Article publicat / Published article:

Barolli, A. [et al.] (2016) Interface and results visualization of WMN-GA simulation system: evaluation for exponential and Weibull distributions considering different transmission rates. "Computer standards & interfaces". Vol. 44. pp.150-158. Doi: 10.1016/j.csi.2015.04.003

Interface and Results Visualization of WMN-GA Simulation System: Evaluation for Exponential and Weibull Distributions Considering Different Transmission Rates

Admir Barolli, Vincenzo Loia

DI - University of Salerno, 84084, Fisciano, Italy

Tetsuya Oda

Graduate School of Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811–0295, Japan

Leonard Barolli

Department of Information and Communication Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811–0295, Japan

Fatos Xhafa

Technical University of Catalonia, Department of Languages and Informatics Systems, C/Jordi Girona 1-3, 08034 Barcelona, Spain

Makoto Takizawa

Hosei University, 3-7-2 Kajino-Machi, Koganei-Shi, Tokyo 184-8584, Japan

Abstract

Wireless Mesh Networks (WMNs) are currently attracting a lot of attention due to their ability to provide cost-efficient broadband wireless connectivity. In this paper, we present the interface of WMN-GA system, which is based on Genetic Algorithms (GAs). We evaluate the performance of WMN-GA simulation system for Exponential and Weibull distributions considering different transmission rates. We present the visualization of the simulation results for

[*]Corresponding author

Email addresses: admir.barolli@gmail.com, loia@unisa.it (Admir Barolli, Vincenzo Loia), oda.tetsuya.fit@gmail.com (Tetsuya Oda), barolli@fit.ac.jp (Leonard Barolli), fatos@lsi.upc.edu (Fatos Xhafa), makoto.takizawa@computer.org (Makoto Takizawa)

different generations. As evaluation parameters, we consider the Packet Delivery Ratio (PDR), throughput and delay metrics. For simulations, we used ns-3 simulator and Hybrid Wireless Mesh Protocol (HWMP). We evaluate the performance of WMNs for different distributions by sending multiple Constant Bit Rate (CBR) flows and different transmission rates in the network. From simulation results, we found that for 10, 20 and 30 number of connections, the PDR is less than 60% when the transmission rate is more than 1200 kbps for Exponential distribution. The PDR for Weibull distribution is higher than Exponential distribution. For different number of connections, the throughput is increased with the increase of transmission rate. The throughput of Exponential distribution is higher than Weibull distribution. With increasing of number of connections and transmission rate, the delay is increased. For 10 connections, the delay is very small until the transmission rate is 800 kbps. The delay of Exponential distribution is smaller than Weibull distribution.

Keywords: Interface; Visualization; Wireless Mesh Network; HWMP; GA; ns-3; Simulation.

---

1. Introduction

The Wireless Mesh Networks (WMNs) are currently attracting a lot of attention due to their importance for providing cost-efficient broadband wireless connectivity. The WMNs can be seen as a special type of wireless ad-hoc networks.

WMNs are based on mesh topology, in which every node (representing a server) is connected through wireless links to one or more nodes, enabling thus the information transmission in more than one path. The path redundancy is a robust feature of mesh topology. Compared to other topologies, mesh topology does not need a central node, allowing networks based on it to be self-healing. These characteristics of networks with mesh topology make them very reliable and robust networks to potential server node failures.

There are a number of application scenarios for which the use of WMNs is a very good alternative to offer connectivity at a low cost. It should also mentioned that there are applications of WMNs which are not supported directly by other types of wireless networks such as cellular networks, ad hoc networks, wireless sensor networks and standard IEEE 802.11 networks. There are many applications of WMNs in Neighboring Community Networks, Corporative Networks, Metropolitan Area Networks, Transportation Systems, Automatic Control Buildings, Medical and Health Systems, Surveillance and so on.

In WMNs, the mesh routers provide network connectivity services to mesh client nodes. The good performance and operability of WMNs largely depends on placement of mesh routers nodes in the geographical deployment area to achieve network connectivity, stability and client coverage.

In this work, we present the interface of WMN-GA system, which is based on Genetic Algorithms (GAs). We evaluate the performance of WMN-GA simulation system for Exponential and Weibull distributions considering different transmission rates. We present the visualization of the simulation results for different generations. As evaluation parameters, we consider the Packet Delivery Ratio (PDR), throughput and delay metrics. For simulations, we use ns-3 simulator and Hybrid Wireless Mesh Protocol (HWMP).

The structure of the paper is as follows. In Section 2, we discuss the related work. In Section 3, we make an overview of HWMP routing protocol. In Section 4, we present the implemented WMN-GA simulation system. In Section 5, we show the description od ns-3 and path loss model. In Section 6, we show the simulation results. Finally, conclusions and future work are given in Section 7.

2. Related Work

Until now, many researchers performed valuable research in the area of multihop wireless networks by computer simulations and experiments [1]. Most of them are focused on throughput improvement and they do not consider mobility [2].

Several heuristic approaches are found in the literature for node placement problems in WMNs [3, 4, 5].

In [3], the authors investigate the role of gateway placement on network throughput for realistic configurations of WMNs. They show that the position of the gateway significantly bears on network throughput. It is hence important to optimize its placement. The authors propose several heuristics to optimally position a single gateway in WMN and compare their relative performance in terms of network throughput with respect to the exact solution, which is obtained through cumbersome computations.

In [4], the author presents an algorithm for the gateway placement problem in Backbone Wireless Mesh Networks (BWMNs). Different from existing algorithms, the proposed algorithm incrementally identifies gateways and assigns mesh routers to identified gateways. The algorithm can guarantee to find a feasible gateway placement satisfying Quality-of-Service (QoS) constraints, including delay constraint, relay load constraint and gateway capacity constraint. Experimental results show that its performance is as good as that of the best of existing algorithms for the gateway placement problem. But, the proposed algorithm can be used for BWMNs that do not form one connected component, and it is easy to implement and use.

In [5], the authors deal with the deployment of Mesh Nodes (MNs) in WMNs. They show that because it is difficult to place the MNs in a regular pattern in the real deployment, finding the optimal locations in the deployment environment is of much interest for the service providers. For a given possible locations for the MNs and the user density in the deployment environment, they aim to find the locations of the MNs to be used that maximizes the coverage and the connectivity of the network together. Due to high computational complexity of the exhaustive searching algorithm, an efficient local searching algorithm is proposed. Numerical results show that, the local search algorithm can give close to optimal performance with much lower time complexity than exhaustive searching.

As node placement problems are known to be computationally hard to solve for most of the formulations [6, 7], GAs have been recently investigated as effective resolution methods. However, GAs require the user to provide values for a number of parameters and a set of genetic operators to achieve the best GA performance for the problem [8, 9, 10].

In our previous work [11, 12, 13], we considered the application of GAs for scheduling and resource allocation in grid systems, and the mesh router nodes placement in WMNs. In [11, 12], we considered the tuning struggle strategy in genetic algorithms for scheduling in computational grids and carried out an experimental study on GAs for resource allocation on grid systems. In [13], we proposed and evaluated the GAs for near-optimally solving problems. We considered two-fold optimization: the maximization of the size of the giant component in the network and the user coverage. Several GA operators have been considered in implementing GAs in order to find the configuration that works best for the problem. We have experimentally evaluated the proposed GAs using a benchmark of generated instances varying from small to large size. The experimental results showed the efficiency of the GAs for computing high quality solutions of mesh router nodes placement in WMNs.

3. Overview of HWMP Routing Protocol

The IEEE 802.11s draft defines a default routing protocol called the Hybrid Wireless Mesh Protocol (HWMP). Every IEEE 802.11s compliant device is required to implement HWMP and to be capable of using it. HWMP is located on layer 2, this means, it uses MAC addresses.

The nodes of a WMN are called Mesh Points (MPs) in IEEE 802.11s. A MP is an IEEE 802.11 station that has mesh capabilities in addition to the basic station functionality. This means that it can participate in the mesh routing protocol and can forward data frames on behalf of other MPs according to the IEEE 802.11s standard. The MPs can be end customer devices such as laptops as well as infrastructure devices such as Access Points (APs).

The MPs with additional AP functionality are called Mesh AP (MAPs). Conventional WLAN clients, which are non-mesh IEEE 802.11 stations (STAs), can connect through the MAPs to the WMN. The MPs with additional portal functionality are called Mesh Portals Points (MPPs). They can bridge data frames to other IEEE 802 networks, especially to a wired network such as an Ethernet.

The IEEE 802.11s WMNs will be applicable to a large variety of usage scenarios [14]. The four most important usage scenarios are:

- residential for wireless home networks;
- office for wireless networks in office environments;
- campus/community/public access for wireless backhaul meshes for Internet access;
- public safety for flexible and fast setup of wireless communications for emergency staff.

The HWMP is a hybrid routing protocol. It has both reactive components and proactive components. The foundation of HWMP is an adaptation of AODV [15] to radio-aware link metrics and MAC addresses. It is the basic, reactive component of HWMP. The on-demand path setup is achieved by a path discovery mechanism that is very similar to the one of AODV. If a MP needs a path to a destination, it broadcasts a Path REQuest message (PREQ) into the WMN. The MPs will rebroadcast the updated PREQ whenever the received PREQ corresponds to a newer or better path to the source. Similarly, the requested destination MP will respond with a Path REPly message (PREP) whenever a received PREQ corresponds to a newer or better path to the source. Intermediate MPs that have already a valid path to the requested destination, can respond with a PREP, if the Destination Only flag (DO flag) is not set. Depending on the new Reply and Forward flag (RF flag), they can also rebroadcast the updated PREQ. This will result in a current path metric in addition to the fast path discovery.

The proactive component of HWMP is the extension with a proactive routing tree to specially designated MPs. Any MP that is configured to be a root MP, will periodically broadcast proactive PREQ messages or Root ANNouncement messages (RANNs) into the WMN, which will create and maintain a tree of paths to the root MP. There are three different, configurable mechanisms for the proactive tree-building available in HWMP.

4. Implemented WMN-GA Simulation System

The proposed and implemented WMN-GA system is based on GA. In this Section, we present briefly GA and then the proposed WMN-GA simulation system.

4.1. Genetic Algorithm

GAs have shown their usefulness for the resolution of many computationally hard combinatorial optimization problems. They are, of course, a strong candidate for efficiently solving mesh router nodes placement problem in WMNs. For the purpose of this work we have used the template given in Algorithm 1.

As can be seen from the template, several parameters intervene in the GAs: population size, intermediate population size, number of evolution steps, crossover probability, mutation probability and parameters for replacement strategies. On the other hand, there are the (families of) genetic operators: crossover operators, mutation operators, selection operators and replacement operators. As there are potentially large range values for parameters and different versions of operators, their tuning becomes crucial to the GA's performance.

4.1.1. Selection Operators

In the evolutionary computing literature we can find a variety of selection operators, which are in charge of selecting individuals for the pool mate. The operators considered in this work are those based on *Implicit Fitness Re-mapping* technique. It should be noted that selection operators are generic ones and do not depend on the encoding of individuals.

- *Random Selection*: This operator chooses the individuals uniformly at random. The problem is that a simple strategy does not consider even the

Algorithm 1 GA Template

Generate the initial population P^0 of size μ; $t = 0$.

Evaluate P^0;

while not termination-condition **do**

Select the parental pool T^t of size λ;

$T^t := Select(P^t)$;

Perform crossover procedure on pairs of individuals in T^t with probability p_c; $P_c^t := Cross(T^t)$;

Perform mutation procedure on individuals in P_c^t with probability p_m; $P_m^t := Mutate(P_c^t)$;

Evaluate P_m^t ;

Create a new population P^{t+1} of size μ from individuals in P^t and/or P_m^t ; $P^{t+1} := Replace(P^t; P_m^t)$

$t := t + 1$;

end while

return Best found individual as solution;

fitness value of individuals and this may lead to a slow convergence of the algorithm.

- *Best Selection*: This operator selects the individuals in the population having higher fitness value. The main drawback of this operator is that by always choosing the best fitted individuals of the population, the GA converges prematurely.

- *Linear Ranking Selection*: This operator follows the strategy of selecting the individuals in the population with a probability directly proportional to its fitness value. This operator clearly benefits the selection of best endowed individuals, which have larger chances of being selected.

- *Exponential Ranking Selection*: This operator is similar to Linear Ranking but now probabilities of ranked individuals are weighted according to an exponential distribution.

- *Tournament Selection*: This operator selects the individuals based on the result of a tournament among individuals. Usually winning solutions are the ones of better fitness value but individuals of worse fitness value could be chosen as well, contributing thus to avoiding premature convergence. Particular cases of this operator are the *Binary Tournament* and $N-$*Tournament Selection*, for different values of N.

4.1.2. Crossover Operators

The crossover operators are the most important ingredient of GAs. Indeed, by selecting individuals from the parental generation and interchanging their *genes*, new individuals (descendants) are obtained. The aim is to obtain descendants of better quality that will feed the next generation and enable the search to explore new regions of solution space not explored yet.

There exist many types of crossover operators explored in the evolutionary computing literature. It is very important to stress that crossover operators depend on the chromosome representation. This observation is especially important for the mesh router nodes problem, since in our case, instead of having strings we have a grid of nodes located in a certain positions. The crossover operator should thus take into account the specifics of mesh router nodes encoding. We have considered the following crossover operators, called *intersection operators* (denoted `CrossRegion`, hereafter), which take in input two individuals and produce in output two new individuals (see Algorithm 2).

4.1.3. Mutation Operators

The mutation operator is crucial for preventing the search from getting stuck in local optima by doing small local perturbations to the individuals of the population. Again, the definition of the mutation operators is specific to encoding of the individuals of the concrete problem under study. We defined thus several specific mutation operators as follows.

- *SingleMutate*: Select a mesh router node in the grid area and move it to another cell of the grid area. After the move is done, network connections

Algorithm 2 Crossover Operator

1: **Input**: Two parent individuals P_1 and P_2; values H_g and W_g for height and width of a small grid area;

2: **Output**: Two offsprings O_1 and O_2;

3: Select at random a $H_g \times W_g$ rectangle RP_1 in parent P_1. Let RP_2 be the same rectangle in parent P_2;

4: Select at random a $H_g \times W_g$ rectangle RO_1 in offspring O_1. Let RO_2 be the same rectangle in offspring O_2;

5: Interchange the mesh router nodes: Move the mesh router nodes of RP_1 to RO_2 and those of RP_2 to RO_1;

6: Re-establish mesh nodes network connections in O_1 and O_2 (links between mesh router nodes and links between client mesh nodes and mesh router nodes are computed again);

7: **return** O_1 and O_2;

are computed again.

- *RectangleMutate*: This operator selects two "small" rectangles at random in the grid area, and swaps the mesh routers nodes in them. Certainly, in this case the modification of the individual is larger than in the case of SingleMutate.

- *SmallMutate*: This operator chooses randomly a router and moves it a small (*a priori* fixed) numbers of cells in one of the four directions: up, down, left or right in the grid.

- *SmallRectangleMutate*: This operator is similar to SmallMutate but now we select first at random a rectangle and then all routers inside the rectangle are moved with a small (*apriori* fixed) numbers of cells in one of the four directions: up, down, left or right in the grid.

Again, after the mutation is done, network connections (the links between routers and links between routers and users) are re-computed.

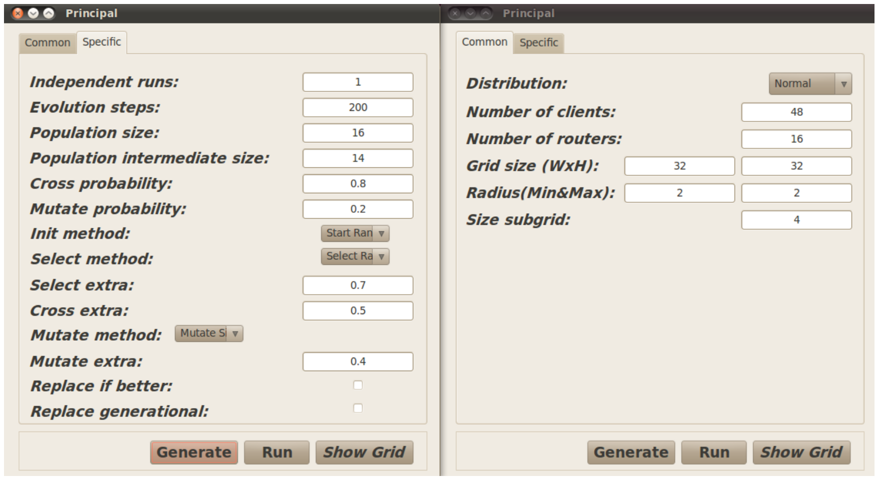


Figure 1: GUI of WMN-GA simulation system.

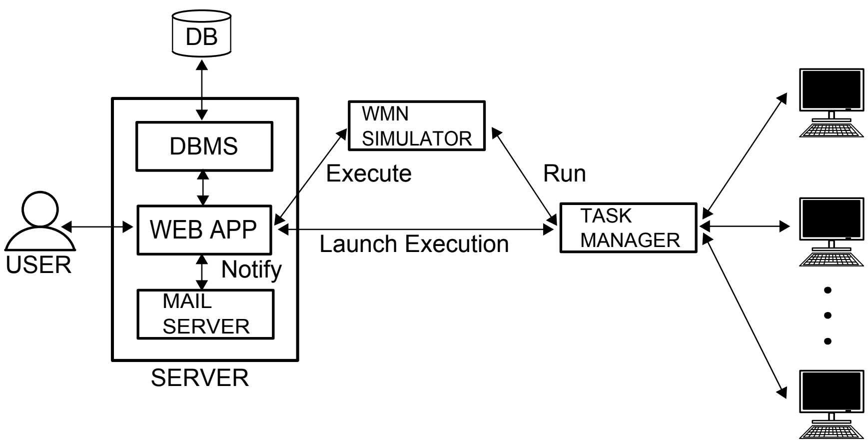


Figure 2: System structure for Web interface.

4.2. GUI of WMN-GA Simulation System and Web Interface

We have implemented the GUI of WMN-GA simulation system, which can generate instances of the problem using different distributions of clients and mesh routers. The GUI 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

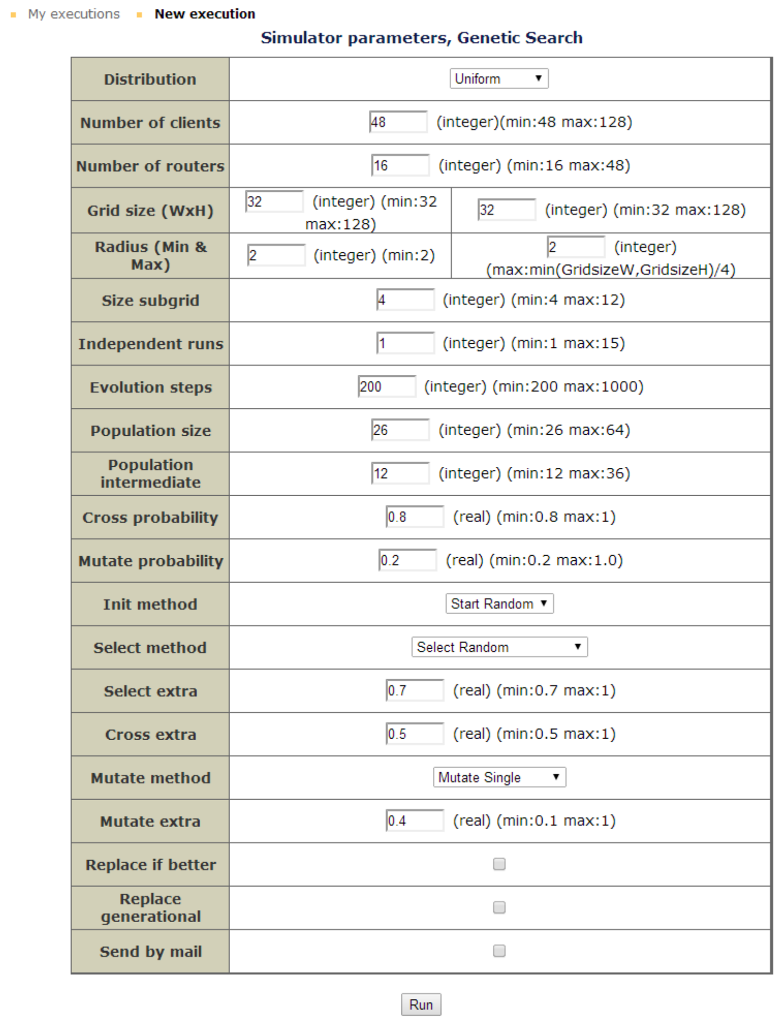


Figure 3: Web interface of WMN-GA simulation system.

transmission distance and the size of sub-grid. 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.

We also have implemented the Web interface. The Web application [16] follows a standard Client-Server architecture and is implemented using LAMP (Linux + Apache + MySQL + PHP) technology (see Fig. 2). Remote users (clients) submit their requests by completing first the parameter setting. The parameter values to be provided by the user are classified into three groups, as follows.

- Parameters related to the problem instance: These include parameter values that determine a problem instance to be solved and consist of number of router nodes, number of mesh client nodes, client mesh distribution, radio coverage interval and size of the deployment area.
- Parameters of the resolution method: Each method has its own parameters. In Fig. 3 are shown the the GUI of Web Interfaces for the parameter setting of GA.
- Execution parameters: These parameters are used for stopping condition of the resolution methods and include number of iterations and number of independent runs. The former is provided as a total number of iterations and depending on the method is also divided per phase (e.g., number of iterations in a exploration). The later is used to run the same configuration for the same problem instance and parameter configuration a certain number of times.

5. ns-3 and Path Loss Model

5.1. ns-3

The ns-3 simulator 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.

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 the wireless network interface controller based on the IEEE 802.11 standard [17].

5.2. Log-distance Path Loss Model

The log-distance path loss model is a radio propagation model that predicts the path loss a signal encounters inside a building or densely populated areas over distance. This propagation model is applicable for indoor propagation modelling. Log-distance propagation loss model is formally expressed as:

$$L = L_0 + 10nlog_{10}(\frac{d}{d_0}),$$

where:

- n: the path loss distance exponent,

- d_0: reference distance [m],
- L_0: path loss at reference distance [dB],
- d: distance [m],
- L: path loss [dB].

When the path loss is requested at a distance smaller than the reference distance, the value of Tx power is returned.

6. Simulation Results

In this section, we present the simulation results. 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 1. The area size is considered 640m $\times$ 640m (or 32 units $\times$ 32 units). The the number of mesh routers is 16 and the number of mesh clients 48. We used HWMP routing protocol and sent multiple CBR flows over UDP. The pairs source-destination are the same for all simulation scenarios. We made simulations for different number of connections (10, 20 and 30 connections). Log-distance path loss model and constant speed delay model are used for the simulation and other parameters are shown in Table 2. We consider the connectivity between mesh routers and conduct simulations using ns-3 simulator. The simulations in ns-3 are done for 5 and 200 generations.

In Fig. 4, we show the simulation results for Size of Giant Component (SGC), which indicates the number of connected mesh routers, and the Number of Covered Mesh Clients (NCMC) for Exponential and Weibull distributions, respectively. First, we optimize the SGC. We can see that after few generations, all 16 routers are connected with each other. Then, we optimize the position of routers in order to cover as many mesh clients as possible. Here, we consider Exponential and Weibull distributions of mesh clients, because these distributions are similar with mesh clients concentrated in hot-spot environments. We

Table 1: Input parameters of WMN-GA.

Parameters	Values
Number of clients	48
Number of routers	16
Grid width	32 units
Grid height	32 units
Independent runs	10
Number of Generations	200
Population size	4096
Selection Method	Linear Ranking
Crossover rate	80 %
Mutate Method	Single
Mutate rate	20 %
Distribution of Clients	Exponential and Weibull

can see that by increasing the number of generations, the number of covered clients is increased. After 200 generations, 47 mesh clients are covered.

In Fig. 5 and 6, we show the visualization of data generated by WMN-GA simulation system for Exponential and Weibull distributions, respectively. In Fig. 5(a) and Fig. 6(a), we show the visualized data for 5 generations. While, in Fig. 5(b) and Fig. 6(b), we have shown the optimized position of visualized data for 200 generations. The simulation results show that all 16 routers are connected and 47 mesh clients are covered. In, Fig. 7(a) and Fig. 7(b), we show the visualization of data generated by WMN-GA Web interface for Exponential and Weibull distributions, when the radius of communication distances is $2 \times 2 : 8 \times 8$ (min : max).

We used PDR, throughput and delay metrics for performance evaluation. In Fig. 8, we show the simulation results of PDR vs. transmission rate for 200 generations and number of connections 10, 20 and 30. From the results, we can

Table 2: Simulation parameters.

Parameters	Values
Area Size	640m×640m
MAC	IEEE 802.11s
Propagation Loss Model	Log-distance Path Loss Model
Propagation Delay Model	Constant Speed Model
Number of Mesh routers	16
Number of Mesh clients	48
Number of Connections	10, 20, 30
Transport Protocol	UDP
Application Type	CBR
Packet Size	1024 bytes
Source Node ID	Random
Destination Node ID	Random
Simulation Time	650 sec

see that when the number of connections is 10 the PDR is higher than other cases. With the increasing of the transmission rate, the PDR is decreased. For number of connections 10, 20 and 30, the PDR is less than 60% when the transmission rate is more than 1200 kbps for Exponential distribution. The PDR for Weibull distribution is higher than for Exponential distribution. When transmission rate is higher, the PDR decreases because of packet collision and congestion. In this case, the queue of each mesh router get full and many packets are dropped, so the efficiency of the network is decreased.

In Fig. 9, we show the simulation results of throughput vs. transmission rate for 200 generations and number of connections 10, 20 and 30. Based on the number of connections, the total data rate transmitted in the network changes. The theoretical throughput is calculated by the following equation:

$$\text{Theoretical throughput} = \text{Transmission rate} \times \text{numCon}.$$

For different number of connections, the throughput is increased linearly with the increasing of the transmission rate. The throughput of Exponential distribution is higher than Weibull distribution.

The simulation results of delay vs. transmission rate for 200 generations are shown in Fig. 10. With increasing of the number of connections and transmission rate, the delay is increased. For 10 connections, the delay is very small until the transmission rate is 800 kbps for Exponential distribution. The delay of Exponential distribution is smaller than Weibull distribution.

7. Conclusions

In this paper, we presented the interface of WMN-GA system and evaluated the performance of WMN-GA simulation system for Exponential and Weibull distributions considering different transmission rates. We present the visualization of the simulation results for different generations. As evaluation parameters, we considered the PDR, throughput and delay metrics. For simulations, we use ns-3 simulator and Hybrid Wireless Mesh Protocol (HWMP). The topologies of WMN are generated using WMN-GA system with area size 640m×640m, 16 mesh routers and 48 mesh clients.

The clients are distributed in the grid area using Weibull and Exponential distributions which are similar with a real hot-spot scenario.

We carried out the simulations using ns-3 simulator. We transmitted multiple CBR flows over UDP. For simulations, we considered different number of connections (10, 20 and 30), HWMP protocol, log-distance path loss model, constant speed delay model and different transmission rates.

From simulation results, we concluded as follows.

1. For 10, 20 and 30 number of connections , the PDR is less than 60% when the transmission rate is more than 1200 kbps for Exponential distribution. The PDR for Weibull distribution is higher than Exponential distribution.
2. For different number of connections, the throughput is increased linearly with the increasing of the transmission rate. The throughput of Exponential distribution is higher than Weibull distribution.

3. For 10 connections, the delay is very small until the transmission rate is 800 kbps for Exponential distribution. The delay of Exponential distribution is smaller than Weibull distribution.

In the future, we would like to make extensive simulations to evaluate other network topologies, different density of nodes, different distribution of mesh clients and different grid sizes.

Acknowledgement

This work is supported by a Grant-in-Aid for Scientific Research from Japanese Society for the Promotion of Science (JSPS). The authors would like to thank JSPS for the financial support.

References

[1] E. Nordström, APE - a large scale ad hoc network testbed for reproducible performance tests, Master's thesis, Uppsala University (2002).

[2] R. Draves, J. Padhye, B. Zill, Comparison of routing metrics for static multi-hop wireless networks, in: SIGCOMM '04, 2004, pp. 133–144.

[3] S. N. Muthaiah, C. P. Rosenberg, Single gateway placement in wireless mesh networks, in: 8th International IEEE Symposium on Computer Networks, 2008, pp. 4754–4759.

[4] M. Tang, Gateways placement in backbone wireless mesh networks, International Journal of Communications, Network and System Sciences 2 (1) (2009) 44–50.

[5] A. Franklin, C. Murthy, Node placement algorithm for deployment of two-tier wireless mesh networks, in: IEEE GLOBECOM-2007, 2007, pp. 4823–4827.

[6] A. Lim, B. Rodrigues, F. Wang, Z. Xu, k-Center problems with minimum coverage, Theoretical Computer Science 332 (2005) 1–17.

[7] J. Wang, B. Xie, K. Cai, D. Agrawal, Efficient mesh router placement in wireless mesh networks, in: MASS-2007, 2007, pp. 9–11.

[8] T. Vanhatupa, M. Hannikainen, T. Hamalainen, Genetic algorithm to optimize node placement and configuration for wlan planning, in: 4th International Symposium on Wireless Communication Systems, 2007, pp. 612–616.

[9] J. Denzinger, J. Kidney, Evaluating different genetic operators in the testing for unwanted emergent behavior using evolutionary learning of behavior, in: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2006, pp. 23–29.

[10] M. Odetay, Empirical study of the interdependencies of genetic algorithm parameters, in: 23rd EUROMICRO Conference, New Frontiers of Information Technology, 1997, pp. 639–643.

[11] F. Xhafa, B. Duran, A. Abrahamy, K. Daha, Tuning struggle strategy in genetic algorithms for scheduling in computational grids, Neural Network World 18 (3) (2008) 209–225.

[12] F. Xhafa, L. Barolli, A. Durresi, An experimental study on genetic algorithms for resource allocation on grid systems, Journal of Interconnection Networks 8 (4) (2007) 427–443.

[13] F. Xhafa, C. Sanchez, L. Barolli, E. Spaho, Evaluation of genetic algorithms for mesh router nodes placement in wireless mesh networks, Journal of Ambient Intelligence and Humanized Computing 1 (4) (2010) 271–282.

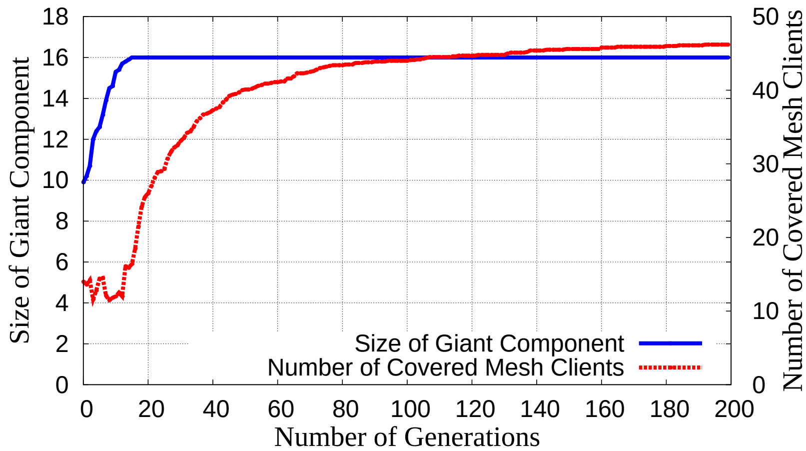
[14] W. Conner, Ieee 802.11 tgs usage models, IEEE P802.11 Wireless LANs, Document IEEE 802.11-04/0662r16,.

[15] C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (aodv) routing, IETF Experimental RFC 3561.

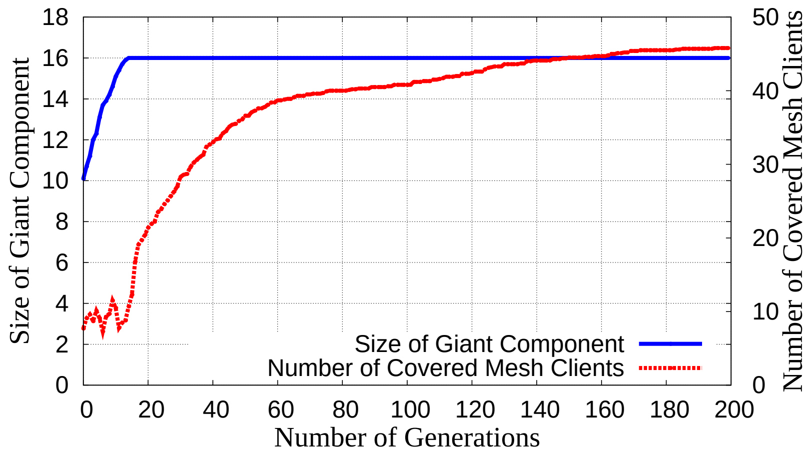
[16] T. Oda, A. Barolli, E. Spaho, L. Barolli, F. Xhafa, Analysis of mesh router placement in wireless mesh networks using friedman test, Proc. of The

28th IEEE International Conference on Advanced Information Networking and Applications (AINA 2014) (Victoria, Canada, May 13-16, 2014.) pp. 282–288,.

[17] I. Std.802.11-2007, Wireless lan medium access control (mac) and physical layer (phy) specifications, IEEE Computer Society.

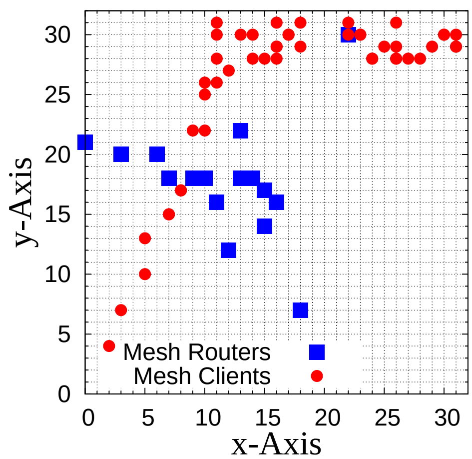


(a) Exponential distribution

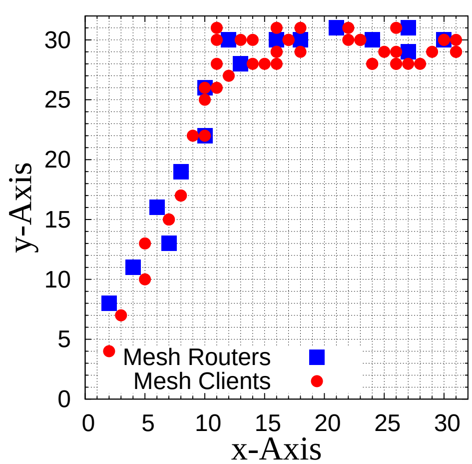


(b) Weibull distribution

Figure 4: Number of covered mesh routers and number of covered clients vs. number of generations.

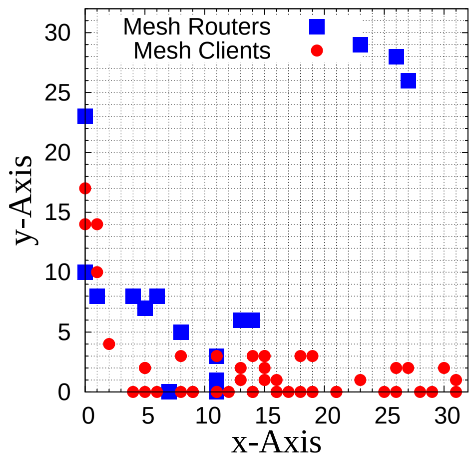


(a) Number of generations: 5 (12, 17)

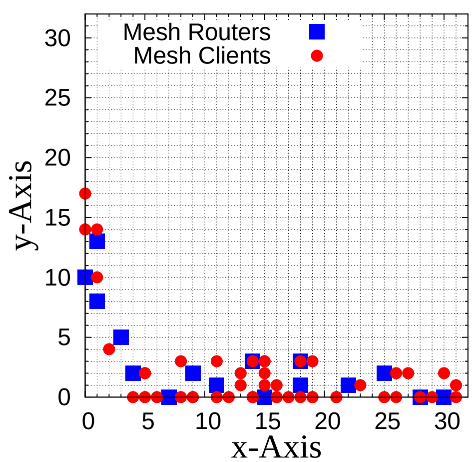


(b) Number of generations: 200 (16, 47)

Figure 5: Optimized location of mesh routers by WMN-GA, (m, n): m is number of connected mesh routers, n is number of covered mesh clients (Exponential distribution).

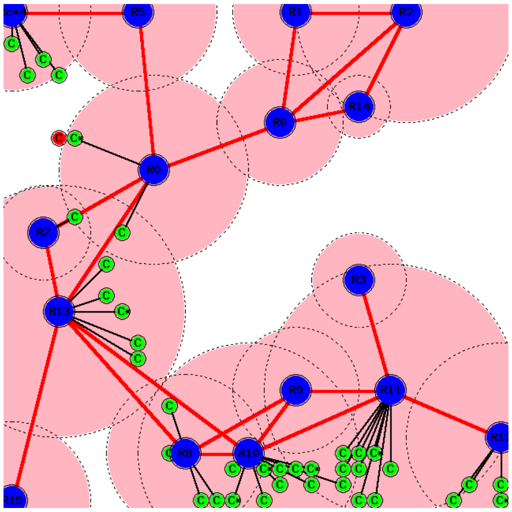


(a) Number of generations: 5 (12, 12)

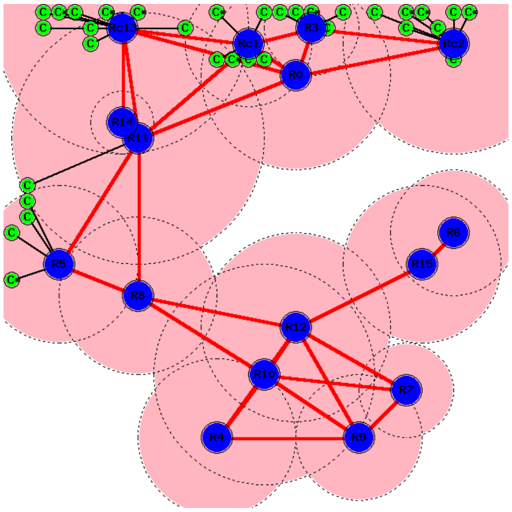


(b) Number of generations: 200 (16, 47)

Figure 6: Optimized location of mesh routers by WMN-GA, (m, n): m is number of connected mesh routers, n is number of covered mesh clients (Weibull distribution).

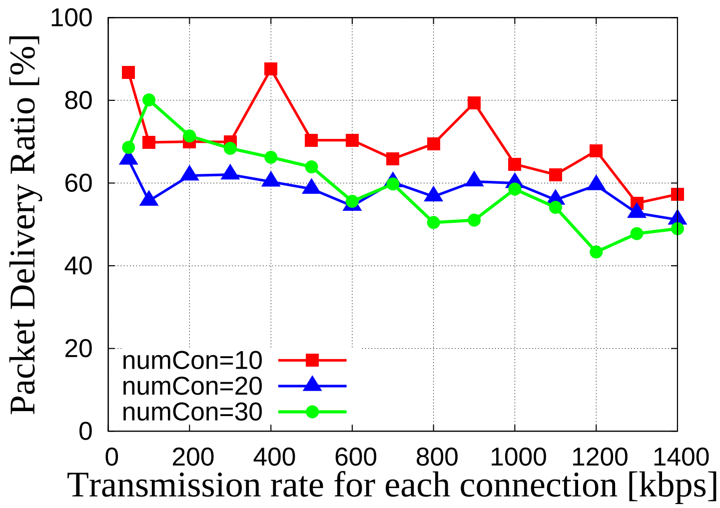


(a) Exponential distribution.

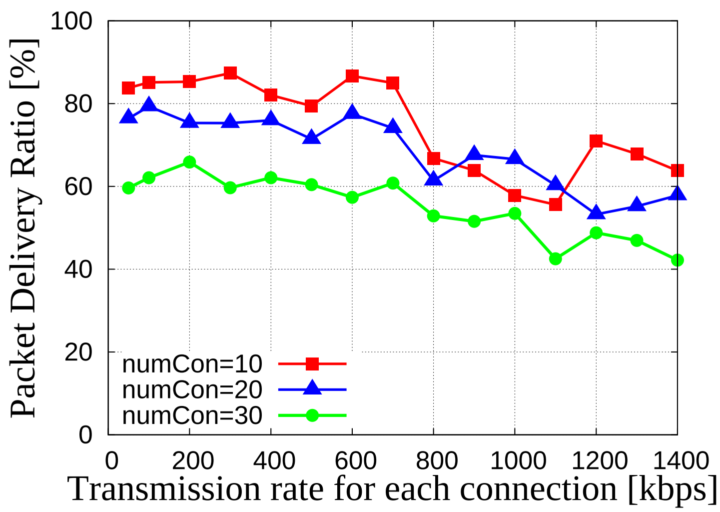


(b) Weibull distribution.

Figure 7: Visualization of nodes using WMN-GA Web interface for 2×2:8×8.

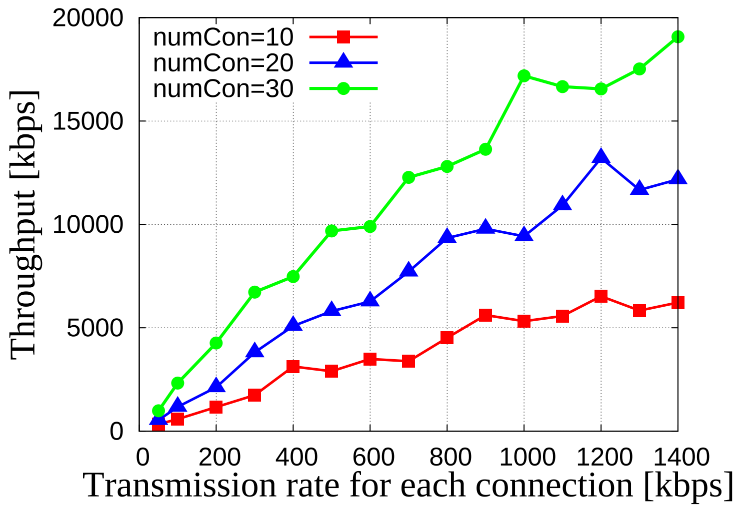


(a) Exponential distribution

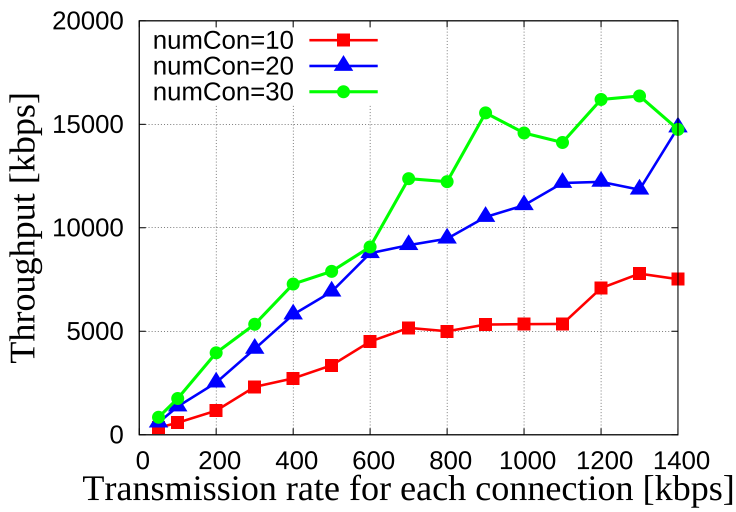


(b) Weibull distribution

Figure 8: Results of average PDR for no. of generations 200.

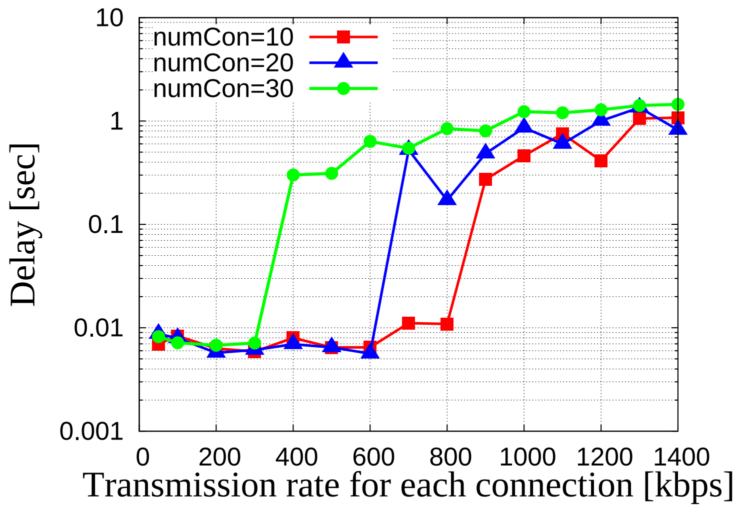


(a) Exponential distribution

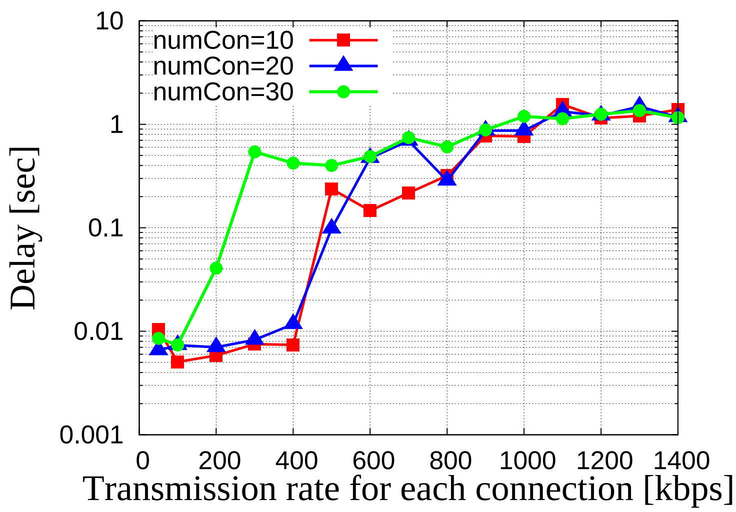


(b) Weibull distribution

Figure 9: Results of average throughput for no. of generations 200.



(a) Exponential distribution



(b) Weibull distribution

Figure 10: Results of average delay for no. of generations 200.