

Research Article

Simulation of Efficiency in Mobile Ad Hoc Networks using OMNeT++

¹Varun Manchikalapudi and ²Sk. Khadar Babu

¹School of Computing Science and Engineering,

²School of Advanced Sciences, VIT University, Vellore, India

Abstract: A network is a group of two or more computer systems linked together. There are many types of computer networks which are categorized based on topology, protocol and architecture. A Mobile Ad hoc Network (MANET) is a self-configuring infrastructure less network of mobile devices connected by wireless. Ad hoc networks maintain an unfair behavior in flow control especially when considered in the case of IEEE 802.11 Mac layer. Introducing efficiency in 802.11 is not an easy task. It reduces the overall global throughput. The network is to be designed in such a way that it deals with the fairness and throughput by maximizing aggregate throughput. Such kind of network design can be efficiently implemented on an evolving simulation tool named OMNeT++.

Keywords: MANET, OMNeT++, throughput

INTRODUCTION

A Mobile Ad hoc Network (MANET) is a self-configuring infrastructure less network of mobile devices connected by wireless. Ad hoc means "for this purpose". A Mobile Ad-hoc Network (MANET) is an ad-hoc network but an ad-hoc network is not a MANET. Each device in a MANET is free to move independently in any direction and will therefore change its links to other devices frequently.

MANET challenges: A MANET environment has to overcome certain issues of limitation and inefficiency. It includes.

The wireless link characteristics are time-varying in nature: There are transmission impediments like fading, path loss, blockage and interference that add to the susceptible behavior of wireless channels. The reliability of wireless transmission is resisted by different factors.

Limited range of wireless transmission: The limited radio band results in reduced data rates compared to the wireless networks. Hence optimal usage of bandwidth is necessary by keeping low overhead as possible.

Packet losses due to errors in transmission: MANETs experience higher packet loss due to factors such as hidden terminals that results in collisions, wireless channel issues (high Bit Error Rate (BER)), interference and frequent breakage in paths caused by mobility of nodes, increased collisions due to the presence of hidden terminals and uni-directional links.

Route changes due to mobility: The dynamic nature of network topology results in frequent path breaks.

Frequent network partitions: The random movement of nodes often leads to partition of the network. This mostly affects the intermediate nodes.

Fairness issues in ad hoc networks have been deeply studied for a couple of years. Several mechanisms and protocols have been proposed to solve the fairness issues. There exist two main approaches in the literature. One approach is based on information exchanges between stations and/or knowledge of the topology. The other approach is topology independent and does not required any information exchanges. Pre-describe a mechanism for translating a given fairness model into its corresponding collision resolution backoff algorithm that probabilistically achieves the fairness objective but Requires an efficient Collision avoidance Scheme (as RTS/CTS) to be efficient (Melih *et al.*, 2005).

Our aim is to find the best trade-off between fairness and global throughput. As far as we know, only one paper deals with the trade-off between these two notions, but the proposed algorithm requires knowledge of the topology and an exchange of flow of information between nodes. However, it appears from the literature that designing a MAC protocol, fair and efficient in terms of global throughput that does not require any knowledge of the topology or specific information from other nodes than those provided by the MAC 802.11 protocol and the data traffic in the network is still a real challenge.

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, queuing 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 (Stuart, 2014). There are extensions for real-time simulation, network emulation, alternative programming languages (Java, C#), database integration, SystemC integration and several other functions. OMNeT++ is a powerful tool and much more useful than just TCP/IP network simulation.

OMNeT++ is an object-oriented modular discrete event network simulation framework. An OMNeT++ model is defined by a textual network description. A network description contains declarations of simple module types, describes compound module types and contains a network definition that instantiates a compound module (Andras, 2010a). In OMNeT++ the textual model description is compiled into C++ code and linked into the simulator executable. This fact has the following important consequence: when one creates a compound module dynamically, its sub-modules and internal connections can be automatically built out by the compiled network description code. In the OMNeT++ simulator, run-time efficiency is guaranteed by the fact that the textual model description is translated into C++ and used in its compiled form. This makes it very fast to build up the model internally when the simulation starts (Andras, 2010b).

A Discrete Event System (DES) is a system where state changes (events) happen at discrete instances in time and events take zero time to happen. It is assumed that nothing (i.e., nothing interesting) happens between two consecutive events, that is, no state change takes place in the system between the events. This is in contrast to continuous systems where state changes are continuous (Andras, 2010a). Systems that can be viewed as discrete event systems can be modeled using discrete event simulation, DES. Computer networks are usually viewed as discrete event systems. Some of the events are start of a packet transmission, end of a packet transmission, expiry of a retransmission timeout (Andras, 2010b).

MATERIALS AND METHODS

Existing system:

Mad mac: The approach of MadMac is to provide a schedule on the packets but topology independent and with no extra information than the one provided by 802.11. Of course, a perfect schedule is difficult to

obtain with these constraints but the results will show that we obtain good performances.

The basic scheme: The idea behind the proposed protocol comes from the following remarks:

- If an active node senses activity on the channel, then it means that it is not alone on the channel and that at least two stations (including itself) send packets on the radio medium (Razafindralambo and Gue´rin-Lassous, 2008).
- If an active node experiences one or more collisions on its packets, then we can derive the same conclusion: at least two stations (including itself) send packets on the radio medium (Razafindralambo and Gue´rin-Lassous, 2008).

The second statement differs from the first one in the sense that the detected competing stations are not necessarily in communication or in carrier sensing range (Razafindralambo and Gue´rin-Lassous, 2008). However, we can say that, from the point of view of the node that experiences collisions, they share the medium since the station cannot successfully send its packets due to interfering transmissions. Note that, considering only the sensing activity and/or the experienced collisions, a node cannot deduce how many nodes compete with it. To approximate this number, other operations are required like capturing useful data (the source and the destination for instance) in control and data packets. However, it seems difficult to exactly deduce this number as soon as a carrier sensing mechanism is used. Since we do not want to use and send extra information, each node can only deduce, with these two statements, whether it shares the medium (in the general sense) with at least one another node.

If the first statement is true, a boolean variable called ACT is set to 1. If the second statement is true, another boolean variable called COL is set to 1 (Congduc, 2011). Since the share is not permanent, these variables are updated periodically. We consider a period of Delta Slot roughly equal to a transmission of 10 packets (data of 1000 bytes sent at 1 Mbps). At the beginning of each Delta Slot, the ACT and COL variable are reset to 0. The Delta Slot period behaves as a sliding window. When ACT or COL is equal to 1 for a node, this one considers that it shares the medium with one or more stations and reduces its MAC throughput by 2 by introducing a waiting time before each new packet to send. The goal of this waiting time is to introduce an alternate schedule between the competing nodes (Razafindralambo and Gue´rin-Lassous, 2008).

Performance of one hop networks: The first simulations have been performed on the simple scenarios where communications take place between

nodes that are in communication range of each other (Congduc, 2011). In these scenarios there is no fairness issue and the goal is to compare the global through-put of MadMac, PNAV, MBFAIR and 802.11 in this classical configuration (Fig. 1).

Proposed system: In order to overcome the drawbacks in the existing system and existing proposed protocols, we consider the basic routing protocols for ad hoc networks and implemented using new simulation tool OMNeT++ (Pradeep Kumar, 2014). This tool helps to design network topography in such a way that the data transmission in the ad hoc networks is done efficiently without the decrease in the global throughput. Any of the existing routing protocol for ad hoc networks is considered in the topography.

OMNeT++ is simulation tool which uses C/C++ as the frontend code and the Network Descriptor (NED) (Andras, 2010a), which is utilized to create the topography of the network. This tool also consists of header file (.h), initialization file (.ini) and cc file for writing the C++ code. The topography is designed in the NED file and the C++ code is written in the .cc file in which the headers are included from the .h file and are interlinked with .inifiles (Andras, 2010b).

While running the simulation of topography the event window will be tracked down parallelly in OMNeT++ tool. The efficiency of the network

topography can be analyzed using the data retrieve while simulation. Here the throughput analysis, the number of packets sent to the number of packets received can be calculated, in this way the variations in the efficiency of the network topography can be studied and analyzed (Andras, 2010a).

RESULTS AND DISCUSSION

The result for the proposed system is as follows.

Figure 2 shows the basic topography of 19 hosts along with a server and how packet transmission is done between a single host and the server directly. Here the host wants to transmit data to the server, so it transmits the packet in the broadcasting channel by maintaining the global throughput thus maintaining the efficiency.

The packets can not only be transmitted directly to the server but can be transmitted with the help of other hosts also. When a particular host wants to transmit the data to the server it can transmit it to the nearby host which forwards the packets. This can be clearly seen in the Fig. 3.

When many different hosts in the topography sends packets to the server via in-between nodes a collision arises near the server by decreasing the global throughput and efficiency as shown Fig. 4.



Fig. 1: A simple illustration of the basic scheme of Mad Mac

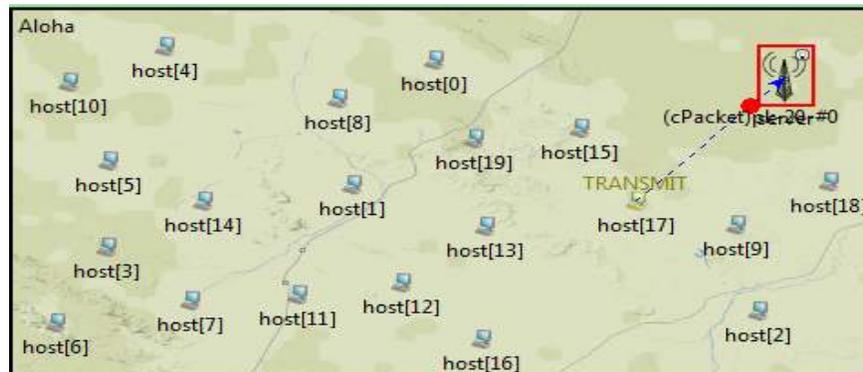


Fig. 2: An ad hoc network with hosts and servers

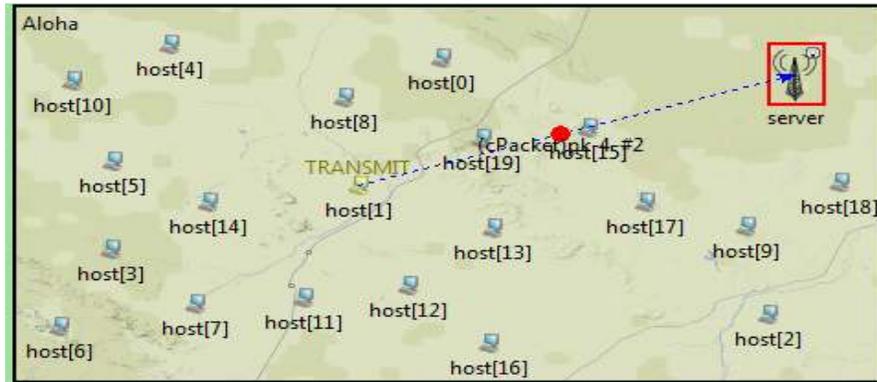


Fig. 3: Transmission via in-between nodes

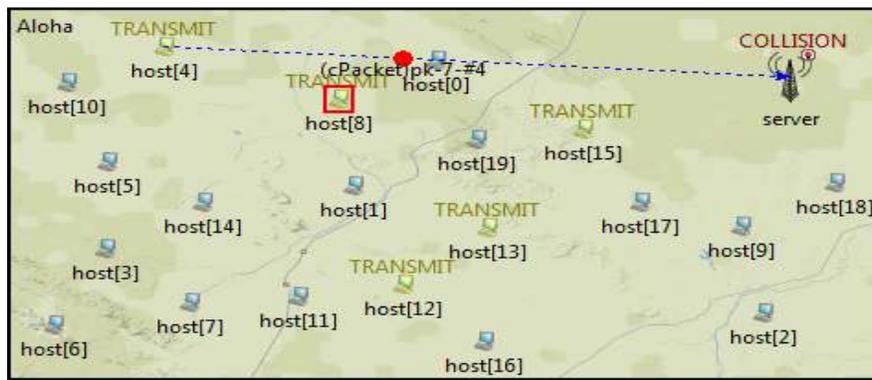


Fig. 4: Collision near the server

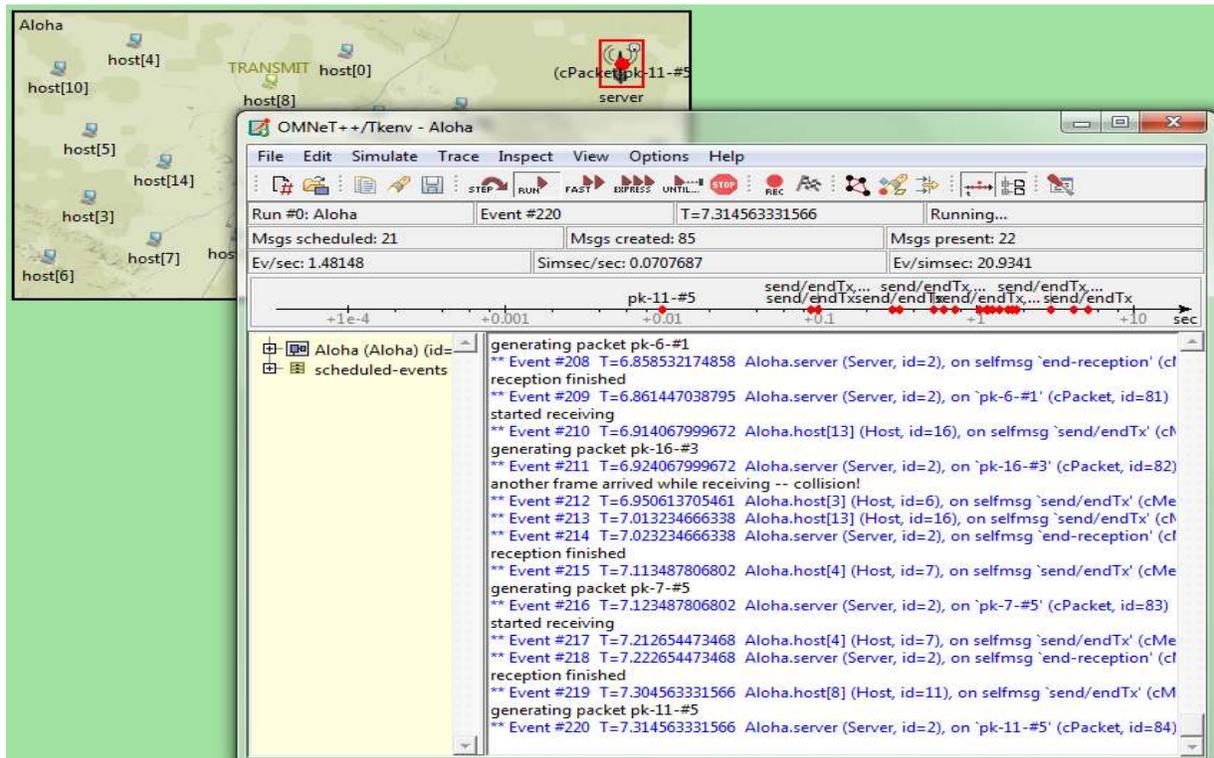


Fig. 5: Scheduling of the packets

The scheduling of the packets from different hosts to the server can be viewed with the help of “Tknev” file in the OMNeT++ simulator. The complete log is present in this file where the scheduling can be controlled in order to prevent the collision near the server. By controlling the scheduling of the packets flow from different hosts to the server efficiency can be maintained and there will be no decrease in the global throughput (Fig. 5).

CONCLUSION

This project briefly concludes how the efficiency in ad hoc networks is maintained without decrease in global throughput with the help of simulation tool OMNeT++. Since maintaining efficiency in 802.11 with an increase in global throughput is not an easy task, we designed a network with the help of OMNeT++ in such a way that it maintains the efficiency without any decrease in global throughput. Thus the efficiency is maintained in ad hoc networks with the help new simulation tool and existing protocols for ad hoc.

REFERENCES

- Andras, V., 2010a. OMNET ++ User Manual version 4.6. Retrieved from: <http://www.omnetpp.org/doc/omnetpp/manual/usman.html#fig:ch-o+vreview:modules>.
- Andras, V., 2010b. OMNeT++Discrete Event Simulation System. Retrieved from: <http://www.ewh.ieee.org/soc/es/Nov1999/18/ned.htm>.
- Congduc, P., 2011. A Video Sensor Simulation Model with OMNET++. LIUPPA labs, University of Pau, France.
- Melih, O., R. Andrea, K. Kishore and S. Christian, 2005. Efficient broadcasting and gathering in wireless ad-hoc networks. Proceeding of the 8th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAAN), DOI: 10.1109/ISPAAN.2005.43.
- Pradeep Kumar, T.S., 2014. VIT Chennai. Retrieved from: <http://www.slideshare.net/tspradeepkumar/simulators-for-wireless-sensor-networks-omnet>.
- Razafindralambo, T. and I. Gue´rin-Lassous, 2008. Increasing fairness and efficiency using the MadMac protocol in ad hoc networks. *Ad Hoc Netw.*, 6(3): 408-423.
- Stuart, M., 2014. Retrieved from: <http://www.finmars.co.uk/blog/8-omnet-general-network-simulation>.