
Barolli, A. [et al.] (2016) Performance analysis of WMNs by WMN-GA simulation system for different WMN architectures and TCP congestion-avoidance algorithms considering uniform distribution. *2016 19th International Conference on Network-Based Information Systems, NBiS 2016, Technical University of Ostrava, Ostrava, Czech Republic, 7-9 September 2016*. IEEE. Pp. 22-28. Doi: http://dx.doi.org/10.1109/NBiS.2016.18.

Barolli, A. [et al.] (2016) Performance analysis of WMNs by WMN-GA simulation system for different WMN architectures and TCP congestion-avoidance algorithms considering uniform distribution. *2016 19th International Conference on Network-Based Information Systems, NBiS 2016, Technical University of Ostrava, Ostrava, Czech Republic, 7-9 September 2016*. IEEE. Pp. 22-28. Doi: http://dx.doi.org/10.1109/NBiS.2016.18.

Performance Analysis of WMNs by WMN-GA Simulation System for Different WMN Architectures and TCP Congestion-Avoidance Algorithms Considering Uniform Distribution

Admir Barolli*, **Tetsuya Oda**†, **Ilir Shinko**‡, **Leonard Barolli**†, **Fatos Xhafa**§, **Makoto Takizawa**¶

**Department of Information Technology,*
Aleksander Moisiu University of Durres,
L.1, Rruga e Currilave, Durres, Albania
E-mail: admir.barolli@gmail.com

†*Department of Information and Communication Engineering,*
Fukuoka Institute of Technology (FIT),
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811–0295, Japan
Email: oda.tetsuya.fit@gmail.com, barolli@fit.ac.jp

‡*Faculty of Information Technologies*
Polytechnic University of Tirana,
Bul. "Dëshmorët e Kombit", "Mother Theresa" Square, Nr. 4, Tirana, Albania
E-mail: ishinko@fti.edu.al

§*Technical University of Catalonia*
Department of Languages and Informatics Systems
C/Jordi Girona 1-3, 08034 Barcelona, Spain
E-mail: fatos@lsi.upc.edu

¶*Hosei University,*
3-7-2, Kajino-Machi, Koganei-Shi, Tokyo 184–8584, Japan
Email: makoto.takizawa@computer.org

***Abstract*—In this paper, we evaluate the performance of two Wireless Mesh Networks (WMNs) architectures considering throughput, delay, jitter and fairness index metrics. For simulations, we used ns-3, Distributed Coordination Function (DCF) and Optimized Link State Routing (OLSR). We compare the performance for Transmission Control Protocol (TCP) Tahoe, Reno and NewReno for uniform distribution of mesh clients by sending multiple Constant Bit Rate (CBR) flows in the network. The simulation results show that for both WMN architectures, the PDR values of TCP congestion-avoidance algorithms are almost the same. For Hybrid WMN architecture, the throughput of TCP Reno is better than other algorithms. However, for I/B WMN, the throughput of TCP Tahoe is higher than other algorithms. The delay and jitter of TCP NewReno are a little bit lower compared with other algorithms. The I/B WMN architecture, the fairness index of TCP congestion-avoidance algorithms is almost the same.**

***Keywords*-Genetic Algorithms, Wireless Mesh Networks, NS-3, Network Architecture, OLSR, Multiple Flows, TCP Congestion-Avoidance Algorithm, SGC, NCMC, Uniform Distribution.**

I. INTRODUCTION

Wireless Mesh Networks (WMNs) [1] are important networking infrastructures. These networks are made up of wireless nodes, organized in a mesh topology, where mesh routers are interconnected by wireless links and provide Internet connectivity to mesh clients.

WMNs distinguish for their low cost nature that makes them attractive for providing wireless Internet connectivity. Moreover, such infrastructure can be used to deploy community networks, metropolitan area networks, municipal and, corporative networks, and to support applications for urban areas, medical, transport and surveillance systems.

The main issue of WMNs is to achieve network connectivity and stability as well as QoS in terms of user coverage. This problem is very closely related to the family of node placement problems in WMNs [2]–[5], among them, the mesh router mesh nodes placement. We consider the version of the mesh router nodes placement problem in which we are given a grid area where to deploy a number of mesh router nodes and a number of mesh client nodes of fixed positions (of an arbitrary distribution) in the grid area. The objective is to find a location assignment for the mesh routers to the cells of the grid area that maximizes the network connectivity and client coverage.

As node placement problems are known to be computationally hard to solve for most of the formulations [6], [7], Genetic Algorithms (GAs) has been recently investigated as effective resolution method.

In our previous work [8]–[10], we used mesh router nodes placement system that is based on Genetic Algorithms (GAs) to find an optimal location assignment for mesh routers in the grid area in order to maximize the network connectivity and client coverage.

In this work, we use the topology generated by WMN-GA system and evaluate by simulations the performance of uniform distribution of mesh clients considering TCP congestion-avoidance algorithms, two architectures and Distributed Coordination Function (DCF) protocol by sending multiple Constant Bit Rate (CBR) flows in the network. For simulations, we use ns-3 and Optimized Link State Routing (OLSR). As evaluation metrics we considered packet delivery ratio (PDR), throughput, delay, jitter and fairness.

The rest of the paper is organized as follows. Architectures of WMNs are presented in Section II. In Section III, we show the description and design of the simulation system. In Section IV, we discuss the simulation results. Finally, conclusions and future work are given in Section V.

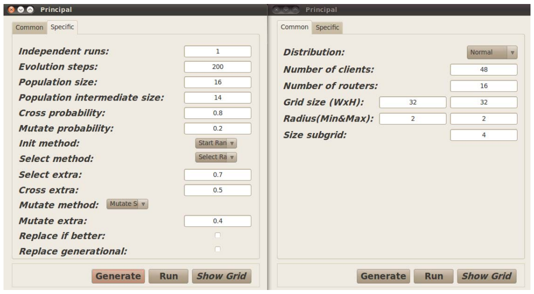
II. ARCHITECTURES OF WMNS

In this section, we describe the architectures of WMN. The architecture of the nodes in WMNs [11]–[14] can be classified according to the functionalities they offer as follows:

Infrastructure/Backbone WMNs: This type of architecture (also known as infrastructure meshing) is the most used and consists of a grid of mesh routers which are connected to different clients. Moreover, routers have gateway functionality thus allowing Internet access for clients. This architecture enables integration with other existing wireless networks and is widely used in neighboring communities.

Client WMNs: Client meshing architecture provides a communications network based on peer-to-peer over client devices (there is no the role of mesh router). In this case we have a network of mesh nodes which provide routing functionality and configuration as well as end-user applications, so that when a packet is sent from one node to another, the packet will jump from node to node in the mesh of nodes to reach the destination.

Hybrid WMNs: This architecture combines the two previous ones, so that mesh clients are able to access the network through mesh routers as well as through direct connection with other mesh clients. Benefiting from the advantages of the two architectures, Hybrid WMNs can connect to other networks (Internet, Wi-Fi, and sensor networks) and enhance the connectivity and coverage due to the fact that mesh clients can act as mesh routers.



Independent runs: 1
Evolution steps: 200
Population size: 16
Population intermediate size: 14
Cross probability: 0.8
Mutate probability: 0.2
Init method: Start Ran
Select method: Select Ra
Select extra: 0.7
Cross extra: 0.5
Mutate method: Mutate S
Mutate extra: 0.4
Replace if better: ☐
Replace generational: ☐

Distribution: Normal
Number of clients: 48
Number of routers: 16
Grid size (WxH): 32 32
Radius(Min&Max): 2 2
Size subgrid: 4

Figure 1. GUI tool for WMN-GA system.

III. SIMULATION SYSTEM DESCRIPTION AND DESIGN

A. GUI of WMN-GA System

The WMN-GA system can generate instances of the problem using different distributions of client and mesh routers.

The GUI interface of WMN-GA is shown in Fig. 1. The left site of the interface shows the GA parameters configuration and on the right side are shown the network configuration parameters.

For the network configuration, we use: distribution, number of clients, number of mesh routers, grid size, radius of transmission distance and the size of subgrid.

For the GA parameter configuration, we use: number of independent runs, GA evolution steps, population size, population intermediate size, crossover probability, mutation probability, initial methods, select method.

B. Positioning of mesh routers by WMN-GA system

We use WMN-GA system for node placement problem in WMNs. A bi-objective optimization is used to solve this problem by first maximizing the number of connected routers in the network and then the client coverage. The input parameters of WMN-GA system are shown in Table I. In Fig. 2, we show the location of mesh routers and clients for first generations and the optimized topologies generated by WMN-GA system for uniform distribution.

In Fig. 3 are shown the simulation results of Size of Giant Component (SGC) and Number of Covered Mesh Clients (NCMC) vs. number of generations. After few generations, all routers are connected with each other.

Then, we optimize the position of routers in order to cover as many mesh clients as possible. We consider uniform distribution of mesh clients. The simulation results of SGC and NCMC are shown in Table II.

C. Simulation Description

We conduct simulations using ns-3 simulator. The simulations in ns-3 are done for number of generations 1 and 200. The area size is considered 640 $[m]\times 640$ $[m]$ (or

Table I
INPUT PARAMETERS OF WMN-GA SYSTEM.

Parameters	Values
Number of clients	48
Number of routers	16, 24, 32
Grid width	32 [$units$]
Grid height	32 [$units$]
Independent runs	10
Number of generations	200
Population size	64
Selection method	Linear Ranking
Crossover rate	80 [%]
Mutate method	Single
Mutate rate	20 [%]
Distribution of clients	Uniform

Table II
EVALUATION OF WMN-GA SYSTEM.

Number of mesh routers	Uniform Distribution	
	SGC	NCMC
16	16	21
20	20	22
24	24	27
28	28	33
32	32	35

Table III
SIMULATION PARAMETERS FOR NS-3.

Parameters	Values
Area Size	640 [m]×640 [m]
Distributions of mesh clients	Uniform distribution
Number of mesh routers	16
Number of mesh clients	48
PHY protocol	IEEE 802.11a
Propagation loss model	Log-distance Path Loss Model
Propagation delay model	Constant Speed Model
MAC protocols	DCF
Maximum queue size	400
Routing protocol	OLSR
Transport protocol	TCP
TCP version	Tahoe, Reno, NewReno
Application type	CBR
Packet size	1024 [$Bytes$]
Number of source nodes	10
Number of destination node	1
Transmission current	17.4 [mA]
Receiving current	19.7 [mA]
Simulation time	600 [sec]

32 [$units$]×32 [$units$]) and the number of mesh routers is from 16 to 32. We used DCF, OLSR protocols and sent multiple CBR flows over different TCP congestion-avoidance algorithms. The pairs source-destination are the same for all simulation scenarios. Log-distance path loss model and constant speed delay model are used for the simulation and other parameters are shown in Table III.

D. NS-3

The ns-3 simulator [15] is developed and distributed completely in the C++ programming language, because it better facilitated the inclusion of C-based implementation code. The ns-3 architecture is similar to Linux computers, with internal interface and application interfaces such as network interfaces, device drivers and sockets. The goals of ns-3 are set very high: to create a new network simulator aligned with modern research needs and develop it in an open source community. Users of ns-3 are free to write their simulation scripts as either *C++ main()* programs or *Python* programs. The ns-3's low-level API is oriented towards the power-user but more accessible "helper" APIs are overlaid on top of the low-level API.

In order to achieve scalability of a very large number of simulated network elements, the ns-3 simulation tools also support distributed simulation. The ns-3 support standardized output formats for trace data, such as the pcap format used by network packet analyzing tools such as tcpdump, and a standardized input format such as importing mobility trace files from ns-2 [16].

The ns-3 simulator is equipped with *Pyviz* visualizer, which has been integrated into mainline ns-3, starting with version 3.10. It can be most useful for debugging purposes, i.e. to figure out if mobility models are what you expect, where packets are being dropped. It is mostly written in Python and it works both with Python and pure C++ simulations. The function of ns-3 visualizer is more powerful than network animator (*nam*) of ns-2 simulator.

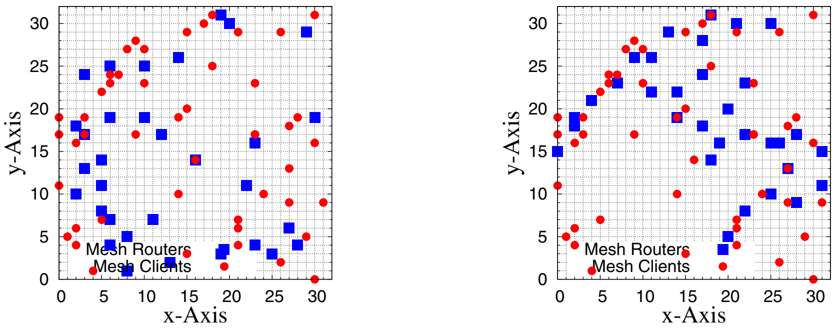
The ns-3 simulator has models for all network elements that comprise a computer network. For example, network devices represent the physical device that connects a node to the communication channel. This might be a simple Ethernet network interface card or a more complex wireless IEEE 802.11 device.

The ns-3 is intended as an eventual replacement for popular ns-2 simulator. The ns-3's wifi models a wireless network interface controller based on the IEEE 802.11 standard [17]. The ns-3 provides models for these aspects of 802.11:

1) Basic 802.11 DCF with infrastructure and ad hoc modes.
2) 802.11a, 802.11b, 802.11g and 802.11s physical layers.
3) QoS-based EDCA and queueing extensions of 802.11e.
4) Various propagation loss models including Nakagami, Rayleigh, Friis, LogDistance, FixedRss, and so on.
5) Two propagation delay models, a distance-based and random model.
6) Various rate control algorithms including Aarf, Arf, Cara, Onoe, Rraa, ConstantRate, and Minstrel.

E. Overview of DCF Protocol

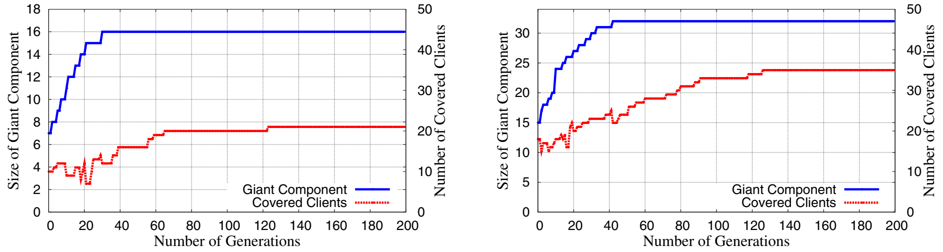
DCF is a random access scheme based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) scheme. A legacy DCF station with a packet to send will first sense the medium for activity. If the channel is idle for a Distributed Inter-Frame Space (DIFS), the station



(a) Number of generations: 1 (12, 15)

(b) Number of generations: 200 (32, 35)

Figure 2. (m, n): m is SGC, n is NCMC for uniform distribution.



(a) Number of mesh routers: 16

(b) Number of mesh routers: 32

Figure 3. SGC and NCMC vs. number of generations for uniform distribution.

will attempt to transmit after a random back-off period. This period is referred as the Contention Window (CW). The value for the CW is chosen randomly from a range $[0, 2^n - 1]$, i.e.

$$CW_{min} \leq CW \leq CW_{max} \tag{1}$$

where n is PHY dependent. Initially, CW is set to the minimum number of slot times CW_{min}, which is defined per PHY in microseconds [18]. The randomly chosen CW value, referred as the back-off counter, is decreased each slot time if the medium remains idle. If during any period the medium becomes busy, the back-off counter is paused and resumed only when the medium becomes idle. On reaching zero, the station transmits the packet in the physical channel and awaits an acknowledgment (ACK). The transmitting station then performs a post back-off, where the back-off procedure is repeated once more. This is to allow other stations to gain access to the medium during heavy contention.

If the ACK is not received within a Short Inter-Frame Space (SIFS), it assumes that the frame was lost due to collision or being damaged. The CW value is then increased exponentially and the back-off begins once again for retransmission. This is referred as the Automatic Repeat Request (ARQ) process. If the following retransmission attempt fails, the CW is again increased exponentially, up until the limit CW_{max}. The retransmission process will repeat for up to 4 or 7 times, depending on whether the short retry limit or long retry limit is used. Upon reaching the retry limit the packet is considered lost and discarded. The retry limit is manufacturer dependent and can vary considerably.

F. Overview of OLSR Routing Protocol

The OLSR protocol [19] is a pro-active routing protocol, which builds up a route for data transmission by maintaining a routing table inside every node of the network. The routing table is computed upon the knowledge of topology information, which is exchanged by means of Topology Control (TC) packets.

OLSR makes use of HELLO messages to find its one hop neighbours and its two hop neighbours through their responses. The sender can then select its Multi Point Relays (MPR) based on the one hop node which offer the best routes to the two hop nodes. By this way, the amount of control traffic can be reduced. Each node has also an MPR selector set which enumerates nodes that have selected it as an MPR node. OLSR uses TC messages along with MPR forwarding to disseminate neighbour information throughout the network. Host Network Address (HNA) messages are used by OLSR to disseminate network route advertisements in the same way TC messages advertise host routes.

G. Overview of TCP Congestion-Avoidance Algorithms

TCP is transport layer is the reliable connection orientated protocol that provides reliable transfer of data between the nodes [20]. It ensures that the data is reached the destination correctly without any loss or damage. The data is transmitted in the form of continuous stream of octets. The reliable transfer of octets is achieved through the use of a sequence number to each octet. Another aspect of TCP is the tree way handshakes mechanism to establish a connection between the nodes [21]. Furthermore, TCP uses the port assignment as an addressing mechanism to differentiate each connection for the cases of more TCP connection between nodes are required. After the introduction of first version of TCP several different TCP variants exist. The most famous implementation of TCP are Tahoe, Reno and NewReno.

Modern TCP implementations contain a number of algorithms aimed at controlling network congestion while maintaining good user throughput. Early TCP implementations followed a go-back model using cumulative positive acknowledgment and requiring a retransmission timer expiration to re-send data lost during transport. These TCPs did little to minimize network congestion. In our study we concentrate on three TCP congestion-avoidance algorithms [21]: TCP Tahoe [22], TCP Reno [23] and TCP NewReno [24].

1) TCP Tahoe: The Tahoe TCP implementation added a number of new algorithms and refinements to earlier implementations. The new algorithms include Slow-Start, Congestion Avoidance, and Fast Retransmit [22]. The refinements include a modification to the round-trip time estimator used to set retransmission timeout values. All modifications have been described elsewhere [22], [25]. The Fast Retransmit algorithm is of special interest in this paper because it is modified in subsequent versions of TCP. With Fast Retransmit, after receiving a small number of duplicate acknowledgments for the same TCP segment (dup ACKs), the data sender infers that a packet has been lost and retransmits the packet without waiting for a retransmission timer to expire, leading to higher channel utilization and connection throughput.

2) TCP Reno: TCP Reno retains the basic principle of Tahoe, such as slow starts and the coarse grain retransmit timer. However it adds some intelligence over it so that lost packets are detected earlier and the pipeline is not emptied every time a packet is lost [26] Reno requires that we receive immediate acknowledgement whenever a segment is received. The logic behind this is that whenever we receive a duplicate acknowledgment, then his duplicate acknowledgment could have been received if the next segment in sequence expected, has been delayed in the network and the segments reached there out of order or else that the packet is lost. If we receive a number of duplicate acknowledgements then that means that sufficient time have passed and even if the segment had taken a longer path, it should have gotten to the receiver by now [27]. There is a very high probability that it was lost. So Reno suggests an algorithm called “Fast Retransmit” [28].

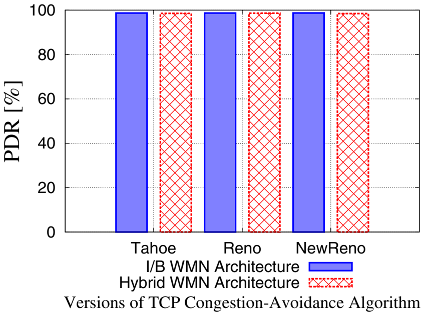


Figure 4. Results of average PDR.

3) TCP NewReno: NewReno is a slight modification over TCP Reno. It is able to detect multiple packet losses and thus is much more efficient that Reno in the event of multiple packet losses. Like Reno, NewReno also enters into fast-retransmit when it receives multiple duplicate packets, however it differs from Reno in that it does not exit fast-recovery until all the data which was out standing at the time it entered fast recovery is acknowledged. Thus it overcomes the problem faced by Reno of reducing the cwnd multiples times. The fast-transmit phase is the same as in Reno. The difference in the fast recovery phase which allows for multiple re-transmissions in NewReno. Whenever NewReno enters fast recovery it notes the maximums segment which is outstanding. The fast-recovery phase proceeds as in Reno, however when a fresh ACK is received then there are two cases: If it ACK’s all the segments which were outstanding when we entered fast recovery then it exits fast recovery and sets cwnd to ssthresh and continues congestion avoidance like Tahoe. If the ACK is a partial ACK then it deduces that the next segment in line was lost and it re-transmits that segment and sets the number of duplicate ACKS received to zero. It exits fast recovery when all the data in the window are acknowledged [24].

IV. SIMULATION RESULTS

We used the PDR, throughput, delay, jitter and fairness metrics to evaluate the performance of WMNs for two architectures considering TCP congestion-avoidance algorithms and uniform distribution of mesh clients.

In Fig. 4, we show the simulation results of PDR. For both

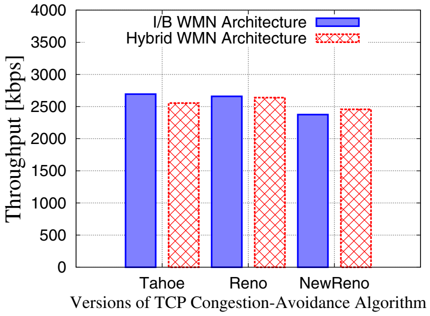


Figure 5. Results of average throughput.

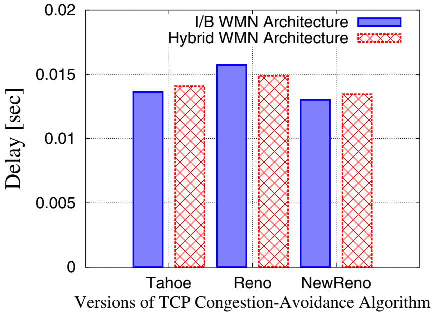


Figure 6. Results of average delay.

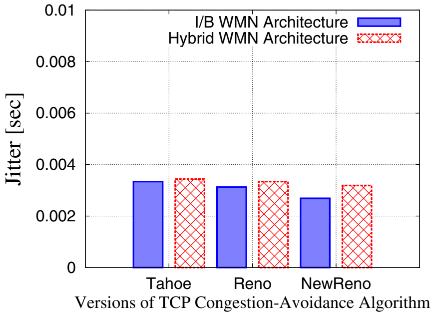


Figure 7. Results of average jitter.

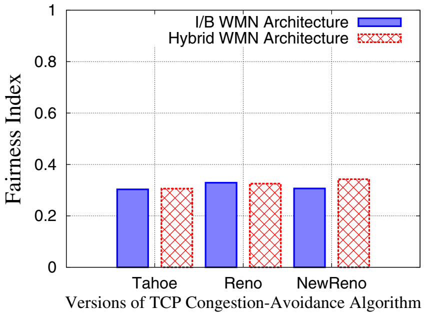


Figure 8. Results of fairness index.

WMN architectures, the PDR values of TCP congestion-avoidance algorithms are almost the same.

In Fig. 5, we show the simulation results of throughput. For Hybrid WMN architecture, the throughput of TCP Reno is better than other algorithms. However, for I/B WMN, the throughput of TCP Tahoe is higher than other algorithms.

In Fig. 6 and Fig. 7, the delay and jitter of TCP NewReno are a little bit lower compared with other algorithms.

In Fig. 8, we show the fairness index. For I/B WMN architecture, the fairness index of TCP congestion-avoidance algorithms is almost the same. However, for Hybrid WMN, the fairness index of TCP NewReno is higher than other algorithms.

V. CONCLUSIONS

In this work, we presented WMN-GA system and applied it for node placement problem in WMNs. We evaluated the performance of WMNs by WMN-GA simulation system for uniform distribution of mesh clients considering DCF, OLSR and different TCP congestion-avoidance algorithms.

From the simulations we found that:

- For both WMN architectures, the PDR values of TCP congestion-avoidance algorithms are almost the same.
- For Hybrid WMN architecture, the throughput of TCP Reno is better than other algorithms. However, for I/B WMN, the throughput of TCP Tahoe is higher than other algorithms.
- The delay and jitter of TCP NewReno are a little bit lower compared with other algorithms.
- For I/B WMN architecture, the fairness index of TCP congestion-avoidance algorithms is almost the same.

In the future work, we would like to implement other systems and compare the performance with the proposed system.

REFERENCES

[1] I. F. Akyildiz, X. Wang, W. Wang, "Wireless Mesh Networks: A Survey", In Computer Networks, Vol. 47, No. 4, pp. 445-487, 2005.

[2] A. Franklin, C. Murthy "Node Placement Algorithm for Deployment of Two-Tier Wireless Mesh Networks", Proc. of IEEE GLOBECOM-2007, pp. 4823-4827, 2007.

[3] S. N. Muthaiah and C. Rosenberg, "Single Gateway Placement in Wireless Mesh Networks", In Proc. of 8th International IEEE Symposium on Computer Networks, Turkey, pp. 4754-4759, 2008.

[4] M. Tang, "Gateways Placement in Backbone Wireless Mesh Networks", International Journal of Communications, Network and System Sciences, Vol. 2, No.1, pp. 45-50, 2009.

[5] T. Vanhatupa, M. Hännikäinen and T.D. Hämäläinen, "Genetic Algorithm to Optimize Node Placement and Configuration for WLAN Planning", In Proc. of 4th International Symposium on Wireless Communication Systems, pp. 612-616, 2007.

[6] A. Lim, B. Rodrigues, F. Wang and Zh. Xua, "k-Center Problems with Minimum Coverage", Theoretical Computer Science, Vol. 332, No. 1-3, pp. 1-17, 2005.

[7] J. Wang, B. Xie, K. Cai and D. P. Agrawal, "Efficient Mesh Router Placement in Wireless Mesh Networks", MASS, Pisa, Italy, pp. 9-11, 2007.

[8] T. Oda, A. Barolli, F. Xhafa, L. Barolli, M. Ikeda, M. Takizawa, "WMN-GA: A Simulation System for WMNs and Its Evaluation Considering Selection Operators", Journal of Ambient Intelligence and Humanized Computing (JAIHC), Springer, Vol. 4, No. 3, pp. 323-330, June 2013

[9] M. Ikeda, T. Oda, E. Kulla, M. Hiyama, L. Barolli and M. Younas, "Performance Evaluation of WMN Considering Number of Connections Using NS-3 Simulator", The Third International Workshop on Methods, Analysis and Protocols for Wireless Communication (MAPWC 2012), pp. 498-502, Victoria, Canada, November 12-14, 2012.

[10] T. Oda, D. Elmazi, A. Barolli, S. Sakamoto, L. Barolli, F. Xhafa, "A Genetic Algorithm Based System for Wireless Mesh Networks: Analysis of System Data Considering Different Routing Protocols and Architectures", Journal of Soft Computing (SOCO), Springer, Published online: 31 March 2015, DOI: 10.1007/s00500-015-1663-z, pp. 1-14, 2015.

[11] F. Xhafa, C. Sanchez, and L. Barolli, "Locals Search Algorithms for Efficient Router Nodes Placement in Wireless Mesh Networks", in International Conference on Network-Based Information Systems (NBiS), pp. 572-579, 2009.

[12] T. Oda, A. Barolli, E. Spaho, L. Barolli, F. Xhafa, "Analysis of Mesh Router Placement in Wireless Mesh Networks Using Friedman Test", Proc. of The 28th IEEE International Conference on Advanced Information Networking and Applications (IEEE AINA), pp. 289-296, Victoria, Canada, May 2014,

[13] T. Oda, S. Sakamoto, A. Barolli, M. Ikeda, L. Barolli, F. Xhafa, "A GA-Based Simulation System for WMNs: Performance Analysis for Different WMN Architectures Considering TCP", 2014 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), pp. 120-126, Guangzhou, China, November 2014.

[14] T. Oda, A. Barolli, E. Spaho, F. Xhafa, L. Barolli, M. Takizawa, "Evaluation of WMN-GA for Different Mutation Operators", International Journal of Space-Based and Situated Computing (IJSSC), Inderscience, Vol. 2. No. 3, pp. 149-157, 2012.

[15] "ns-3", https://www.nsnam.org/.

[16] "The Network Simulator-ns-2", http://www.isi.edu/nsnam/ns/.

[17] IEEE 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Computer Society Std., 2007, online available: http://standards.ieee.org/getieee802/download/802.11-2007.pdf.

[18] IEEE-SA, "IEEE 802.11 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999.

[19] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", RFC 3626 (Experimental), 2003.

[20] R. Kaur, G. S. Josan, "Performance Evaluation Of Congestion Control Tcp Variants In Vanet Using Omnet++", International Journal of Engineering Research and Applications (IJERA), Vol. 2, No. 5, pp. 1682-1688, 2012.

[21] K. Fall, S. Floyd, "Simulation-based comparisons of Tahoe, Reno and SACK TCP", ACM SIGCOMM Computer Communication Review, Vol. 26, No. 3, pp. 5-21, 1996.

[22] V. Jacobson, "Congestion Avoidance and Control", SIGCOMM '88, pp. 314-329, 1988.

[23] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control", RFC 2581, 1999.

[24] T. Henderson, S. Floyd, A. Gurtov, Y. Nishida, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 6582, 2012.

[25] W. R. Stevens, "TCP/IP Illustrated, Vol. 1: The Protocols", Addison Wesley, 1994.

[26] L. Subedi, M. Najiminaini, and L. Trajkovi, "Performance Evaluation of TCP Tahoe, Reno, Reno with SACK, and New Reno Using OPNET Modeler", Simon Fraser University Vancouver, British Columbia Canada, 2004.

[27] F. Anjum and L. Tassiulas, "Comparative study of various TCP versions over a wireless link with correlated losses", IEEE/ACM Transactions on Networking, Vol. 11, No. 3, pp. 370-383, 2003.

[28] V. Jacobson, "Modified TCP Congestion Avoidance Algorithm", ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt, 1990.