Sindh Univ. Res. Jour. (Sci. Ser.) Vol.44 (3) 491-496 (2012)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Implementation of the Dynamic Source Routing Protocol in Network Simulator 2 – An Overview

H. ZAFAR, L. HASAN, A. KHATTAK*, Z. MUFTI, S. JAN**

Department of Computer Systems Engineering, University of Engineering and Technology, Peshawar, Pakistan.

Received 12th May 2012 and Revised 06st August 2012

Abstract: A Mobile Ad Hoc Network (MANET) is a collection of mobile wireless nodes that can utilize multi-hop radio relaying and are capable of operating without necessarily using any pre-existing infrastructure. In MANETs, one of the key issues is how to route packets efficiently. Many routing protocols for such networks have been proposed. Amongst the most popular ones are Dynamic Source Routing (DSR), Ad Hoc On-demand Distance Vector (AODV) and Optimized Link State Routing (OLSR), which has the status of IETF Experimental RFCs. This paper presents an overview of the implementation of the DSR protocol in Network Simulator 2 (NS-2). A number of simulation-based experiments are presented to raise confidence in the model behavior and assumptions. The work can act as a primer for NS-2 beginners to use NS-2 for communication network performance evaluation based on simulation.

Keywords: Dynamic Source Routing, IETF, Mobile Ad Hoc Network, Network Simulator 2.

1. INTRODUCTION

Mobile Ad Hoc Networks (IETF MANET, 2012) are characterized by lack of infrastructure and frequent topological changes. Due to quick and economically less demanding deployment, MANETs find applications in several areas. Some of these include: military applications, collaborative and distributed computing, emergency operations, wireless sensor networks, wireless mesh networks and hybrid wireless network architectures. Design of appropriate routing protocols for MANETs is one of the major challenges.

The recognition of the requirements such as distributed implementation, adaptability to changing topology, efficient capacity utilization, energy conservation, freedom from loops and unidirectional link support has led to a large body of work on the subject of ad hoc routing. A large number of competing protocols have emerged. Some of them have become Internet Engineering Task Force (IETF) Request for Comments (RFCs); others have draft status in the IETF, while many have been dropped by the research community.

The techniques discussed in this paper are based on Dynamic Source Routing (DSR) routing protocol (Johnson *et al.,* 2007). The reasons for choosing DSR among the other routing protocols are as follows:

- In a range of scenarios, source routing is shown to outperform table-driven approaches (Broch *et al.,* 1998).
- DSR is widely considered as a benchmark source-based scheme (Broch *et al.,* 1998) (Tachtatzis and Harle, 2008) and standard, validated implementations of this protocol are available with the Network Simulator version 2 (NS-2) (NS-2, 2012).
- When the flow-state is disabled, the end-to-end path is appended to the packet's header. This feature permits more flexible monitoring and more flexible tracing of the path, when the packet is propagated from its source to destination (Tachtatzis and Harle, 2008).
- Intermediate nodes need not maintain up-to-date routing information in order to route the packets that they forward; the packets themselves already contain all the routing decisions.
- It is convenient to build multiple paths using source routing, since the destination node knows the complete path of all available routes.

2. MATERIALS AND METHODS

Dynamic Source Routing Protocol

Dynamic Source Routing (DSR) (Johnson *et al.,* 2007) is a reactive protocol. It has a flat structure, with all nodes treated equally by the routing algorithm. In DSR routing, the source node appends the complete routing path to each data packet before transmitting. Additionally, each node uses a caching technique to maintain the route information. Generally, routing construction in DSR comprises two major phases; the route discovery and the route maintenance. Prior to sending data packets, a source node consults the route

++Corresponding Author: haseeb@nwfpuet.edu.pk +92-91-9216590
*Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan.
**Department of Computer Science and Information Technology, University of Engineering and Technology, Peshawar, Pakistan.

cache to determine the path availability to the destination. If a valid routing path exists, it will be included inside the data packet, and if not, the source node invokes the route discovery by broadcast a route request (RREQ) packet. A RREQ packet is distinguished by using a unique number along with the address of both the source and destination. At every intermediate node, the received RREQ packet is compared against the route cache to match the route information for the destination. If such information is not available or has not been previously determined, the node appends its own address to the route record field of the RREQ before forwarding to its neighbors.

The communication overhead is reduced by removing duplicate RREQ packet and concatenating route information at intermediate nodes. Duplicate packets are prevented from being propagated if the RREQ contents matches the destination address appeared in the node's route cache. When the RREQ packet reaches the intended destination or an intermediate node that has valid path to the destination, a route reply (RREP) packet is generated and returned. The content of a RREP packet typically comprises the complete list of address in which the RREQ packet has traversed through. If the source of RREP is an intermediate node instead of the destination node, the address installed in the RREP is the list of nodes address traversed concatenated with the address from the intermediate node's cache.

There are three possible paths that can be followed by the RREP packet back to the source node. The first case is the destination or intermediate node already has a predetermined route to the source node in its route cache. Second and the most common method is in a reverse manner, to follow the path list in the route record field, assuming that the path between the pair of source and destination consists solely of symmetrical links. The third possibility is that there may be potential asymmetrical links along the routing path. In such cases, the RREP packet propagation is lost and a new route discovery is initiated to find another route to the source by the destination node. The RREP packet is piggybacked on the RREQ (by the destination node) and broadcast to the neighbors. During route maintenance, a routing path breakage is typically handled by the source node. When the data link layer detects link disconnection, a route error (RERR) packet that reversely follows the list of path in the route cache is unicast back to the source node. At each node, all routes associated with the broken links are removed from the route cache.

DSR has the advantages of being able to use asymmetric links and generating no additional routing overhead if the network topology remains unchanged. Moreover, it is easily implemented and supports multipath routing. The main disadvantage of DSR lies in the use of source routing. More specifically, the size of RREQ packets can increase as they propagate through the network. Moreover, if the route is sufficiently long, the routing information embodied in a packet can become large in relation to data carried by the packet. These issues compromise the scalability of DSR.

DSR is used in many performance comparisons, evaluation studies, and is used as the benchmark for other newer improved protocols. Furthermore, DSR is used as the reference protocol in investigations to obtain more general improvements in MANET performance (Broch *et al.*, 1998) (Tachtatzis and Harle, 2008).

Simulation Approach

Simulation packages are widely used to develop a simulation model. Such packages are particularly valuable in having the potential to reduce time to develop the model as a result of enhanced development, debugging and execution environment. Many packages also have excellent visualization feature. There are several well-known packages that can be used for the simulation of MANETs such as NS-2 (NS-2, 2012), GloMoSim (GloMoSim, 2012), QualNet (QualNet, 2012), OPNET (OPNET, 2012) and OMNeT++ (OMNeT++, 2012). Full comparisons between such packages are presented by (DiCaro, 2003), (Halvardsson and Lindberg, 2004) and (Begg et al., 2006). This work adopts the NS-2 as simulation platform for several reasons:

- It is widely utilized in the MANETs domain.
- It is an open source.
- It has no license cost.
- It provides both good manuals and tutorials.
- The source code can be compiled on different platforms, e.g. UNIX and Windows.
- Many wireless extensions have been contributed from the UCB Daedelus, the CMU Monarch projects and Sun Microsystems (DiCaro, 2003).
- New modifications can be easily included.

Implementation and simulation under NS-2 consists of 4 steps:

- Implement the protocol by adding a combination of C++ and OTcl (OTcl, 2012) code to NS-2's source base.
- Describe the simulation in an OTcl script.
- Start the NS-2 simulation engine.
- Analyze the generated trace files that can be used to do data processing (goodput, end-to-end delay) and to visualize the simulation with a program called Network Animator (NAM).

The NS-2 simulation process adopted for this work is summarized in **(Fig. 1)**.

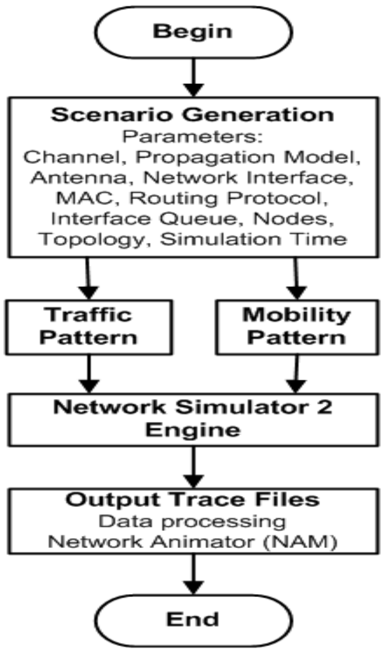

Fig. 1. NS-2 simulation process

Simulation Implementation

This section provides detailed descriptions of the network model, mobile node model, DSR routing process model state machine and mobility process model implemented in NS-2.

Network Model: **(Fig. 2)** shows a screenshot of NAM depicting the network model. The scenario contains 50 mobile nodes; a value common in the literature (Broch et al., 1998) (Tachtatzis and Harle, 2008) and used here for comparison. The nodes can move around the specified area and communicate over wireless links with a transmission range of 250 m.

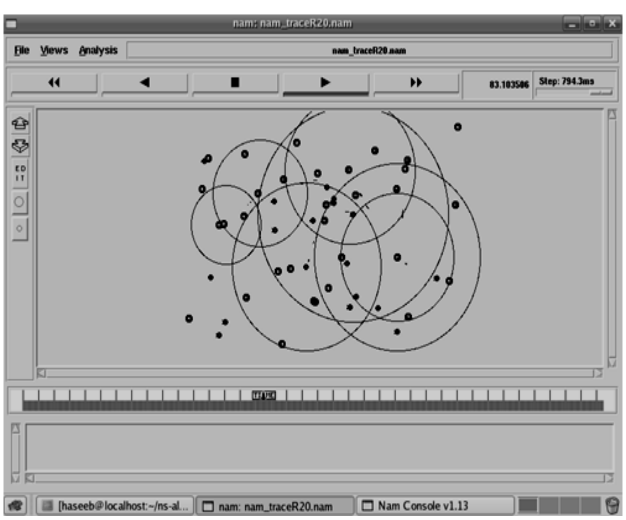

Fig. 2. Network model

Mobile Node Model: **(Fig. 3)** shows the DSR mobile node model (CMU, 1998), which simulates the protocol stack. Below are the details of modules that make up the DSR mobile node model:

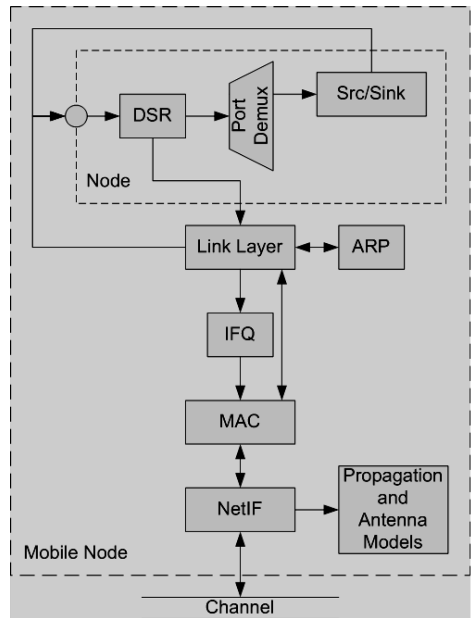

Fig. 3. Mobile node model

- *Src/Sink module:* Packets sent by the source (Src) on the mobile node are handled by the mobile node's entry point, which passes them to the DSR agent. Once generated, packets are sent to the immediate lower layers. On the other hand, the sink agent will only receive packets through the port demuliplexer (port demux), if this is the final destination.
- *DSR module:* This module is used to discover and maintain the routing information. Receiving a data packet from the node's entry point, the module first checks its cache. If there is a route path towards the destination node in cache, the module forwards the data packet to the next node. Otherwise, the module executes the DSR algorithm to discover a route path to destination node. The DSR process model is discussed in the next sub-section.
- *LL class and ARP module:* The link-layer (LL) class is responsible for simulating the data link protocols. The LL class uses an Address Resolution Protocol (ARP) to determine the hardware addresses of the neighboring nodes and map IP addresses to their correct interfaces.
- *IFQ class:* An Interface Queue (IFQ) class 'CMUPriQueue' (CMU, 1998) is used to queue all routing and data packets prior to being transmitted by MAC layer. The IFQ has a maximum size of 50 packets, maintaining a queue with two priorities each served in a first-in first-out (FIFO) order.

Routing packets are assigned a higher priority than data packets.

- *MAC module:* This module is an implementation of the IEEE 802.11 standard (IEEE 802.11, 2007) the protocol commonly used in the MANET modeling community, and an appropriate candidate protocol to be used in the proposed applications.
- *Propagation and antenna models:* The two-ray ground reflection model (Murthy and Manoj, 2004) and a unity gain omni-directional antenna is used in order to make the results obtained from this study comparable with the work of the other researchers who have adopted the same approach. The antennas of the transmitter and the receiver are placed at a height of 1.5 meters above the ground. The antenna parameters for type *Antenna/OmniAntenna* are summarized in **(Table 1)**.

Table 1. Antenna parameters

Parameter	Tcl class variable	Value
Position X (meters)	X_	0
Position Y (meters)	Y_	0
Position Z (meters)	Z_	1.5
Transmitting Antenna Gain	Gt_	1.0
Receiving Antenna Gain	Gr_	1.0

- *NetIF module:* The wireless network interface (NetIF) uses the characteristics of the 914MHz Lucent WaveLAN Direct Sequence Spread Spectrum (DSSS) radio (Tuch, 1993). WaveLAN is modeled as a shared-media radio with minimum range of 250m and nominal bit rate of 2Mbps. The configuration parameters that make the wireless interface *Phy/WirelessPhy* model the Lucent WaveLAN radio interface is given in **(Table 2)**.

Table 2. Wireless interface parameters

Parameter	Tcl class variable	Value
Raw bit rate (bps)	Rb_	2*1e6
Power of transmission (W)	Pt_	0.2818
Frequency (Hz)	freq_	914e+6
System loss factor	L_	1.0
Carrier sense threshold (W): min power required to detect another node's transmission	CSThresh_	1.559e-11
Receive threshold (W): min power required to receive a packet	RXThresh_	3.652e-10
Capture threshold (dBm): signal ratio required to maintain receiver capture of incoming packet in face of collision	CPThresh_	10.0

DSR Process Model State Machine: The DSR process **(Fig. 4)** implements the DSR algorithm. The role of each state in this finite state machine is described below (WCTG, 2001):

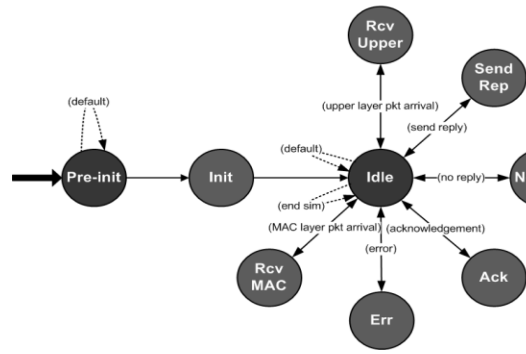

Fig. 4. DSR process model state machine

- *Pre-init State:* This state pre-initializes the DSR process model by managing the DSR address of the current node and by checking that this address is valid within the network.
- *Init State:* This state initializes every variable, statistic, cache and user parameter that is used by the DSR process model. Once the initialization setup is accomplished, the process transits to the idle state.
- *Idle State:* This is the default state where the process waits for an event.
- *Rcv Upper State:* This state handles every packet generated by the upper layer that the DSR protocol must carry to a given destination. The current state first extracts the destination address from the received packet, then checks the cache. If a route path exists in its cache, the current state forwards it to the next hop node towards destination. Otherwise, the current state saves the packets to a waiting queue and initiates a route discovery process by sending an RREQ packet.
- *Rcv MAC State:* This state handles every packet received from the IEEE 802.11 link layer. The packet is then processed according to its type: RREQ, RREP, data or RERR.
- *Send Rep:* This state occurs when the current (destination) node sends a scheduled RREP packet to the source node using the reverse path identified within the RREQ packet.
- *No Rep:* This state is called when the RREP wait timer expires. That means that the source node still did not receive a RREP to its RREQ. Thus, the previous step of the route discovery has failed and that a new request packet must be generated by the node. If the maximum number of retries is reached, the discovery process for that destination is aborted. Consequently, any data packets waiting for the route is dropped from the buffer.
- *Ack State:* This state handles every acknowledgement coming from the IEEE 802.11 link layer. An acknowledgement confirms that the current link used is valid, thus allowing the node to send any data packets through this link.

- *Err State:* This state occurs when the current node detects a link break to a neighboring node. It sends a RERR packet to its upstream node towards source.

Mobility Process Model: The mobility process model **(Fig. 5)** implements a random waypoint mobility model (Broch *et al.,* 1998) (Saad and Zukarnain, 2009). The random waypoint mobility model is widely used model when evaluating MANETs (Broch *et al.,* 1998) (Johnson *et al.,* 2007) (Perkins *et al.,* 2003) (Tachtatzis and Harle, 2008) and hence is considered in this work. Each state of the model is described below.

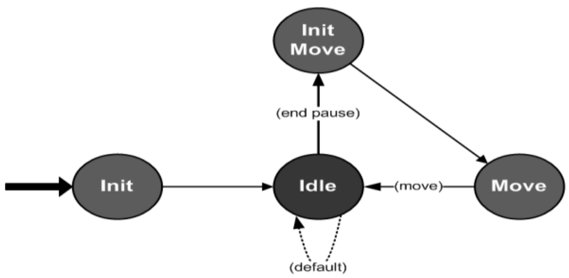

Fig.5. Mobility process model

- *Init State:* In the initialization state, each node picks a random location inside the simulation area.
- *Idle State:* This is the default state where the node remains stationary for 'pause time' seconds.
- *Init Move State:* In this state, the node selects a random destination in the simulation area and a random speed distributed uniformly between 0 and maximum speed.
- *Move State:* In this state, the node travels toward the chosen destination at the selected speed. Upon arrival, the node returns to the idle state and pauses for a specified time period before restarting the process.

3. RESULTS AND DISCUSSION

DSR Model Validation

For simulation results to be credible the simulation models in use must undergo verification and validation (Balci, 1997) (Carson, 2002) (Sargent, 2003). DSR model validation is carried out with the same simulation parameters as stated The simulation parameters are summarized in **(Table 3)**.

Table 3. Simulation Parameters

Parameter	Value
NS-2 Version	2.33
DSR Implementation	NS-2 Default
Transmitter range	250m
Nominal channel bandwidth	2Mbps
Simulation time	900sec
Number of nodes	50
Pause time	0, 30, 60, 120, 300, 600 and 900sec
Terrain size	1500x300m^2
Traffic type	CBR
Packet rate	4 packets/sec
Packet size	64 bytes
Number of sources	20
Maximum speed	20 m/s
Number of runs	10

(Table 4) shows the goodput and routing overhead (Zafar *et al.,* 2011). **(Fig. 6)** and **(Fig. 7)** depicted from **(Table 4)** represents the average of 10 random mobility and traffic scenarios for the given pause time, and the error bars represent the corresponding confidence interval of 95% (Zafar *et al.,* 2012). The results are at the same level when compared to the reported results as both simulations are conducted using same simulation package, i.e. NS-2. DSR delivers almost all data packets, regardless of pause time, with the goodput rising to 100% at pause time 900 (a stationary network). Similarly, the routing overhead is very small at pause time 900, rising only slowly as pause time decreases (as the average node mobility rate in the network increases). The slight difference in routing overhead can be due to the dynamic nature of the network underlying (and possibly undocumented) parameter settings and not the schemes being compared.

Table 4. Goodput and routing overhead as a function of pause time

Pause time (seconds)	Goodput (%)		Routing overhead (Packets)	
	DSR	DSR [Broch+]	DSR	DSR [Broch+]
0	99.548	98.2	26076	24500
30	99.628	98.3	26100	25000
60	99.692	98.5	24000	23000
120	98.83	98.6	17400	16500
300	99.78	98.1	7333.6	8000
600	99.92	99.5	4619.8	3800
900	100	100	544.8	800

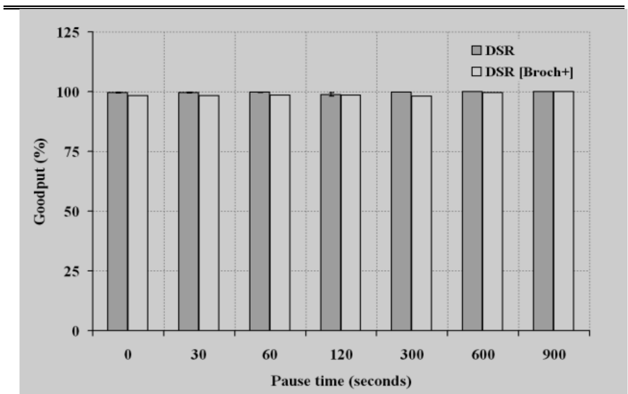

Fig. 6. Goodput as a function of pause time

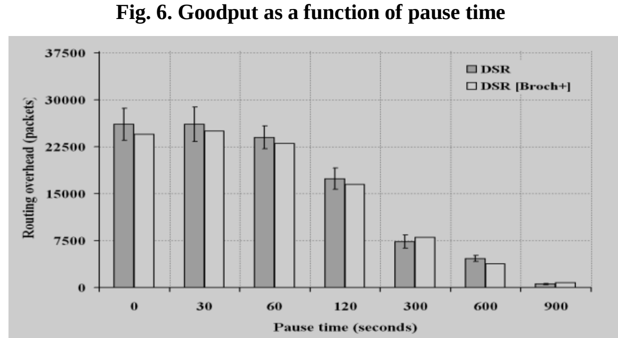

Fig. 7. Routing overhead as a function of pause time

4. CONCLUSION

The paper reviewed the DSR protocol implementation in NS-2. The network model, mobile node model, DSR routing process model state machine, mobility process model are described. The paper presented results of simulation-based experiments in order to ensure that the implementation of the DSR protocol inside the simulator is faithful to the protocol's specifications. DSR implementation is validated by comparing the simulation results with that of a known baseline.

REFERENCES:

Balci O. (1997) Verification, validation and accreditation of simulation models. Proc. Winter Simulation Conference: 135–141.

Begg L., W. Liu, K. Pawlikowski, S. Perera and H. Sirisena (2006) Survey of Simulators of Next Generation Networks for Studying Service Availability and Resilience. Technical Report, University of Canterbury, New Zealand.

Broch J., D.A. Maltz, D.B. Johnson, Y. Hu and J. Jetcheva (1998) A performance comparison of multi-hop wireless ad hoc network routing protocols. Proc. Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, Dallas, TX: 85–97.

Carson J.S. (2002) Model verification and validation. Proc. Winter Simulation Conference: 52–58.

CMU Monarch Project's Wireless and Mobility Extensions to NS (1998) http://www.monarch.cs.rice.edu/cmu-ns.html.

DiCaro G.A (2003) Analysis of simulation environments for mobile ad hoc networks. Technical Report, Dalle Molle Institute for Artificial Intelligence, Manno, Switzerland.

GloMoSim (Global Mobile Information Systems Simulation Library) (2012) http://pcl.cs.ucla.edu/projects/glomosim/.

Halvardsson M. and P. Lindberg (2004) Reliable Group Communication in Military Ad Hoc Network. Technical Report, Vaxjo University.

IEEE 802.11 (2007) IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

IETF MANET Working Group (2012) **Error! Hyperlink reference not valid.**.

Johnson D.B., Y. Hu and D.A. Maltz (2007) Dynamic Source Routing in Ad Hoc Wireless Networks. RFC 4728Pp.

Murthy C.S. and B.S. Manoj (2004) Ad hoc Wireless Networks Architectures and Protocols. Prentice Hall, NJ, USA.

NS-2 (The Network Simulator 2) (2012) http://www.isi.edu/nsnam/ns/.

OMNeT++ (Objective Modular Network Testbed in C++) Community Site (2012) http://www.omnetpp.org/.

OPNET (Optimized Network Engineering Tool) (2012) http://www.opnet.com/.

OTcl - Object Tcl Extensions (2012) http://web.mst.edu/~tk424/portal/teaching/EE401-IntComm/readingMaterials/OTcl-Object_Tcl_Extensions.pdf.

Perkins C.E., E.M. Royer and S. Das (2003) Ad hoc On-demand Distance Vector (AODV) Routing. RFC 3561.

QualNet, Scalable Network Technologies (2012) http://www.scalable-networks.com/.

Saad M. and Z. Zukarnain (2009) Performance Analysis of Random-Based Mobility Models in MANET Routing Protocol. European Journal of Scientific Research, 32(4): 444–454.

Sargent R.G. (2003) Verification and validation of simulation models. Proc. Winter Simulation Conference: 37–48.

Tachtatzis C. and D. Harle (2008) Performance Evaluation of Multi-path and Single-path Routing Protocols for Mobile Ad hoc Networks. Proc. 2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Edinburgh: 173–180.

Tuch B. (1993) Development of WaveLAN, an ISM Band Wireless LAN. AT&T Technical Journal, 72 **(4):** 27–33.

WCTG (2001) Simulation Model for the DSR MANET Routing Protocol. National Institute of Standards and Technology Group, National Institute of Standards and Technology, Gaithersburg, Maryland, **Error! Hyperlink reference not valid.**.

Zafar H., M. Zuhairi, D. Harle and I. Andonovic (2012) A Review of Techniques for the Analysis of Simulation Output. IETE Technical Review, 29**(3):** 223–228.

Zafar H., N. Alhamahmy, D. Harle and I. Andonovic (2011) Survey of Reactive and Hybrid Routing Protocols for Mobile Ad Hoc Networks. International Journal of Communication Networks and Information Security (IJCNIS), **(3):** 193–216.