The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 17, No. 3, pp.29-45, Jun. 30, 2017. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2017.17.3.29
JIIBC 2017-3-4

최대 수명을 갖는 AODV 라우팅 프로토콜 실험 설계

Experimental Design of AODV Routing Protocol with Maximum Life Time

김용길*, 문경일**

Yong-Gil Kim*, Kyung-Il Moon**

요 약 애드 혹 센서 네트워크는 분산형 구조와 구축으로 특징지어지며 센서 네트워크는 낮은 이동성과 엄격한 에너지 요구 조건 등을 제외하고는 애드 혹 네트워크의 기본적인 특징을 모두 갖추고 있다. 기존 프로토콜은 내결함성, 분산 컴퓨팅, 견고성, 확장성 및 신뢰성과 같은 특성 간에 서로 다른 보완성을 제공한다. 지금까지 제안된 무선 프로토콜은 매우 제한되어있어 일반적으로 단일 기지국 또는 센서 데이터 수집에 중점을 두었다. 그러한 제약을 가지는 주된 이유는 네트워크 활동을 유지하기 위해 최대 수명을 유지하기 때문에 네트워크 수명은 애드 혹 네트워크에서 중요한 설계 기준이며 모든 노드가 라우터 역할을 수행하여 에너지 부족인한 일부 노드가 동작하지 않으면 다른 노드로 통신할 수 없다. 본 논문에서는 네트워크 노드의 에너지 통신을 최적화하기 위한 실험적인 애드 혹 주문형 거리 벡터 라우팅 프로토콜을 제안 한다 부하 분산은 경로 선택 단계에서 소진된 노드의 선택을 피하고 노드 간 에너지 사용의 균형을 유지하고 네트워크 수명을 극대화한다. 전송 제어 단계에서는 신호 전송 범위를 증가시키는 높은 전송 전력의 선택과 홉 수를 줄이고 네트워크 연결 비용의 부담을 줄이는 낮은 전력 수준 사이의 균형이 필요하다.

Abstract Ad hoc sensor network is characterized by decentralized structure and ad hoc deployment. Sensor networks have all basic features of ad hoc network except different degrees such as lower mobility and more stringent energy requirements. Existing protocols provide different tradeoffs among some desirable characteristics such as fault tolerance, distributed computation, robustness, scalability and reliability. wireless protocols suggested so far are very limited, generally focusing on communication to a single base station or on aggregating sensor data. The main reason having such restrictions is due to maximum lifetime to maintain network activities. The network lifetime is an important design metric in ad hoc networks. Since every node does a router role, it is not possible for other nodes to communicate with each other if some nodes do not work due to energy lack. In this paper, we suggest an experimental ad-hoc on-demand distance vector routing protocol to optimize the communication of energy of the network nodes.The load distribution avoids the choice of exhausted nodes at the route selection phase, thus balances the use of energy among nodes and maximizing the network lifetime. In transmission control phase, there is a balance between the choice of a high transmission power that lead to increase in the range of signal transmission thus reducing the number of hops and lower power levels that reduces the interference on the expense of network connectivity.

Key Words : Ad Hoc On-Demand Distance Vector, Routing Protocol, Wireless Ad Hoc Network

*정회원, 조선이공대학교 컴퓨터보안과
**정회원, 호남대학교 공과대학 컴퓨터공학과(교신저자)
접수일자: 2017년 4월 8일, 수정완료: 2017년 5월 8일
게재확정일자: 2017년 6월 9일

Received: 8 April, 2017 / Revised: 8 May, 2017 /
Accepted: 9 June, 2017
**Corresponding Author: kimoon@honam.ac.kr.
department of computer engineering, Honam university, Korea

Ⅰ. Introduction

Wireless ad-hoc network has been widely discussed for many years. It consists of two or more devices equipped with wireless communications and networking capability. Such devices can communication with other nodes that immediately within their radio range or one that is outside the radio range. The wireless ad-hoc network does not have gateway, and all nodes work as the gateway. The decentralized functionality of wireless ad hoc networks is suitable for a variety of applications where central nodes are not used. Even though the overall capacity of such networks has some practical restrictions, it may advance the scalability of wireless ad hoc networks compared to wireless managed networks[9]. Ad hoc networks can have minimal configuration and quick deployment in case of emergency situations such as natural disasters or military conflicts. The presence of adaptive routing protocols makes ad hoc networks to be formed quickly. Wireless ad-hoc networks can be classified as mobile ad-hoc network, wireless mesh network and wireless sensor network[14]. The mobile ad-hoc Network has a self-configuring structure of mobile devices connected by wireless links. Each device moves independently in anyway, and changes its links to other devices frequently[11]. It forwards traffic unrelated to its own use, and therefore is a router. The primary issue of the mobile ad-hoc Network is to equip each device to maintain the information needed to properly route traffic. Such network operates by itself or is connected to the larger network. The mobile ad-hoc network is a kind of wireless ad hoc networks that has a routable network on top of a link layer. Wireless 802.11/Wi-Fi network has made the mobile ad-hoc networks a popular research field. Different protocols have been evaluated based on measure such as the packet drop rate, the overhead by the routing protocol, end-to-end packet delays and network throughput[10][12]. The wireless mesh network is regarded as a special type of wireless ad-hoc network. It is a communications network consisted of radio nodes with a mesh topology. It often consists of mesh clients, mesh routers and gateways. The mesh clients are usually laptops, cell phones and other wireless devices while the mesh routers forward traffic to and from the gateways which need not connect to the network. The wireless mesh network often has a more planned configuration, and deploys to provide cost effective connectivity over a certain area. The mesh routers may be mobile, and be moved according to specific demands arising in the network. Since the mesh routers have not the restricted resources compared to other nodes in the network, they can be exploited to perform more resource intensive functions. In this way, the wireless mesh network differs from an ad-hoc network. Mesh networks may involve either fixed or mobile devices. An important application is VoIP. By using a quality of service scheme, the wireless mesh supports local telephone to be routed through the mesh. The wireless sensor network consists of autonomous sensors to manage environmental parameters such as temperature, sound, vibration, pressure and motion. It sends the environmental data through the network to a main location. This network has bi-directional property to manage the sensor activity. It is now used in many application areas such as industrial process control, machine health, environment and habitat, healthcare applications, home automation, and traffic control[3]. The wireless sensor network is consisted of sensor nodes from a few to several hundreds or even thousands. Each node is connected to several sensors. It has typically a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source[7]. A sensor node has various sizes from that of a shoebox down to the grain size of dust. The limited sizes of sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology can vary

from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the network hops can be routing or flooding[4]. Ad hoc On-Demand Distance Vector (AODV) routing protocol is intended for mobile nodes in an ad hoc network[6]. It offers a fast dynamic link adaption, less memory overhead, low network utilization, and determines unicast routes to destinations within the network. It uses destination sequence numbers to ensure loop freedom at all times, avoiding infinity problem related to typical distance protocol[13]. However, it focuses on a single base station or gathering sensor data. This reason is due to maximum lifespan to maintain network activities. The network lifetime is an important factor in the ad hoc networks[8]. Since every node plays a router role, it is not possible for other nodes to communicate with each other if some nodes die early because of energy lack. For protocols that maximize the overall network lifetime, the main focus is to distribute the energy consumption among all nodes in a balanced manner. If the route with maximal energy saving is always chosen for delivery, the subset of nodes along this route will be over utilized and therefore drained in a short period of time which may lead to network partitioning[15]. Energy management is an important design problem while developing a routing and distributed protocol to increase the lifespan of wireless ad-hoc networks[2]. In homogeneous WSN, the sensor nodes have the same lifespan if the energy consumption rate is uniform. However, in heterogeneous WSN, each sensor node has different means in terms of storage, processing, sensing, etc. The heterogeneous sensor node needs a more powerful microprocessor and large memory[1]. Also, it has high bandwidth and longer transmission range and requires more reliable data transmission. Further, it makes special situations like different energy level, line powered, or replaceable battery[5]. This study suggests an experimental energy-specific AODV routing protocol. At the route selection step, the exhausted sensor nodes exclude for load distribution. Thus, the efficient energy use is attained between the sensor nodes and maximizing the network lifespan. In transmission control step, a balance is considered as regards high transmission power that increases in terms of signal transmission. Thus, the number of hops and power levels are decreased by reducing the interference on link connection. In section 2, the AODV related protocol messages are described. Section 3 represents operation scenarios that are related to energy efficient designing in wireless ad-hoc network. The network operations and various message processing are discussed for energy aware protocol. Section 4 shows a wireless ad-hoc network implementation by using MATLAB Simulink, and finally the paper is concluded in section 5.

II. Background

The AODV message types are composed of Route Request (RREQ), Route Reply (RREP) and Route Error (RERR). The messages are received through User Datagram Protocol (UDP), and use normal IP header processing. AODV operation requires the RREQ to be disseminated throughout the network. This dissemination has the range by the time-to-live in the IP header. If the endpoints have valid routes to each other, AODV does not work. The node broadcasts a RREQ to search a route to the destination in case that a route to a new destination is required. A route is determined in case that the RREQ reaches either the destination itself, or an intermediate node with a valid route entry to the destination. The valid route input represents a route entry for the destination whose related sequence number is at least as great as that contained in the RREQ. The route is available by unicasting a RREP back to the RREQ origination. Each node receiving the request caches a route back to the request creator, thus the RREP can be unicast from the destination to that creator[1]. Nodes manage the link status of next hops in the active routes. If a link break

is appeared in the active route, a RERR message is used to notify other nodes with the link loss occurrence. The RERR message denotes those destinations that are not reachable by way of the broken link. Each node keeps a precursor list that contains the IP address for each its neighbor nodes. The precursor list is very easily acquired during the generation process of a RREP message. If the RREP has a positive prefix length, then the RREQ generator is included among the precursors for the subnet route. AODV is a routing protocol, and it deals with route table management. Route table keeps even for short lived routes, which are generated to store reverse paths originating RREQs. AODV uses some fields with each route table inputs. They are corresponding to destination IP address, destination sequence number, valid destination sequence number flag, other state and routing flags, network interface, hop count, next hop, list of precursors and life time.

Table 1 illustrates a format of RREQ. Field J is join flag, R is repair flag that is reserved for multicast, G is unnecessary RREP flag, D is destination only flag, U is an unknown sequence number, Reserved is ignored on reception, and hop count field is the number of hops from the creator IP address to the node handling the request. RREQ ID is a sequence number identifying the special RREQ when taken in conjunction with the IP address originating node. Destination IP address is the IP address of the destination for which a route is desired. Destination sequence number is the latest sequence number received in the past by the originator for any route towards the destination. Originator IP address is the IP address of node that originated the route request. Originator sequence number is the current sequence one to be used in the route entry pointing towards the originator of the route request.

표 1. RREQ 메시지 형식
Table 1. RREQ Message Format

0				1				2					3	
0	..	8	9	0	1	2	..	0	1	2	3	..	0	1
1		J	R	G	D	U	reserved						hop count	
RREQ ID														
destination IP address														
destination sequence number														
originator IP address														
originator sequence number														

Table 2 illustrates a format of RREP message. Bit A corresponds to the required acknowledgment, and is used when the sent RREP message is unreliable or unidirectional. When the RREP message contains the A bit set, the receiver of RREP is expected to return a RREP acknowledgement message. If prefix size is nonzero, the 5-bit prefix size specifies that the indicated next hop is used for any nodes with the same routing prefix as the requested destination. The prefix size allows a subnet router to supply a route for every host in the subnet by the routing prefix. The subnet router ensures reachability to all hosts sharing the indicated subnet prefix. When the prefix size is nonzero, any routing and precursor data are kept with respect to the subnet route. The hop count field is the number of hops from the originator IP address to the destination IP one. For multicast route requests, it represents the number of hops to the multicast tree object sending the RREP. Destination IP address is the IP address of the destination for which a route is supplied. Destination sequence number is related to the route. Originator IP address is the IP address of node that originated the RREQ for the supplied route. Life time is the time in milliseconds for which nodes receiving the RREP consider the route to be valid.

표 2. RREP 메시지 형식
Table 2. RREP Message Format

0				1				2					3	
0	..	8	9	0	..	8	9	0	1	2	3	..	0	1
2		R	A	reserved				prefix size					hop count	
destination IP Address														
destination sequence number														
originator IP address														
life time														

Table 3 illustrates a RERR message format. Field N is not delete flag, and sets when a node has performed a local link repair, and upstream nodes do not delete the route. Destination count is the number of unreachable destinations included in the message. Unreachable destination IP address is the IP address of the destination that is unreachable due to a link break. Unreachable destination sequence number is the sequence number in the route table for the destination listed in the past unreachable destination IP address field. The RERR message is sent whenever a link break causes one or more destinations to be unreachable from some neighbors of the node. Table 4 illustrates a route reply acknowledgement (RREP-ACK) message format. RREP-ACK is sent in response to a RREP message with the A bit set. It is typically done when there are some risks of unidirectional links preventing the completion of a route discovery cycle.

표 3. RERR 메시지 형식
Table 3. RERR Message Format

0				1		2					3	
0	..	8	9	0	..	0	1	2	3	..	0	1
2		N	Reserved							destination count		
unreachable destination IP address												
unreachable destination sequence number												
unreachable destination IP addresses(Additional)												
unreachable destination sequence numbers(Additional)												

표 4. RREP-ACK 메시지 형식
Table 4. RREP-ACK Message Format

0				1		
0	...	8	9	0	...	5
4		reserved				

Ⅲ. AODV Operation Scenarios

In this section, we present some scenarios under which nodes generate RREQ, RREP and RERR messages for unicast communication towards a destination, and how the message data are handled. In order to process the messages correctly, certain state information has to be maintained in the route table entries for the destinations of interest.

1. Route Table Entries and Route Requests

Every route table has the sequence number for the IP address of destination node. It is updated whenever a node receives new data for the sequence number from RREQ, RREP, or RERR messages. AODV has the destination sequence number to ensure loop freedom of all routes. A destination node increases its sequence number in two situations. It increases immediately its own sequence number before a node originates from a route discovery. This prevents conflicts with the reverse routes previously built towards the RREQ originator. Also, it increases immediately its own sequence number before a destination node originates from the RREP. It updates its own sequence number to the maximum of its current sequence number and the destination sequence number in the RREQ packet. When the destination increases its sequence number, it treats the sequence number value as an unsigned number. If the sequence number has already been assigned to be the largest possible number, then has a zero value. The negative numbers is not relevant to the AODV sequence numbers. When a node receives an AODV control packet from a neighbor, or updates a route for a particular destination, it uses the route table for the destination input. The route table input is generated in case that there is no corresponding input for that destination. The sequence number is either determined from the data belonged to the control packet, or the valid sequence number field has a false value. The route is only updated if the new sequence number is either higher than the destination sequence number in the route table, or the sequence numbers are equal. However, the hop count increment is smaller than the existing hop count in the routing table. Also, the route can be updated in case that the sequence number is unknown. The life time field of the routing table is either determined from the control packet, or it is set to the active route timeout. This route is used to

send any data packets, and satisfies any dominant requests. Whenever a route is used to send a data packet, its active route life time field of the source, destination and next hop towards the destination is not less than the current time plus active route timeout. Since the route between each pair is assumed to be symmetric, the active route life time for the past hop is not less than the current time plus the active route timeout. A node disseminates a RREQ when it needs a route to a destination and has no available one. It can happen in case that the destination is previously unknown to the node, or that a past valid route to the destination terminates or is notified as invalid case. The destination sequence number in the RREQ message is the last known destination one for this destination and is received from the destination sequence number field in the routing table. If the sequence number is unknown, the unknown sequence number flag is set. The originator sequence number is the node itself sequence number, which is increased prior to insertion in a RREQ. The RREQ ID field is increased by one from the last RREQ ID used by the current node. Each node maintains only one RREQ ID. The hop count field is set to zero. Before broadcasting the RREQ, the originating node buffers the RREQ ID and the originator IP address of the RREQ for path discovery time. When the node receives the packet again from its neighbors, it will not reprocess and send the packet again. An originating node often assumes to have bidirectional communications with a destination node. In such cases, the originating node does not have a route to the destination node. The destination node has also a route back to the originating node. Thus, a RREP generation by an intermediate node must be accompanied for delivery to the originating node through some actions that notify the destination about a route back to the originating node. The originating node selects this work in the intermediate nodes through the G flag setting. A node should not originate more than RREQ rate limit messages per second. After broadcasting a RREQ, a node waits for a RREP. If a route is not received within network traversal time, the node tries again to find a route by broadcasting another RREQ, up to a maximum of RREQ retry time at the maximum TTL value. Each new attempt must increase and update the RREQ ID. For each try, the TTL field of the IP header is set according to the specified way, in order to control over how far the RREQ is disseminated. Data packets waiting for a route are buffered, and the buffering is a queue structure. If a route discovery has been attempted RREQ retry times at the maximum TTL without receiving any RREP, all destined data packets must be dropped from the buffer and a destination unreachable message is delivered to the application. To reduce the network congestion, retries use a binary exponential back off through a source node at route discovery for a single destination. The first time a source node broadcasts a RREQ, it waits network traversal time for the reception of a RREP. If a RREP is not received within that time, the source node sends a new RREQ. When the RREP waiting time is computed after sending the second RREQ, the source node uses a binary exponential back off. Hence, the RREP waiting time corresponding to the second RREQ is twice of network traversal time. If a RREP is not received within this time period, another RREQ may be sent, up to the RREQ retry time additional attempts after the first RREQ. For each additional attempt, the waiting time for the RREP is multiplied by two, so that the time conforms to a binary exponential back off. To avoid unnecessary dissemination of RREQ, the originating node uses an expanding ring search method. In an expanding ring search, the originating node initially uses a TTL start in the RREQ packet IP header and sets the timeout for receiving a RREP to ring traversal time. The TTL value used in the ring traversal time is equal to the TTL value in the IP header. If the RREQ times are out without a corresponding RREP, the originator broadcasts the RREQ again by TTL increment. It continues until the TTL set in the RREQ reaches TTL threshold. The timeout for receiving a RREP is the ring

traversal time. When it has all retries traverse the entire ad hoc network, it can be achieved by configuring TTL start and TTL increment both to be the same value as the network range. The hop count of an invalid routing table is corresponding to the last known hop count to that destination in the routing table. When a new route to the same destination is required at a later time, the TTL in the RREQ IP header is set to the hop count plus TTL increment. Thereafter, following each timeout the TTL is changed by TTL increment until TTL threshold is reached. Once TTL is set to the network diameter, the timeout for waiting for the RREP is set to the network traversal time. A finished routing table input must not be erased before current time plus delete period. Otherwise, the soft state related to the route is disappeared.

2. Route Reply Generation

When a node receives a RREQ, it first creates or updates a route to the past hop without a valid sequence number, then tries to determine whether it has received a RREQ with the same originator IP address and RREQ ID within at least the last path discovery time. If such a RREQ is received, the node discards the newly received RREQ. The followings describe some actions taken for RREQs that are not discarded. First, it increments the hop count value in the RREQ by one, to account for the new hop through the intermediate node. Then the node searches for a reverse route to the originator IP address, using longest-prefix matching. When necessary, the route is created, or updated using the originator sequence number from the RREQ in its routing table. This reverse route is needed if the node receives a RREP back to the node that originated the RREQ identified by the originator IP address. When the reverse route is created or updated, the following steps on the route are also carried out: Originator sequence number is compared to the corresponding destination sequence number in the route table and copied if greater than the existing value. Valid sequence number field is set to true. Next hop in the routing table is the node from the received RREQ. It is obtained from the source IP address in the IP header and is often not equal to the originator IP address field in the RREQ message. Hop count is copied from the RREQ message. Whenever a RREQ message is received, the life time of the reverse route input for the originator IP address is set to be the maximum of existing lifetime and minimal lifetime, where minimal lifetime is: (current time) + 2(network traversal time) -2(hop Count)×(node traversal time). The current node can use the reverse route to forward data packets in the same way as for any other route in the routing table.

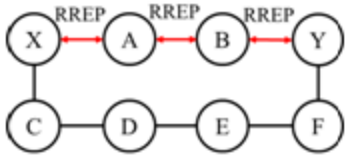

그림 1. 전달 및 전송 경로
Fig. 1. Propagation and Forward path

Reverse path is available to send a RREP message when a broadcast RREQ packet reaches at a node having a route to the destination. While transmitting this message, the forward path can be set. Once the forward path is set (see Fig.1), the data transmission can be started. Data packets to be transmitted are buffered locally and transmitted in a queue when a route is setting. After a RREP was forwarded by a node, it can receive another RREP. This RREP can be discarded or forwarded according to its destination sequence number as the followings: If it has a greater destination sequence number, the route must be structured again, and it is forwarded. If the past destination sequence numbers and this RREP is equal, but it has a lesser hop count, it should be selected and forwarded. Otherwise all later incoming ones are ignored. The path for route request and route reply is as Fig 2 from source (X) to destination (Y).

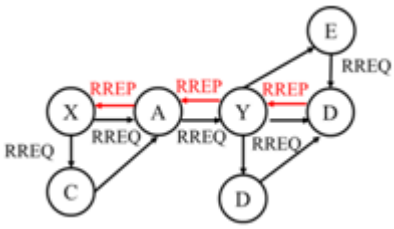

그림 2. 경로 발견
Fig. 2. Route discovery

The node updates and broadcasts the RREQ to the related address on each of its configured interfaces if a node does not generate a RREP following the processing rules, and if the incoming IP header has TTL larger than 1. To update the RREQ, the TTL or hop limit field in the outgoing IP header is decreased by one, and the hop count field is incremented by one, to account for the new hop through the intermediate node. Lastly, the destination sequence number for the requested destination is set to the maximum of the corresponding value received in the RREQ message, and the destination sequence value currently maintained by the node for the requested destination. However, the forwarding node does not change its maintained value for the destination sequence number, even if the value received in the incoming RREQ is larger than one currently maintained by the forwarding node. If a node generates a RREP, then the node discards the RREQ. It might turn out that the destination does not receive any of the discovery messages if the intermediate nodes reply to every transmission of RREQs for a certain destination. In this case, the destination does not learn of a route to the originating node from the RREQ messages. It causes the destination to initiate a route discovery. In order that the destination learns of routes to the originating node, the originating node sets the G flag in the RREQ if for any reason the destination is likely to need a route to the originating node. If a RREP is returned with the G flag setting by intermediate node, an unnecessary RREP is unicasted to the destination node. If the generating node is the destination itself, it increments its own sequence number by one in case that the sequence number in the RREQ packet is equal to that incremented value. Otherwise, the destination does not change its sequence number before generating the RREP message. The destination node places its sequence number into the destination sequence number field of the RREP, and enters the value zero in the hop count field of the RREP. If the node generating the RREP is not the destination node, but instead is an intermediate hop along the path from the originator to the destination, it copies its known sequence number for the destination into the destination sequence number field in the RREP message. The intermediate node updates the forward route input by placing the last hop node into the precursor list for the forward route entry. The intermediate node also updates its route table for the node originating the RREQ by placing the next hop towards the destination in the precursor list for the reverse route input. It places its distance in hops from the destination count field in the RREP. The lifetime field of RREP is computed by subtracting the current time from the termination time in its route table. The unnecessary RREP is to be sent to the desired destination containing the following values in the RREP message fields. Hop count as indicated in the node's route table entry for the originator, destination IP address (IP address of the node that originated the RREQ), destination sequence number (Originator sequence number from the RREQ), Originator IP address (IP address of the destination node in RREQ) and lifetime. The unnecessary RREP is then sent to the next hop along the path to the destination node, just as if the destination node had already issued a RREQ for the originating node and this RREP was produced in response to that RREQ. The RREP that is sent to the originator of the RREQ is the same whether or not the G bit is set. When a node receives a RREP message, it searches for a route to the previous hop by using longest prefix matching. When needed, a route is created for the previous hop, but without a valid sequence number. Next, the node

then increments hop count value by one to account for the new hop through the intermediate node. Upon comparison, the existing entry is updated only in the following circumstances. The sequence number in the routing table is marked as invalid in route table input. Destination sequence number in the RREP is greater than the destination sequence number and the known value is valid, or the sequence numbers are the same, but the route is marked as inactive, or the sequence numbers are the same, and the new hop count is smaller than the hop count in route table entry. If the route table input to the destination is created or updated, then the following actions occur: (1) the route has an active state. (2) The destination sequence number has a valid marking. (3) The next hop is assigned in the route input to be the node from which the RREP is received, which is indicated by the source IP address field in IP header. (4) The hop count is set to the value of new hop count. (5) The finished time has the current time plus the value of the life time in RREP message, and the destination sequence number is the destination sequence number in RREP message.

The current node can subsequently use this route to forward data packets to the destination. If the current node is not the node indicated by the originator IP address in the RREP message and a forward route has been created or updated as described the above, the node controls its route table input for the originating node to find the next hop for the RREP packet, and then forwards the RREP towards the originator using the data in that route table. If a node forwards a RREP over a link that is likely to have errors or be unidirectional, the node must set the A flag to require that the recipient of the RREP acknowledge receipt of the RREP by sending a RREP-ACK message back. When any node transmits a RREP, the precursor list for the corresponding destination node is changed by adding to it the next hop node to which the RREP is forwarded. Also, at each node the reverse route used to forward a RREP has its life time changed to be the maximum of existing life time and current time plus active route timeout. Finally, the precursor list for the next hop to the destination is changed to involve the next hop to the source[15]. A RREP transmission fails if the RREQ transmission triggering the RREP occurs over a unidirectional link. If no other RREP generated from the same route discovery reaches the node that originated the RREQ message, the originator tries again route discovery after a timeout. If valid action is not taken, it can happen even when bidirectional routes between originator and destination do exist. Link layers cannot detect the presence of such unidirectional links in case of using broadcast transmissions for the RREQ. In AODV, any node acts on only the first RREQ with the same RREQ ID and ignores any subsequent RREQs. Suppose that the first RREQ arrives along a path that has one or more unidirectional links. A subsequent RREQ arrives via a bidirectional path, but it is ignored. To prevent it, when a node detects that its transmission of a RREP message has failed, it remembers the next hop of the failed RREP in an invalid list. Such failures can be detected via the absence of a link layer or network layer acknowledgment. A node ignores all RREQs received from any node in its invalid list. Nodes are removed from the invalid list after an invalid list timeout period. This period should be set to the upper bound of the time it takes to perform the allowed number of route request retry attempts. A node offers connectivity information by broadcasting local messages. A node should only use the messages if it is part of an active route. Every message interval times, the node checks whether it has sent a broadcast within the last message interval. If it has not, it may broadcast a RREP with TTL one. The RREP message fields set as follows: Destination IP address, Destination sequence number, Hop count zero and Life time. A node determines connectivity by learning for packets from its neighbor set. If it has received a message from a neighbor within the past delete period, and does not receive any packets for more than allowed message loss multiplied the message interval time for that neighbor, the link to this

neighbor is currently lost. Whenever a node receives a certain message from a neighbor, the node has an active route to the neighbor, and create one if necessary. If a route already exists, then the lifetime for the route increases to be at least allowed message loss multiplied by the message interval. The current node begins using this route to forward data packets. Routes created by the messages and not used by any other active routes have empty precursor lists and do not trigger a RERR message, if the neighbor moves away and a neighbor timeout occur.

3. Connectivity and Maintenance

Each forwarding node keeps track of its continued connectivity to its active next hops as well as neighbors that have transmitted the messages. A node maintains accurate data about its continued connectivity to these active next hops, using one or more of the available link or network layer mechanisms. Any suitable link layer notification, such as those provided by IEEE 802.11, can be used to determine connectivity, each time a packet is transmitted to an active next hop. If layer-2 notification is not available, passive acknowledgment is used when the next hop is expected to forward the packet, by listening to the channel for a transmission attempt made by the next hop. If transmission is not detected within next hop wait time or the next hop is the destination, one of the following methods are used to determine connectivity: Receiving any packet from the next hop, or a RREQ unicast to the next hop, asking for a route to the next hop, or an ICMP Echo Request message unicast to the next hop. If a link to the next hop cannot be detected by any of these methods, the forwarding node must assume that the link is lost, and take corrective action. Generally, route error and link breakage processing requires the following steps:

Invalidate existing routes

List affected destinations

Determine that neighbors may be affected

Deliver an appropriate RERR to such neighbors

A RERR message can be broadcast, unicast, or iteratively unicast to all precursors. Even when the RERR message is iteratively unicast to several precursors, it is considered to be a single control message for the purposes of the description in the text that follows. A node should not generate more than RERR rate limit messages per second. A node initiates processing for a RERR message in three cases. The first case is to detect a link break for the next hop of an active route in its routing table while transmitting data. The node first makes a list of unreachable destinations consisting of the unreachable neighbor and any additional destinations in the local routing table that use the unreachable neighbor as the next hop. The second is to get a data packet destined to a node for which it does not have an active route and is not repairing. There is only one unreachable destination, which is the destination of the data packet that cannot be delivered. The third is to receive a RERR from a neighbor for one or more active routes. The list is consisted of those destinations in the RERR for which there exists a corresponding entry in the local routing table that has the transmitter of the received RERR as the next hop. Some of the unreachable destinations in the list could be used by neighboring nodes, and it may therefore be necessary to send a new RERR. The RERR should contain those destinations that are part of the created list of unreachable destinations and have a non-empty precursor list. The neighboring nodes received the RERR messages are all those that belong to a precursor list of at least one of the unreachable destination in the newly created RERR. In case there is only one unique neighbor received the RERR, the RERR is unicast toward that neighbor. Otherwise the RERR is typically sent to the local broadcast address with unreachable destinations, and their corresponding destination sequence numbers, included in the packet. The destination count field of RERR packet indicates the number of unreachable destinations included in the packet. Just before transmitting the RERR, certain updates are made on the routing table that may affect

the destination sequence numbers for the unreachable destinations. For each one of these destinations, the corresponding routing table input is updated as follows. (1) The destination sequence number of this routing entry, if it exists and is valid, is incremented for the cases one and two above, and copied from the incoming RERR in case three above. (2) The entry is invalidated by marking the route entry as invalid. (3) The Lifetime field is updated to current time plus delete period. Before this time, the entry should not be deleted. Note that the Lifetime field in the routing table plays dual role – for an active route it is the expiry time, and for an invalid route it is the deletion time. If a data packet is received for an invalid route, the lifetime field is updated to current time plus delete period. When a link break in an active route occurs, the node upstream of that break chooses to repair the link locally if the destination was no farther than maximum of repair TTL hops away. To repair the link break, the node increments the sequence number for the destination and then broadcasts a RREQ for that destination. The TTL of RREQ is initially set to the following value: max (min repair TTL, 0.5 * number of hops to the sender of currently undeliverable packet) + Local TTL. Thus, local repair attempts are often invisible to the originating node, and always have TTL (^{3} minimum repair TTL + Local TTL). The node initiating the repair waits the discovery period to receive RREPs in response to the RREQ. During local repair data packets must be buffered. If the repairing node has not received a RREP at the end of the discovery period (or other control message creating or updating the route) for that destination, it proceeds by transmitting a RERR message for that destination. On the other hand, if the node receives one or more RREPs (or other control message creating or updating the route to the desired destination) during the discovery period, it first compares the hop count of the new route with the value in the hop count field of the invalid route table entry for that destination. If the hop count that recently determined the route to the destination is greater than the hop count of the past known route, the node considers a RERR message for the destination with the N bit set. Then it proceeds by updating its route table entry for that destination.

A node that receives a RERR message with the N flag set must not delete the route to that destination. The only action is the retransmission of the message, if the RERR arrived from the next hop along that route, and if there are one or more precursor nodes for that route to the destination. When the originating node receives a RERR message with the N flag set, if this message came from its next hop along its route to the destination then the originating node may choose to reinitiate route discovery. Local repair of link breaks in routes sometimes results in increased path lengths to those destinations. Repairing the link locally is likely to increase the number of data packets that are able to be delivered to the destinations, since data packets will not be dropped as the RERR travels to the originating node. Sending a RERR to the originating node after locally repairing the link break may allow the originator to find a fresh route to the destination that is better, based on current node positions. However, it does not require the originating node to rebuild the route, as the originator may be done, or nearly done, with the data session. When a link breaks along an active route, there are often some unreachable destinations. The node that is upstream of the lost link begins an immediate local repair for only one destination towards which the data packet was traveling. Other routes with the same link are marked as invalid, but the node handling the local repair may flag each such newly lost route as locally repairable. This local repair flag is reset in the route table when the route times out. Before the timeout occurs, these other routes are repaired as needed when packets arrive for the other destinations. Hence, they are repaired as needed. If a data packet does not arrive for the route, such route is not repaired. Alternatively, the node that is dependent on local congestion begins the local repair process for the other routes without waiting for new packets to arrive. By proactively

repairing the routes that have broken due to the loss of the link, incoming data packets for those routes are not subject to the delay of repairing the route and can be immediately forwarded. However, repairing the route before a data packet is received for it runs the risk of repairing routes that are no longer in use. Therefore, depending upon the local traffic in the network and whether congestion is being experienced, the node elects to proactively repair the routes before a data packet is received. Otherwise, it can wait until a data is received, and then commence the repair of the route. A node involved in the ad hoc network must do some actions after reboot as it might lose all sequence number records for all destinations. But, the neighboring nodes using it as an active next hop may be existed. It can generate the routing loops. To prevent it, each node must wait for the delete period before transmitting any route discovery messages. If it receives a RREQ, RREP, or RERR control packet, it should generate some route entries given the sequence number in the control packets, but must not forward any control packets. If the node receives a data packet for some other destination, it broadcasts a RERR and sets the waiting timer again to finish after current time plus the delete period. It can be shown that its neighbors are not used as an active next hop any more until the rebooting node gets out the waiting step and becomes an active router again[16]. Its own sequence number gets updated once it receives a RREQ from any other node, as the RREQ always carries the maximum destination sequence number seen en route. If no such RREQ arrives, the node must initialize its own sequence number to zero. Because AODV should work smoothly over wired, as well as wireless, networks, and because it is likely that AODV uses some wireless devices, the particular interface with the arrived packets must be known to AODV whenever a packet is received. This includes the reception of RREQ, RREP, and RERR messages. Whenever a packet is received from a new neighbor, the interface on which that packet was received is recorded into the route table entry for that neighbor, along with all the other appropriate routing information. Similarly, whenever a route to a new destination is known, the interface through which the destination can be reached is also recorded into the route table of the destinations. When some interfaces are available, a node retransmitting a RREQ message broadcasts again the message on all interfaces configured for operation in ad-hoc network, except those on which it is known that all of the nodes neighbors have already received the RREQ. For some broadcast media, it is presumed that all nodes on the same link receive a broadcast message at the same time. When a node needs to transmit a RERR, it transmits only it on those interfaces that have neighboring precursor nodes for that route.

4. Route Discovery and Extension

The reverse path uses for sending a RREP message when a broadcast RREQ packet arrives at a node with a route to the destination. The forward path is setting up during transmitting this RREP. Data packets waiting to be transmitted are buffered locally and transmitted in a queue when a route is set up. After a RREP was forwarded by a node, it can receive another RREP. This new RREP is either discarded or forwarded, depending on its destination sequence number. If new RREP message has a greater destination sequence number, then the route must be restructured, and RREP is forwarded. If the past destination sequence numbers and new RREP messages are the same, but the new RREP has a lesser hop count, this new RREP should be chosen and forwarded. Otherwise all later incoming RREP messages are discarded. The path for route request and route reply is from source to destination. In order for a subnet router to operate the AODV protocol for the whole subnet, it has to maintain a destination sequence number for the entire subnet. In any such RREP message sent by the subnet router, the prefix size field of the RREP message must be set to the length of the subnet prefix. Other nodes sharing the subnet prefix

must not issue RREP messages, and must forward RREQ messages to the subnet router. The RREP process that gives routes to subnets is the same as processing for host-specific RREP messages. Every node that receives the RREP with prefix size creates or updates the route table input for the subnet, including the sequence number by the subnet router. Then, in the later the node can use the information to avoid sending future RREQs for other nodes on the same subnet. When a node uses a subnet route, a packet is routed to an IP address on the subnet that is not assigned to any existing node in the ad hoc network. In this case, the subnet router returns ICMP host unreachable message to the sending node. Upstream nodes receiving such an ICMP message records the data that the particular IP address is unreachable, but do not invalidate the route entry for any matching subnet prefix. If several nodes are defined by the subnet prefix in the subnet advertise reachability to the subnet, the node with the lowest IP address is elected to be the subnet router, and all other nodes stop advertising reachability. The behavior of default routes is not defined in this specification. The routes selection sharing prefix bits is according to longest match first. If the contact points to the other networks can act as subnet routers for any relevant networks within the external routing domains, then the ad hoc network can maintain connectivity to the external routing ranges. The external routing networks use the ad hoc network defined by AODV. Thus, a contact point to an external network must act as the subnet router for every subnet within the external network for which the infrastructure router can provide reachability. It includes the need for maintaining a destination sequence number for that external subnet. If multiple infrastructure routers offer reachability to the same external subnet, they have to cooperate to provide consistent AODV semantics for ad hoc access to those subnets.

All extensions to the RREQ and RREP messages appear after the message data, and have the format as in table 5. Length denotes the length of the specific data, not including the type and length fields of the extension in bytes. The rules for extensions are spelled out more fully, and conform to the rules for handling IPv6 options. The message interval extension may be appended to a RREP message with time-to-live (= 1), to be used by a neighboring receiver in determine how long to wait for subsequent such RREP messages. The default values for some important factors are associated with AODV protocol works. A particular mobile node may change the certain factors such as the network diameter, route timeout, allowed message loss, RREQ retries, and possibly the message interval. These factors affect the performance of the protocol. Changing the node traversal time changes also the node estimate of the network traversal time, and so can only be done with suitable knowledge about the behavior of other nodes in the ad hoc network. The configured value for the route timeout must be at least the path discovery time twice.

표 5. 메시지 확장 형식
Table 5. Message Extension Format

0				1				2			3	
0	..	8	9	0	..	5	...	0	1	...	0	1
2		Length						Specific data				

The minimum repair time-to-live is the last known hop count to the destination. If special messages are used, then the active route timeout factor value is more than the allowed message loss multiplied by the message interval. For a given active route timeout value, it requires some adjustment to the value of the message interval, and consequently uses of the message interval extension in the messages. The time-to-live value is the field value in the IP header while the expanding ring search is doing. The purpose of timeout buffer is to provide a buffer for the timeout so that if the RREP is delayed due to congestion, a timeout is less possible to occur while the RREP is route back to the source. To delete this buffer, set the timeout buffer zero. The delete period factor is intended to provide an upper bound on the time for which an

upstream node A can have a neighbor B as an active next hop for destination D, while B has invalidated the route to D. Beyond this time B can delete the route to D. The determination of the upper bound depends somewhat on the characteristics of the underlying link layer. If the messages are used to determine the continued availability of links to next hop nodes, the delete period factor must be at least the allowed message loss multiplied by the message interval. If the link layer feedback is used to detect loss of link, the delete period factor must be at least the active route timeout. If the messages are received from a neighbor but data packets to that neighbor are lost, there needs more concrete assumptions about the underlying link layer. It is supposed that such asymmetry cannot persist beyond a certain time, say, a multiple K of the message interval. In other words, a node will invariably receive at least one out of K subsequent messages from a neighbor if the link is working and the neighbor is sending no other traffic. Covering all possibilities, the delete period is set as the following: 5×max (Active route timeout, Message interval). The network diameter measures the maximum possible number of hops between two nodes in the network. The node traversal time is a conservative estimate of the average one hop traversal time for packets and should include queuing delays, interrupt processing times and transfer times. The active route timeout is set to a longer value if link-layer indications are used to detect link breakages such as in IEEE 802.11 standard. The time-to-live start is set to at least 2 if the messages are used for local connectivity information. The AODV performance is sensitive to the chosen values of these constants, which depend on the characteristics of the underlying link layer protocol, radio technologies etc. The invalid list timeout is suitably increased if an expanding ring search is used. In such cases, it has the following value, and it is to account for possible additional route discovery attempts: ((time-to-live threshold – time-to-live start) / (time-to-live increment) + (RREQ retries) + 1)×(Network traversal time)

Ⅳ. Simulation

The simulation results were carried out using Simulink toolbox of MATLAB. Wireless Ad hoc network with random topology and seven nodes was simulated. The model structure of AODV is as shown in Fig. 3. The Fig. 4 shows the random topology for wireless ad hoc network used for simulation. Two tasks were created in each node to handle AODV send and receive actions, respectively. The AODV send task is activated from the application code as a data message should be sent to another node in the network. The AODV receive task handles incoming AODV control messages and forwarding of data messages. Communication between the application layer and the AODV layer is handled using Simulink mailboxes. The AODV send task operates according to the following Procedure Send:

```
Procedure Send {//Check the routing table for a route to the
destination.
   If (a valid route exists) {
      Forward data message to next hop on route.
      Update expiry time of route entry.}
   Else {
      Initiate route discovery by broadcasting RREQ message.
      Buffer data message until route has been established.}
   Else if (notified of established new route) {
      Send all buffered data messages to destination.}
}
```

The AODV receive task performs the following Procedure Receive:

```
Procedure Receive {
   If (receiving data message) {
      Update expiry timer for reverse route entry to source.
      If (this node is the destination) {
         Pass data message to application.}
      Else {
         Forward data message to next hop on route.
         Update expiry timer of route entry.}
   }
    Else {
       Switch (message type) {
           Case RREQ {
                If (first time this RREQ is received) {
                   Enter RREQ in cache.
                   Create or update route entry to source.
                   Check the routing table for a route to the
destination.}
                   If (a route exists) {
                          Send RREP message back towards
source.}
                    Else {
                          Update and rebroadcast the RREQ.}
                }
           }
           Case RREP {// Check the routing table for a route to
the destination.
```

```
                If (no route exists) {
                    Create route entry to destination.}
                Else if (route entry exists but should be updated)
{
                    Update route entry to destination.}
                If (this node is the original source) {
                    Notify the AODV send task about the new
route.}
                Else if (route to destination was created or
updated) {
                    Update reverse route entry towards source.
                    Propagate RREP to next hop towards
source.}
            }
            Case RERR {
                Find and invalidate all affected route entries.
                Propagate the RERR to all previous hops on the
routes.
            }
    }
}
```

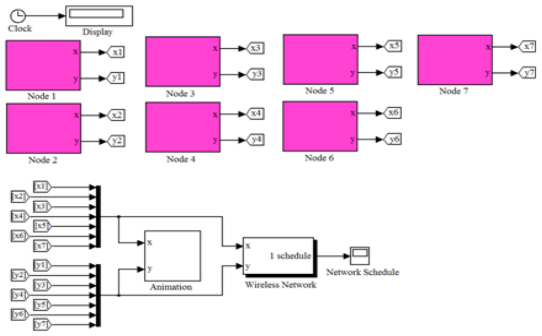

그림 3. Simulink 모델 구조
Fig. 3. Simulink Model Structure

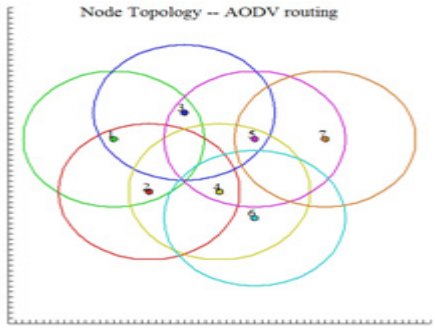

그림 4. 노드 무작위 토폴로지
Fig. 4. Node Random Topology

Each node also contains a periodic task, responsible for broadcasting hello messages and determines local connectivity based on hello messages received from neighboring nodes. Finally, each node has a task to handle timer expiry of route entries. In the simulation scenario, node 1 sends data periodically to node 7 with period 0.5. The initial route that is established is 1 ® 3 ® 5 ® 7. At time t = 3, node 5 starts to move which eventually leads to the route breaking. At time t = 10, node 6 repairs the route by moving in between node 4 and 7. The results in table 6 show the path of communication, time at which the various message are sent like data packets, hello messages, node expiry time and status of messages (see Fig 5.).

표 6. 실험 결과
Table 6. Simulation Results

Wireless network data: Transmit power is: -8.00 dbm <==> 0.16 mW Receiver threshold is: -48.00 dbm <==> 1.58e-005 mW Maximum signal reach is calculated to: 12.89 m
Time: 0.0002; Application in Node#1 wants to send to Node#7 Data: 0.018504 Size: 4 No (valid) route exists Buffering message 1 Time: 0.000894; A new route has been established between Node#1 and Node#7 --- 1 3 5 7 1 data messages in buffer Sending buffered message 1 to Node#7 Application in Node#7 receiving data: 0.018504 Buffer emptied Time: 0.5002 Application in Node#1 wants to send to Node#7 Data: 0.82141 Size: 4 Route exists in table --- 1 3 5 7 ... Time: 19.0002 Application in Node#1 wants to send to Node#7 Data: 0.50281 Size: 4 Route exists in table --- 1 2 4 6 7 Application in Node#7 receiving data: 0.50281 Node#3 lost connection to Node#5 Time: 19.5002 Application in Node#1 wants to send to Node#7 Data: 0.70947 Size: 4 Route exists in table --- 1 2 4 6 7 Application in Node#7 receiving data: 0.70947

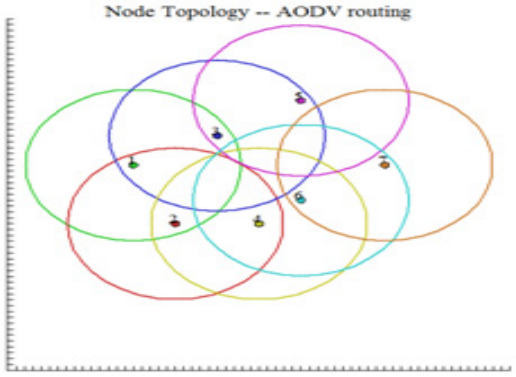

그림 5. 노드 토폴로지
Fig. 5. Node Topology

표 7. 라우팅 테이블
Table 7. Routing Table

Destination	7	1	1	1	1	1	1
Next hop	2	1	1	2	3	4	6
Hops	4	1	1	2	2	3	4
DestSeq	1	10	10	10	10	10	10
Expiry	22.5002	22.5003	16.5003	22.5004	16.5004	22.5005	22.5005
Neighbor	-	4	5	6	7	7	-
State	Valid	Valid	Invalid	Valid	Invalid	Valid	Valid

The performance of routing protocol is evaluated using several performance metrics to calculate best path for routing the packet to its destination. These metrics are a standard measurement that could be number of hops, which is used by the routing algorithm to determine the optimal path for the packet to its destination. Packet Delivery Ratio (PDR) is the ratio of the number of data packets received by the receivers verses the number of data packets supposed to be received. This number presents the effectiveness of a protocol. End-to-end delay denotes how long it took for a packet to travel from the source to the receiver (see Table 7). The lengths of vector sent and received can be used to determine how many messages that where lost due to the delay in detecting and propagating the information about the broken link back to the source node. The messages sent at times 8.0002, 8.5002, and 9.0002 are lost. The message interval determines who fast the network will respond to broken links (and also the bandwidth overhead). Changing the message interval factor to decrease the number of lost data messages, only two messages are lost.

V. Conclusions

Wireless ad hoc networks are formed by sensor nodes without the support of fixed infrastructure. Each node performs as a host as well as host to forward the data packets. The nodes are adjustable based on the local environments. The ad hoc networks are used in many situations where temporary network connection is needed. In this paper, we suggested an experimental AODV routing protocol to enhance the network lifetime. Here, the wireless ad hoc networks were used in temporary network connections. They were formed by sensor nodes without the support of fixed infrastructure. The proposed model has been simulated using Matlab Simulink. AODV routing protocol is a reactive routing protocol which enables multi-hop routing between participating mobile nodes in an ad-hoc network. AODV increases the packet delivery ratio with increase in power and decrease in number of hops. For the AODV security considerations, it does not specify any special security measures. Route protocols are main targets for impersonation attacks. In networks where the node membership is unknown, it is difficult to detect the impersonation attacks. AODV control messages must be protected by the authentication techniques when there is a danger of such attacks. While AODV does not place restrictions on the authentication mechanism used for this purpose, IPsec AH is an appropriate selection for cases that the nodes share an appropriate security association. In particular, RREP messages must be authenticated to avoid creation of spurious routes to a desired destination. RERR messages must be also authenticated in order to prevent malicious nodes from disrupting valid routes between nodes that are communication entities.

References

[1] N. Bhanushali, P. Thakkar and P. Shete, “Impact of hello interval on performance of AODV protocol,” International Journal of Computer Applications (0975 – 8887), 116(23):33-37, 2015.

[2] K. Chaturvedi and K. Shrivastava, “Mobile Ad-Hoc Network: A Review,” International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences, Vol.1, Issue 1, 29-33, 2013.

[3] O. Ertan, E. Cem and D. Hakan, “Quality of deployment in surveillance wireless sensor networks,” International Journal of Wireless

Information Networks, 12(1):61-67, 2005.

[4] J. A. Gay-Fernandez, M. Garcia Sánchez, I. Cuinas, A. V. Alejos, J. G. Sanchez, and J. L. Miranda-Sierra, "Propagation analysis and deployment of a wireless sensor network in a forest," Progress In Electromagnetics Research, Vol. 106, 121-145, 2010.

[5] S. K. Indumathi and G. M. K. Nawaz, "Susceptibility of ad hoc protocols to covert channels (an analysis and countermeasures)," 3rd International Conference on Information and Financial Engineering, vol.12, 61-65, 2011.

[6] Y. Jahir, M. Atiquzzaman, H. Refai, and P. G. Lopresti, "AODVH: ad hoc on-demand distance vector routing for hybrid nodes," Computer and Information Technology (ICCIT) 2010 13th International Conference on, pp. 165-169, 2010.

[7] B. Karthikeyan, C. A. Gunter, and D. Obradovic, "Fault origin adjudication," In Proceedings of the Workshop on Formal Methods in Software Practice, Portland, OR, August 2000.

[8] P. Kumar, M. P. Singh, U. S. Triar, and S. Kumar, "Energy band based clustering protocol for wireless sensor networks," International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3, ISSN (Online): 1694-0814, 2012.

[9] S. J. Lee, Elizabeth M. Belding-Royer, and Charles E. Perkins, "Scalability study of the ad hoc on-demand distance vector routing protocol," International Journal of Network Management, 13(2):97-114, 2003.

[10] H. Liu and Z. Shang, "Comparing the performance of the ad hoc network under attacks on different Routing Protocol" International Journal of Security and Its Applications. 9(6): 195-208, 2015.

[11] G. Lucas, J. G. V. Luis, F. Buiati, J. B. M. Sobral, and E. Camponogara, "Self-configuration and self-optimization process in heterogeneous wireless networks," Sensors (Basel); 11(1): 425-454, 2011.

[12] K. M. Mahesh and R. D. Samir, "Ad hoc on-demand multipath distance vector routing," Wireless Communications and Mobile Computing, 6: 969-988, 2006.

[13] P. P. Naidu and M. Chawla, "Extended ad hoc on demand distance vector local repair trial for MANET," International Journal of Wireless & Mobile Networks, 4(2):235-250, 2012.

[14] A. A. Radwan, T. M. Mahmoud and E. H. Houssein, "Performance measurement of some mobile ad hoc network routing protocols", International Journal of Computer Science Issues, 8(1):107-112, 2011.

[15] D. Rajitha and J. Ravichander, "Lifetime enhancement routing algorithm for mobile ad hoc networks," International Conference on Recent Advances in Communication, VLSI & Embedded Systems. ISSN (Online): 2349-0020, ISSN (Print): 2394-4544, 2014.

[16] Y.G. Kim and K.I. Moon, " Adaptive Time Delay Compensation Process in Networked Control System," International Journal of Advanced Smart Convergence, Vol. 5, No 3:34-46, 2015.

저자 소개

김 용 길(정회원)

- 1990년 : 호남대학교 전산통계학과 졸업(이학사)
- 1992년 : 광주대학교 대학원 컴퓨터학과 졸업(공학석사)
- 2014년~현재 : 조선이공대학교 컴퓨터보안과 조교수

<주관심분야 : 네트워크보안, 통신시스템, 정보보호>

문 경 일(정회원)

- 1982년 : 서울대학교 계산통계학과 졸업(이학사)
- 19880년 : 서울대학교 대학원 계산통계학과 졸업(이학석사)
- 1991년 : 서울대학교 대학원 계산통계학과 졸업(이학박사)
- 1987년~현재 : 호남대학교 컴퓨터공학과 교수

<주관심분야 : 지능 시스템, 복잡성 과학>