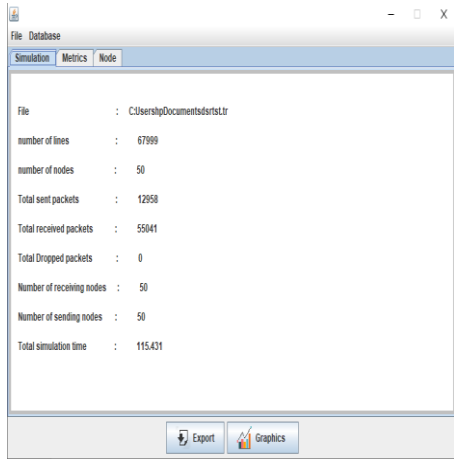


Fig. 3. General Information

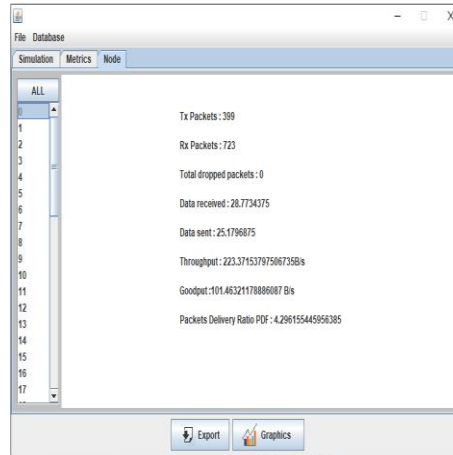


Fig.4. Specific node metrics

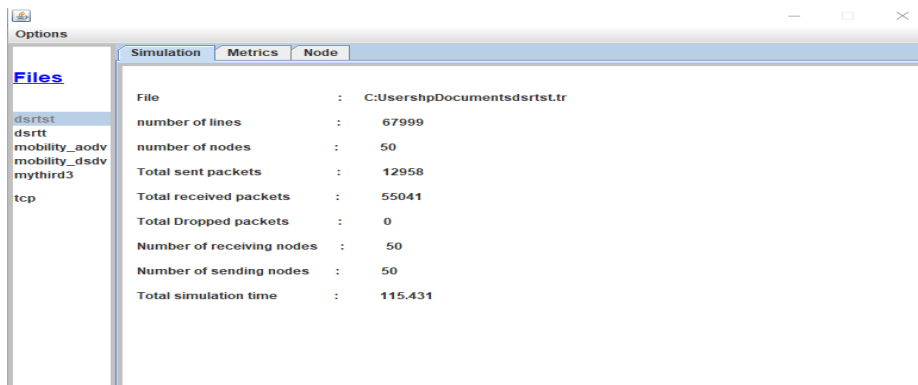


Fig. 4. Selection of a file in the previously analyzed database

6.2 Chart plotting

At this point the user can draw various graphs in the form of a curve or a histogram as shown in Fig.5, and Fig.6. Once all parameters are set, graphics can be drawn for general simulation or for a particular node. The graphics provided by this tool are the packet delivery rate and the bit rate (bits / s) versus time. Finally, it is also possible to export graphics as an image.

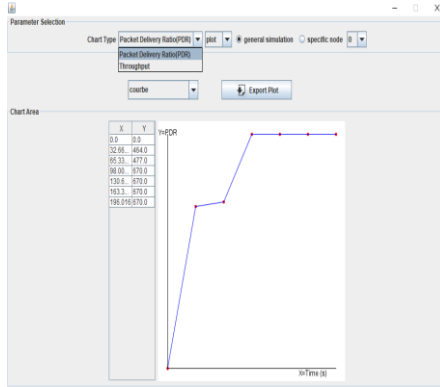


Fig. 5. Chart plotting

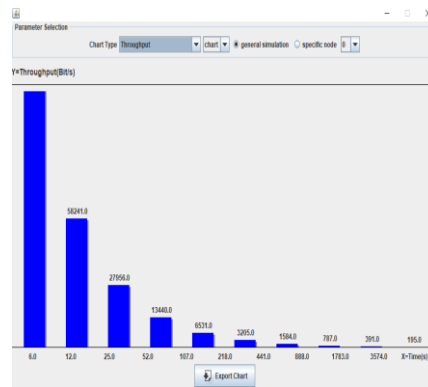


Fig.6. Bar chart

6.3 NsAMAS Results

In this section, we provide timing results in relation to the basic procedures described in section 5, which include reading the trace file, analyzing, processing and finally transferring its contents to the database local. We have shown that software not only manages to handle trace file .tr, but it does so extremely quickly. Therefore, we present the results in Figure 7 regarding the above processes.

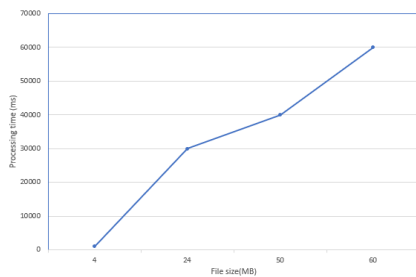


Fig. 7. Processing time result for the trace file .tr



Fig. 8. File processing time comparison

Based on the above results, it is clear that our NsAMAS software operate even with very large trace files to keep the processing time very low. And we notice that the analysis time increases with the size of the file.

We also provide a comparison of NsAMAS with other popular simulation analysis tools. These tools are described in Section 2 and are some of the best-known tools for dealing with simulated trace files. One of their main drawbacks is the time taken to open the trace file. The comparison shows that our NsAMAS software gave good results in processing time. Figure 8.

7 Conclusion and Future Work

In this work, we have presented a new NsAMAS tool to automate the analysis of “.tr” trace files of simulations of VANETs networks carried out with NS-3 using the concept of a multi-agents system. This new tool aims to offer an easy-to-use interface. use whereby users can load simulation trace files to facilitate the analysis procedure.

We have used simulation .tr trace files to produce measurements, statistics, and graphs to help students and researchers study the performance of their VANET networks under NS-3.

Building on the success of this software, we intend to upgrade it to calculate other parameters, such as transmission capacity and power consumption. We also intend to use other big data tools to analyze the file and draw another diagram.

References

1. Weingärtner, E., vom Lehn, H., Wehrle, K., 2009. A performance comparison of recent network simulators. In Proceedings of the IEEE International Conference on Communications 2009 (ICC 2009), Dresden, Germany, IEEE.
2. Christos Bouras, Savvas Charalambides, Michalis Drakouelis, Georgios Kioumourtzis, “Simulation Design and Execution the Case of TRAFIL”, International Conference on Simulation and Modeling Methodologies, Technologies and Application, 2013.
3. S. Siraj, A. Gupta, and R. Badgular, “Network simulation tools survey,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 1, no. 4, pp. 199–206, 2012.
4. <https://www.nsnam.org/> Accessed 11-July -2020.
5. Simulator Network 2, “Ns-2: <https://www.isi.edu/nsnam/ns.>” Accessed 4-Oct-2020.
6. I. Dorathy and M. Chandrasekaran, “Simulation tools for mobile ad hoc networks: a survey,” Journal of applied research and technology, vol. 16, no. 5, pp. 437–445, 2018.
7. <https://www.nsnam.org/docs/manual/html/tracing.html> Accessed 10 -July-2020
8. https://www.nsnam.org/docs/release/3.9/tutorial/tutorial_23.html Accessed 10-J July-2020
9. H. Qian, W. Fang, Jtrana: a java-based ns2 wireless trace analyzer, 2008.
10. J. Malek, K. Nowak, Trace graph-data presentation system for network simulator ns, in: Proceedings of the Information Systems – Concepts, Tools and Applications (ISAT 2003), Poland, September 2003.
11. C. Cicconetti, E. Mingozzi, and C. Vallati, “A 2 k· r factorial analysis tool for ns2measure,” in Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools, p. 4, ICST (Institute for Computer Sciences, Social-Informatics and, 2009.
12. The Mathworks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA, 2012.
13. Aliff Umair Salleh, Zulkifli Ishak, Norashida Md. Din, Md Zaini Jamaludin, Trace Analyzer for NS-2, in: Proceedings of the 4th Student Conference on Research and Development (SCoReD 2006), Shah Alam, Selangor, Malaysia, 27–28 June, 2006.
14. Uma R Pujeri, V Palaniswamy, “Trace Analyzer for NS3” Volume7, Number5, October 2015.
15. J.Ferber Les systèmes multi-agents : un aperçu général. Technique et Science Informatiques, vol.16 n°8, pp. 979-1012, 1997.
16. J. Ferber. Les Systèmes Multi Agents : vers une intelligence collective, InterEditions, 1995.