IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-1, Jan 2019 E-ISSN: 2347-2693

MCMSim Test-Bed for Multi-Channel MAC Framework For MANET

Rohit Kumar[1*], Kavita Taneja[1], Harmunish Taneja[2]

[1]Department of Computer Sc. & Applications, Panjab University, Chandigarh, India
[2]Department of IT, DAV College Sec-10, Panjab University, Chandigarh, India

[*]*Corresponding Author: rohitbhullar@gmail.com, Tel.: +91-9802-039898*

Available online at: www.ijcseonline.org

Accepted: 26/Jan/2019, Published: 31/Jan/2019

Abstract— With the rising computational capacity requirements and ubiquitous computing, newer and powerful hardware and software platforms are being created. These newer powerful architectures and platforms are significantly different from their conventional predecessor architectures. These new platforms need to be evaluated, tested and validated for performance measurement before the product release. In the area of mobile ad hoc networks the use of multiple communication channels has emerged as a powerful new hardware platform which can increase the capacity manifold. But one major problem faced by researchers is that the traditional simulation tools don't support these improved hardware platforms. So to carry out performance analysis of these platforms new test-beds and simulators are required to be developed. In the proposed work a software simulation test-bed for multi-channel Medium Access Control framework for mobile ad hoc Networks (MANET) has been developed and is named MCMSim.

Keywords— Mobile ad hoc networks, test-beds, NS-3, Multi-channels, Network, Simulation Tools

I. INTRODUCTION AND RELATED WORK

Multiple types of simulators are available for testing and validating protocols, techniques for mobile ad hoc networks. Traditional simulators are able to provide conventional execution environment for common type of tasks on a given platform. But when the core network enabling technology or framework is changed then these simulators doesn't prove to be sufficient. In order to test new platform either new enhancements are required to be made to the existing simulators or new test-beds or simulators are required to be developed [1,23].

In the proposed work a new framework/platform has been used to provide performance improvement for MANET and other wireless technologies. Existing simulators [1-20] doesn't supports number of features like multiple type of rendezvous modes and most commonly parallel rendezvous mode, multiple channels, type of mobility and mobility control required for newer platforms and many other configurable performance parameters. In the proposed work NS-3 compliant software test-bed has been created to capitalize the performance benefits of multiple channels and other hardware improvements provided by the new underlying framework. Some existing and primarily used simulators for mobile ad hoc networks has been presented in table 1.

Table 1:. Commercial and Open Source Simulators

	Network simulators name
Commercial	OPNET, QualNet
Open source	NS2, NS3, OMNeT++, SSFNet, J-Sim

1.1 Test beds and Issues

Test beds are one of the most credible mechanism for establishing and measuring network performance for newly introduced frameworks. Test beds tries to provide on field like capability and environment for testing the underlying network, associated software and hardware components. But the major limiting factor of test-beds is that they are cost and efforts intensive and may be out of reach for average developers. Other problem with test beds is scalability especially when the test bed is largely made of hardware components. Installing new hardware and integrated circuits may not be possible all the time and this may hinder testing of large scale networks. Historically as per the literature [23] maximum capacity of test-bed developed so far is limited few hundreds of nodes. But with the rising requirements for large networks this capacity is not sufficient.

II. TEST-BED STRUCTURE AND CREATTION

On the sidelines of NS-3 the test-bed has been written and developed in C++ programming Language and is able to cater to all needs and requirements of multiple wireless channels. The test-bed can be easily integrated with NS-3 and produces output similar to NS-3 in the form of trace files. These trace files are easily readable and can be read and analyzed easily by number of filtering tools like AWK to produce quantifiable results. Output graphs can also be directly generated from such trace files. The output file structure produced by the test-bed is on lines with pcap (packet capture files) which are largely used and universally accept network trace file types.

2.1 Features of Test-Bed MCMSim:

The test-bed developed has been named MCMSim and is a powerful and state of art tool written in C++ to utilize the potential of multi-channel communication platform. The test-bed provides ambient environment to test potential and capabilities of multi-channels especially in context of mobile ad hoc networks. The MCMSim can provide number of capabilities and capacities. Each node or mobile device participating in communication is equipped with numerous capabilities. These capabilities are briefly shown in figure 1. Some of the capabilities of a each node provided by MCMSim are:

Device Information: In multi-channel environment a host or mobile device can be configured in number of ways. This function allows user to know all details about a given host and helps in error diagnosis significantly

Sending Data in Multiple Modes: A mobile device can transmit data using multiple modes depending upon the traffic quantity and density of devices.

Receive Data in Multiple Modes: A mobile device can receive data using multiple modes depending upon the consent given to a particular sender.

Mobility Support: Random walk model is directly supported as mobility control mechanism. Other modes are also supported with minor alterations and configurations.

Collision Detection: Time out mechanism are used to identify loss of frames due to collisions and other reasons.

Enabling and Disabling of a Channel: A mobile device can switch a channel ON or OFF depending upon the network load and is done automatically by the algorithms used for implementing the system.

Noise Control: To observe the network environment under different noise and interference levels this feature can be used. Packet drop rate is configured to check this impact.

Delay Control: The amount of delay can be controlled as a simulation parameter to take care of propagation and frame processing over heads.

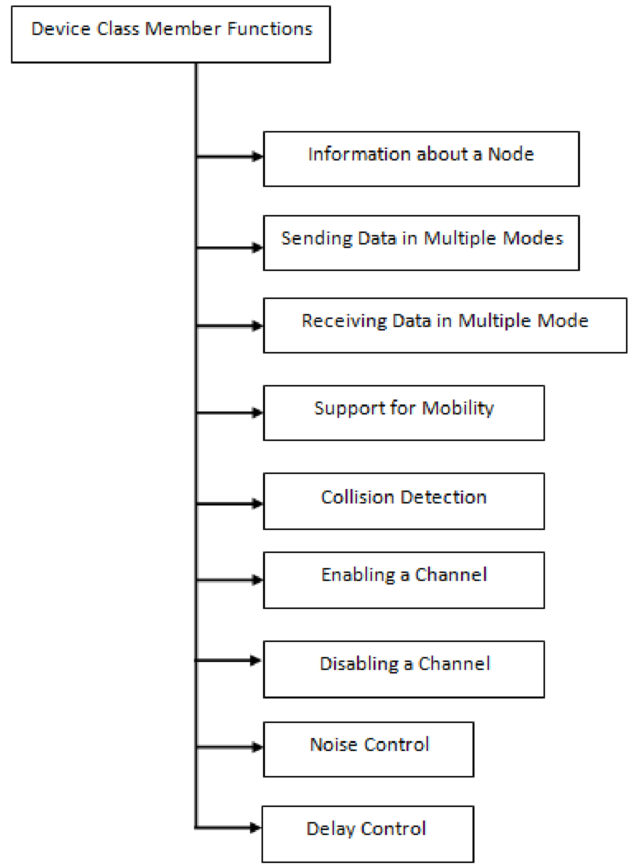

Figure 1: Capability of a Mobile Device.

2.2 Major Exception Handled

MCMSim is able to detect number of exception which can occur during network operation. Some very common exceptions handled by the test-bed are shown in figure 2.

Buffer Overflow: Each device maintains incoming and outgoing queues to hold incoming and outgoing data. These queues can get overflowed if limit is breached. In the event of buffer overflow this exceptions occurs and are recorded in log files.

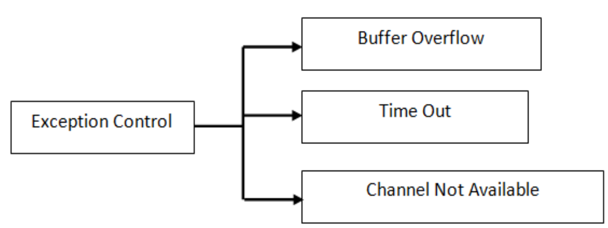

Figure 2: Common Exceptions Handled by MCMSim.

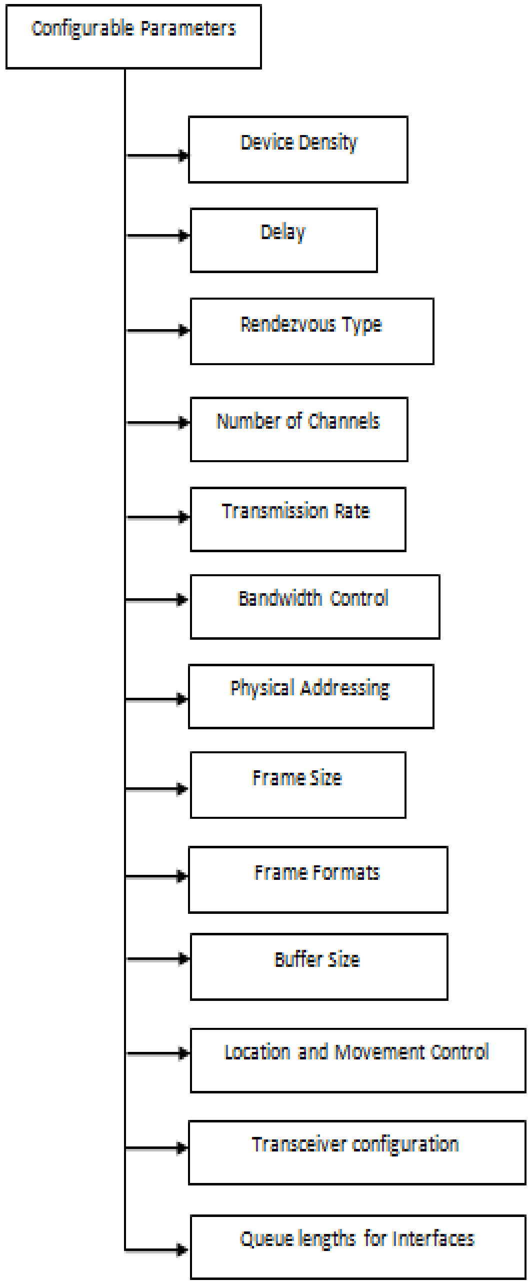

Figure 3: Configurations Offered by MCMSim.

Time out and Channel Not Available: These are other two kinds of exceptions handled by the system. Time out is related to re-transmissions of lost frames. Channel not available exception occurs at the time of connection establishment if channel with desired parameters is not available for communication.

2.3 Configurable Parameters:

MCMSim provides a flexible set of environment to operate the network and to observe the impact of numerous parameters. Figure 3 presents major parameters which a user can configure in the test-bed. Among these all parameters number of channels to be used, type of rendezvous and bandwidth control are unique feature of the test-bed. One sample fragment of code for interface class is shown in figure 4.

```
class interface
{
    public:
    int datarate;
    int queuelenght;
    string buffer[5];
    float delay;
    float waitingtime;

    //public:
    interface ()
    {
     queuelenght=0;
     delay=0.2;
     waitingtime=0;
     datarate=5;
    }
}

    void transcieverinfo()
```

Figure 4: Class Structure for an Interface.

Output produced by the MCMsim system in the form for trace file. Figure 5 and 6 presents typical output snapshot of trace files produced by the system. Figure 5 presents the trace file for the sender whereas figure 6 presents the trace file for receiver. Here both the sender and receivers operates in high traffic mode and are performing parallel data transmissions on different channels. The proposed system offers number of operating modes for different environmental conditions and corresponding output trace files.

III. Comparison with Simulators

In this section MCMSim has been compared with other simulation tools used for simulating mobile ad hoc networks. Each simulator have its own specializations and advantages over other simulators. In this section all major simulators used have been compared with each other. For comparison, numbers of parameters have been considered raging from granularity to market share of all these simulators. Table 2, 3 and 4 presents the comparative analysis of all these simulators.

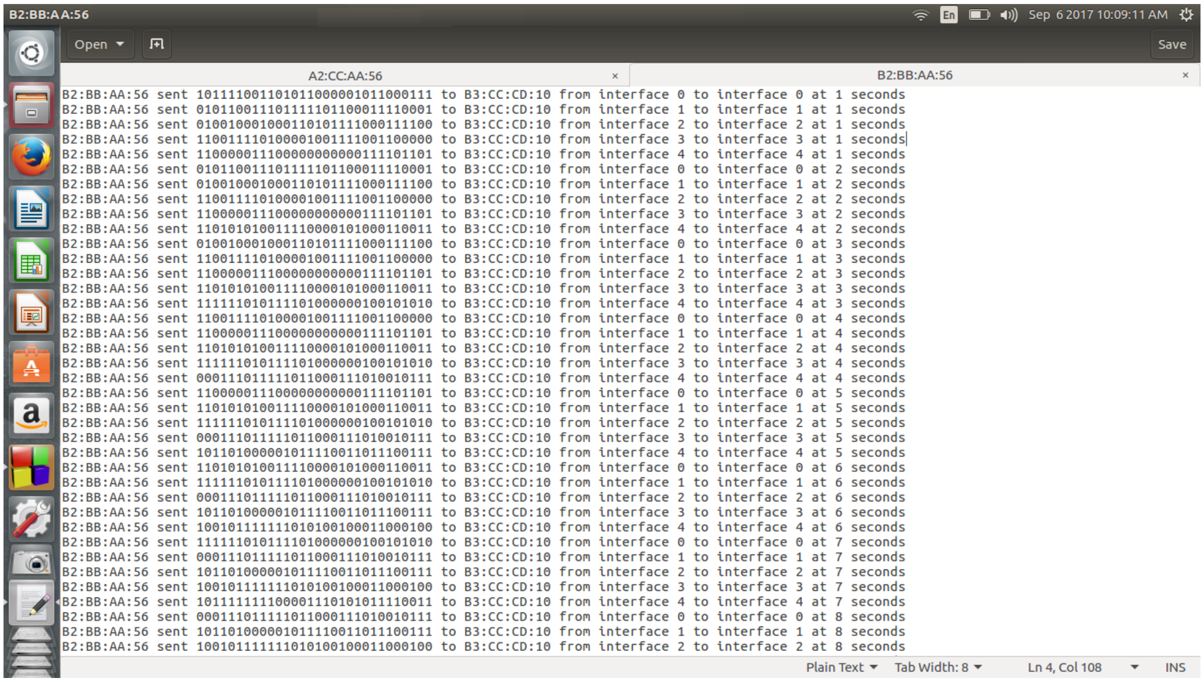

Figure 5: Output Snapshot at Sender Side for High Traffic Density.

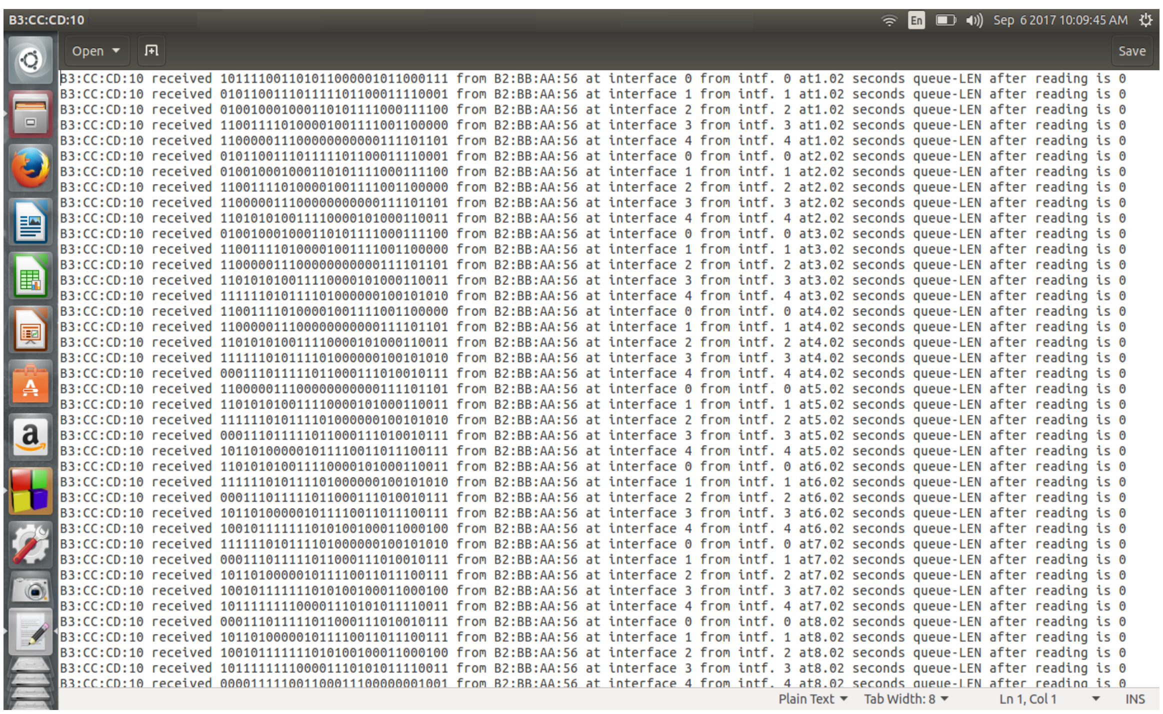

Figure 6: Output Snapshot at Receiver Side for High Traffic Density.

Granularity

Granularity, we define finest < finer < fine < medium < application-level.

Table 2: Comparison of Simulators

Name of Tools	Granularity	Mobility
MCMSim	Finest	Supported
ns-2	Finest	Supported
NS-3	Finest	Supported
DIANEmu	Application-level	No
Glomosim	Fine	Supported
GTNets	Fine	No
J-Sim	Fine	Supported
Jane	Application-level	Native
NAB	Medium	Native
OMNet++	Medium	No
OPNet	Fine	Supported
QualNet	Finer	Supported
SWANS	Medium	Supported

Table 3: Interfaces and Parallelism Support:

Name of Tools	Parallelism	Interface
ns-2	No	C++/OTCL
MCMSim	Yes	C++
DIANEmu	No	Java
Glomosim	SMP/beowulf	Parsec (C-based)
GTNets	SMP/beowulf	C++
J-Sim	RMI-based	Java
Jane	No	Java
NAB	No	OCaml
OMNet++	MPI/PVM	C++
OPNet	Yes	C
pdns	beowulf	C++/OTCL
QualNet	SMP/beowulf Parsec (C-based)	Parsec (C-based)
SWANS	No	Java

Table 4: Popularity Analysis

Name	Popularity	Name Popularity License
ns-2	88.8%	Open source
GloMoSim	4%	Open source
OPNet	2.61%	Commercial
QualNet	2.49%	Commercial
OMNet++	1.04%	Free for academic and educational use
NAB	0.48%	Open source
J-Sim	0.45	Open source
SWANS	0.3%	Open source
GTNets	0.13	Open source
pdns	< 0.1%	Open source
DIANEmu	< 0.1%	Free
Jane	< 0.1%	Free
MCMSim	<0.1	Open Source

IV. Conclusion and Future Scope

Comprehensive evaluation of a network is an important task tfor validation and benchmarking of given ad hoc network. As it can be seen that number of tools are available for ad hoc network design, testing and evaluation. Each of these tools offers some distinctive features like Graphical user interface, ease of network setup and many more. Some of the key challenges for tools in ad hoc network which are of utmost importance include network size, node density capacity, parameters configuration, support for new protocol development, support for new packet formats and much more. Some tools like NS-3, NS-2 offers number of these features but are not able to support multi-channel networks in full operational modes. MCMSim takes care all operational requirements required by multi-channel based framework and work at par with the standard simulation tools.

References

[1] Hortelano, Jorge, et al. "Castadiva: a test-bed architecture for mobile ad hoc networks." *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*. IEEE, 2007.

[2] Rampfl, Sebastian. "Network simulation and its limitations." *Proceeding zum Seminar Future Internet (FI), Innovative Internet Technologien und Mobilkommunikation (IITM) und Autonomous Communication Networks (ACN)*. Vol. 57. 2013.

[3] Breslau, Lee, et al. "Advances in network simulation." *Computer* 33.5 (2000): 59-67.

[4] Bajaj, Lokesh, et al. "Glomosim: A scalable network simulation environment." *UCLA computer science department technical report* 990027.1999 (1999): 213.

[5] Lin, Guolong, Guevara Noubir, and Rajmohan Rajaraman. "Mobility models for ad hoc network simulation." *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. Vol. 1. IEEE, 2004.

[6] Tian, Jing, et al. "Graph-based mobility model for mobile ad hoc network simulation." *Simulation Symposium, 2002. Proceedings. 35th Annual*. IEEE, 2002
[7] Heidemann, John, et al. "Effects of detail in wireless network simulation." *Proceedings of the SCS Multiconference on Distributed simulation*. 2001.
[8] Fujimoto, Richard M., et al. "Large-scale network simulation: how big? how fast?." *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on*. IEEE, 2003.
[9] Beshay, Joseph D., et al. "Wireless networking testbed and emulator (winetester)." *Computer Communications* 73 (2016): 99-107.
[10] Klein, Michael. *Dianemu: A java based generic simulation environment for distributed protocols*. Universität Karlsruhe, Fakultät für Informatik, 2003.
[11] Riley, George F. "Simulation of large scale networks II: large-scale network simulations with GTNetS." *Proceedings of the 35th conference on Winter simulation: driving innovation*. Winter Simulation Conference, 2003.
[12] Sobeih, Ahmed, et al. "J-Sim: a simulation and emulation environment for wireless sensor networks." IEEE wireless communications 13.4(2006): 104-119.
[13] Hogie, Luc, Pascal Bouvry, and Frédéric Guinand. "An overview of manets simulation." *Electronic notes in theoretical computer science* 150.1 (2006): 81-101.
[14] Kumar, Rohit, Kavita Taneja, and Harmunish Taneja. "Performance Evaluation of MANET Using Multi-Channel MAC Framework." *Procedia computer science* 133 (2018): 755-762.
[15] Taneja, Kavita, Harmunish Taneja, and Rohit Kumar. "QoS Improvement in MANET Using Multi-Channel MAC Framework." (2018)..
[16] Varga, András, and Rudolf Hornig. "An overview of the OMNeT++ simulation environment." *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.
[17] Prokkola, Jarmo. "Opnet-network simulator, *URL http://www. telecomlab.oulu.fi/kurssit/521365A tietoliikennetekniikansimuloinnitjatyokalut/Opnetesittel*, 7 (2006).
[18] Ahmed, Sheeraz, Muhammad Bilal, and Umer Farooq. "Performance Analysis of various routing strategies in Mobile Ad hoc Network using QualNet simulator." *Emerging Technologies, 2007. ICET 2007. International Conference on*. IEEE, 2007.
[19] Barr, Rimon. "Swans-scalable wireless ad hoc network simulator." *User Guide, 2004.*
[20] Issariyakul, Teerawat, and Ekram Hossain. *Introduction to network simulator NS2*. Springer Science & Business Media, 2011.
[21] Riley, George F., and Thomas R. Henderson. "The ns-3 network simulator." *Modeling and tools for network simulation*. Springer, Berlin, Heidelberg, 15-34, 2010.
[22] V. Kumar and R. Kumar, "An Optimized Multichannel MAC Scheme with Dynamic Control Channel Interval in Dense VANET" *International Journal of Information Technology, 2019.*
[23] Brown, Timothy, et al. "Test bed for a wireless network on small UAVs." *AIAA 3rd" Unmanned Unlimited" Technical Conference, Workshop and Exhibit*. 2004.

