Breadth Fixed Gossip: a Route Discovery Mechanism in Ad Hoc Network

Qurrotul Aini\textsuperscript{1,2}, Achmad Affandi\textsuperscript{1}, Eko Setijadi\textsuperscript{1}, Takuya Terasawa\textsuperscript{3}, Mauridhi Hery Purnomo\textsuperscript{1}

Abstract – The lack of information from a root node to neighbors becomes a particular constraint in a message delivery. While the transmission range of node needs to be considered also for ensuring a delivered packet message to them. Therefore, a route discovery mechanism is a good start to make a reliable path in an unknown environment. The objective of this study is to explore the implementation of Breadth First Search, fixed radius model, and gossip algorithms (Breadth Fixed Gossip) on the ad hoc network. Simulations were conducted to obtain the value of three metrics i.e. saved retransmission, transmission failure, and hop counts. These results were compared with Depth First Search and Gossip performance. It indicates that saved retransmission metric Breadth Fixed Gossip has a 58 times better performance, reduces transmission failure by 20%, and requires 50% more hops at transmission range below 15 than Depth First Search and Gossip. Copyright © 2016 Praise Worthy Prize S.r.l. - All rights reserved.

Keywords: Route Discovery, Breadth First Search, Gossip, Fixed Radius, Wireless, Ad Hoc Network

Nomenclature

| Symbol | Description |
|--------|-------------|
| $g$    | Gossip function |
| $p$    | Forwarding probability |
| $p_c$  | Critical probability |
| $G=(V,E)$ | Graph model with node set $V$ and edge set $E$ |
| $O(b^d)$ | Time complexity of depth $d$ |
| $r$    | A node range transmission |
| $m \times n$ | Simulation area |
| $l_{ij}$ | Link between node $i$ and node $j$ |
| $D_i$  | Distance of node $i$ |
| $x_i$  | Coordinate of $x$ at node $i$ |
| $x_0$  | Coordinate of $x$ at initial or root node |
| $y_i$  | Coordinate of $y$ at node $i$ |
| $y_0$  | Coordinate of $y$ at initial or root node |
| $t_i$  | Number of retransmission at route $i$ |
| $h_i$  | Number of hop at route $i$ |
| $p_s$  | Number of successful in transmitting a packet message from a root node |
| $n_i$  | Node $i$ |
| $p_{\text{flooding}}$ | Probability of flooding |
| $l_{16}$ | Link between a root node and node 16 |
| $Tx$   | Transmission range |

I. Introduction

A network is a group of people, a system or organization that aims at achieving the purpose of exchanging information through a wire or wireless media.

Given the increasing need for data communication and the lack of infrastructure availability, a wireless ad hoc network is an option to overcome the obstacles that spontaneously occur. The ad hoc network issue is a route discovery process, i.e. to establish and select the shortest route of messages/information distribution that is included in a routing protocol. To select a route, a root (source) node should identify, recognize, and record the neighbor nodes as a reference to select appropriate nodes which could relay the packet data to the destination. Moreover, a routing protocol can identify and prevent the attacks [1], such as geographic routing approaches, which investigated the similarities and differences based on design attributes and attack protection [2].

The discovery process is a complex issue when it is coupled with node mobility where each node moves arbitrarily moves to join or leave a group. Recently, several techniques related to route discovery have been proposed. To minimize and optimize a route rediscovery process, an attempt is offered based on cluster and compared with the existing routing protocols such as AODV. Referring to simulation results, the effectiveness and network performance of Quality of Service (QoS) improved [3]. Likewise QoS-DSR considers bandwidth constraint in MANET to select the appropriate route [4]. Also, [5] offers a routing method to minimize energy consumption to enhance the network life time. Ref. [6] presents Routing-IP model as an advance investigation to reduce energy consumption of each node in network. Moreover, it compared QoS performance with other models. Meanwhile, [7] it proposes a route selection...
based on fuzzy logic with hybrid optimization of Genetic Algorithm and Hill Climbing algorithm.

The availability of a communication network provided by ad hoc network is slightly reliable than the existing network infrastructure. Moreover, this network has dynamic movement, topology, limitation of energy resources for communication and computation. Therefore, the algorithm for this network should be robust against network dynamics and utilize minimum energy resources. This situation and requirements lead us to Gossip algorithm. It is a part of the mechanisms that has been proposed in routing protocols. This protocol evolved in combination with the existing method that offers probability technique to determine routes for information delivery. Gossip algorithm is a particular method for probabilistic broadcast on flooding technique [8]. Implementing this algorithm in ad hoc network is described as follows. It starts from a root node and conducts a searching process Depth First Search (DFS) algorithm, followed by computing the probability of the links among nodes [9]. The objective of DFS is looking for a destination on a deep branch.

In the large-sized network, the connection among the nodes becomes complex. Knowledge of members of the network’s nodes is required. With DFS search, the possibility of one branch from a root node can be infinite, therefore, the chosen route may not be guaranteed as an optimal route. When compare with DFS, BFS is easy to implement, because it finds out first the solution at the same level, if not then it will be searched in a higher level.

Another problem arises when a node is selected as a relay node by the previous node, so it must ensure that the relay node must be in the transmission range. Therefore, the authors propose the BFS search techniques combined with a fixed radius model and probabilistic broadcast. Fig. 1 depicts the route from the root to destination (node 25), through 5 relay nodes.

![Fig. 1. A route discovery from root to destination node](image)

The objective of this work is to propose a simple mechanism of route discovery in order to select an appropriate node as a relay node to disseminate information from source to destination in ad hoc network by combining Breadth First Search, fixed radius model, and Gossip algorithm. Our contributions in this work are:

1. Discovering a route that is available for source node to transmit the packet information
2. Selecting the appropriate the relay node when out of transmission range, and
3. Performing evaluation of the proposed a route discovery mechanism in metrics of transmission failure, hop counts, and saved retransmission.

The remainder of the paper is organized as follows: in Section 2, we present related work in particular development of Gossip algorithm and DFS. Meanwhile, in Section 3, Breadth Fixed Gossip (BFG) as the proposed route discovery mechanism is described in details. Simulation and results are presented in Section 4. And the last, Section 5 presents the conclusions of this study.

II. Related Work

This section describes the existing route discovery mechanisms based on Gossip algorithm and Depth First Search. Also, it denotes the benefit and drawback of both algorithms.

II.1 Gossip Algorithm

For the realization of topology in MANET, there are some models in random graphs. Fixed radius model is one of the models in which some points are placed randomly and which correspond to percolation theory. A key aspect in ad hoc network in disseminating information is random mobility in particular when nodes have partial knowledge of the network (e.g. route history, node location and schedule, mobility pattern) and nodes are outside of the transmission range. This situation forced the root node in network to select the appropriate node that is able to disseminate information to destination nodes or to other network members. Many attempts to achieve better performance by combining several algorithms are proposed, since it will affect other matters in performance metric, i.e. end-to-end delay or energy consumption. Therefore, selecting the appropriate node and a route discovery in the proposed mechanism should be simple and reliable.

The term Gossip, first stated by Demers et al. in 1987, has been implemented for distributed system [10]. The idea of Gossip was used for the implementation of updates distribution of databases replica in a distributed system, which evolved into the algorithm. Therefore, the information exchange randomly occurs when a node selects other members randomly.

The benefit of Gossip has been used and developed into a protocol that was implemented in sensor networks. Some research areas that become trending topics are data dissemination and reliable broadcast. Gossip routing protocol consists of a probability function of the input parameters [11]:

---

1. Gossip algorithm
2. Depth First Search (DFS)
3. Breadth Fixed Gossip (BFG)
4. Simulation and results
5. Conclusions of this study
The value of Gossip function \( g(\cdot) \) lies in 0 and 1. Generally, the Gossip function known as Gossip probability is based on percolation theory and random graphs.

Ref. [12] introduced percolation theory that learns geographic connectivity. It means that two locations are connected if they are near each other, and unconnected other than that. In a square lattice, a cluster is shown as a collection of occupied (nearest) neighboring sites, marked with the black dot [12]. This theory describes two models in the two-dimension square lattice, i.e., site and bond percolation [13].

In site percolation, it is considered that a lattice site is open (marked in black) with probability \( p \) and closed with probability \((1 - p)\). Fig. 2(a) illustrates site percolation. Meanwhile, in bond percolation model (Fig. 2(b)), each open edge of a lattice (marked in bold) is a probability \( p_c \). \( p_c \) is the value of probability that allows a node to reach its neighbors. Based on previous studies, \( p_c \) value is \( 0.583 [14] - [16] \).

\[
g(\cdot) = p \quad (1)
\]

\[
g(\cdot) \rightarrow [0,1] \quad (2)
\]

Dimakis et al. [20] developed routing protocol based on Gossip and geographic information. This algorithm intended to improve energy consumption in the distributed nodes.

Dwivedi, Sharma and Sharma [21] investigated the variant of Gossip based Sleep Protocol (GSP) i.e. Adaptive GSP, Traffic-aware GSP and Battery-aware GSP in energy conservation.

Other studies are Gossip protocols involving a fuzzy method [22] and considering the adaptation of failures and network topology changes [23].

In previous studies, Gossip routing protocol has evolved to many applications and various methods to achieve better performances in ad hoc network. The advantages of Gossip algorithm can be listed as follows [24]:

- Gossip is easy to implement and selects the first partner to submit the information.
- It converges in rapid speed and can automatically adjust to the network.
- Gossip is a robust algorithm from changes and failures in the network.

Meanwhile, the drawback of Gossip algorithm is that it cannot estimate the size of a network in practice, therefore, we did not know how many rounds are needed to run the algorithm to approach the real value [25].

### II.2 Depth First Search

A route discovery begins with searching algorithm because of unknown state in ad hoc networks. A searching algorithm is able to know the connection among members of the network and finds the shortest path to the desired node. DFS is one of the search algorithms used for graphs and trees.

It explores a branch (from a root) as deep as possible and uses stack data-structure. It allows the algorithm to be applied in both iterative and recursive forms.

The graph in a network can be modeled as \( G = (V,E) \), where \( V \) denotes the set of nodes (vertices) and \( E \) denotes the set of edges. Fig. 3 depicts the DFS algorithm, which starts from 0 (as root) and the red arrows indicate the search order [26].

The obvious advantages of DFS are shown as follows [27]:

- The memory requirement is linear to graph search, because it needs to store a stack of nodes from the root to the current node.
- Time complexity of depth \( d \) is \( O(b^d) \), i.e. DFS is time-limited rather than space-limited.
- When DFS finds solutions without deeply exploring a path, time and space are smaller.

The disadvantage of DFS is the possibility that the infinite graph can generate an infinite tree. Therefore, to solve this problem, depth \( d \) must be limited, but if \( d \) is too small, it may fail to find a solution. Also, DFS does not guarantee to find a solution if more than one solution exists [27].

![Site and bond percolation](image)

(a) Site percolation  
(b) Bond percolation

Figs. 2. Site and bond percolation [14]
### Algorithm 1 Depth First Search

**Input:** Graph $G = (V,E)$

**Output:** DFS tree

Mark each vertex in $V$ with 0 as a mark of being "unvisited"

`count ← 0`

for each vertex $v$ in $V$ do

- if $v$ is marked with 0

  `dfs(v)`

`dfs(v)`

`count ← count + 1`

Mark $v$ with `count`

For each vertex $w$ in $V$ adjacent to $v$ do

- If $w$ is marked with 0

  `dfs(w)`

---

**III. Breadth Fixed Gossip Mechanism**

This section proposes Breadth Fixed Gossip (BFG) mechanism for route discovery in ad hoc network. BFG is an approach that considers search algorithm and fixed radius model as initial step and Gossip algorithm as last determination step. The objective of this mechanism is achieving a route discovery and make successful delivery of a message from a root to a destination.

Breadth First Search (BFS) is a graph searching technique, which analyzes semantic graphs, decides the relationship of two vertices in a graph, and looks for the shortest path. Moreover, BFS could find paths between two vertices in a certain range [28]. The previous study shows that BFS is a search strategy with efficient memory usage and preventing node regeneration being it easy to implement [29].

BFS is used to illustrate the node graph in a network and their relationship. The objective of this graph is to determine how to reach the goal.

BFS starts a search from a single root assigned to level 0, continues to visit the neighbors sequentially on the right side first, i.e. 13, 14, and 5 assigning them as level 1 (see Fig. 4). Then, it proceeds to the next level, i.e. 4, 10, 2, and 11 as level 2. The process continues until the last level, i.e. 15 and 3 as level 3, node 9 and 8 as level 4, node 12 as level 5. This is illustrated in Fig. 4. Furthermore, it creates a list that records the entire nodes connectivity in a network [30]. This property is a significant matter for network routing, because the shortest path from root to destination can constructed and detected.

Thus, the implementation of BFS in wireless ad hoc networks becomes challenging because:

- Network size, number of members, and their connectivity are unknown.
- It overcomes DFS problem i.e. ensuring a minimal solution if some solutions exist that meet requirements such as relay node in transmission range, the shortest path, minimum hop count, and successful transmission exist.

How BSF can select one of the several solutions is described in detail using Fig. 4. First, a root searches the connectivity between the members of entire network and determines the level for each node. After locating the destination, the root knows that there are some relay nodes (node 13, 14, and 5) in its transmission range (fixed radius model). BFS can handle this situation and select node 13 as relay node because it is on the shortest path to node 4 (destination).

---

**Algorithm 2 Breadth First Search [31]**

**Input:** Graph $G = (V,E)$

**Output:** BFS tree

`count ← 0`

mark each vertex in $V$ with 0

for each vertex $v ∈ V$ do

- if $v$ is marked with 0

  `bfs(v)`

`bfs(v)`

`count ← count + 1`

mark $v$ with `count`

initialize queue with $v$

while queue is not empty do

  `a ← front of queue`

for each vertex $w$ in $V$ adjacent to $a$ do

- If $w$ is marked with 0

  `count ← count + 1`

mark $w$ with `count`

add $w$ to the end of the queue

Remove $a$ from the front of the queue

---

**Fig. 3. DFS Visualization**

**Fig. 4. Selected route of network members in envSize 10x10**
In the fixed radius model, $r$ indicates a node’s communication range and scattered in $m \times n$ areas. A link between node $i$ and $j$, denoted as link $l_{ij}$, has been added to the graph when it is smaller or equal to $r$, like $l_{1,4}$ in Fig. 5.

Two kinds of neighbors should be considered, i.e. reachable and unreachable neighbors where nodes are located in fixed position. The distance of reachable neighbors per root is defined as:

$$D_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \leq r$$

(3)

$D_i$ denotes distance to node $i$, $x_0$ and $y_0$ are $x$ and $y$ coordinates of the root looking for the relay node in its transmission range.

Figs. 6 show the flowchart of BFG mechanism. The flowchart consist of the main flowchart (Fig. 6(a)), the subroutine BFS (Fig. 6(b)), and the subroutine Gossip (Fig. 6(c)).

### IV. Simulation and Results

Simulations are performed five times each in Matlab 7.10 to evaluate the performance of BFG, then the average of five running simulations was determined.

To prove BFG’s significant performance, the simulation results have been compared with the performance of DFS+Gossip. Simulations were conducted following three metrics:

- **Saved retransmission (SR):** the number of transmission that could be saved when a link probability is higher than its flooding probability.

$$SR = \sum_{i=1}^{n} t_i$$

(4)

where $t_i$ = number of retransmission at route $i$, $i = 1,2,...,n$.

- **Hop counts (HC):** the number of hops from a root node to a destination node:

$$HC = \sum_{i=1}^{n} h_i$$

(5)

where $h_i$ = hop route $i$, $i = 1,2,...,n$. 

(a) Main flowchart of BFG

(b) BFS Subroutine

(c) Gossip Subroutine

Figs. 6. Flowchart of BFG mechanism
Transmission failure (TF): a percentage of failure in transmitting a packet message from a source node to its neighbors in the network.

\[ TF = 100\% - \frac{p_s}{(n-1)} \]  

(6)

where \( p_s \) denotes a number of successful transmissions of a packet message from a root node and \( n \) as a number of nodes in network. The complete set of simulation parameters are shown in Table I.

| Simulation Parameters | Values             |
|-----------------------|--------------------|
| Number of nodes       | 15 – 200           |
| Transmission range    | 1 – 70             |
| Environment size      | 10x10, 50x50, 100x100 |

At the beginning of the simulation, 15 nodes are generated in 10x10 field with transmission (Tx) range = 3. Node 1 is the root which has role as the source node. Then, BFS works to determine the level of each node in the entire network. First, BFS determines the 0 level and 1 starts from node 1 (Fig. 7). Note that 0 level means that node is the root itself or it is unreachable from the root. Then, node 8 and 15 are indicated as level 2 (Fig. 8).

Finally, level 3 is determined at node 9, 7 and 14 (Fig. 9(a)), and level 4 at node 2 (Fig. 9(b)). After BFS finishes its searching, the next step is specifying the relay node to distribute the packet message.

Considering that the destination is node 10, it can be determined that delivering packet message results in a failure because node 10 is level 0, which means unconnected. In this case, a fixed radius model has been implemented to support BFS, however, node 10 is outside of root’s transmission range (Fig. 10).

Meanwhile, when the packet message is delivered to node 2 (destination), based on BFS algorithm and fixed radius model, the root will be connected to node 2 through node 13, 15, and 14 (Fig. 11). In other words, the root will send packet message in 4 hops to node 2.
When the number of node is 20 in the same size of field with Tx range = 3, the connection is different. Suppose the destination is node 12. To determine relay nodes to reach node 12, there are several route options (Fig. 12(a)). Node 2, 5, 7, 13, 14, 16, and 19 are the nodes to reach node 12, there are several route options. Suppose the destination is node 12. To determine relay nodes to reach node 12, there are several route options. When the number of node is 20 in the same size of field with Tx range = 40, the increasing node density (envSize vs nodes) in network, a number of SR is greater. Because of the increasing number of nodes in a Tx range, many nodes do not need to transmit the same message. Some important points correspond to Table II:

- No connected nodes at Tx range = 1 although the number of nodes is up to 200.
- Considering Tx range = 40, the increasing node density (envSize vs nodes) in network, a number of SR is greater. Because of the increasing number of nodes in a Tx range, many nodes do not need to transmit the same message.
- While the environment size of network is getting bigger, a number of SR is getting smaller. This can be explained as follows: the number of nodes in network is fixed, but they spread in a larger area, therefore nodes are few in Tx range and SR becomes less.
- At envSize 50×50, SR is available from 80 nodes (Tx range = 5). Meanwhile, at 100×100, SR is available from 80 nodes at Tx range = 10. The results indicate that the node density and Tx range increase, while the SR decreases.

### Table II

| Environment Size 50×50 | Number of Nodes |
|------------------------|-----------------|
|                        | 15 20 30 40 50 60 70 80 90 100 150 200 |
| 1                      | x x x x 0 x x x x 0 x x |
| 5                      | 0 0 0 0 0 0 0 1 1 3 1 8 |
| 10                     | 0 1 0 4 7 6 12 23 34 35 73 105 |
| 20                     | 2 7 9 22 25 30 42 52 68 75 112 168 |
| 30                     | 5 12 17 26 23 40 54 62 72 79 131 180 |
| 40                     | 7 15 23 31 47 57 67 78 89 142 163 |
| 50                     | 9 17 25 33 37 61 71 81 95 145 192 |

| Environment Size 100×100 | Number of Nodes |
|--------------------------|-----------------|
|                         | 15 20 30 40 50 60 70 80 90 100 150 200 |
| 1                       | x x x x 0 x x x x 0 x x |
| 5                       | 0 0 0 0 0 0 0 0 0 x x |
| 10                      | 0 0 0 0 0 0 0 1 1 3 1 18 |
| 20                      | 2 1 0 4 5 2 3 14 24 14 61 105 |
| 30                      | 3 4 17 9 8 13 19 43 25 64 159 |
| 40                      | 7 9 22 25 30 42 52 68 75 112 165 |
| 50                      | 1 13 9 27 26 36 45 60 70 80 120 177 |

| Environment Size 200×200 | Number of Nodes |
|--------------------------|-----------------|
|                         | 15 20 30 40 50 60 70 80 90 100 150 200 |
| 1                       | x x x x 0 x x x x 0 x x |
| 5                       | 0 0 0 0 0 0 0 0 0 x x |
| 10                      | 0 0 0 0 0 0 0 1 1 3 1 18 |
| 20                      | 2 1 0 4 5 2 3 14 24 14 61 105 |
| 30                      | 3 4 17 9 8 13 19 43 25 64 159 |
| 40                      | 7 9 22 25 30 42 52 68 75 112 165 |
| 50                      | 1 13 9 27 26 36 45 60 70 80 120 177 |

x = there is no connected node

### IV.1 Saved Retransmission

This subsection discusses the calculation of saved retransmission (SR) metric. This metric aims at reducing retransmission of message delivery to neighbors to save node’s power. Fig. 13 shows the graph of BFG saved retransmission in transmission range (Tx) up to 5. It shows the maximum number of SR increases as the number of nodes rises in a larger transmission range. At Tx range 5, the number of SR is the highest than others. It can be explained that when there are many nodes within the transmission range, BFG selects the highest probability and farthest node to retransmit. Therefore, the other nodes are not obliged to retransmit. Referring Figs. 12 and Table II, 20 nodes at environment size 50×50, at Tx range = 20, there are four levels to distribute the packet message to the entire network. Thus, SR has been counted when nodes are at level 3 and 4.
Consider 20 nodes at envSize 100×100, Tx range = 30, and the destination is node 17. The neighbors node are 2, 5, 7, 13, 14, 16, and 19 at level 1. Based on Gossip, a root calculate the link probability of neighbors. First, a root visited node 2, node 2 was initially set to 0 becomes 1. Because of \( p^{\text{flooding}} = 0.625 \) and \( p_{2,1} = 0.7117 \), thus \( p_{1,2} \leq p^{\text{flooding}} \), and there is no need to retransmit (i.e., saved retransmission). Second, a root visited node 5, \( p^{\text{flooding}} = 0.625 \), it is still constant and \( p_{5,1} = 0.7837 \). Because \( p_{7,1} \geq p^{\text{flooding}} \), then node 5 no need to retransmit (second saved retransmission). Third, a root visited node 7, \( p^{\text{flooding}} \) is still constant (0.625), but \( p_{7,1} = 0.3871 \). Because \( p_{7,1} \leq p^{\text{flooding}} \), then node 7 retransmits the message. Fourth, node 7 visited node 18, \( p^{\text{flooding}} = 1.6667 \) and \( p_{18,7} = 0.4638 \). Because of \( p_{7,18} \leq p^{\text{flooding}} \), node 18 retransmit. The process will continue until reaching a destination node. The topology configuration is shown in Fig. 14.

Table III shows the maximum SR of DFS+Gossip in various node density and Tx range. Table III shows no correlation between the number of nodes and Tx range. At Tx range 20 up to 40, the maximum SR is constant. Moreover, at Tx range = 50, the maximum SR is 1. But, at Tx range = 5, the maximum SR is up to 15. These results would seem to be contradicted with Table II, for the increasing number of nodes and Tx range. This occurs on a particular node within Tx range, because DFS does not perform the probability calculation for all links connected to a destination. Therefore, it can not select a link that has great potential to reach a destination. This way, the maximum SR is smaller than BFG. Figs. 15 compare and illustrate maximum saved retransmission in BFG and DFS+Gossip. Overall BFG has a better SR than DFS+Gossip, except at Tx range = 5, DFS+Gossip shows a better result. Referring to both results, BFG saved retransmission is 5 up to 58 times better than DFS+Gossip in various node densities and Tx ranges.

**Table III**

| DFS+Gossip Maximum Saved Retransmission |
|-----------------------------------------|
| Environment Size 50×50                  |
| Number of Nodes                        |
| Tx Range                               |
| Number of Nodes                        |
| 15                                     |
| 20                                     |
| 30                                     |
| 40                                     |
| 50                                     |
| 60                                     |
| 70                                     |
| 80                                     |
| 90                                     |
| 100                                    |
| 150                                    |
| 200                                    |

Referring to results of Table IV, we can summarize the points of the investigation.

- Consider Tx range = 1, TF occurs in all envSize because the neighbor nodes are outside of a root’s Tx range. Hence, there is no node which can retransmit a packet message to the destination.
- TF decreased as the number of nodes and Tx range increased.
- There is no TF in Tx range below 20 and 40 at envSize 50×50 and 100×100. This happens because nodes are generated randomly, spread in environment region, and have limited transmission range to connect other nodes.

Table V shows transmission failure of DFS+Gossip in various node densities and Tx ranges. This table indicates that for Tx range up to 5, almost 100% transmissions fail. Meanwhile, from Tx range = 10 up to 20, the increasing number of nodes causes TF decreases. And no TF at Tx range = 20 in all number of nodes. At Tx range 10 and 15, some nodes get TF up to 95%. Means some nodes outside of coverage although it retransmitted by another node. In other words, DFS+Gossip still unable to ensure zero TF although the number of nodes becomes enlarged.
Both methods indicate no TF at Tx range = 20, which means that all members of nodes within the network can receive the message.

- Overall BFG shows better TF at some Tx ranges, considering values over Tx range = 5, TF decreases as a number of nodes increases.
- BFG can reduce TF up to 20% from transmission failure of DFS + Gossip.

### IV.3 Hop Counts

This subsection describes hop count (HC) metric that was used for determining the number of hops of a root node to reach destination. HC calculation based on the highest probability of connected links within the network. This metric is related with many involved nodes, so a lot of energy resource is required to retransmit. Implementation of BFS does indeed make it easier and more efficient to determine the number of hops of nodes required to establish a route (Table VI). Table VI shows the maximum hop count BFG in various node density and Tx range.

![Image of graphs showing hop counts](image)

### Table IV

| Tx Range | 15 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 150 | 200 |
|----------|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| Environment Size | 50x50 |     |    |    |    |    |    |    |    |     |     |     |
| Number of Nodes |     |    |    |    |    |    |    |    |    |     |     |     |
| 1 | x | x | x | x | 1 | x | x | x | 1 | x | 0 |     |
| 5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 9 | 12 | 11 | 18 |     |
| 10 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 9 | 12 | 11 | 18 |     |
| 15 | 4 | 6 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 20 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 30 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 40 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 50 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 2 |     |

Table IV compared with Table V (envSize 50x50) indicates some points as follows:

- At Tx range = 1, both methods show 100% TF in a various number of nodes, because the Tx range of a root does not cover its neighbors.
- Both methods indicate no TF at Tx range = 20, which means that all members of nodes within the network

### Table V

| Tx Range | 15 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 150 | 200 |
|----------|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| Environment Size | 50x50 |     |    |    |    |    |    |    |    |     |     |     |
| Number of Nodes |     |    |    |    |    |    |    |    |    |     |     |     |
| 1 | x | x | x | x | 1 | x | x | x | 1 | x | 0 |     |
| 5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 9 | 12 | 11 | 18 |     |
| 10 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 9 | 12 | 11 | 18 |     |
| 15 | 4 | 6 | 4 | 4 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 20 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 30 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 40 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 50 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 2 |     |

x=no node is connected
There are several points that can be inferred from Table VI:

- Overall calculation of hops begins at Tx range = 5, because below Tx range = 5, most all nodes are unconnected to a root node.
- When the number of nodes and Tx range increases, HC decreases. This relates to the node density, when the node density is increasing, it is only requiring several HC to reach all the member’s nodes of network.
- When the number of nodes increases at a certain Tx range, HC almost decreases.
- At Tx range = 5 and 10, the number of HC is the highest at envSize 50x50 and 100x100.

Considering envSize 50x50, at Tx range = 5, the maximum HC is obtained where the number of nodes is 200. Meanwhile, at Tx range = 15, up to 200 nodes, average is 4 HC. Meanwhile at envSize 50x50, maximum HC require 4 to reach all members of the network at Tx range = 20. Likewise, when Tx range increases two times, it requires HC from 4 till 2 hops only, thus a reduction up to 50% (Table VI).

Table VII shows the maximum HC of DFS+Gossip in various node densities and Tx ranges. There are some points that can be inferred from Table VII:

- Generally, the greater the Tx range, the fewer the number of hops that can reach all network’s member.
- Over Tx range = 20, all number of nodes have a maximum two hops to reach all network members. As range transmission increases, a node can reach longer distances. So, a root node need fewer relay nodes to send a packet message to destination.
- Hop count calculation starts at Tx range = 5, because at Tx range = 1, all nodes within network unconnected.
- Maximum hops is 12, which means that a root node can send a message to all members of the network with 12 hops.
- At Tx range = 50, a maximum of 2 hops can reach all members of network.

| Tx Range | Environment Size 50x50 | Number of Nodes |
|----------|------------------------|-----------------|
| 15       | 1 x x x x x x x x x x x | 15             |
| 20       | 2 2 2 1 1 2 2 2 2 5 5 5 5 5 | 20             |
| 30       | 3 2 3 2 2 2 2 2 2 2 2 2 2 2 2 | 30             |
| 50       | 2 1 1 2 1 2 1 1 1 1 1 1 1 1 1 | 50             |

From Tables VI and VII (envSize 50x50), it can be inferred some points:

- Maximum hops with DFS+Gossip is 12, meanwhile BFG has 18 hops. It indicates BFG needs much time and node relay to reach all member network with the same topology configuration.
- At Tx range = 5, BFG requires more hops compared with DFS+Gossip, which may require extra energy. BFG expect efficient resource. However, it needs further investigation.
- Over Tx range = 15, the number of hops in the two methods are similar. Meanwhile, DFS+Gossip has the number of hops fewer than BFG in below Tx range = 15.

V. Conclusion

BFG is a route discovery mechanism in ad hoc network that applies BFS, fixed radius model and Gossip algorithms aiming at establishing a link to deliver a packet message to the desired destination. This route discovery mechanism focuses on the observed parameter i.e. the number of nodes, Tx range, and node area distribution. The measurement metrics for performance evaluation are transmission failure, hop count, and saved retransmission. Based on the simulation, it is proved that a route discovery is affected by the number of nodes that spread across the network. When the number of nodes increases, the possibility of the establishment of a route to a destination is greater but undeniable that Tx range has an important role, because it limits the ability of each node to send a packet message. Another important finding is that BFG shows the better performance in metric saved retransmission and transmission failure, and a similar number of hops needed at Tx range above 15 than DFS+Gossip. Meanwhile, when Tx range below 15, a number of hops need to pay attention, because BFG require hops more up to 50%.

This study has conducted a preliminary mechanism as a basic for next proposed routing protocol in wireless ad hoc network which associated optimization method in artificial intelligence.

References

[1] Enshaei, M., Mohd Hanapi, Z., Othman, M., A Review: Mobile Ad Hoc Networks Challenges, Attacks, Security, Vulnerability and Routing Protocols, (2014) International Journal on Communications Antenna and Propagation (IRECAP), 4 (5), pp. 168-179.
[2] Adnan, A., Hanapi, Z., Geographic Routing Protocols for Wireless Sensor Networks: Design and Security Perspectives, (2015) International Journal on Communications Antenna and Propagation (IRECAP), 5 (4), pp. 197-211.
[3] Senthil Kumar, R., Sareeshkumar, C., Cluster Based AODV Protocol to Minimizing the Route Rediscovery Process and to Improve the Lifetime of Network in MANET, (2014) International Review on Computers and Software (IRECOS), 9 (1), pp. 38-47.
[4] Alzabin, N., Mesleh, A., Hamed, S., Massadeh, S., AIHeyasat, O., AIQaisi, A., A QoS Based DSR Routing Protocol for MANETs Using Bandwidth, (2014) International Journal on Communications Antenna and Propagation (IRECAP), 4 (5), pp. 151-156.
[5] Ramachandran, P., Murukanantham, D., A Location Based-Energy Aware Routing Approach for Mobile Ad Hoc Networks, (2014) International Journal on Communications Antenna and Propagation (IRECAP), 4 (4), pp. 114-123.
[6] Yakine, F., Idrissi, A., Performance Comparison of ILP Models for QoS Topology Control to Conserve Energy in WANEt, (2015) International Journal on Communications Antenna and Propagation (IRCAP), 5 (16), pp. 328-336.

[7] Kuppuchamy, N., Manimegalai, R., Fuzzy Logic with Hybrid Optimization Approach for Optimal Route Selection in MANET, (2014) International Review on Computers and Software (IReCS), 9 (2), pp. 230-237.

[8] Sasson, Y., Cavin, D. and Schiper, A., Probabilistic Broadcast for Flooding in Wireless Mobile Ad hoc Networks, Proceedings of Wireless Communications and Networking (Page: 1124-1130 Year of Publication: 2003 ISBN:0-7803-7700-1).

[9] X. Wang, Probabilistic Flooding Simulation [Internet]. The MartWorks, Inc. June 27, 2005. Available from: https://www.mathworks.com/matlabcentral/fileexchange/7928-probabilistic-flooding-simulation/content/Gossip.m.

[10] Montresor, A., Intelligent Gossip, Proceedings of the 2nd International Symposium on Intelligent Distributed Computing (Page: 3-10 Year Publication: 2008 ISBN:978-3-540-85257-5).

[11] B. Blywis et al. A survey of flooding Gossip routing and related schemes for wireless multi-hop networks. Berlin: Institute of Computer Science, Freie Universitat Berlin, Germany. Technical Report No. TR-B-11-106, October 10, 2011. Available from: http://www.diss.fu-berlin.de/docs/servlets/MCRFileNodeServlet/FUDOCs_dervate_000000001761/2010-Gossip-Routing.pdf.

[12] D. H. Rothman, Modeling Environmental Complexity [Internet]. Boston: Massachusetts Institute of Technology, MIT OpenCourseWare: November 8, 2011 [cited 2016 Nov 25]. Available from: https://dspace.mit.edu/bitstream/handle/1721.1/99739/12-086-fall-2011/contents/lecture-notes/MIT12_86F11_percolation.pdf.

[13] S. R. Broadbent and D. R. Hammersley, Percolation processes I. crystals and mazes, (1957 Cambridge Philosophical Society, 53, pp. 629–641.

[14] Y. Sasson, D. Cavin and A. Schiper, Probabilistic Broadcast for Flooding in Wireless Mobile Ad hoc Networks, (2003) Wireless Communications and Networking Conference, 2, pp. 1124-1130.

[15] P. Erdos, A. Ren yi, On The Evolution of Random Graphs, (1960) Publications of the Mathematical Institute of the Hungarian Academy of Sciences, 5, pp. 17–61.

[16] Krishnamachari, B., Wicker, S.B., Bejar, R., Phase transition phenomena in wireless ad-hoc networks, Proceedings of Symposium on Ad-Hoc Wireless Networks. (Page: 2921 - 2925 Year Publication: 2001 ISBN:0-7803-7206-9).

[17] Z. Haas, F. Halpern, and L. Li, Gossip-Based Ad Hoc Routing, (2006) IEEE Transactions on Networking, 14 (3), pp. 479-491.

[18] Y. Li, K. Moavenni, O. Fieder, Regional Gossip Routing for Wireless Ad Hoc Networks, (2005) Mobile Networks and Applications, 10 (1), pp. 61-77.

[19] Z. Xu, Z. Shen, H., Adaptive Gossip-based Routing Algorithm, Proceedings of IEEE International Conference on Performance, Computing, and Communications (Page: 323-324 Year of Publication: 2004 ISBN: 0-7803-8396-6).

[20] A. D. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, Geographic Gossip: Efficient Averaging for Sensor Networks, (2008) IEEE Transactions on Signal Processing, 56 (3), pp. 1205–1216.

[21] C. P. Dwivedi, S. Sharma and V. Sharma, A Survey on Gossip-based Energy Conservation for Wireless Ad Hoc Networks, (2012) International Journal of Computer Technology and Electronics Engineering, 2 (4), pp. 31–36.

[22] Fauji, S., Kalpakis, K., A gossip-based energy efficient protocol for robust in-network aggregation in wireless sensor networks, Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops (Page: 166-171 Year of Publication: 2011 ISBN: 978-1-61284-888-4).

[23] AlShawi, I. S., Yan, L., Pan, W., Luo, B., A Fuzzy-Gossip routing protocol for an energy efficient wireless sensor networks, Proceedings of IEEE Sensors (Page: 1-4 Year of Publication: 2012 ISBN: 978-1-4577-1766-6).

[24] Chiussa, A., Fagnani, F., Schenato, L., Zampieri, S., Gossip algorithms for distributed ranking, Proceedings of American Control Conference (Page: 5468-5473 Year of Publication: 2011 ISBN: 978-1-4577-0080-4).

[25] Kempe, D., Dobra, A., Gehrke, J., Gossip-based computation of aggregate information, Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science (Page: 482-491 Year of Publication: 2003 ISBN: 0-7695-2040-5).

[26] Depth-First Search (DFS) - Algorithm and Data Structures. (2016, April 9). Retrieved from http://www.crazyforcode.com/depth-first-search-dfs-algorithms-data-structures/.

[27] Depth First Search. (2009, December 18). Retrieved from http://intelligence.worldofcomputing.net/ai-search/depth-first-search.html#V-DkcvQpr6s.

[28] Yoo, A., Chow, E., Henderson, K., McLendon, W., Hendrickson, B., Catalyurek, U., A scalable distributed parallel breadth-first search algorithm on BlueGene/L, Proceedings of the ACM/IEEE Super Computing (Page: 25 Year of Publication: 2005 ISBN: 1-59593-061-2).

[29] Zhou, E. A. Hansen, Breadth-first Heuristic Search, (2006) Artificial Intelligence, 170 (4-5), pp. 385-408.

[30] V. K. Akram, O. Dagdeviren, Breadth-First Search-Based Single-Phase Algorithms for Bridge Detection in Wireless Sensor Network. (2013) Sensors, 13, pp. 8766-8813.

[31] Algorithm Chapter 5: Decrease and Conquer. (2012 April 11). Retrieved from http://www.slideshare.net/chidabdu/algorithm-chapter-5.
Takuya Terasawa currently is an Associate Professor at Graduate School of Media Science, Tokyo University of Technology. His research interests are in Computer Network, Internet Technology, Computer Architecture, Parallel Processing, Distributed System, Simulation, Performance Evaluation, and Operating System. E-mail: terasawa@stf.tetu.ac.jp

Mauridhi Hery Purnomo currently is a Professor at Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember Surabaya, Indonesia. His research interests are in Artificial Intelligent Technique for Power System and Biomedical Engineering. E-mail: hery@ee.its.ac.id