Vulnerable Node Detection and Route Recovery in Dynamic Complex Networks with the Ant Colony Optimization

ReisaDewi[1], Tae-Hyong Kim[2]
Computer Eng. Department, Kumoh National Institute of Technology
61 Daehak-ro, Gumi, Gyeongbuk39177 Republic of Korea
[1] reisadewi@gmail.com
[2]taehyong@kumoh.ac.kr (corresponding author)

***Abstract*—Vulnerability is an important issue that needs to be solved in order to optimize the performance of complex networks. Dynamism in the topology of a complex network isan important factor in vulnerability analysis of complex networks.We analyses the vulnerability of dynamic complex networks and deals with vulnerable nodes in such networks by focusing on ad-hoc networks, which are typical dynamic networkssharing the properties of complex networks.This paper represents a node-type model with respect to network vulnerability as a semi-Markov process, and definesthe vulnerability index of ad-hoc networks by throughput measurement and graphical analysis. We propose an algorithm based on the ant colony optimization (ACO) in order to detect vulnerable nodesand to reconstruct a new robust routefor ad-hoc networks. The simulation results show that the proposed algorithm lowers the vulnerability index andreduces vulnerable nodes with maintainingthe throughput of the network.**

Keyword-Vulnerable Node Detection, Route Recovery, Ad-hoc Networks, Ant Colony Optimization

I. INTRODUCTION

Vulnerability is an important issue to address the problem in optimization and robustness of complex networks. There are many researches related to vulnerability, such as vulnerability management [1], network vulnerability identification[2], and detecting and repairing node vulnerability [3]. Vulnerability factors of complex networks are divided into internal factors (e.g. network element, network topology and network policy) and external factors (e.g. infrastructure condition). Some researchers focus on structural robustness of the network properties e.g., degree distribution [4], betweeness [5] and link removal [6]. Those researches show that the more heterogeneous the networks are, the more robust theyare to random failures[7]. However, the networks become more vulnerable if they have highly connected or important nodes, (e.g. scale free networks). Furthermore, the researches regarding relations between network properties and vulnerability are insufficient to maintain robustness of the networks, since real networks evolve dynamically.

An ad-hoc network is an example of complex networks. Ad-hoc networkshave been prominent issuesin computer networks in the past few years due to the advancement of network technology and increased demands of high connectivity. Ad-hoc networks continue to grow into large scale networks, whether theyare fixed networks or mobile networks. The self-configuration or decentralize infrastructure is one of the advantages of ad-hoc network to improve flexibility and robustness of the networks. Information exchange in ad-hoc networks can be done without a server node. Additionally, each node can be a host as well as a router simultaneously [6]. Despite of ease and fast deployment of ad-hoc networks, they did not reduce the complexity of networks. Furthermore, there are some challenges in ad-hoc networks that need to be considered such as dynamic topology where users join and leave frequently, whichcauses rapid change of network routing, limited resources (bandwidth and power), and security problem. Due to these challenges in ad-hoc networks, vulnerability analysis become important in order to maintain the network's performance.

In general, there are three main components in vulnerability of complex networks, resistance (robustness), resilience and adaptive capacity [7]. The resistance is the ability to resist random changes in its system environment, resilience is the ability to recover to its structural property after the occurrence of a perturbation, while adaptive capacity is the system ability to adapt or maintain the function and property. The vulnerability in ad-hoc networks caused by its dynamic topology was attracting the attention of some researchers [6]. With limited resources, highly mobile nodes, and security threats make challengingthe vulnerability analysis in ad-hoc networks. To overcome this problem, some researchers identified the vulnerable nodes in the network using several different algorithms, e.g. multiple-objective optimization, game theory [8], and genetic algorithm. These algorithms are able to find the best solution in relatively short time, for both local and global solution.

This paper proposes an algorithm based on the ant colony optimization (ACO)to detect vulnerable nodes and to recover communication in a network efficiently [9] which finds the optimum local solution to lead the global solution. The ACO is a meta-heuristic algorithm and it is inspired by the foraging behaviour of real ants. Ants

are known for their ability to find the shortest path to their destination and their adaptability to dynamic topology. If ants meet an obstacle in their path, they look for a new path in relatively short time without the need to go back to their nest. This ability to adapt in ants is necessary to respond any vulnerability in the network.

The rest of this paper is organized as follows. Section 2 introduces definitions and preliminaries of vulnerability in complex networks. Section 3 describes modelling and analysis of vulnerability issues. Section 4 and 5 present the proposed algorithm and its evaluation respectively. Finally, section 6 presents the conclusion and direction for future work.

II. PRELIMINARIES AND RELATED WORK

This section presents preliminaries of the vulnerability analysis of complex networks andad-hoc networks. Several recent studies related to the ACO are introduced as well.Vulnerability analysis of complex networks can be divided into three parts: structural, non-structural, and functional vulnerabilities.

A. Structural Vulnerability

Generally, structural vulnerability of complex network is divided into node vulnerability and edge vulnerability [10]. Vulnerability node is measured by node properties, e.g. degree, betweenness [11], clustering coefficient [12] and edge vulnerability by link removal.For directed graph, the*node degree*of a node is the sum of incoming and outgoing edges of that node. As for undirected graph, the node degree is the total edges of the node [13]. The node degree number is calculated using the vulnerability analysis when in the local state but not for global state. Not all the node with higher degree are vulnerable than other nodes.Mishkovski et. al. [5] defined metric of network vulnerability as a normalized average edge *betweenness* of a network, where network is modelled as a simple graph, $G=(V,E)$, where V is a set of vertices or nodes together and E is the edges or lines. The average edge betweenness (b) of graph G is defined as:$b(G) = \frac{1}{|E|}\sum_{i\in E} b_i$. The *clustering coefficient* is a measurement number that uses the node degree to show the nodes that tend to cluster together, i.e., zero when there is no clustering and one for maximal clustering. Maximal clustering happens when the network is consisted of disjoint cliques. Clustering coefficient is one of the vulnerability analyses parameter, because the higher the clustering coefficient is, the more vulnerablethe system is [4]. The objective of *link removal* is to optimize the robustness of the system i.e., to minimize the spread of infection. Eva et.al [14] proposed the quadratically constrained quadratic program (QCQP) algorithm by finding the equivalent set assignment in the network which minimizes a partition cost function.

B. Non-structural and Functional Vulnerability

Non-structural vulnerability consists of management of policy of each node in the network [2,15,16]. For example, the network security policy that defines which port is accessible and policy that defines the properties of each user.Functional vulnerability refers to internal and external factors of the network related to three main aspect: exposure, sensitivity and capacity of response [17]. Exposure is the overlapping factor of the dynamic networks as a result of variability in service. Sensitivity exhibits the emerging properties of dynamics network and capacity of response is the respond to variability in order to maintain the properties.

The vulnerability function of a network N can be state as the impact of a certain disturbance d in the value of the target node n is given by its exposureε, sensitivity φ and capacity σ of respond at time t: $V_{d,n,t}(N)=f(\varepsilon,\varphi,\sigma)$.These classifications of vulnerability analysis in complex network become the base to perform the vulnerability assessments [18]. The vulnerability analysis flow chart is defined in Fig. 1.

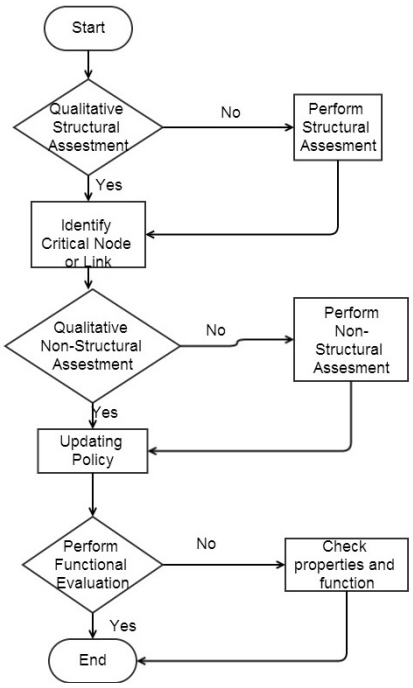


Fig. 1. Vulnerability Analysis Procedure

C. Vulnerability Analysis of Ad-Hoc Networks

There are several vulnerability issues in ad-hoc networks, e.g., failed node, cooperative node, selfish node and malicious node [19].A*failed* node is the node unable to perform the service or application, a *cooperative* node is the node that sends the packet data to the destination node, a *selfish* node is the node that exploits its resources for its personal interest, and a *malicious* node is the node that has intention to disrupt the service in the network. Each of the vulnerability issue in ad-hoc networks is mapped into the vulnerability analysis classification. The vulnerability analysis in ad-hoc networks is defined in TABLE I.

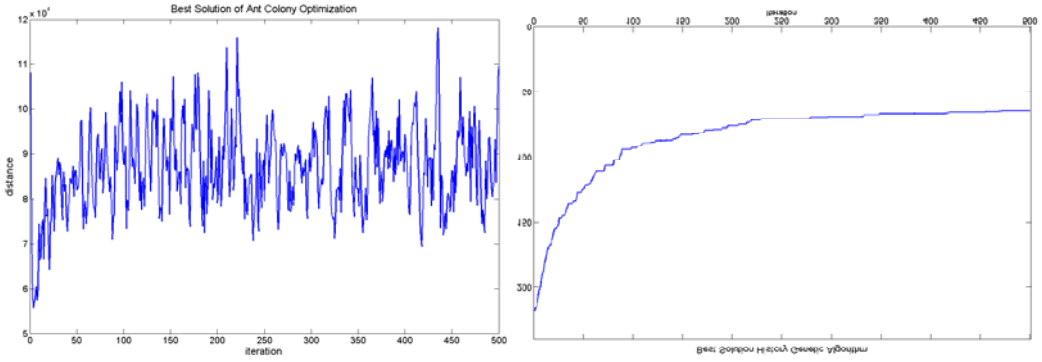
TABLE I. Vulnerability Issues in Ad-hoc Networks

Vulnerability Issue	**Structural Analysis**	**Non-Structural Analysis**	**Functional Analysis**	**Existing Solutions**
Failed Node	- broken link - out of coverage area - H/W breakdown	- S/W defect - authorization failed - authentication failed	- battery run out - no signal	- power constraint - restart - add more coverage area/hotspot
Cooperative Node	- routing issue and path calculation - as hub or bridge in network	- complexity applica-tion - security check	- resource spent fas-ter	- BW utilization - limited packet send to each node - randomize forwarding stra-tegy
Selfish Node	- linked with higher utility node - packet loss higher	- used most of BW - forwarding service to other node	- saving resources for personal interest - do not forward information to next nodes	- watchdog mechanism - identifying and isolation - introduction of billing system - trusted the 3rd party system - sharing reputation information
Malicious Node	- attached with high-er-degree node	- internal attack - DOS and routing attack - interpretation and eavesdropping - black hole attack	- overload packet in the network	- authentication scheme - data integrity system - cryptography method

D. Recent Related Applications of Ant Colony Optimization

Dewi et al. [20] presented the type of the ACO for routing algorithm in mobile ad-hoc networks. The ACO meta-heuristic is targeted towards optimization problems that can be solved as shortest path problems on graphs. The ACO is also used for modelling, the vulnerability detection [21], and performance evaluation [9]. Ants construct solutions to optimization problems by moves around in a graph and use stigmergy to communicate their experiences.

The ACO has been used for problem detection in several application areas. Francesca et al [22] used the ACO to reduce the state explosion problem when looking for deadlocks in complex distribution network. Xu et al. [23] also proposed the ACO for detecting community in bipartite network.Zang et al. used the ACO for fault section location detection in concurrent system [24].



(a) (b)

Fig. 2. Comparison between the ACO (a) and the genetic algorithm (b)

Fig. 2 shows that the genetic algorithm (GA) detects the solution faster compared to the ACO algorithm. However, if there is a change in the networks, the GA becomes slower compared to the ACO in acquiring the solution. Although the GA is the fastest algorithm to find the best solution, when there is a sudden change in the network condition, the GA cannot obtain the best solution due to the local optima problem. In contrast, the ACO algorithm can acquire the best solution even though in dynamic network condition.

III. MODELLING AND ANALYSIS OF VULNERABILITY ISSUES

A. Modelling Node Vulnerability as a Semi-Markov Process

There are some constraints in modelling the vulnerability issue of ad-hoc networks as follows. First, the model is based on node vulnerability issue. The definition of each type of node is based on the packet throughput. Second, the transition of each type of node issues is defined by transition probability that consists of the power consumption rate and the failure rate as the time goes to the infinity. Third, there is no difference between a source node and a destination node, since the process starts from the cooperative state. Finally, the resource is limited, so, as the time goes to infinity, the resource power will go to zero and there is no concept of recharging.

From the constraint above, the state transition for each type of node vulnerability [25] can be state as follows.A cooperative node can remain cooperative or becamea selfish node, a malicious node or a failed node in the future. A failed node has toremain at a failed state. A selfish node can remain selfish, or became a malicious node or a failed node. Amalicious node can remain malicious or became a failed node in the future.

The model parameterѕ of the vulnerability issues are based on (1) the number of packets received and transmitted in period of T, and (2) the cost power function $C_i(t)$.Acquiring data at a nodein T, denoted by $D_A(T)$, is defined as the number of packets received in T minus the number of packets consumed in T at that node. Forwarding data in T at a node, denoted by $D_F(T)$, is defined as the number of packets transmitted in T minus the number of packets initiated in T at that node. Data utilization at a nodein T, denoted by $D_U(T)$, is defined as the number of packets initiated in T plus the number of packets consumed in T at that node. From those definitions, the fraction of forwarding at a node in T, denoted by $C_F(T)$, is defined as: $C_F(T) = D_A(T) - D_F(T) \geq 0$. The condition that $C_F(T)$ equals to zero is called perfect forwarding which means that all acquired data has been forwarded. Nonzero $C_F(T)$ indicates that some data has been ignored at a node without being forwarded. The data utilization rate at a node in T, denoted by $R_U(T)$, is defined as $D_U(T)/D_T(T)$, and the data consuming rate at a nodein T, denoted by $R_C(T)$, is defined as $D_F(T)/ D_A(T)$.By assuming that each node initially starts with the same power energy, the cost power function $C_i(t)$ is defined as $C_i(t) = \rho_i \cdot F/R_i(t)$, where ρ_i is the power consumption rate, F is the full-charge battery capacity of node i and $R_i(t)$ is the remaining power left of node i at time t.If the

cost power function getssmaller than the threshold, the transition may occur.The concept of this packet flow analysis at a nodeis shown in Fig. 3.

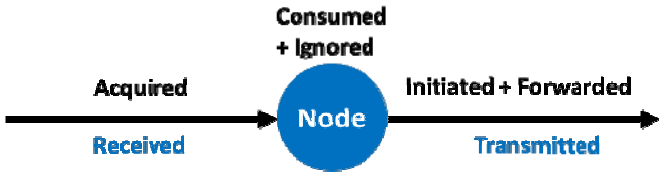


Fig. 3. Packet flow analysis at a node

The vulnerability issues in ad-hoc networks and the state transition is shown is Fig. 4. The notations used are listed as follow. *a*is the probability that a cooperative node remains cooperative in the next state. *b*, *c*, and *d* are the probabilities that a cooperative node becomes selfish, malicious, and failed in the next state, respectively. *e* is the probability that a selfish node remains selfish in the next state. *f* and *g* are the probabilities that a selfish node becomes malicious and failed in the next state, respectively. *h* and *i* are the probabilities that a malicious node remains malicious and becomes failed in the next state, respectively.The semi-Markov process can be denoted by$Z(t)$; $t \leq 0$, where the state space(S)are *C* (cooperative), *S* (selfish), *M* (malicious), and *F* (failed). The semi-Markov process is defined by two matrices$P_{i,j}$ and $Q_{i,j}$, where $P_{i,j}$ is the transition probability matrix from state *i* to state*j*and$Q_{i,j}$is the distribution function matrix of time spent from state *i* to state *j*.The probability of next state of each node and the transition time distribution can be represented as following two matrices:

$$P_{i,j} = \begin{matrix} \\ C \\ S \\ M \\ F \end{matrix} \begin{matrix} \begin{matrix} C & S & M & F \end{matrix} \\ \begin{pmatrix} a & b & c & d \\ 0 & e & f & g \\ 0 & 0 & h & i \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}, \quad Q_{i,j} = \begin{pmatrix} Q_a(t) & Q_b(t) & Q_c(t) & Q_d(t) \\ 0 & Q_e(t) & Q_f(t) & Q_g(t) \\ 0 & 0 & Q_h(t) & Q_i(t) \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

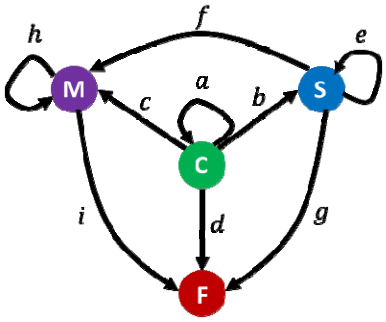


Fig. 4. Semi-Markov model of node vulnerability

From the transition process it can be seen that the final state is the failed node state. The transition probability of the next state ($P_{i,j}$) of semi-Markov process is presented in the following equation, which depends on the current state and the next state transition. The current state of the node $X(t)$ at time *t* is

$$X(t) = \lim_{\epsilon \to} X(t+\epsilon), \text{ for } \epsilon > 0.$$
$$Q_{i,j} = \Pr\{X_{t+1} = j, N_{t+1} - N_t \leq t | X_t = i\}.$$
$$P(X_t = i | X_0 = j) = \frac{\pi_i \mu_i}{\Sigma_{j \in S} \pi_j \mu_j},$$

where X_t and X_{t+1} represent the states of the node at time *t* and *t*+1 respectively, N_t and N_{t+1} represent the transitions at times *t* and *t*+1 respectively, π_i is the stationary probability of state *i* of X_n and as $t \to \infty$.To calculate the $\pi_i$and μ_i, we can start from $P_{i,j}$ and $Q_{i,j}$. As the node changes,each state depends on the time and thepower.Therefore the complete definition of $P_{i,j}$ can be as follows:

$$P_d = P_g = P_i = \max_i(\tfrac{1}{F}, \tfrac{1}{\rho_i}),$$
$$P_c = P_f = P_h = \min_i(\tfrac{1}{F}, \tfrac{R_i(t)}{\rho_i}),$$
$$P_b = P_e = \tfrac{1}{F} \cdot \tfrac{R_i(t)}{R_i(t)-1}, \text{ and}$$
$$P_a = \tfrac{F}{\rho_i R_i(t)}.$$

After determining$P_{i,j}$ and $Q_{i,j}$, we can obtainπ_i from the steady-state probability equation,

$\bar{\pi} = \bar{\pi}p,\ \sum_{i\in S}\pi_i = 1, \pi_i \geq 0$. Therefore,

$\pi_{i,j}(t) = \lim_{\Delta t\to 0}\frac{P(X_n=j, t<T_n\leq t+\Delta t | X_{n-1}=i)}{\Delta t}$,

$\pi_{i,j}(t) = Q_{i,j}\cdot P_{i,j}(t)dt$, and $\mu_i = \sum_{j\in S} P_{i,j}\mu_{i,j}$.

B. Vulnerability Component Analysis

The rules to identify the type of a node in an ad-hoc network as shown in the directed graph of Fig. 5 are based on packet and link availability.

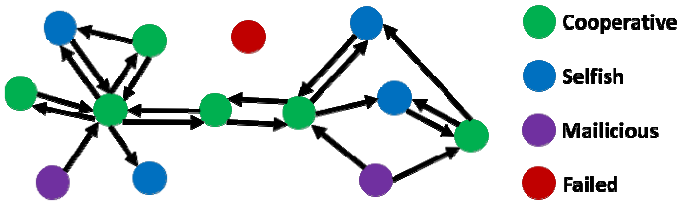


Fig. 5. A sample directed graph for node degree analysis

A cooperative node has the same number of in-degree and out-degree.A failed node has zero in-degree and zero out-degree.A selfish node has more in-degree than out-degree.A malicious node has less in-degree than out-degree. When we denote H_C and H_M the consuming and malicious thresholds respectively, the properties shown in TABLE II hold for eachtype of nodes.

TABLE II. Node type analysis based on packet and link availability

Node type	Property
Cooperative	$R_C(T) > H_C$
Selfish	$0 < R_C(T) < H_C$
Malicious	$R_C(T) > H_M$
Failed	$R_C(T) = 0$

The betweenness of node*k* for directed graphB_k is presented as:

$$2B_k = \sum_i \sum_j \frac{P_{ijk}}{P_{ij}}; i \neq j \neq k.$$

In order to to compute the betweenness of node*k*, the dependency of all pairs must be computed as follows:

$$d_{(i,j)} = d_{(i,k)} + d_{(k,j)}.$$

The value of P_{ikj} will be 0 if$d_{(i,j)} < d_{(i,k)} + d_{(k,j)}$. Based on the degree analysis, each type of nodes appears to have the following property regarding the betweenness. A cooperative node has normal degree of betweenness. A failed node has no betweenness. A selfish node has lower betweenness than a cooperative node. A malicious node has lower betweenness than a selfish node.

The clustering coefficient $C_i$of a vertex v_i in a directed graph$G = (V, E)$, where V and E is the set of vertices and edges respectively, is given by:

$$C_i = \frac{|\{e_{jk}: v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$$

, where $N_i$is the neighbourhood for a vertex $v_i$defined by $N_i = \{v_j: e_{ij} \in E \vee e_{ji} \in E\}$, and k_i is the number of neighbours of vertex v_i. Since a high clustering coefficient indicates a high degree of clustering, we can say that the higher the clustering coefficient, the more robust a network.

Now we define the vulnerability index of an ad-hoc network I_V as the product of the average betweenness and clustering coefficient as follows:

$$I_V = \bar{B}\cdot\bar{C}$$

, where $\bar{B} = \sum_i B_i/|V|$and $\bar{C} = \sum_i C_i/|V|$.

IV. The Proposed Algorithm

A. Graphical Representation of Vulnerability Analysis

Fig. 6 shows a graphical representation of the proposed vulnerability analysis for ad-hoc networks. In the measurement phase, incoming and outgoing packets from each node in the network are measured in the specific period of time. According to the throughput of incoming and outgoing dataat a node, each link between nodes is decided tounidirectional or bidirectional. In the metric phase, vulnerability parameters, the node betweenness

and the clustering coefficient, are calculated with the node degree information. The types of nodes, cooperative, selfish, malicious, and failed, are also decided in this phase. In the vulnerability phase, vulnerable nodes are decided with the node vulnerability decision function and the network vulnerability is also decided with the vulnerability index.

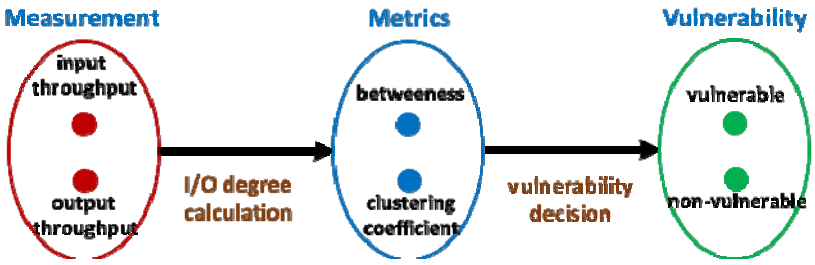


Fig. 6. Graphical representation of the proposed vulnerability analysis

B. Vulnerable Node Detection and Route Recovery with the Ant Colony Optimization

Ants are generated based on two conditions,periodic checking and conditional triggering.First, ants are distributed randomly to check the local and global solution periodically. Second, ants are released if there is an event triggering, for example, the significant change of node degree distribution or big amount of packet lost detected. The key points of ant colony optimization (ACO) in detecting vulnerability nodes are divided into two modes, forward and backward. In the forward mode, ants generated according to the algorithm walk randomlyon a graph. The probability of the next path selection by ants$P_{(i,j)}$willbe shownbelow. In the backward mode, ants find the new solution, then goes back from the new path or new node to the source node leaving the pheromone on the trail. The proposed vulnerable node detection algorithm based on the ACO is depicted in Fig. 7. To choose the next node *j*, the forward mode has the following transition rule to balancing between the exploitation and exploration.

$$\mathrm{j} = \begin{cases} argmax_{h\in\Omega}\{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}\} & if\ q \le q_0 \\ P_{(i,j)} = \dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{h\in\Omega}[\tau_{ih}]^{\alpha}[\eta_{ih}]^{\beta}} & otherwise \end{cases}$$

, where $q_0 \in (0,1)$ is the constant parameter define exploitation and exploration, q is a random number between 0 and 1,$P_{(i,j)}$is the probability of choosing the next node in the forward mode, $h \in \Omega$ is the set of nodes which are feasible to be visited from node i,$\alpha$is the parameter weight of the influence trails, $\beta$is the parameter weight of the visibility, $\eta_{i,j}$is a heuristic value, and $\tau_{i,j}$ is the pheromone level laid between node *i* and *j*.

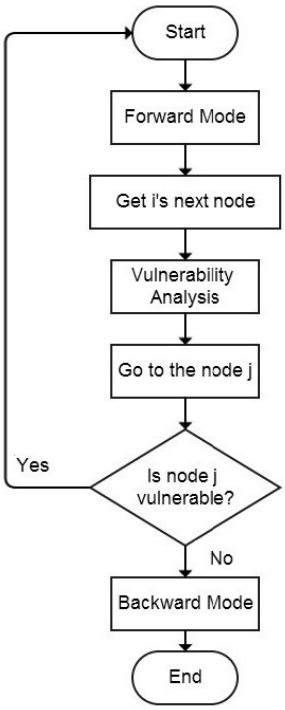


Fig. 7. Flowchart of the proposed vulnerable node detection algorithm based on the ACO

The ACO algorithm was run with the following parameters settings.The value of α and β are set to 1.0 because it is the value of maximum network capacity reached;i.e., a short path is discovered faster. $\tau_{i,j}$ is set to 1/*length*(*S*), where *S* is the set of connections from the source to vulnerable nodeswith circular paths deleted.Finally the evaporation value is set to 0.999.

To reduce the overhead, ants stop when there are feedback-ants from other nodes. The pheromone value and heuristic value are gradually increased when there are feedback-ants. The path with lower pheromone value is reported or removed if there is no pheromone laying at all. The pheromone value and heuristic value are updated as the global solution is found. The periodical check updates the global solution and explore a new path.

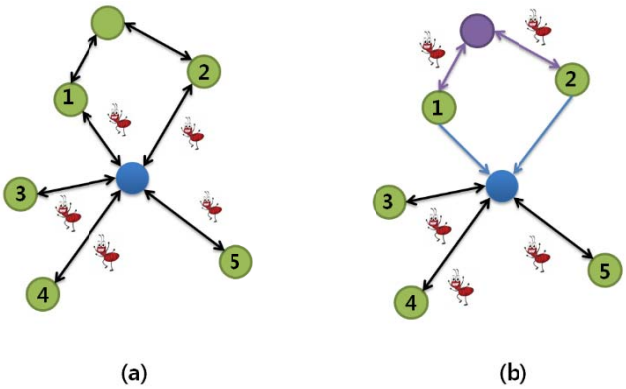


Fig. 8. Detection of selfish nodes by ants ((a) before, (b) after)

In Fig. 8, neighbouring nodes of the selfish node generate the ants to check the network condition. Ants walk randomly through available paths, and the destination node has to send the feedback ant to the source node. If there is no feedback ant, the source node releases other ants to check the available path and makesa new path, if there is no available path. The feedback-ants leave the pheromone trail along the path. The gradually increasing pheromone marks the stronger path and the pheromone on the path to a selfish node is decreased over time because of evaporation.

Fig. 9 presents the algorithm for vulnerable node detection and route recovery. Vulnerable node detection is based on the condition that a degree change ofora throughput change is above the threshold. The fixed candidate boundary and the flexiblelink boundary depend on the node distance and the available link. If there is no available path from the source node to the destination node, the link boundary increases and the new node is linked from the candidate boundary. The link boundary is decreased if the network throughput is above the threshold or by periodic checking.

```
Build the candidate boundary (B_C) table.
Build the link boundary (B_L) table.
Gather connection lists for all nodes and links.
Measurethe throughput and degree change.
whilethe throughput <the threshold do
    The source node (n_S) sends ants to alinked neighbour node.
    if (the receiver node (n_R) ≠ the destination node (n_D) then
        Forward ants to the next linked node.
    else
        Send backward ants to the source node (n_S).
        if(backward-ants travel time < evaporation time) then
            Retain the link.
        else
            Increase the link boundary (B_L).
            Get a new linked node from the candidate boundary (B_C)
            Send ants to the new linked node.
        end if
    end if
end while
```

Fig. 9. Vulnerable node detection and route recovery algorithm

V. EVALUATION

A. Simulation Modelling and Environment

The simulation was performed using Matlab R2012a. A scale free network was built by using 5 nodes as initial network and grows into 100 nodes in the final form, based on the power law degree distribution. The nodes are distributed randomly in 1000 meters by 1000 meters. Fig.10 shows the topology of thegenerated scale free network in the final form.

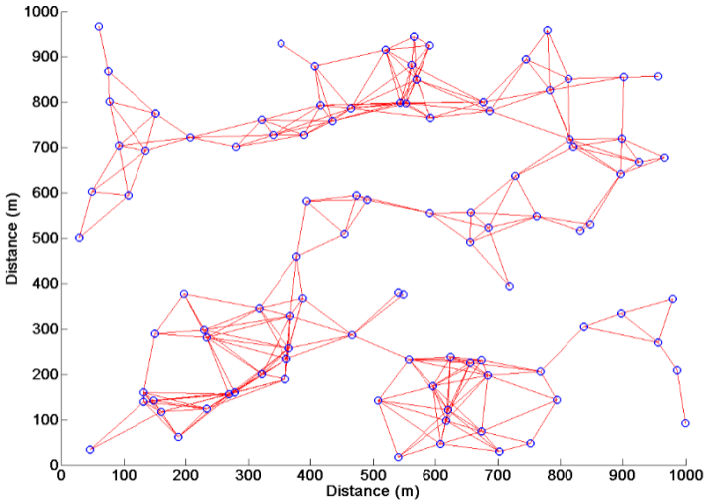


Fig. 10. The topology of the scale free network used in the simulation

The parameter values used in the simulation are presented in TABLE III. The parameters to be measured are the node degree, the betweenness, the clustering coefficient, and the vulnerability index. The simulation was run 50 times to get more real data.

TABLE III. Parameter values used in the simulation

Parameter	Value
Total number of nodes	100
Total area	1000 m × 1000 m
Topology	Scale free network
Initial nodes (seed)	5
User mobility	random
Input data	random
Simulation time	20 minutes

B. Simulation Results

Fig.11 presents a comparison of network throughput between the originalsituation and the situation when the network uses the proposed ACO-based algorithm. The figure shows that the proposed algorithm does not decrease network throughput at all, even if there is no significant throughput increase.

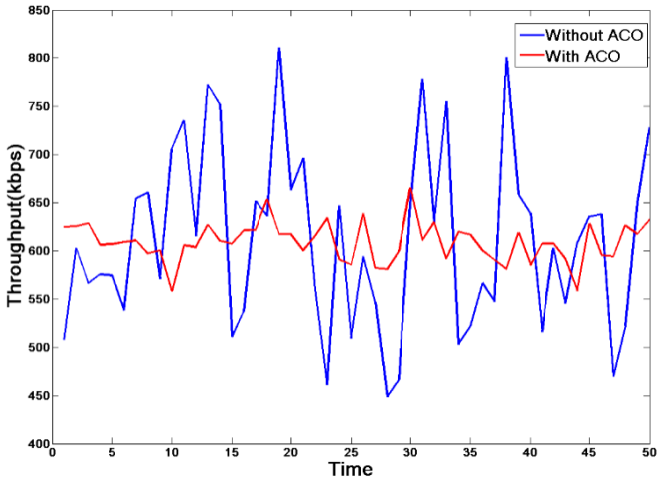


Fig. 11. Comparison of network throughput from the simulation results

Random user mobility and frequent changes of topology produced fluctuationsof the throughput in the scale free networks. The simulation shows that the standard deviation of the throughput in case without the ACO is higher than that in case with the ACO, which means the degree of randomness in case with the ACO is lower than that in case without ACO. This is because the ACO is better in handle the traffic at each node. The ACO seems to be able tohandle user movements and link changes efficiently.

Fig.12 demonstrates the comparison of network vulnerability index between the without-ACO situation and the with-ACO situation. The figure shows that the proposed ACO-based algorithm is able to lower the vulnerability index of a scale free network.

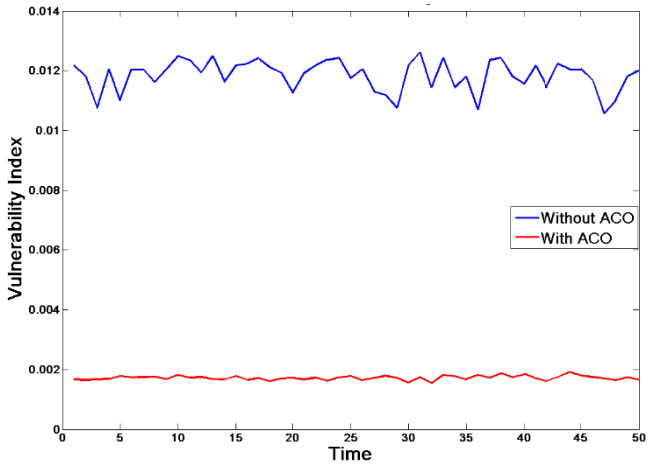


Fig. 12. Comparison of the vulnerability index from the simulation results

Fig. 13 shows the relation between the throughput and the vulnerability index in the simulation results. It can be seen that nodes with higher throughput normally have also higher vulnerability index in the network. However, the proposed algorithm does not show any significant change of the throughput.In terms of the vulnerability index, some nodes are more vulnerable than other nodes. For example, while node numbered 3 has low throughput but high vulnerability index, node numbered 33 has high throughput but low vulnerability index. It should be stated that the factors affecting the node vulnerability are the connections, the betweenness and the clustering coefficient.

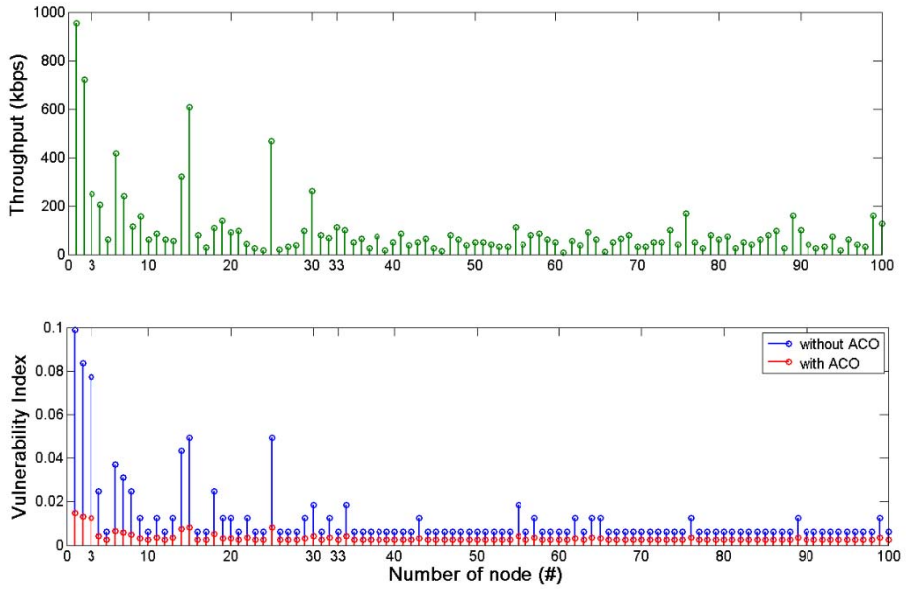


Fig. 13. Throughput and vulnerability index of each node in the simulation

Fig. 14 shows that the proposed algorithmreduced the number of vulnerable node in the network because the algorithm detected vulnerable nodesand found new paths. In some simulation points, there were more vulnerable nodes detected with the proposed algorithm but the algorithm loweredsome vulnerable nodes rapidly by creating new available paths.

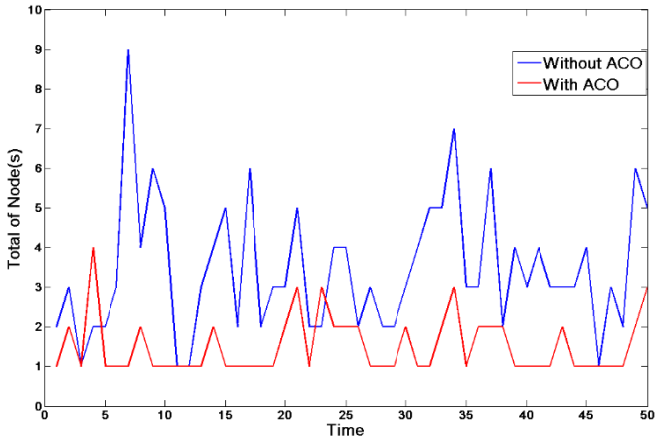


Fig. 14. Comparison of the vulnerability index from the simulation results

C. Discussion

As shown in the simulation results, the proposed algorithm has several advantages. In dynamic networks, the proposed algorithm is able to detect vulnerable nodesand to find new available paths. Additionally, the proposed algorithmhas high adaptability to dynamism in topology, i.e. frequent changes in network topology.

The experiment was conducted bya discrete event simulation in which the topology changes are not continuous with respect to time. Moreover, it is assumed that there is no dead node due to power failure.Each link has the same bandwidth and the throughput is calculated using number of packets; and there is no signal problem.

In order to further improve the validation of the proposed method, simulation using real-time and real-world condition is required in the future. Additionally, combining the current Matlab simulation with a powerful network simulator such as Riverbed Modeler is a good step to increase the credibility of the proposed algorithm.

VI. CONCLUDING REMARKS

This paperpresenteda systematical vulnerability analysis of complex networks using ad-hoc networksand proposed an ACO-based algorithm for vulnerable node detection and route recovery. The vulnerability analysis is composed of three parts; structural analysis, non-structural analysis and function analysis. This paper presented a node-type model with respect to network vulnerability as a semi-Markov process. The vulnerability index of ad-hoc networks was obtained through node throughput measurement and incoming and outgoing

degree calculation, which can indicate that a node is vulnerable or not.Vulnerable nodescan be detected by the ACO approach utilizing both heuristic and pheromone values. Furthermore, when a vulnerable node is detected, the propose algorithm finds a new available path without the vulnerable node. The simulation results have shown that the proposed algorithm based on the ACO lowers the vulnerability index with maintainingthe throughput. For future work, dynamic and adaptive vulnerability thresholdsmay be studied to improve the performance of the proposed algorithm.

ACKNOWLEDGMENT

This paper was supported by Research Fund, Kumoh National Institute of Technology.

REFERENCES

[1] Swami Iyer, Timothy Killingback, BalaSundaram, and Zhen Wang. Attackrobustness and centrality of complex networks. PLoS ONE, 8(4):e59613, 042013.NGM

[2] E. Al-Shaer, W. Marrero, A. El-Atawy, and K. Elbadawi. Network configurationin a box: towards end-to-end verification of network reachability and security.In Network Protocols, 2009. ICNP 2009. 17th IEEE International Conferenceon, pages 123-132, Oct 2009.

[3] K. Daouda, Z. Pascale, and P. Francois. Infrastructure network vulnerability.In Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE), 2011 20th IEEE International Workshops on, pages 305-312, June 2011.

[4] Jianxin Wang, Min Li, Huan Wang, and Yi Pan. Identification of essentialproteins based on edge clustering coeficient. Computational Biology and Bioinformatics, IEEE/ACM Transactions on, 9(4):1070-1080, July 2012.

[5] Igor Mishkovski, Mario Biey, and LjupcoKocarev. Vulnerability of complexnetworks. Communications in Nonlinear Science and Numerical Simulation,16(1):341-349, 2011.

[6] Jayraj Singh, Arunesh Singh, and Raj Shree. An assessment of frequentlyadopted unsecure patterns in mobile ad hoc network: Requirement and security management perspective. International Journal of Computer Applications,24(9):34-39, June 2011. Published by Foundation of Computer Science.

[7] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complexnetworks: Structure and dynamics. Physics Reports, 424(4):175-308, 2006.

[8] Xiaoying Zhang, Chi Guo, and Lina Wang. Using game theory to reveal vulnerability for complex networks. In Computer and Information Technology (CIT),2010 IEEE 10th International Conference on, pages 978-984, June 2010.

[9] Li Guowei, Xu Zhenqiang, and Yuan Peiyan. Performance evaluation of antcolony algorithm with new parameters. In Computational Intelligence and Design (ISCID), 2010 International Symposium on, volume 1, pages 198-200, Oct2010.

[10] Claudio M. Rocco S. and Jos Emmanuel Ramirez-Marquez. Vulnerability metrics and analysis for communities in complex networks. Reliability Engineering& System Safety, 96(10):1360-1366, 2011.

[11] Jing Huang, Xin Ji, and Huiying He. A model for structural vulnerability analysis of shipboard power system based on complex network theory. In ControlEngineering and Communication Technology (ICCECT), 2012 InternationalConference on, pages 100-104, Dec 2012.

[12] M.R. Brust, D. Turgut, C. H C Ribeiro, and M. Kaiser. Is the clusteringcoeficient a measure for fault tolerance in wireless sensor networks? In Communications (ICC), 2012 IEEE International Conference on, pages 183-187,June 2012.

[13] Lifang Guo, Huimin Xu, and K. Harfoush. The node degree for wireless ad hocnetworks in shadow fading environments. In Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on, pages 815-820, June 2011.

[14] Eva A. Enns, Jeffrey J. Mounzer, and Margaret L. Brandeau. Optimal link removal for epidemic mitigation: A two-way partitioning approach. MathematicalBiosciences, 235(2):138- 147, 2012.

[15] Gu Yue-sheng, Zhang Bao-jian, and Yu Zhou. Wireless network security policybased on integrated vulnerability management. In Networking and Digital Society, 2009. ICNDS '09. International Conference on, volume 2, pages 225-228,May 2009.

[16] E.S. Al-Shaer and H.H. Hamed. Firewall policy advisor for anomaly discoveryand rule editing. In Integrated Network Management, 2003. IFIP/IEEE EighthInternational Symposium on, pages 17-30, March 2003.

[17] P. Chopade and M. Bikdash. Structural and functional vulnerability analysisfor survivability of smart grid and scada network under severe emergenciesand wmd attacks. In Technologies for Homeland Security (HST), 2013 IEEEInternational Conference on, pages 99-105, Nov 2013.

[18] C.G. Ghedini and C. Ribeiro. A framework for vulnerability management incomplex networks. In Ultra Modern Telecommunications Workshops, 2009.ICUMT '09. International Conference on, pages 1-8, Oct 2009.

[19] Jianhua Zhang, Xiaoming Xu, Liu Hong, Shuliang Wang, and Qi Fei. Attackvulnerability of self-organizing networks. Safety Science, 50(3):443-447, 2012.

[20] ReisaRahmatuDewi, Tae-Hyong Kim, and Soo-Young Shin. Ant colony optimization based routing algorithms for mobile ad hoc networks (manet): Asurvey. pages 362-365. THE INSTITUTE OF ELECTRONICS ENGINEERSOF KOREA, 2013.

[21] Xie Hui, Wu Min, and Zhang Zhi-ming. Using ant colony optimization to modeling the network vulnerability detection and restoration system. In IndustrialMechatronics and Automation, 2009. ICIMA 2009. International Conferenceon, pages 21-23, May 2009.

[22] G. Francesca, A. Santone, G. Vaglini, and M.L. Villani. Ant colony optimizationfor deadlock detection in concurrent systems. In Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual, pages 108-117,July 2011.

[23] CHEN Ling XU Yongcheng. Community detection on bipartite networks basedon ant colony optimization. Journal of Frontiers of Computer Science andTechnology, 8(3):296, 2014.

[24] Zhang Jingning, Zhou Ren, and Zhong Kai. Application of improved ant colonyalgorithm in fault-section location of complex distribution network. In ElectricUtility Deregulation and Restructuring and Power Technologies (DRPT), 20114th International Conference on, pages 1067-1071, July 2011.

[25] A.H. Azni, R. Ahmad, and Z.A. Mohamad Noh. Correlated node behavior inwireless ad hoc networks: An epidemic model. In Internet Technology AndSecured Transactions, 2012 International Conference for, pages 403{410, Dec2012.