INFO-125
Primljeno/Received: 2017-07-12

UDK: 004.715:004.021:004.73
Preliminary Communication/Prethodno priopćenje

# A BIGENERIC MULTI-PATH ROUTING ALGORITHM FOR WIRELESS MESH NETWORKS

*Shajilin Loret J B*[1], *Ganesh Kumar*[2]
*Department of Computer Science and Engineering, VV College of Engineering, Tamilnadu, India*[1]*;*
*School of Computing Science & Engineering, Galgotias University, Greater Noida, New Delhi, India*[2]

*Abstract*

Routing is the important research issue in the development of Wireless Networks. Multipath routing allows data communication through multiple paths. On the other hand, multi-path routing does not guarantee deterministic transmission. Since one route is available for transferring data from the source node to the destination node. A bigeneric multi-path routing algorithm is planned for significant wireless mesh networks to enhance reliability, also as in impact considering with link failures. The constituted algorithm adopts the modified Dijkstra's algorithm for searching the shortest route from the gateway to each end node. A virtual trail distinct from the regular trail is introduced to realize trail diffusion and updating. The routes used for data point's transmission are selected based on their regular trail values, alleviating the delivery of data points through better routes. Link failures are then treated using route maintenance mechanism. This can be accomplished by increase the accuracy through the already visible route measures collected by the routing protocol. Rate adaptation algorithm is designed to compute the best rate for each wireless link. This modified conclusion aims at providing better routing and rate alternatives. Simulation results show that the proposed algorithm outperforms conventional algorithms in terms of packet delivery ratio, end-to-end delay routing operating cost.

*Keywords*

Multipath routing, Bigeneric path, Gateway, Rate Adaptation, Routing Overhead

## 1. INTRODUCTION

Nowadays, mobile communications like wireless technology, plays great role in the real world. Advance creations of hardware along with software technologies leads to proficient wireless mobile equipment. The classifications of wireless systems are /**1**/, namely, Infrastructure networks and Infrastructure-less networks. The convergence of this two approaches results in wireless mesh networks (WMNs) that guarantees larger flexibility, increased reliability, and enhanced performance. However, there are still numerous challenges in designing of wireless mesh networks (WMNs).

Video communications in WMN are typically Internet oriented and therefore the traffic is either from end user to Internet gateway (IGW) or vice versa /**2**/. In most WMNs deployed currently, the main focus of the routing protocol is to find a single best possible route from the source to the destination. Consequently, certain links might be heavily loaded whereas many others are significantly underutilized. Such a phenomenon could be a breach of traffic engineering principle and will depreciate the overall performance of the network. In this article, a bigeneric multi-path routing algorithm known as Rate Adaptive DWMNet is proposed for industrial wireless mesh networks that combines Dijkstra's algorithm /**3**/ and also the Rate Adaptive Link Estimation based ant colony optimization (ACO) algorithm /**4**/. This was derivative from hybrid routing algorithm of /**5**/. Dijkstra's algorithm gets the first route setup, while the ACO algorithm is responsible for route exploration and maintenance. Rate Adaptive DWMNet aims to achieve load balance additionally as reliable and deterministic transmission with minimal flooding problem. Another mechanism based on a joint approach for solving these two important issues in WMNs such as automatic rate adaptation and routing metric assignment /**6**/ is also presented. The idea is to use a cooperative cross-layer method, called ROUTE selection. VANETs emerged to turn the attention of researchers in the field of wireless communications /**7**/. The taxonomy of various clustering and routing techniques in WSNs based on networks /**8**/. The routing protocols are categorized as energy efficiency computational complexity, path establishment and network structure /**9**/. Combined network topology information and transform-domain analysis in the window, the specious traffic components identified in /**10**/. Temporal-credential-based mutual authentication scheme introduced in /**11**/.

## 2. PROPOSED ROUTING ALGORITHM

The bigeneric multi-path routing algorithm for industrial wireless mesh network, hereby known rate adaptive DWM Net, which combines the improved Dijkstra's algorithm and ACO approach. The enhanced Dijkstra's algorithm achieve the route setup, while the ACO algorithm is accountable for route exploration and maintenance. Following the initialization of a network, the network manager implements the enhanced Dijkstra's algorithm to compute the shortest path to every end node. Afterwards, the remaining routes are searched by employing Rate Adaptive Link Estimation based ACO in the route exploration stage that is executed concurrently with data communication. The method of route maintenance will respond effectively to some topological changes in an correct way. The bigeneric architecture not only improves effectiveness in route setup, in addition to that it makes efforts on redundant routes searching and the link failure maintenance. Routing information is stored within the routing tables and data transmission is directed by the information from this routing tables.

### 2.1. Routing tables

In industrial wireless mesh networks, all nodes preserve a series of tables, which control the communications, which is performed by the nodes. These tables like neighbor table and graph table collect the statistics on communications. In the suggested algorithm, trail tables are added as extension tables for all nodes.

- Neighbor table: This refers to the list of nodes that a node is capable to communicate with directly. The neighbor table of every node is populated after the initial period of a network.

- Graph table: Graphs in networks are accustomed to route messages from their source to their destination nodes. Each node doesn't know the whole route, however only knows subsequent hop destination legal for propagating a packet. Data transmission relies on the route with graph ID listed in the graph table. The regular trails of routes in graph table are utilized as the index for route selection.

Trail table: This records trail information regarding routes from node $x$ to the gateway $g$ through the intermediate node y. The trail table contains information of the intermediate nodes on this route and the regular trail value $R_{xy}^{g}$ or a virtual trail value $V_{xy}^{g}$. At the same time every route can't have both the regular and the virtual trail value. The regular trail is used to estimate the goodness of routes and to decide which route will be used for data transmission. The virtual trail signifies the routes which will possibly relay the data, and then guides ants to sample these attainable routes in networks /**12**/. Through information bootstrapping the virtual trail is obtained by employing the goodness values reported by neighbor nodes throughout route exploration. In the route exploration and maintenance, the bootstrapped value is stored briefly within trail table. The goodness of route is stated as the converse of the cost of a pathwhich depends on the metrics. Here the shortest paths are preferred; therefore the distance between two nodes is a significant metric.

### 2.2. First route setup by enhanced Dijkstra's algorithm

In networks, all end nodes should send field data to the gateway. The gateway exchanges supervision messages with all the end nodes. After the network initialization, an enhanced Dijkstra's algorithm is adopted for first route setup by low calculation complexity and less overhead. Instead of permitting each node to launch its own search process function is executed through the network manager. Additionally, the enhanced Dijkstra's algorithm used Fibonacci heap /**13**/ to implement Dijkstra's algorithm to raise searching efficiency.

An industrial wireless mesh network will be described as a simple connected graph, $G = (V, E)$, where $V$ and $E$ denote the set vertices and also the set of edges, correspondingly. The simple weight of each edge is delineated by the distance between two nodes, as the length of the edge. $g$refers the gateway, and $dt(v)$ is that the distance between the *gateway* and node $v$. The enhanced Dijkstra's algorithm is given in subsection and it works as follows:

Throughout algorithm operation, every vertex is either in unlabeled or labeled state. At first, $d(g) = 0$and $d$[illegible]$v$[illegible] , the state of each and every vertices are unlabeled (lines 1 to 4). The Fibonacci heap stores all the vertices, and $dt(v)$ is the key of each vertex (line 5). The state of $v$ is converted into labeled (lines 6 to 8) and also the vertex $v$ with the minimum key is extracted from the heap. For each edge $(v, u)$, if the state of $u$ be unlabeled and $dt(v) + dt(v, u)$ is smaller than $dt(u)$, then all $dt(u)$ is replaced by$dt(v) + dt(v, u)$ and also the key of $u$ ismodified consequently (lines 9 to 14). This algorithm is repeated until the heap is empty. In this mode, the shortest routes from the gateway to each node are found out.

**Algorithm 1:Enhanced Dijkstra's algorithm**

```
Enhanced Dijkstra(G, g, v)
1. for each vertex v in G
  2. distance[v] = infinite; parent[v] = NULL;
     state[v] = unlabeled;
  3. for gateway g in G
    4. distance[g] = 0;
    5. insert all the vertexes into distance Fibonacci
       heap Q;
      6. while (!Q.empty())
        7. v = Extract-Min(Q);
        8. state[v] = labeled;
        9. for each edge (v, u)
          10 if (state[u] = unlabeled)
           11 if(distance[u]>distance[v]
                + distance[v, u])
           12 distance[u]=distance[v] do
                + distance[v, u];
          13 parent[u] = v;
14. Decrease-Key(u)
```

Previously the first shortest route from node x to the gateway is found, trail tables of all nodes on this route and also the graph table of node x are populated. The trail at this route could be a regular trail $R_{xy^g}$ , and this route is added to the graph table for data transmission.

After the search process is completed, the network manager will hold the information of the possible shortest routes for every node. Afterwards, the network manager sends the associated routing information to the nodes so as to populate their respective graph tables. From the information within the graph tables, the nodes will identify the graph ID, the receiver node, and also the next node. This technique can be called as target direction query. Here, if someone asks some directions for a specific place to another one, they will provide him some directions; then, at subsequent crossing, this person will ask some other person for assistance. This technique will decrease the memory capacity of each node, by avoiding the redundant multi-hop search process. Every node has one of the shortest route to the gateway for sending data packets.

### 2.3. Route exploration by Rate Adaptive Link Estimation based ACO

In this phase multiple redundant routes for data transmission are accomplished. These processes extend available routing information to a mesh node of multiple routes based on initial route setup period. Two sub-processes are included in route exploration namely trail diffusion and trail updating. In trail diffusion the obtainable trail information is spread over the network by employing "keep-alive" messages periodically. The aim of trail updating is to regulate and update trail information all through route sampling by utilizing ants.

In this study, trail diffusion proceeds as soon as the shortest routes are discovered; it's then concluded when the regular trail information of all routes are populated through the method of trail updating. A route exploration cycle is formed by this process, and more over the next cycle is activated by topological changes or the timer.

### 2.3.1. "Keep-alive" message

In industrial wireless mesh networks, each node exchanges its information with every other node by employing "keep-alive" messages. The unique intention of this message is to look for quiescent links, find out new neighbors, and also to maintain time synchronization. In order to avoid power consumption and network flood because of numerous messages, "Keep-alive" messages have to be sent once per minute.

In this structure, "keep-alive" message is expanded to take account of the routing information for trail diffusion. This "keep-alive" message plays an important role in trail diffusion, and as such, some properties require adjustment. In the route discovery period, as a node gets new information of paths to the gateway, the node sends "keep-alive" messages toward its neighbors. And the time interval of the transmission of those messages should be more than 1 s. Each message carries information concerning five routes to the gateway at most. Since this step stops, the properties of "keep-alive" message are rearranged to their default configuration.

### 2.3.2. Trail updating

When a node receives a "keep-alive" message about route trail, the resulting bootstrapped value begins the trail updating of routes. There are two possibilities cases concerning trail updating: routes without regular trail and routes with regular trail. Similar to these two cases, the bootstrapped value will be utilized for route exploration and route updating, correspondingly.

**a) Route exploration**

If only a bootstrapped value $B_{xy}^{g}$ is present rather than a regular trail value within the trail table of node $x$,$B_{xy}^{g}$ *indicates* a possible new route. But, this new routehas never been usedwholly, therefore it is potentially unreliable. Hence node x updates its trailinformation as given as: $V_{xy}^{g}$ [illegible] $B_{xy}^{g}$ . Once a virtual pathis allotted to a route, proactive forward ants are sent out by the source node that may be a typical mode of operation in the Rate Adaptive Link Estimation based ACO algorithm.

Proactive forward ants are unicast instead of broadcast toward the destination. In this process, a proactive forward ant takes a new routing decision at the intermediary node $m$ employing the probability $P_{xm}^{g}$ as outlined in equation (3). There are regular trails or the virtual trails on routes from node $x$ to the gateway. Hence, $P_{xm}^{g}$ is decided by the actual circumstances of route trail. When a proactive ant arrives at a node that provides no routing information concerning the gateway suggests the route does not reside. Afterwards, the forward ant is removed. On the other hand, the network manager records the route taken by means of the proactive onward ant once it arrives at the gateway. Thus the virtual path is investigated by this manner, and also the regular path is chosen on this route. As a result, other routes from $x$ to $g$ which contain regular path is also involved in trail updating. Since the number of routes is being increased, the offered route for data transmission also increases for a full mesh.

By using multiple routes in ant forwarding, the throughput is improved as the load is spread more evenly over theobtainable network resources. This approach to choose the next-hop node throughout the advancing process is similar to the sensible routing policies analyzed in /**14**/.

**b) Route updating**

If node $x$ contains a regular path value in its trail table for a route from $x$ to the gateway y, in that case this route may possibly have been sampled by ants within the past or is taken into account the shortest route. Thus, this route is reliable. $B_{xy}^{g}$ is treated like an update of the goodness estimate of this reliable route and is employed to replace $R_{xy}^{g}$ when $B_{xy}^{g}$ is better than $R_{xy}^{g}$. In this manner, trails on current routes are kept up-to-date. At the same time, the graph tables ought to be updated in addition. Trail updating doesn't last as long as the whole network lifetime. A loop is terminated when all routes within the network have regular trails.

Once the network is initialized, the network manager allows the enhanced Dijkstra's algorithm to calculate the shortest path to all end nodes. When the shortest route of each node is found, the trail tables of all nodes on this route and graph table of this node are populated. Route exploration includes two sub-processes, that is, trail diffusion and trail updating. Throughout the process of trail diffusion, a node will derive the bootstrapped value. On the other hand, the network manager records the route taken by the proactive forward ant once it arrives at the gateway. In this way, the regular trail is allocated on this route. If there's a regular trail value for a route, the bootstrapped value can be treated like an update of the goodness estimate of this route and is employed to replace the regular trail on this route when the bootstrapped value is better than the regular trail. In this manner, trails on current routes are kept up-to-date. In this article, trail diffusion proceeds immediately the shortest routes are discovered; it's then terminated when the regular trail information of all routes is populated through the method of trail updating.

### 2.4. Link Quality Estimation for ROUTE Selection

In order to compute the metric $RS_{ab}$ its necessary to know the value of Expected Transmission Count (ETX) for $a \rightarrow b$ in every available transmission rate. As the original formulation of ETX depends on broadcast probes, this value is just computed at one rate. Thus the Route Selection needs a different approach for computing ETX value.

The most straight forward solution is to simply manipulate the transmission rate for broadcast frames, in order that Route Selection can send the ETX probes at every obtainable rate. By this way, Route Selection would have statistical data to compute ETX in all rates. But, considering, for example, the IEEE 802.11b/g mixed mode (widely utilized in commercial devices), there are twelve available rates. Therefore, this strategy would significantly increase network overhead.

To avoid such an overhead increase, Route Selection adopts a different approach, shown in ETX algorithm.

This approach relies on a process of renovation of the links success probabilities. This renovation happens in two steps:

1) the average SNR (Signal-to-Noise Ratio) of the link is estimated utilizing the information provided by probe packets;

2) the average SNR is employed to estimate the link success probability in all rate, that is later used to compute ETX for each rate.

Both steps need the knowledge of a function that relates SNR and also the success possibility of a link. while defining a closed phrase for such a function isn't trivial, previous works have collected data through experiments and simulations for all transmission rates employed in the IEEE 802.11b/g mixed mode /**15**/. The flow chart of Rate Adaptive DWM Net is showed in figure 1.

These data is accustomed to build a table relating four physical measures includes SNR, transmission rate, frame size, and PER (Packet Error Rate). For example, this table is employed, by the DEI802.11-mr module, an enhanced performance of the IEEE 802.11 standard for the ns-2 simulator.

By this pre-computed table, it's possible to assume the SNR of a connection from the packet error rate computed with probe packets at the basic rate (as done by the ETX metric). The transmission rate and also the frame size (probe size) are noted from the transmission of the probe packets. The PER is approximated by the value measured by employing the probes. Thus, the table might be consulted to find a value to the SNR. On the other hand, by means of this SNR estimate and selecting a target rate for a new table, consult returns an estimate for the link's PER.

1) Conversion Issues: The suggested technique for converting link's PER presents a problem. The function which relates SNR and PER (for a given rate and a given frame size) has an asymptotic behavior intended for both low and high values of SNR. As SNR increases, PER approaches 1. On the other hand, as SNR decreases, the value of PER approaches to 0. For the extreme cases PER equals to 0 or 1 but are never reached, because, independently of how high or low the SNR is, there's always a chance of failure or success respectively.

**Algorithm 2:** Selection of the mainly useful statistic algorithm.

```
Input:lossProb.
Output:usedProbandusedRate.iflossProb[3] < 1 then
        UsedProb, ←lossProb[3]usedRate←54 Mbps
else
iflossProb[2] < 1 then usedProb ← lossProb [2]
usedRate ←36 Mbps
else
iflossProb[1] < 1 then usedProb ← lossProb[1]
usedRate←18 Mbps
else
usedProb←lossProb[0]usedRate← 1 Mbps
end ifd if end if
return usedProb, usedRate
```

When Route Selection has to calculate the metric for a link, it first chooses one of the four probabilities utilizing Algorithm 2. Within the code, *lossProb* is an array, containing the link error rates estimated at each of the four transmission rates. This simple algorithm chooses when the most appropriated statistics for the current link the one related to the higher probe rate, so that the PER value can be less than 1. As a result the extreme values such as 0 and 1 are avoided by improving the exactness of the SNR estimate.

2) Table Limitations: The conversion table utilized by Route Selection associates four physical quantities: SNR, transmission rate, frame size and PER. When this data is organized as a table, there are, two limitations:

a)Two physical quantities stored in the table assumes continuous values such as SNR.

Therefore, to represent all possible values of both physical quantities, the table would need an infinite number of entries. Because this may not be possible, there must be a minimal granularity for these physical quantities.

b) If the table contains a low granularity, it will become too extensive, causing the consults to be computationally expensive.

To avoid both limitations, Route Selection doesn't use directly the table, though instead a set of functions it proposes for interpolating data in the table.

Figure 1 compares the original data and also the obtained curves for four different rates, considering 1500-byte packets. Although this work considers the transmission rates available for the IEEE 802.11b/g standard, it argues that the method represented here can be applied to transmission rates offered in other standards like IEEE 802.11n.The above route selection method is also provides better results on computation.

### 2.5. Route selection for data transmission

Once all the routes have regular trails, after that the cycle of route exploration ends. But for deterministic transmission, not all explored routes are utilized for data transmission. In this paper, the network manager reserves top five routes for every node based on the trail values of routes. If node *x* doesn't have five routes, then all discovered routes are selected for data transmission. Each of these top five routes is indicated by a graph ID, a regular trail value, and also information concerning the next-hop node on this route. The information on these routes for the data transmission is stored in graph tables of nodes along these routes.

Preceding to each data transmission, node *x* selects the route whose graph ID is *k* from the graph table employing the probability $Pi^k$. The number of routes used for data transmission is outlined beforehand owing to the fact that the quality of the routes varies with every node. In this mode, the data transmission can be dispersive for load stability. So, choosing the five best routes and setting $\tau2$ as five guarantees deterministic communication, and balance the path load in the networks, thus transmitting the data through relatively better routes. As a result, these five routes will undertake data transmission equally.

### 3. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our proposed Rate adaptation based hybrid multipath algorithm for
different network sizes under simulated wireless communication channel conditions using the NS2 network simulator. In our simulations, we assume that single source trying to send multi path dependent packet transfers via 60
fixed nodes.

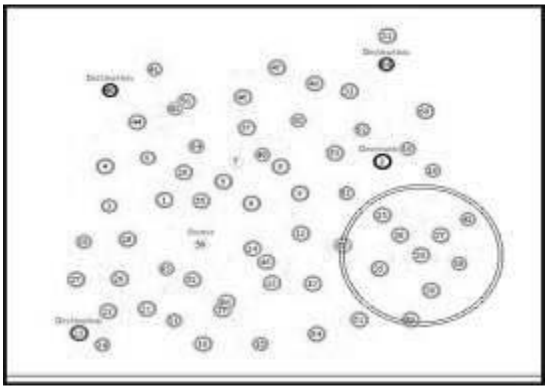

**Figure1. Represent the initial deployment of the wireless mesh node to simulation**

We let nodes to be uniformly randomly distributed in a square cell whereby the maximum distance of a pair of connected nodes is set to 50 meter. We also assume that several number of non-functioning are located at 30 meters from their associated source nodes. The WMN data rate is set to 10 Mb/s using a next-generation IEEE 802.11n WLAN me This evaluation was conducted on both simulated using the implementations discussed in Section II.

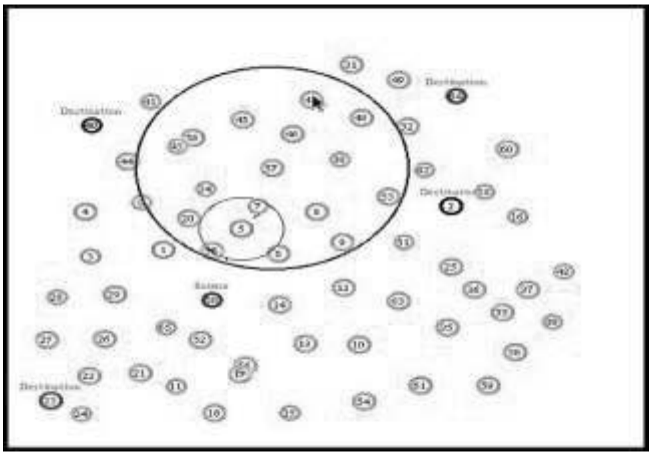

**Figure2 . Represent the source based multipath packet**

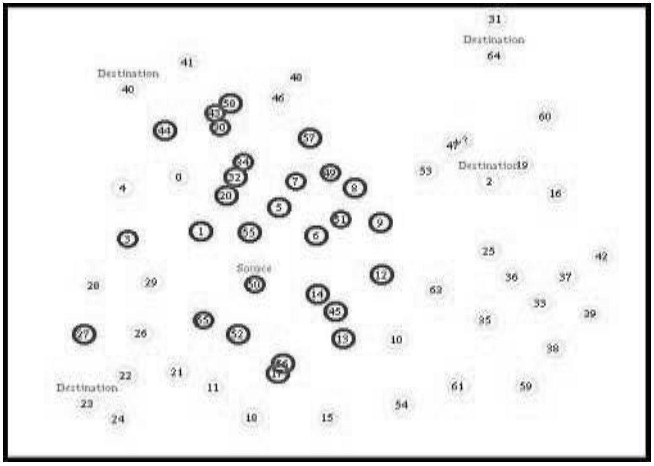

**Figure 3. Represent the available multi path nodes that can be used transfers across the destinations in wireless meshwork**

We evaluate the throughput of the proposed algorithm using standard configurations. In this configuration, we have a 60-node network where single node and multiple destinations are deployed. Figure 2 shows, the high number of multi-receivers, the number of available channels is 3, and the randomly generated link loss ratio on each link value is between 10 and 90%.

In figure 3 it shows, the number of multi-receivers already used for the path is shown in red, where the yellow nodes can be used for the obtainable paths. For each link packet delivery ratio is a randomly generated value between 10 and 94%.

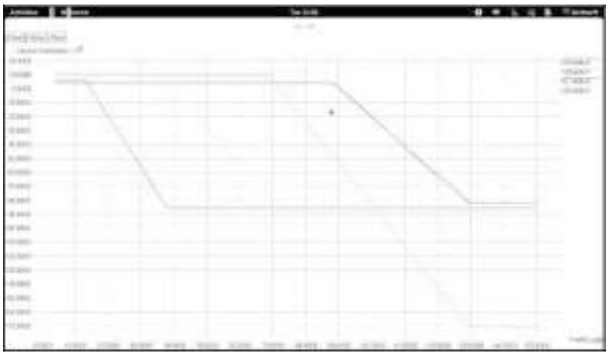

**Figure 4. Represent the overall throughput of the source nodes used during the communication in wireless mesh network**

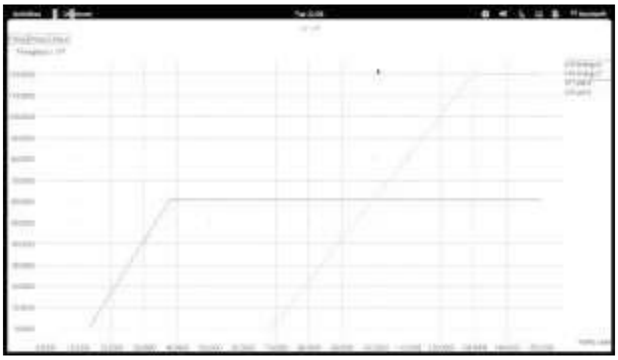

**Figure5. Represent the total number of packet sent during the communication in wireless mesh network**

The performance graph in figure 4 represents the comparison of overall system throughput. From the figure it is observed that the overall throughput is high for the Path which has the highest packet delivery ratio.

We evaluate the total number of packet sent from the source to the destination by measuring the total packets sent to reach its destination. For these experiments, we used 60 nodes and 10 channels, and we varied the number of multi-receivers from 5 to 50. The packet delivery ratio of each link is randomly generated between 10 and 92%.

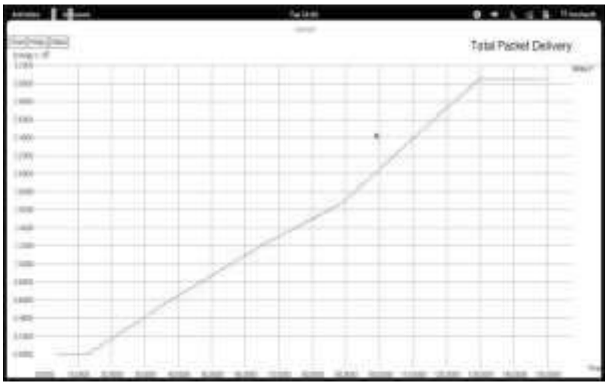

**Figure6. Represent the overall energy spent during the communication in wireless mesh network**

The investigation of the routes which is chosen by each routing metric yielded new interesting insights, though the most frequent route chosen by each metric.

From the results, we can state that the rate adaptation mechanism is able to achieve more throughput than the static algorithm with the consideration of Ant colony transformation. This is because when the link quality is estimated throughout the route matrices, the initially built path may not be capable to provide the best multicast paths for the current application. Here the rate adaption gives enough adjust mechanism to the system which can adapt to the time varying environment.

## 4. CONCLUSION

This paper investigates an efficient hybrid routing approach using ant colony transformation and Rate adaptation matrices in the link-heterogeneous mesh networks where throughput maximization is most important. Compared with previous techniques, our work more accurately models multipath path selection based on rate adaptation in WMNs. The wireless mesh network is modeled as a routine trace collection which can be used for selecting the best path from the available structure. The projected method is simulated with heterogeneous model in order to identify and understand the performance of the overall system.

Future work will be focused on self-adaptive learning based mechanism which can be used to increase the performance in the dissimilar networks. This system is analyzed with various channel based communications for better performance.

*Notes*

/1/ J. H. Schiller, *Mobile communications*. Pearson Education, 2003.

/2/ N. Nandiraju, D. Nandiraju, L. Santhanam, B. He, J. Wang, and D. P. Agrawal, "Wireless mesh networks: current challenges and future directions of web-in-the-sky," *IEEE Wirel. Commun.*, vol. 14, no. 4, pp. 79–89, 2007.

/3/ E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.

/4/ Y. Lu, G. Zhao, and F. Su, "Adaptive ant-based dynamic routing algorithm," in *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, 2004, vol. 3, pp. 2694–2697.

/5/ Y. Zuo, Z. Ling, and Y. Yuan, "A hybrid multipath routing algorithm for industrial wireless mesh networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2013, no. 1, p. 82, 2013.

/6/ D. Passos and C. V. N. Albuquerque, "A joint approach to routing metrics and rate adaptation in wireless mesh networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 999–1009, 2012.

/7/ S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *J. Netw. Comput. Appl.*, vol. 37, pp. 380–392, 2014.

/8/ S. Tyagi and N. Kumar, "A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 36, no. 2, pp. 623–645, 2013.

/9/A. M. Zungeru, L.-M. Ang, and K. P. Seng, "Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison," *J. Netw. Comput. Appl.*, vol. 35, no. 5, pp. 1508–1536, 2012.

/10/ D. Jiang, Z. Xu, P. Zhang, and T. Zhu, "A transform domain-based anomaly detection approach to network-wide traffic," *J. Netw. Comput. Appl.*, vol. 40, pp. 292–306, 2014.

/11/ K. Xue, C. Ma, P. Hong, and R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 316–323, 2013.

/12/ F. Ducatelle, G. Di Caro, and L. M. Gambardella, "Using ant agents to combine reactive and proactive strategies for routing in mobile ad-hoc networks," *Int. J. Comput. Intell.*

*Appl.*, vol. 5, no. 2, pp. 169–184, 2005.

/13/ M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM*, vol. 34, no. 3, pp. 596–615, 1987.

/14/ E. Gelenbe, "Sensible decisions based on QoS," *Comput. Manag. Sci.*, vol. 1, no. 1, pp. 1–14, 2003.

/15/ J. P. Pavon and S. Choi, "Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement," in *Communications, 2003. ICC'03. IEEE International Conference on*, 2003, vol. 2, pp. 1108–1113.

# BIGENERIČKI VIŠESTAZNI ALGORITAM ZA USMJERAVANJE ZA BEŽIČNE MREŽE

*Shajilin Loret J B*[1], *Ganesh Kumar*[2]

*Odjel za računarstvo i inženjerstvo, VV Koledž za inžinjerstvo, Tamilnadu, Indija*[1];
*Sveučilište Galgotias, Greater Noida, New Delhi, Indija*[2]

*Sažetak*

Usmjeravanje je važan istraživački problem u razvoju bežičnih mreža. Višestazno usmjeravanje omogućuje podatkovnu komunikaciju kroz više puteva. S druge strane, višestruko usmjeravanje ne jamči deterministički prijenos, budući da je jedna ruta dostupna za prijenos podataka iz izvornog čvora do odredišnog čvora. Konstituirani algoritam primjenjuje modificirani algoritam Dijkstra za traženje najkraćeg puta od pristupnika do svakog krajnjeg čvora. Algoritam prilagodbe stope dizajniran je za izračunavanje najbolje brzine za svaku bežičnu vezu. Ovaj modificirani zaključak ima za cilj pružiti bolje usmjeravanje i ocjenjivati alternative. Rezultati simulacije pokazuju da predloženi algoritam nadmašuje konvencionalne algoritme.

*Ključne riječi*

Višesmjerno usmjeravanje, bigenerički put, izlaz, prilagodbe stope