단일 노드 결합시 MANET 자동 네트워킹 프로토콜의 메시지 복잡도 분석

정회원 김 상 철*

Message Complexity Analysis of MANET Address Autoconfiguration-Single Node Joining Case

Sang-Chul Kim* ***Regular Member***

요 약

네트워크 계층의 MANET 라우팅 프로토콜이 개발될 때 가장 주요한 관심사는 라우팅 오버헤드 (Overhead)의 축소이다. 본 논문에서는 MANET 에서 단일 노드가 자동 네크워킹 프로토콜에 의해 IP 주소를 획득할 때의 라우팅 오버헤드 (메시지 복잡도)를 정량적으로 분석하는 수학적 방법을 제안한다. MANET에서 자동 네크워킹 프로토콜은 IP 주소 할당시 IP 주소 충돌회피에 사용된다. 메시지 복잡도는 Upper Bound에 의해 수학적으로 정량화되고, 실제 주소 할당을 컴퓨터 시뮬레이터로 구현하여, 제안된 Upper Bound와 시뮬레이션 결과를 비교, 분석한다. 메시지 복잡도의 Upper Bound는 Worst Case 시나리오 분석에 의해 유도된다.

Key Words : Ad hoc Network, Address Autoconfiguration Protocols, Complexity, Duplicate Address Detection

ABSTRACT

This paper proposes a novel method to perform a quantitative analysis of message complexity and applies this method in comparing the message complexity among the mobile ad hoc network (MANET) address autoconfiguration protocols (AAPs). To obtain the upper bound of the message complexity of the protocols, the *O*-notation of a MANET group of N nodes has been applied. The message complexity of the single node joining case in Strong DAD, Weak DAD with proactive routing protocols (WDP), Weak DAD with on-demand routing protocols (WDO), and MANETconf has been derived as $n(mO(N)+O(t))$, $n(O(N)+O(t))$, $n(O(N)+2O(t))$, and $nO((t+1)N)+O(N)+O(2)$ respectively. In order to verify the bounds, analytical simulations that quantify the message complexity of the address autoconfiguration process based on the different conflict probabilities are conducted.

Ⅰ. Introduction

MANETs are self-organizing wireless networks where mobile nodes have routing capabilities to be able to forward packets to communicate with one another over multi-hop wireless links without any fixed communication infrastructure, such as a base station or an access point. Therefore, it is essential for all nodes to be able to perform the operations required for configuration of unique addresses to

※ This research is supported by Seoul R&DB program of Korea.
* 국민대학교 컴퓨터공학과 (sckim7@kookmin.ac.kr)
논문번호 : KICS2006-10-443, 접수일자 : 2006년 10월 19일, 최종논문접수일자 : 2007년 5월 18일

execute proper routing of data packets in a *MANET*. Address autoconfiguration is an important issue in *MANET*s since address pre-configuration is not always possible. *MANET*s currently depend on the mechanism of checking *IP* addresses of nodes to decide whether the connection and identification of nodes participating in a *MANET* are established or not.[1]

In conventional networks, address autoconfiguration can be classified as either a stateless or stateful protocol.[2] The stateless approach is used when a network is not especially required to control the exact *IP* address assignments provided that the addresses are unique and routable. The stateful approach is used when a network demands exact *IP* address assignments.[3] Dynamic Host Configuration Protocol (*DHCP*) is an example of a stateful protocol where a *DHCP* server assigns unique addresses to unconfigured nodes and keeps state address information in an address allocation table. However, in stateless protocols, a node can select an address by itself and verify its uniqueness in a distributed manner using duplicate address detection (*DAD*) algorithms.[4] By using *DAD* algorithms, a node in a *MANET*, which lacks an *IP* address in the *MANET*, can determine whether a candidate address selected by itself is available or not.

A node already equipped with an *IP* address also depends on *DAD* in order to protect its *IP* address from being accidentally used by another node in the *MANET*.[5] Based on the conventional method stated in [6], *DAD* can be classified as Strong *DAD* and Weak *DAD*. Strong *DAD* uses an address discovery mechanism where a node randomly selects an address and requests the address within a *MANET* by checking if the address is being used in the *MANET*. Based on a reply for the claimed request, which needs to arrive at the node within a finite bounded time interval, the node can detect an address duplication in the *MANET*.[1] Weak *DAD* is proposed by [6], where ad hoc routing protocols are used to detect address duplication by modification of the routing protocol packet format.

MANET routing protocols can be classified into proactive and on-demand. Proactive routing protocols using periodic neighbor discovery messages and topology update messages give route information to each node before a node sends data packets to a destination. The Fisheye Scope Routing (*FSR*)[7], Topology Broadcast Based on Reverse Path Forwarding (*TBRPF*)[8], Fuzzy Sighted Link State Routing (*FSLS*)[9], Optimized Link State Routing Protocol (*OLSR*)[10], and Landmark Ad Hoc Routing (*LANMAR*)[11] are currently being developed as *MANET*proactive routing protocols.

On-demand routing protocols such as Dynamic Source Routing (*DSR*)[12] and the Ad hoc On Demand Distance Vector (*AODV*)[13] issue route discovery mechanism messages only when a node needs to send data to a destination node and it does not have an active source route to a destination. Because these protocols do not use any periodical message exchange, such as the neighbor discovery message used in proactive routing protocols, they do not hold any route information at each node before a node sends data towards a destination node. Therefore, they need Route Request and Route Reply messages to find and maintain a route when it is needed.

As a stateful protocol, *MANET*conf [14] uses a mutual exclusion algorithm for a node to acquire a new *IP* address. Therefore, if a requester wants to acquire an *IP* address, the *IP* address should be approved by all nodes in a *MANET*.

In other related research, Weniger and Zitterbart summarized the current approach and future directions of address autoconfiguration in *MANET*s [4]. Jeong *et al*. studied hybrid ad hoc *IP* address autoconfigurations in [5]. The authors of [15] proposed an *IP* address configuration for Zeroconf. In [16] Mohsin and Prakah introduce an *IP* address assignment method for *MANET*s. In [17] Zhou and Mutka investigated prophet address allocation for large scale *MANET*s. Additionally, Weniger proposed a passive autoconfiguration for *MANET*s in [2]. Since the autoconfiguration protocols in the above section have been composed of key steps introduced in Strong *DAD*, Weak *DAD* and *MANET*conf based on authors' knowledge, this paper focuses on formalizing how to analyze and compare the message

complexity of the key steps used in Strong *DAD*, Weak *DAD* and *MANET*conf. Therefore, other protocols related with these formulations can adopt the formulations proposed in this paper in order to get quantitative analysis and comparison among the protocols.

Based on the many considering factors of a *MANET*, the reduction of routing overhead is a main concern when a *MANET* routing protocol is developed. Therefore, one essential measure of the quality of a *MANET* routing protocol is the scalability in regards to an increase of the *MANET* nodes. Message complexity is defined as a performance measure where the overhead of an algorithm is measured in terms of the number of messages needed to satisfy the algorithm's request. The authors of [18] use the message complexity and synchronization delay to measure the performance of a mutual exclusion algorithm which is used to effectively share resources in distributed systems. In [19], Shen uses the message complexity to statistically measure the performance of the Cluster-based Topology Control (*CLTC*) protocol. The authors in [20] calculate the storage complexity and communication complexity to analyze the scalability of various *MANET* routing protocols and introduce the routing overhead of periodically updated *LS* messages, which follow the order of $O(N^2)$, where N indicates the number of nodes in a *MANET*. However, the detailed investigation to derive the upper bound of *LS* messages has not been justified by a mathematical form and currently the message complexity analysis and comparison among the *IP AAP*s for *MANET*s has not been conducted yet. Therefore, in this paper, the upper bounds of the message complexity of the *IP AAP*s for *MANET*s are derived.

The analytical framework used in deriving the upper bound of the message complexity, which is represented in this paper, can be widely adapted to a wide variety of protocols. The general methodology of analysis is based on [21], which uses a flowchart to analyze the time complexity of an image segmentation algorithm based on the recursive shortest spanning tree (*RSST*). The authors of [22] point out that time complexity is one of the most important factors to measure or compare the performance of different algorithms, and therefore, should be considered when an algorithm is being developed. Based on the complexity analysis method of [21], the message complexity of *MANET* address autoconfiguration algorithms is investigated. Each node strictly follows a *procedure*, which is a sequence of *steps* in the algorithm, where each step guides a node to make a general decision such as whether to generate a message or not, whether to take a same procedure or not (which is called a *recursive procedure*), whether to branch to a different procedure or not, and whether to stop a step or not. The method of adding the upper bounds of the time complexity measured at each step can be adapted in the proposed algorithm since *MANET* address autoconfiguration algorithms are composed of a sequence of discrete distinctive procedures where each step has its own message complexity.

Therefore, by adding the message complexity measured at each step, the message complexity of a procedure can be calculated. Correspondingly, the method of adding the time complexity measured at each node to get the time complexity of n nodes can be adapted in the proposed algorithm since *MANET* address autoconfiguration algorithms are composed of recursive procedures. Therefore, by adding the message complexity measured at each procedure for each node, the message complexity of a *MANET* operation can be calculated.

This paper is organized as follows. Section II presents a system model that is used in the derivations and analysis of the following sub-sections and introduces the approach method used in ana-

Table 1. Acronym table [*: variable]

Acronym	Message	Acronym	Message
AB	Abort	*IQ*	Initiator Request
AC	Address Cleanup	*LS*	Link State
AD	Advertised	*NR*	Neighbor Reply
AE	Address Error	*NQ*	Neighbor Query
AL	Allocated	*RR*	Route Reply
AO	Allocation	*RQ*	Route Request
AP	Address Reply	*RT*	Requester Request
AQ	Address Request	*M*	*DAD* retry count limit*
IR	Initiator Reply	*n*	retry count limit*

lyzing the message complexity in this paper. Subsections present several *Lemmas* and their proofs used in deriving the message complexity of Strong *DAD*, Weak *DAD*, and *MANET*conf, respectively. Section III contains numerical results and performance analysis. Section IV states the conclusion. The acronyms of messages and nomenclatures of the retry count variables used in this paper are summarized in Table 1.

Ⅱ. Message Complexity Analysis

A *MANET* is represented as an undirected graph $G(\boldsymbol{V}, \boldsymbol{E})$ where $\boldsymbol{V}$ is a finite nonempty set of nodes, which can be represented as $\boldsymbol{V}= \{ V_1^G, V_2^G, \cdots, V_W^G \}$ where $|\boldsymbol{V}|=W$ and $\boldsymbol{E}$ is a collection of pairs of distinct nodes from $\boldsymbol{V}$ that form a link, which can be represented as $\boldsymbol{E}= \{ E_1^G, E_2^G, \cdots, E_W^G \}$ [23].

***Definition* 1** In a free tree $P(V,E)$, broadcasting an *Address Query* (e.g., *AQ* message in Strong *DAD*, *LS* and *RQ* messages in Weak *DAD*, or *IQ* message in *MANET*conf) message by a node is defined as a *trial*.

Ⅰ. A *success trial* is defined as an event in which after a node broadcasts an *Address Query* message, it does not receive any *Address Reply* message (e.g., *AP* message in Strong *DAD*, *AE* in Weak *DAD*, or negative *IR* message in *MANET*conf) within a specific time period.

Ⅱ. A *failure trial* is defined as an event in which after a node broadcasts an *Address Query* message, it receives at least one *Address Reply* message within a specific time period.

Ⅲ. A *successful IP verification procedure* is defined from m consecutive success trials.

A. Therefore, for a node to get a verified *IP* address, the node has to perform a sequence of m independent trials where each trial has to become a *success trial*.

B. In Strong *DAD*, m is defined as a positive number which is greater than one (m>1).

C. In Weak *DAD* and *MANET*conf, since m is set to one (m=1), the *successful IP verification procedure* is same as the *success trial*.

Ⅳ. An *IP verification procedure* including any *failure trial* results in a *failure IP verification procedure*.

A. In Strong *DAD*, a *failure IP verification procedure* is composed of consecutive x-1 times of *success trials* and a *failure trial* at the x^{th} *trial* where x = 1,2,, m.

B. n Weak *DAD* and *MANET*conf, since m is set to one (m=1), the *failure IP verification procedure* is same as the *failure trial*.

Ⅴ. A *session* is defined as a sequence of *successful* or *failure procedures*. The maximum number of procedures executed in the *session* is limited by n in Strong *DAD*, Weak *DAD*, and *MANET*conf.

A. When computingthe upper bound in Strong *DAD*, the worst case of a *successful session* composes of n-1 consecutive *failure IP verification procedures* and a *successful IPverification procedure* at the n^{th} *IP verification procedure*. A *failure session* is composed of *n failure IP verification procedures*.

When computing the upper bound in Weak *DAD* and *MANET*conf, the worst case of a *successful session* composes of n-1 consecutive *failure trials* and a *success trial* at the n^{th} *IP verification procedure*. A *failure session* composes of *n failure trials*. □

In this paper, the most common flooding method is used to broadcast an *Address Query* message where every node retransmits an *Address Request* message to its entire one-hop neighbors whenever it receives the first copy of the *Address Query* message [24]. Since each member node in a free tree will relay the *Address Query* message initiated at node V_i, assuming that the duplicated packet discard scheme is applied, the maximum number of nodes relaying an *Address Query* message is N-1, where the rule of discarding duplicated messages at a node is adopted. Therefore, the maximum number of *Address* Query messages

broadcasted or relayed in the free tree is N, which can be represented as $O(N)$. The above content can now be generalized into the following definition.

***Definition* 2** For a *MANET* with N nodes, $O(N)$ is the upper bound of the maximum number of broadcasted or relayed *Address Query*messages when a node broadcasts the *Address Query* message.

***Lemma* 1** For a *MANET* routing tree with t nodes in the maximum length path, $O(t)$ is the upper bound of the maximum number of unicasted or relayed *Address Reply*messages when a node unicasts an *Address Reply* message.

Proof. Since each member node in a path of $d(j,i)$ hops relays an *Address Reply* message initiated by an *Address Reply* source node, the maximum number of nodes relaying an *Address Reply* message is t-2 where the rule of discarding duplicated messages at a node is adapted and the node V_i does not relay an *Address Reply* message. Therefore, the maximum number of *Address Reply* messages unicasted or relayed in the free tree is t-1, where the message complexity bound can be represented as $O(t)$. □

2.1 Strong DAD

In order to derive the upper bound of the message complexity of the Strong *DAD* protocol, the flowchart of Strong *DAD*, as shown in Fig. 1, is used.

To compute the upper bound of the message complexity, a scenario where a node experiences a *failure IP verification procedure* is considered. Since the procedure is composed of a total of (m-1) number of *success trials* and a *failure trial* at the m^{th} *trial*, the message complexity of a *failure IP verification procedure* can be represented as $mO(N)+O(t)$. Based on the above result, the following *Lemma* is given.

***Lemma* 2** In an *IP* address verification procedure, $mO(N)+O(t)$ is the upper bound of the maximum number of broadcasted/relayed *AQ* messages and unicasted/relayed *AP* messages when a node

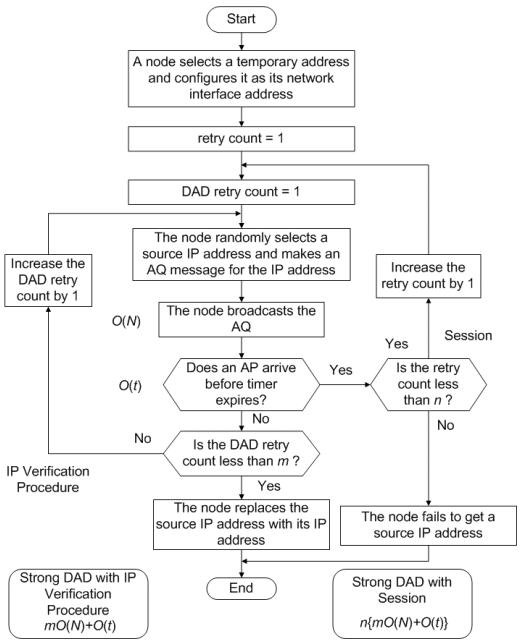


Fig. 1. The flowchart of Strong DAD operations

needs to verify its *IP* address in a *MANET* with the Strong *DAD* protocol.

Proof. The *IP verification procedure* including a *failure trial* at the m^{th} *trial* is composed of m-1 *success trials*, which gives (m-1)$O(N)$ number of broadcasted or relayed *AQ* message based on *Definition* 2, and a *failure trial* at the m^{th} *trial*, which gives $O(N)$ number of broadcasted or relayed *AQ* message based on *Definition* 2, and $O(t)$ unicasted or relayed *AP* message based on *Lemma* 1. Therefore, the message complexity of the *failure IP verification procedure* can be represented as (m-1)$O(N)+O(N)+O(t)$, which sums the upper bound of the maximum number of broadcasted, unicasted, and relayed *AQ* and *AP* messages in m-1 *success trials* and a *failure trial* at the m^{th} *trial*. Rearranging (m-1)$O(N)+O(N)+O(t)$ yields $mO(N)+O(t)$. □

***Lemma* 3** In a session, $n(mO(N)+O(t))$ is the upper bound of the maximum number of broadcasted/relayed *AQ* messages and unicasted/relayed *AP* messages using the Strong *DAD* protocol.

Proof. Strong *DAD* has a *session* and the maximum number of retries of the *IP verification procedure* is limited by n in the *session*. Since the

session consists of *n* maximum number of *IP verification procedures* and the upper bound of the maximum number of *IP verification procedures* is $mO(N)+O(t)$, based on *Lemma* 2, the message complexity of the session can be represented as $n(mO(N)+O(t))$. □

2.2 Weak DAD

In order to derive the upper bound of the message complexity of the Weak *DAD* protocol, the flowchart of Weak *DAD*, as shown in Fig. 2, is used.

In *WDP*, nodes periodically broadcast *LS* messages to inform other nodes of the network topology. In *WDO*, only when a source node needs to send data to a destination node where the source node does not have a route to the destination, the source node broadcasts a *RQ* message to find a route to a destination node and a node which is the destination node or a node having a fresh enough route unicasts a *RR* messages in response to the *RQ* message.

When a node finds an *IP* address that is duplicated with an entry in its routing table after investigating an *IP* address in a *LS*, *RQ*, or *RR* message, the node takes additional steps to inform other nodes of the duplicated address[6]. In such a case, the node that was already using the *IP* address will unicast an *AE* message to the node that has the duplicated *IP* address[5]. If a node does not find any duplicated *IP* address after investigating an *IP* address in a *LS*, *RQ*, or *RR* message, the node normally relays the *LS*, *RQ*, or *RR* message. Based on the above specifications, the following *Lemmas* can be derived.

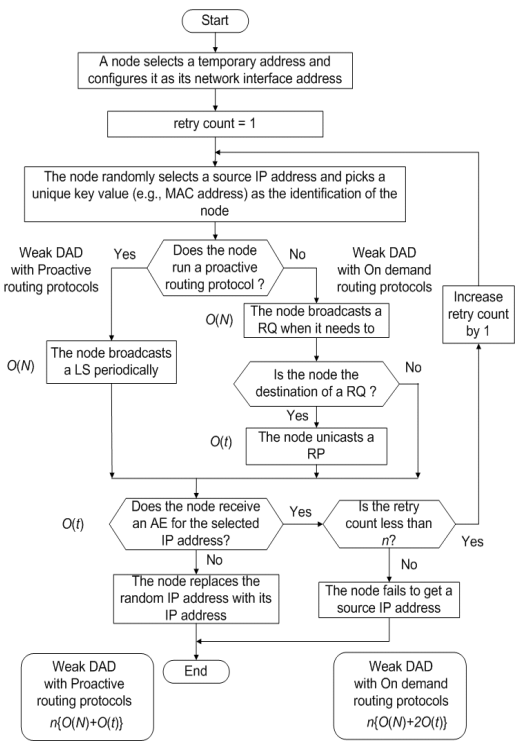


Fig. 2. The flowchart of Weak DAD operations

***Lemma* 4** In an *IP verification procedure*, $O(N)+O(t)$ is the upper bound of the maximum number of broadcasted/relayed *LS* messages and unicasted /relayed *AE* messages when a node needs to verify its *IP* address in a *MANET* using *WDP*.

Proof. The maximum number of messages occurs when the *IPverification procedure* results in a *failure trial*. Since, the *failure trial* gives $O(N)$ number of broadcasted or relayed LS messages based on *Definition* 2, and $O(t)$ unicasted or relayed *AP* message based on *Lemma* 1, the message complexity of the *failure trial* can be represented as $O(N)+O(t)$, which sums the upper bound of the maximum number of broadcasted and relayed *LS* messages and unicasted and relayed *AE* messages. □

***Lemma* 5** In a session, $n(O(N)+O(t))$ is the upper bound of the maximum number of broadcasted/ relayed *LS* messages and unicasted /relayed *AE* messages using *WDP*.

Proof. *WDP* has a *session* and the maximum number of retries of the *IP verification procedure* is limited by *n* in the *session*. Since the *session* consists of *n* maximum number of *IPverification procedures* and the upper bound of the maximum number of an *IP verification procedure* is $O(N)+O(t)$ based on *Lemma* 4, the message complexity of the *session* can be represented as $n(O(N)+O(t))$, where *n* is the number of retry count of the *IP verification procedures*. □

In *WDO*, a node broadcasts or relays a *RQ* message and it can unicast a *RP*message if it is a destination node based on the normal routing pro-

cedure. In addition, it unicasts an *AE* message when a node finds a duplicated *IP* address. Based on the above results, the following *Corollaries* that are similar to the *WDP* case are given.

***Corollary* 1** In an *IP verification procedure*, $O(N)+2O(t)$ is the upper bound of the maximum number of broadcasted/relayed *RQ* messages and unicasted/relayed *RP*messages and *AE* messages when a node wants to verify its *IP* address in a *MANET* using *WDO*.

Proof. The maximum number of messages occurs when the *IPverification procedure* results in a *failure trial*. Since, the *failure trial* gives $O(N)$ number of broadcasted or relayed *RQ* messages based on *Definition* 2, and $2O(t)$ unicasted or relayed *RP* messages and *AE* messages based on *Lemma* 1, the message complexity of the *failure trial* can be represented as $O(N)+2O(t)$, which sums the upper bound of the maximum number of broadcasted and relayed *RQ* and unicasted and relayed *RP* and *AE* messages.

***Corollary* 2** In a session, $n(O(N)+2O(t))$ is the upper bound of the maximum number of broadcasted/relayed *RQ* messages and unicasted/relayed *RP* messages and *AE* messages in *WDO*.

Proof. *WDO* has a *session* and the maximum number of retries of the *IP verification procedure* is limited by *n* in the *session*. Since the *session* consists of *n* maximum number of *IP verification procedures* and the upper bound of the maximum number of an *IP verification procedure* is $O(N)+2O(t)$ based on *Corollary* 1, the message complexity of the *session* can be represented as $n(O(N)+2O(t))$ where *n* is the number of retry count of the *IP verification procedure*. □

2.3 MANETconf

In order to derive the upper bound of the message complexity in *MANET*conf, the flowchart as shown in Fig. 3 is used. When a node (which is a *Requestor*) tries to join a *MANET* and to obtain a verified *IP* address, it broadcasts a *NQ* message to its neighbors. When the *Requestor* does not receive any *NR* messages before the neighbor reply timer expires, it repeats broadcasting the *NQ* message by a threshold number. After finishing the repetition, the *Requestor* decides that there is only one node and configures itself with an *IP* address. The *Initialization* procedure of *MANET*conf described above is not considered into the message complexity since the message complexity is focused on the procedures of a single node joining into a *MANET* group.

If the *Requestor* receives *NR* messages, the *Requestor* selects an *Initiator* and unicasts a *RR* message to the *Initiator*. The message complexity of unicasting the *RR* message can be represented as $O(1)$. After receiving a *RR*message, the *Initiator* broadcasts an *IQ* message to all nodes of the *MANET* group in order to verify the *IP* address of the *Requestor*. The message complexity of broadcasting the *IQ* message can be represented as $O(N)$ based on *Definition* 2. Recipient nodes will reply with an affirmative or a negative response through the *IR* message, to the *Initiator*. The message complexity of unicasting the *IR* message by all nodes in the *MANET* group can be represented as $O(tN)$, since all *N* nodes unicast *IR*messages and each *IR* message has the message complexity $O(t)$ based on *Lemma* 1. If the *Initiator* receives positive *IR*

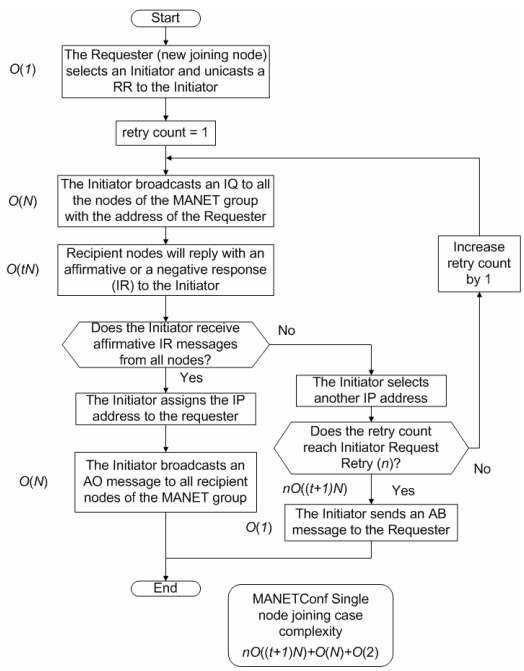


Fig. 3. The flowchart of MANETconf operations

messages from all the recipient nodes, it broadcasts an *AO* message to all the recipient nodes of the *MANET* group. The message complexity of broadcasting the *AO* message can be represented as $O(N)$ based on *Definition* 2. If the *Initiator* receives negative *IR* messages from the recipient nodes, it selects another *IP* address and repeats the step of broadcasting *IQ* and receiving *IR* messages until the retry count reaches the *Initiator Request Retry* which is set to n in this paper. Based on the above results, the following *Lemma* can be derived.

***Lemma* 6** In an *IP verification procedure* of a single node joining case, $O((t+1)N)$ is the upper bound of the maximum number of broadcasted/relayed *IQ* messages and unicasted/relayed *IR* messages when a node needs to verify its *IP* address in a *MANET* with *MANET*conf.

Proof. The maximum number of messages occurs when the *IP verification procedure* results in a *failure trial*. Since, the *failure trial* gives $O(N)$ number of broadcasted or relayed *IQ* messages based on *Definition* 2, and $O(tN)$ unicasted/relayed *IR* messages based on *Lemma* 1, the message complexity of the *failure trial* can be represented as $O(N)+O(tN)$, which sums the upper bound of the maximum number of broadcasted and relayed *IQ* and unicasted and relayed *IR* messages. This can be rearranged as $O((t+1)N)$. □

Therefore, the message complexity of broadcasting an *IQ* message and receiving *IR* messages until the retry count reaches n can be represented as $nO((t+1)N)$. After n times of repetitions, if the initiator receives negative *IR* messages, it sends *AB* messages to the *Requestor*. The message complexity of unicasting the *AB* message can be represented as $O(1)$. Therefore, the message complexity of a single node joining case can be represented as $nO((t+1)N)+O(N)+O(2)$ where $O(2)$ indicates the message complexity of unicasting *RR* and *AB*messages. Based on the above results, the following *Lemma* can be derived.

***Lemma* 7** In a session of a single node joining case, $nO((t+1)N)+O(N)+O(2)$ is the upper bound of the maximum number of broadcasted or relayed *IQ* and *AO* messages and unicasted or relayed *IR*, *RR*, and *AB* messages in *MANET*conf.

Proof. *MANET*conf has a *session* and the maximum number of retries of the *IP verification procedure* is limited by n in the *session*. Since the *session* consists of n maximum number of *IP verification procedures*and the upper bound of the maximum number of an *IPverification procedure* is $O((t+1)N)$ based on *Lemma* 6, the message complexity of the *session* can be represented as $nO((t+1)N)+O(N)+O(2)$ where n is the number of *IP verification procedures*, $O(N)$ indicates the message complexity of broadcasting the *AO* message and $O(2)$ indicates the message complexity of unicasting *RR* and *AB* messages. □

Ⅲ. Numerical Results

In order to analyze the message complexity of each address autoconfiguration protocol, a standalone *MANET* environment is needed, where the *MANET* nodes have no connection to an external network like the Internet [14]. Therefore, a computer simulator was developed where nodes are randomly distributed with uniform density in a network area of $1km^2$. A discrete-event simulator was developed in *Matlab* in order to verify the various network topologies and to calculate the message complexity of each address autoconfiguration protocol. The random node generator and simulator performance was verified (for the numbers of nodes 100, 125, 150, and 175) so that the average number of nodes per cluster as well as several specs in the *ADB* algorithm [19] matched with the results in [19], which was performed by *QualNet*, with less than a 1% difference for almost all cases.

In our analysis, the conflict probability is defined as the probability in which the *IP* address that a node requests to use is already in use in the *MANET* group. The conflict probability depends on the size of the address and the number of nodes in a *MANET* group [2]. The author of [2]

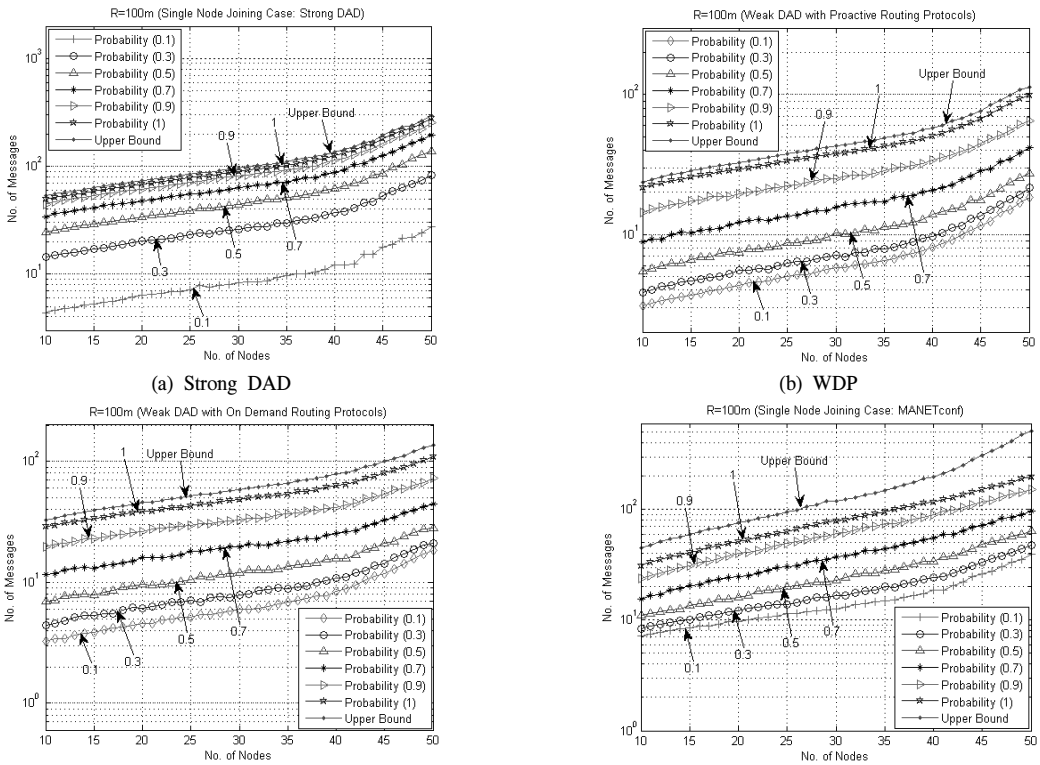
calculates the conflict probability, which is as high as 50% when an address space size of 16 bits is used as *MANET* local addresses in a network of 300 nodes.

In the graphs to follow, the message complexity is shownfor the conflict probabilities of 0.1, 0.3, 0.5, 0.7, 0.9, and .999 (which is denoted as 1 in the graphs to follow). When the conflict probability is almost one, the selected or reselected *IP* addresses will almost always conflict with one of the *IP* addresses in the *MANET* group. This event results in the worst case message complexity that has been obtained through the derivations in the former chapters.

It can be expected that in the simulation of *WDO*, having a different occurrence probability of unicasting a *RP* message at a certain conflict probability will result a different message complexity value. Therefore, for simplicity, in the simulation experiments to follow, it is assumed that the occurrence probability of unicasting a *RP* message in a node is not zero, and is the same as the conflict probability of the requested *IP* address for simplicity.

The most common flooding method used in the simulation is to have every node retransmit an *Address Request* message to all of its one-hop neighbors whenever it receives the first copy of the *Address Request* message [24]. Dijkstra's shortest path algorithm at each node is used to calculate the number of hops in unicasting or relaying an unicasted *Address Reply* message from a destination node to a source node. The transmission range of the nodes changes the number of hops. The upper bound of the message complexity is calculated based on the derived equation of the former chapter, where themaximum number of nodes in a reverse path at each unicast case is used to calculate $O(t)$ in each upper bound equation.

In the Strong *DAD* protocol, five is used for *retry count limit* (n) and three is used for *DAD retry count limit* (m). In the Weak *DAD* and



(a) Strong DAD

(b) WDP

(c) WDO

(d) MANETconf

Fig. 4. Message complexity

*MANET*conf protocols, five is used for *retry count limit* (*n*) and one is used for *DAD retry count limit* (*m*). In addition, 100*m*is selected as the transmission range of nodes. The number of nodes is varied from 10 to 50 for the transmission range.

3.1 Message Complexity Analysis

Fig. 4 shows the message complexities of the Strong *DAD*, WDP, WDO, and *MANET*conf protocols based on the different conflict probabilities. The horizontal axis represents the number of nodes in the network area and the vertical axis indicates the number of messages for each case. As the conflict probability increases, it is shown that the number of messages to resolve the duplicated *IP* address also increases.

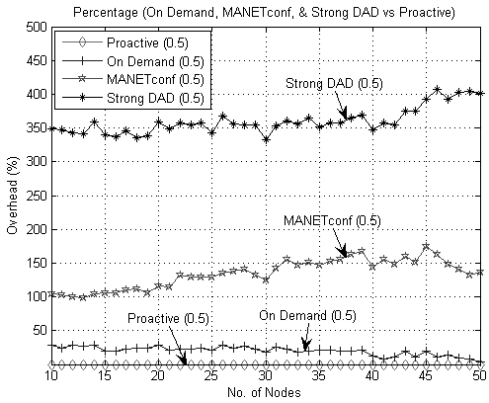
Fig. 4 shows that as the conflict probability approached 1 the message complexity approaches the derived theoretical message complexity upper bounds. However for the case of *MANET*conf, due to the difference in the consideration of the stateful addressing procedures, a small difference between the upper bound and the p=1 case can be observed.

3.2 Percentage Overhead Analysis

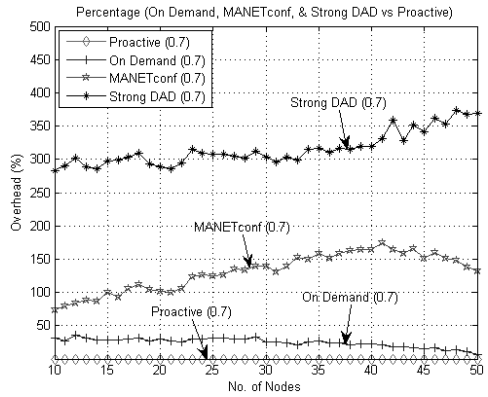
Fig. 5 provides a comparison of the percentage overhead among the *WDP*, *WDO*, *MANET*conf, and Strong *DAD* for a conflict probability of 0.5, 0.7, and 0.9, respectively. Table 2 and 3respectively compares the maximum and average overhead percentage based on the data obtained from Fig. 5.

It is shown that as the number of nodes increases, the message complexity of *WDO* tends to converge to the message complexity of *WDP*.

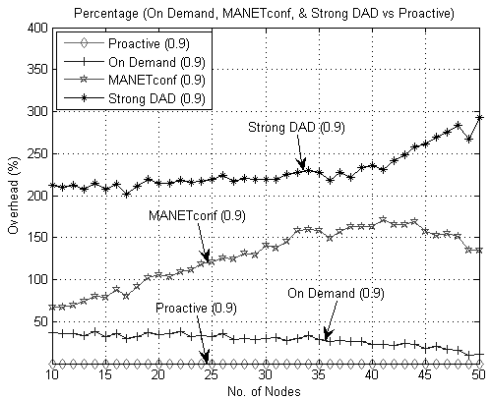
For the conflict probability of 0.5, 0.7, 0.9 and 1, *WDP* has the smallest message complexity and Strong *DAD* has the largest message complexity. However, since proactive routing protocols depend on a periodic message to update the network topology and on demand routing protocols do not need a periodic message, it is not fair for the two routing protocols to be compared usingthe number of messages. The result above can be used only in the case where the Weak *DAD* protocol uses *MANET*routing protocols for nodes to configure its *IP* address and solve the duplicate *IP* address detection.



(a) p=0.5



(b) p=0.7



(c) p =0.9

Fig. 5. Percentage overhead comparison

Table 2. Maximum overhead percentage [%]

p	*WDO*	*WDP*	*MANETconf*
0.5	29.30	174.04	407.18
0.7	36.15	174.56	374.24
0.9	38.41	171.28	292.58
1	33.71	136.64	180.99

Table 3. Average overhead percentage [%]

p	*WDO*	*WDP*	*MANETconf*
0.5	20.22	134.43	360.08
0.7	24.86	130.92	315.46
0.9	28.70	127.74	229.43
1	26.31	97.89	142.30

Based on the results on the average overhead percentage, since Strong *DAD* uses a *DAD retry count limit* m=3, it can be expected that the percentage overhead of Strong *DAD*will be three times larger than the one of *WDP*. Therefore, the average overhead percentages with conflict probability of 0.5, 0.7, and 0.9 tend to follow the expected result (behavior).

However, the average overhead percentage with a conflict probability very close to 1 does not follow the expected result (behavior). The maximum or average percentage overhead of the message complexity in the case of the conflict probability of one has a little difference value between *MANET*conf (97.89%, in the case of average percentage overhead) and Strong *DAD* (142.30%, in the case of average percentage overhead), which means that when the conflict probability is close to one, there is not much difference in the message complexity between *MANET*conf and Strong *DAD*.

Based on the percentage overhead of the message complexity of *MANET*conf, unicasting by all nodes causes approximately 135% (175%, from the results of maximum overhead percentage) more overhead than unicasting by a single node in the single node joining case. Based on the percentage overhead of the message complexity of *WDO*, another unicasting mechanism causes approximately 29% (39%, from the result of the maximum overhead percentage) more overhead compared to one of a single unicasting mechanism in the single node joining case.

Ⅳ. Conclusion

The main objective of this paper is to propose a novel method to perform a quantitative analysis of message complexity and to compare the message complexity among the *MANET AAPs*. To conduct a quantitative analysis of message complexity, the analysis of the worst case scenario is conducted in this paper. The main contributions of this paper are based upon the following accomplishments.

The original publications on the *AAPs* had many incomplete parts making them impossible to use on practical *MANET*s. Therefore, the first objective of the research was to complete the *AAPs* by filling in the missing gaps in the processing procedure to make the protocols operational. The missing procedures that were filled in have been developed based on the most logistic procedures staying as close/similar as possible to the original protocol procedures.

1. By introducing the *retry count limit* (n) of a *session* in Strong *DAD*, the possibility of resulting in an infinite loop has been removed. The original Strong *DAD* does not define the maximum number of retries of the *IP verification procedure*.
2. By adapting the mechanism of the replying AE message, introduced in [6], the Weak *DAD* protocol is equipped to properly react when solving duplicated *IP* address situations.
3. In *MANET*conf, the duplicated address node with the higher *Partition Identity* will become the *Requestor* asking its neighboring node to become its *Initiator*.

In addition, best to the authors' knowledge, except for *MANET*conf computer simulation, none of the *AAPs* have been investigated in reference to their complexity and scalability in *MANET* based operations. Therefore, research was conducted to provide a detailed derivation of the single node joining message complexity and extends the results to scalability and complexity analysis.

Table 4 summarizes the message complexity of a single node joining case in Strong *DAD*, *WDP*, *WDO*, and *MANET*conf.

Based on the simulation results and analysis of the message complexity in Tables 2 and 3, when nominal n, m, t, N values and transmission range have been assigned in a single node joining case with $p \leq 1$, the message complexity can be com-

Table 4. Comparison of the message complexity

*AAP*s	Message Complexity
Strong *DAD*	$n(mO(N)+O(t))$
WDP	$n(O(N)+O(t))$
WDO	$n(O(N)+2O(t))$
*MANET*conf	$nO((t+1)N)+O(N)+O(2)$

pared as follows: WDP < WDO < *MANET*conf < Strong *DAD*.

In the view point of the message complexity, when a *MANET* area is composed of a high conflict probability, Weak *DAD* with *MANET* routing protocols becomes a more suitable protocol than *MANET*conf and Strong *DAD* since Weak *DAD* with *MANET* routing protocols provides both routing and address autoconfiguration and much less message complexity compared to *MANET*conf and Strong *DAD*.

References

〔1〕 C. E. Perkins, J. T. Malinen, R. Wakikawa, E. M. Royer, and Y. Sun, "*IP* address autoconfiguration for ad hoc networks," *IETF draft*, 2001, http://www.cs.ucsb.edu/~ebelding/txt/autoconf.txt.

〔2〕 K. Weniger, "PACMAN: passive autoconfiguration for mobile ad hoc networks," *IEEE J. Select. Areas Commun.*, vol. 23, no.3, pp.507-519, Mar. 2005.

〔3〕 S. Thomson, and T. Narten, "*IP*v6 Stateless Address Autoconfiguration," *RFC 2462*, Dec. 1998.

〔4〕 K. Weniger, and M. Zitterbart, "Address autoconfiguration in mobile ad hoc networks: Current Approaches and Future Directions," *IEEE Network*, pp. 6-11, Jul./Aug. 2004.

〔5〕 J-. P. Jeong, J. Park, H. Kim, and D. Kim, "Ad hoc *IP* address autoconfiguration," *IETF draft*, July 2004.

〔6〕 N. H. Vaidya, "Weak duplicate address detection in mobile ad hoc networks," *Proc. ACM MobiHoc 2002*, pp. 206-216, June 2002, Lausanne, Switzerland.

〔7〕 M. Gerla, X. Hong, and G. Pei, "Fisheye State Routing Protocol (FSR) for Ad Hoc Networks," *Internet Draft*, June, 2002, http://www.ietf.org/proceedings/02jul/I-D/draft-ietf-*MANET*-fsr-03.txt.

〔8〕 R. Ogier, M. Lewis, and F. Templin, "Topology Broadcast Based on Reverse Path Forwarding (TBRPF)," *Internet Draft*, April. 2003, http://www.potaroo.net/ietf/ old-ids/draft-ietf-*MANET*-tbrpf-08.txt.

〔9〕 C. A. Santiváñez, R. Ramanathan, and I. Stavrakakis, "Making link-state routing scale for ad hoc networks," *Proc. of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, Long Beach, CA, USA, 2001.

〔10〕 T. Clausen, Ed., and P. Jacquet, Ed., "Optimized Link State Routing Protocol (OLSR)," *RFC 3626*, *The Internet Society*, Oct., 2003.

〔11〕 M. Gerla, X. Hong, L. Ma, and G. Pei, "Landmark Routing Protocol (LANMAR) for Large Scale Ad Hoc Networks," *Internet Draft*, Nov. 2002, http://www.ietf.org/proceedings/01dec/I-D/draft-ietf-*MANET*-lanmar-02.txt.

〔12〕 D. Johnson, D. A. Maltz, and Y.-C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," *Internet Draft*, April, 2003, http://www.potaroo.net/ietf/all-ids/draft-ietf-*MANET*-dsr-09.txt.

〔13〕 C. Perkins and E. Royer, "Ad hoc On Demand Distance Vector (AODV) Routing," *RFC 3561*, *The Internet Society*, July, 2003.

〔14〕 S. Nesargi and R. Prakash, "*MANET*conf: Configuration of hosts in a mobile ad hoc network," *Proc. IEEE Infocom 2002*, June 2002, New York, USA.

〔15〕 S. Cheshire and B. Aboba, "Dynamic Configuration of *IP*v4 Link-Local Address," *RFC 3927*, *The Internet Society*, Mar. 2005.

〔16〕 M. Moshin and R. Prakash, "*IP*address assignment in a mobile ad hoc network," *Proc. IEEE Milcom 2002*, pp. 856-861, Oct. 2002.

〔17〕 H. Zhou, L. M. Ni, and M. W. Mutka, "Prophet Address Allocation for Large Scale *MANET*s," *Ad Hoc Networks Journal*, vol. 1, issue 4,

pp. 423-434, Nov. 2003.

〔18〕 G. Cao and M. Singhai, “A delay-optimal quorum-based mutual execution algorithm for distributed systems,” *IEEE Trans. Parell and Distributed Systems*, vol. 12, no.12, pp. 1256-1268, Dec. 2001.

〔19〕 C-. C. Shen, and C. Srisathapornphat, and R. L. Z. Huang, and C. Jaikaeo, and E. L. Lloyd, “CLTC: A cluster-based topology control framework for ad hoc networks,” *IEEE Trans. Mobile Computing*, vol. 3, no.1, pp. 18-32, Jan.-Mar. 2004.

〔20〕 X. Hong, K. Xu, and M. Gerla, “Scalable Routing Protocol for Mobile Ad Hoc Networks,” *IEEE Network*, pp.11-21, Jul./Aug. 2002.

〔21〕 S. H. Kwok and A. G. Constantinides, “A fast recursive shortest spanning tree for image segmentation and edge detection,” *IEEE Trans. Image Processing*, vol. 6, no.2, pp. 328-332, Feb. 1997.

〔22〕 A. Boukerche, S. Hong, and T. Jacob, “An efficient synchronization scheme of multimedia streams in wireless and mobile systems,” *IEEE Trans. Parell and Distributed Systems*, vol. 13, no.9, pp. 911-923, Sep. 2002.

〔23〕 J. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1998.

〔24〕 M. Sheng, J. Li, and Y. Shi, “Relative Degree Adaptive Flooding Broadcast Algorithm for Ad Hoc Networks,” *IEEE Trans. on Broadcasting*, vol. 51, no. 2, pp. 216-222, June 2005.

김 상 철 (Sang-Chul Kim) 정회원

1994년 2월 : 경북대학교 전자 공학과 졸업

1998년 8월 : 창원대학교 컴퓨터 공학과 석사

1994년 3월~2000년 7월 : 삼성 항공, 삼성 SDS 시스템 엔지니어

2005년 12월 : 미국 Oklahoma State University Electrical & Computer Eng. Ph.D.

2006년 2월 : 미국 Univ. of Nevada Las Vegas, Computer Science, Post Doctor

2006년 3월~현재 : 국민대학교 컴퓨터 공학과

<관심분야> 디지털 유무선 통신공학, 로봇공학