25

Comparison of *DP* Effects in *MANET AAP*s with Link Error

Sang-Chul Kim
School of Computer Science, Kookmin University,
861-1, Chongnung-dong, Songbuk-gu, Seoul, 136-702
Korea

1. Introduction

A *MANET* consists of a set of mobile nodes where mobile nodes have routing capabilities to forward packets. Each mobile host becomes a member of a self-organizing wireless network, where one another communicate over multi-hop wireless links, without relying on a fixed communication infrastructure, such as a base station or an access point. It is essential that all nodes are able to perform the operations required for the configuration of unique addresses to execute proper routing of data packets in a *MANET*.

Address auto-configuration is an important issue, since address pre-configuration is not always possible in MANETs. MANETs currently depend on checking the IP addresses of nodes to decide if the connection and identification of nodes participating in a *MANET* are established. In conventional networks, address auto-configuration is categorized as either a stateless or a stateful protocol. When a network is not especially required to control the exact IP address assignments if the addresses are unique and routable, the stateless approach is used.

In contrast, the stateful approach is used when a network demands exact IP address assignments. 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 and verify its uniqueness in a distributed manner using *DAD* algorithms. Using *DAD* algorithms, a node in a *MANET*, which lacks an IP address in the *MANET*, can determine if a candidate address it selects is available. A node already equipped with an IP address also depends on *DAD* to protect its IP address from being accidentally used by another node in the *MANET*.

Based on the conventional method [1], *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*, checking if the address is being used in the *MANET*. Based on a reply to the claimed request, which needs to arrive at the node within a finite bounded time interval, the node can detect address duplication in the *MANET*.

Weak *DAD* is proposed, where ad hoc routing protocols are used to detect address duplication by modification of the routing protocol packet format. *MANET* routing

Source: Convergence and Hybrid Information Technologies, Book edited by: Marius Crisan,
ISBN 978-953-307-068-1, pp. 426, March 2010, INTECH, Croatia, downloaded from SCIYO.COM

protocols can be classified as 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. On-demand routing protocols issue route discovery mechanism messages, only when a node needs to send data to a destination node.
Since these protocols do not use a periodical message exchange, such as the neighbor discovery message used in proactive routing protocols, they do not hold route information at each node before a node sends data towards a destination node. Therefore, they need route request (*RQ*) and route reply (*RR*) messages to find and maintain a route when it is needed. Based on the above observation, the advantages and disadvantages of the proactive and on-demand routing protocols can be summarized as follow.
The main advantage of proactive routing protocols is that whenever a node sends a data packet, it obtains the route information to a destination searching its route table. Therefore, the route is already known and can be used immediately. In addition, there is no delay time in determining the route in the source node. However, a portion of the network resources in MANETs should be allocated to handle the periodic neighbor discovery and topology update messages, and this increases network traffic load.
The main advantage of on demand routing protocols is the reduction of network traffic overhead, as no messages are exchanged before the start of data communication. However, the delay caused by the route discovery mechanism to find a route to a destination could be a significant factor when considering *MANET*'s routing performance. As the node population and mobility increase, the routing control overhead in the *MANET* area also increases. This is a dominant factor to be considered in limited wireless bandwidth. The scalability issues in *MANET*'s proactive and on-demand routing protocols have been studied.

2. Related work

In regards to the mobility factor in MANETs, it is indicated in that the rate of link failure, due to node mobility, is the main concern of routing in ad hoc networks. *MANET* nodes move around according to their mobility scenarios, while they perform routing procedures simultaneously. Many papers deal with mobility patterns and mobility-based frameworks.
A broadcast request can be issued at any time by any host with a packet to be delivered to the entire network. A single transmission sent by each node will be received by all nodes within the node's transmission range. All other nodes need to cooperate to propagate the packet by rebroadcasting it. In [2], it is indicated that wireless ad hoc networks prefer localized algorithms and power-efficient network topologies, since a wireless ad hoc network has its own unavoidable limitations, where nodes have been powered by batteries and limited memory, in contrast to wired networks. The authors of [3] address the Lucent *WaveLAN* IEEE 802.11 wireless network interface consuming the power of 1,327 and 967 *mW* respectively when it transmits and receives at a transmission rate of 2 *Mbps*. [4] considers reduction of the number of broadcast messages, in which the authors focus on the concept of efficiency that is represented as the number of forward nodes, rather than reliability that is represented as the percentage of nodes receiving the broadcast packet [4].
One of possible methods to reduce broadcast redundancy is to perform the *AAP* and routing operations simultaneously. Passive Auto-configuration for Mobile Ad Hoc Networks (*PACMAN*) [5] uses routing protocol traffic to assign IP addresses. Since it uses routing messages to implement address configuration, it does not have control overhead to

implement *PACMAN*. The author of [5] indicates that even though IPv6 has sufficient address space to provide a unique IP address, it needs the IPv6 stateless address auto-configuration (*SSA*), since there is no hardware ID (e.g., 48 bits IEEE medium address control (*MAC*) address) that is truly globally unique. The author of [6] analyzes various address auto-configuration protocols for *MANET* and introduces the necessary routing protocols to enable reliable detection of all conflicts.

Much recent research has been conducted to reduce broadcast redundancy, since blind flooding in a wireless ad hoc networks has high cost and excessive redundancy [7]. The authors address two research approaches, probabilistic and deterministic, to obtain an efficient broadcast [7]. The probabilistic approach uses no or limited neighbor information and requires high broadcasts to maintain an acceptable packet delivery ratio. However, the deterministic approach finds the list of forward nodes to guarantee full network coverage.

In [8], a node does not forward a broadcast packet if a self-pruning algorithm is satisfied based on neighborhood information. Even though only a set of nodes forward the broadcast packet, this process guarantees complete network delivery. Self-pruning-based broadcast protocols [8] collect neighborhood topology information based on the *Hello* message and form a connected dominating set via forward nodes. *DP* [9] also offers a promising approach to reduce redundant transmissions caused by blind flooding. It is considered as an approximation to the minimum flood tree problem. The self-pruning algorithm uses the information of one-hop neighboring nodes; however, the *DP* algorithm utilizes two-hop neighborhood information.

Due to the multitude of factors to be considered for a *MANET*, the reduction of the routing overhead is the main concern during the development of a *MANET* protocol. One essential measure of the quality of a *MANET* protocol is its scalability with regard to an increase in the number of *MANET* nodes. Message complexity is defined where the overhead of an algorithm is measured in terms of the number of messages needed to satisfy the algorithm's request. This chapter proposes a novel idea where the *AAP*s (Strong *DAD*, Weak *DAD* and MANETconf) are able to perform routing. Therefore, the proposed algorithm can perform the *AAP* operation and routing simultaneously. In addition, since it is not well known how much improvement can be achieved when the *DP* algorithm substitutes the conventional blind flooding in the *MANET AAP*s, the performance is investigated in reference to complexity and scalability.

Therefore, the next goal of this chapter is to obtain a quantitative ratio of percentage reduction when the *DP* algorithm is used in *MANET AAP*s for the broadcast operation. Research was conducted to provide a detailed simulation of a single node joining message complexity and extends the results to scalability and complexity analysis. This chapter adopts the analysis of the worst case scenario [10] to conduct a quantitative analysis of message complexity.

The remainder of this chapter is organized as follows: Section 3 describes a detailed explanation of the proposed algorithm, particularly the concept of *AAP*s routing capability to reduce redundant transmission. In addition, it describes the proposed architecture of *AAP* algorithms. Section 4 addresses the numerical experiments and results. Finally, Section 5 summarizes our work and concludes the chapter.

3. Proposed algorithm

The proposed algorithm can be described in three sections. The first section introduces the procedures to enable *AAP*s to have routing capability, by creating new messages. The

second section illustrates how the *DP* algorithm substitutes blind flooding. The last section includes pseudocode to describe the detailed operation of the proposed *AAP* algorithms.

3.1 *AAP* with routing capability

In a standalone *MANET*, where a *MANET* has no connection to an external network, such as the Internet, the following two procedures are essential for each node in a *MANET* to be configured as a normal node in a conventional method. First, each node performs an *AAP* to obtain a unique IP address for proper routing of data packets in a *MANET*. Second, each node performs a *MANET* routing protocol to inform other nodes of the network topology and to send data packets towards a destination. This section addresses the procedure of a new *AAP* algorithm, where the routing capability has been implemented. Consequently, it is shown that the proposed algorithm reduces the complexity and solves scalability issues.

In the conventional approach where the *AAP* and routing are used separately, the messages can be classified into four categories. The message categories are: neighbor discovery (*Hello*), topology update (*TU*), address request (*AQ*) and address reply (*AR*). *Hello* and *TU* messages are designed for routing operation and *AQ* and *AR* messages are developed for *AAP* operation.

A new classification method, based on a forwarding method and a periodicity of message, can be proposed as follows. From the forwarding method, the message can be classified as broadcast, local broadcast, and unicast messages. From the periodicity, the message can be classified as periodical (implemented in the proactive *MANET* routing protocols) and non-periodical (implemented in the ondemand *MANET* routing protocols) messages.

Based on the above method, the *Hello* message is classified as a local broadcast message and a periodical message. *TU* message has the property of broadcast and periodical or non-periodical message, depending on routing protocols. *AQ* message is classified as broadcast and non-periodical message. *AR* message is unicast and non-periodical message. The summary of message property used in *MANET AAP* and *MANET* routing protocols are shown in Table 1.

Message	Forwarding Method	Periodicity	Used in	Prop. Algorithm
Hello	Local Broadcast	periodical	Routing	Yes
TU	Broadcast	non- or periodical	Routing	Yes
AQ	Broadcast	non-periodical	*AAP*	No
AR	Unicast	non-periodical	*AAP*	No

Table 1. Property of Messages

Two messages - *Hello* and *TU* - have been newly suggested in the proposed algorithm *MANET AAP*s to have the routing capability. As in the conventional use of the *Hello* message, the *Hello* message is designed only for neighbor discovery in the proposed algorithm. *TU* message has several different options, such as topology update, address request, and address reply. The following steps describe the process for the *TU* message to have routing capability and address auto-configuration. The topology update option gives mobile nodes the ability to implement routing capability.

In the *MANET* proactive routing protocols, *TU* message is generated periodically. In the *MANET* on-demand routing protocols, *TU* message is issued non-periodically, since on-demand message is randomly triggered, only when nodes find a route and respond by sending a route reply to the corresponding route request. The following procedure enables

the capability of address auto-configuration in the *TU* message. Whenever a node requires triggering an *AQ* message for a new joining node to be equipped with a unique IP address, it broadcasts *TU* message with the option of address request. In addition, it follows the periodic (when *TU* message is used in the proactive *MANET*) or non-periodic (when *TU* message is used in the on-demand *MANET*) property of *TU* message. That is, there might be some delay to generate *TU* message, until the next periodic (or non-periodic) time is issued.
When one of the nodes in a *MANET* detects a duplicated IP address, when the *TU* message with the option of address request is propagated into *MANET*, it responds by generating the *TU* message with the option of address reply. The *TU* message can broadcast, however, in the case of relaying the option of address reply, it unicasts the forwarding method where it follows the reverse path of the *TU* message with the option of the address request. A node waits until the next periodic (or non-periodic) time to generate and transmit the *TU* message with the option of route reply. Since non-periodicity does not guarantee triggering the *TU* message in a limited time, a node waits until a certain threshold time to generate or relay a non-periodic *TU* message. If a node does not have an event to trigger transmission of the *TU* message within the threshold time, the node autonomously generates a *TU* message.

3.2 *AAP* with *DP*

The detailed procedure of the proposed algorithm is described to implement the *DP* algorithm, to reduce the number of broadcast messages. The broadcast storm problem is a serious issue in a *MANET*. Hence, several algorithms are introduced to reduce the number of broadcast messages. The authors of [8] concluded that finding a minimum flood tree that gives the minimum number of forward nodes is proven to be *NP*-complete. They argued that even though a minimum flood tree is constructed, the maintenance cost of the tree in a mobile environment is too high to be useful in practice.
The *DP* algorithm [9] can reduce redundant transmission using 2-hop neighborhood information. Total dominant pruning (*TDP*) and partial dominant pruning (*PDP*) algorithms, introduced in [8], are proposed to overcome some deficiencies of the *DP* algorithm.
Since a source node knows the list of forward nodes, based on its neighboring nodes selected using the *DP*, *TDP* or *PDP* algorithm, all the neighboring nodes do not need to rebroadcast a packet issued by the source node. In contrast, all the neighboring nodes rebroadcast a packet issued by the source node in blind flooding. *DP*, *TDP* and *PDP* algorithms can reduce the total number of rebroadcasted packets and re-broadcast nodes compared to blind flooding. Adopting the *DP* algorithm can evaluate performance for the decision by nodes to rebroadcast packets in the proposed *AAP* algorithm that enables routing.
The following section describes the basic differences between blind flooding, self pruning and *DP* algorithms. Let us define $N(v)$ as the set of adjacent nodes of node v [8] [9]. $N(N(v))$ is defined as the set of nodes that is located within two-hops from node v [9] [10]. Due to the use of the periodic *Hello* message that informs the neighboring nodes of the presence of a node, self pruning and *DP* methods can collect the neighboring information periodically. Therefore, each node can construct its own neighboring list.
In self pruning, when a receiver node (r) receives a packet that piggybacks a neighboring list of a sender node (s), the receiver node r calculates if the set of $N(r) - N(s) - r$ is empty. If the set is empty, the receiver node r does not rebroadcast the packet, since $N(r)$ is covered by the sender node s. Otherwise, the receiver node r rebroadcasts the packet.

In conventional blind flooding, the receiver node r always rebroadcasts the packet, even though the set of $N(r) - N(s) - r$ is empty. This increases broadcast redundancy.
In *DP*, a sender node selects adjacent nodes in $B(s, r)$ (that equals $N(r) - N(s)$) that rebroadcast the packet, so that all nodes in $U=N(N(r)) - N(s) - N(r)$ receive the packet. The adjacent nodes also determine the forward list to complete flooding.
While self-pruning uses direct neighbor information only, *DP* uses neighborhood information up to two hops. The pruning methods require extra control overhead, since they use the periodic *Hello* messages for each node to get network topology information. Since nodes in a *MANET* use the periodic *Hello* messages in a normal (stable) status, the pruning methods can utilize the advantage of the periodic *Hello* messages.
Fig. 1 shows an example of the *DP* algorithm where node 2 is a source node. One-hop neighboring node set is represented as x and two hop neighboring node set is represented as y.

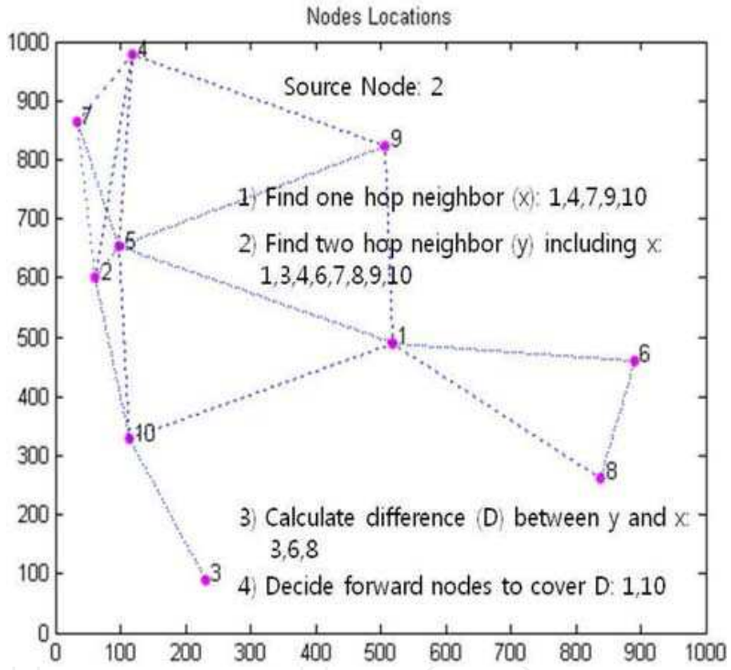

Fig. 1. An Example of Dominant Pruning

3.3 Proposed *AAP* algorithms

Strong *DAD* uses an address discovery mechanism, where a node randomly selects an address and requests the address within a *MANET*, checking if the address is being used in the *MANET*. Based on the reply to the claimed request, which needs to arrive at the node within a finite bounded time interval, the node can detect address duplication in the *MANET*. Weak *DAD* is used to detect address duplication by modification of the routing protocol packet format. MANETconf 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*.
Figs. 2, 3, and 4 show the pseudo code for Strong *DAD*, *WDP*, *WDO*, and MANETconf operations in the simulation respectively, where the newly proposed messages in this chapter are used with its option to implement the autoconfiguration process. In the

conventional broadcast (and its simulation), the most common flooding method is used to broadcast a *TU* (*AQ*: Address Request option) message where every node retransmits a *TU* (*AQ* option) message to its entire one-hop neighbors whenever it receives the first copy of the a *TU* (*AQ* option).

```
Start
Step 01: A node selects a temporary address
         and configures it as its network interface address
Step 02: n=0; (Set retry count (n) =0)
Step 03: m=0; (Set DAD retry count (m) = 0)
Step 04: n++; (Increase the retry count (n) by 1)
Step 05: m++; (Increase the DAD retry count (m) by 1)
Step 06: The node randomly selects a source IP address
         and makes a TU (AQ) message for the IP address
Step 07: The node broadcasts the TU (AQ)
Step 08: if (all MANET nodes receive the TU (AQ) == TRUE)
Step 09:    if (a TU (AP) arrives to the node before timer expires
                == TRUE)
Step 10:       if ((retry count <= n ) == TRUE)
Step 11:        goto Step 4;
Step 12:       else
Step 13:        goto Step 21;
Step 14:    else
Step 15:       if ((DAD retry count <= m ) == TRUE)
Step 16:          The node replaces the source IP address with its IP address
              break;
Step 17:       else
Step 18:          goto Step 5;
Step 19: else
Step 20: goto Step 7;
Step 21: The node fails to get a source IP address
End
```

Fig. 2. Strong *DAD* Operations

However, in the proposed algorithm (and its simulation), the *DP* algorithm is used to replace the conventional flooding algorithm. *Dijkstra*'s shortest path algorithm at each node is used to calculate the number of hops in unicasting or relaying a unicast *TU* (*AP*: Address Reply option) from a destination node to a source node. In Strong *DAD*, the retry count limit (*n*) is five and for *DAD* the retry count limit (*m*) is three. In Weak *DAD* and MANETconf protocols, the retry count limit (*n*) is five and for *DAD* retry count limit (*m*) is one.

4. Numerical results

In the simulation, a single node joining case in the largest sub-network, among several partitioned sub-networks, is considered to perform the evaluation. The computer-based simulator was written to implement the proposed algorithm. In the simulator, the forward node list (*F*) implemented by the *DP* algorithm has been selected to rebroadcast messages. Since only the forward nodes in the neighboring list can broadcast the *TU* message, it is shown that the message complexities of the proposed *AAP*s are significantly reduced, compared to the blind flooding method.

```
Start
Step 01: A node selects a temporary address
         and configures it as its network interface address
Step 02: n=0; (Set retry count (n) =0)
Step 03: n++; (Increase the retry count (n) by 1)
Step 04: The node randomly selects a source IP address and picks a
         unique key value (e.g., MAC address) as the identification of the node;
Step 05: if (Proactive routing protocol is used == TRUE)
Step 06:   The node broadcasts a TU (LS) periodically
Step 07:   if (all MANET nodes receive the TU (LS)
                == TRUE)
Step 08:     if (the node receives a TU (AE) for the selected IP address
                == TRUE)
Step 09:       if (retry count <= n)
Step 10:         goto Step 3;
Step 11:       else
Step 12:         The node fails to get a source IP address, goto End
Step 13:     else
Step 14:       The node replaces the source IP address with
                    its IP address, goto End
Step 15:   else
Step 16:     goto Step 6;
Step 17: else
Step 18:   The node broadcasts a TU (RQ) whenever it needs to
Step 19:   if (all MANET nodes receive the TU (RQ))
Step 20:     if (the node is the destination of a TU (RQ))
Step 21:       The node unicasts a TU (RP).
Step 22:     else
Step 23:       goto Step 8
Step 24:   else
Step 25:     goto Step 18
End
```

Fig. 3. Weak *DAD* Operations

```
Start
Step 01: A requester (new joining node) selects an initiator and
         unicasts Hello (RR) to the initiator
Step 02: n=0; (Set retry count (n) =0)
Step 03: n++; (Increase the retry count (n) by 1)
Step 04: The initiator broadcasts a TU (IQ) to all the nodes of the MANET group
       with the address of the requestor
Step 05: if (all MANET nodes receive the TU (IQ) == TRUE)
Step 06:   Recipient nodes reply with an affirmative or
             a negative response (TU (IR)) to the initiator
Step 07: else
Step 08:   goto Step 4;
Step 09: if (the initiator receives affirmative TU (IR) messages
             from all nodes == TRUE)
Step 10:   The initiator assigns the IP address to the requestor
Step 11:   The initiator broadcasts a TU (AO) message to all
             recipient nodes of the MANET group, goto End
Step 12: else
Step 13:   The initiator selects another IP address
Step 14: if (retry count <= n)
Step 15:   The initiator sends a TU (AB) message to the requestor,
             goto End
Step 16: else
Step 17:   goto Step 3;
End
```

Fig. 4. *MANETconf* Operations

A system model that is used to analyze the proposed algorithm follows the system model introduced in [10]. For a given link error probability of P_e, the retransmission count limit value R can be defined based on the network manager's desired setting, some optimal criteria, and/or the mobile node's priority. For a given link error probability, the average number of transmissions (T_N) required for success ful reception is provided in (1). This can be used as a reference value for the retransmission count limit value R.

$$T_N = \frac{1}{1 - P_e}, \ for \ 0 \leq P_e < 1 \tag{1}$$

Since a link error can stop message propagations, a node that experiences link errors needs to rebroadcast the messages to its neighboring nodes. It is assumed that a node is able to learn of transmission failure using acknowledgments from the lower layers. Based on the detected link error probability, a network controller can set the retransmission count limit R to a desired value. A standalone *MANET* environment is needed to compare the message complexity between the conventional *AAP*s and the proposed *AAP*s, where the *MANET* nodes have no connection to an external network, such as the Internet.

A computer-based 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* to verify the various network topologies and to calculate message complexity. The random node generator and simulator performance were verified (number of nodes: 100, 125, 150, and 175) so that the average number of nodes per cluster as well as several of the specifications in the adaptive dynamic backbone (*ADB*) algorithm [10] matched the results in [10]. This was performed on *QualNet*, with less than a 1% difference in most cases. In our analysis, the conflict probability (P_c) 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.

The blind flooding used in the simulation, which is compared to the *DP* algorithm, is to have every node retransmit a message to all of its one-hop neighbors, whenever it receives the first copy of the message. In addition, the node transmission range is selected to be $150m$. The number of nodes is varied from 10 to 100.

In the computer simulation, the values of 0.25 and 0.5 are used for P_e; P_c of 0.5 is used in the following graphs. Corresponding to each of P_e values of 0.25, and 0.5, the retransmission count R has been set to 1.33 and 2 respectively, based on (1).

Figs. 5 and 6 show the simulation of message complexities between the conventional *AAP* algorithms and the proposed algorithm, when P_e values of 0.25 and 0.5 are used respectively. Even if the *DP* method reduces message complexity, it can be shown that message complexity linearly increases with increases in link error probability. The messages used in the conventional *AAP* and the proposed algorithm are different. It is assumed that the messages lengths are the same. For example, the conventional Strong *DAD* uses *AQ* and *AP* messages; however, the proposed Strong *DAD* uses the *TU* message with the option of address request and the *TU* message with the option of address reply.

In the graph, the x axis shows the number of nodes in a *MANET* and the y axis shows the message complexities of the conventional *AAP* and the proposed algorithm. It can be shown that at the conflict probability of 0.5, until the node number is 55, conventional Strong *DAD* has the largest message complexity, after the node number exceeds 55, conventional MANETconf has the largest message complexity and *WDP* with the proposed algorithm has the least message complexity.

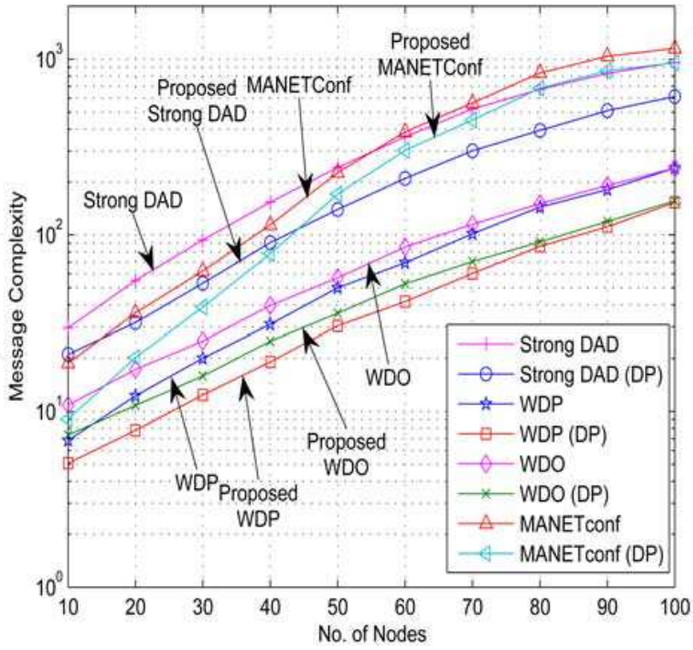

Fig. 5. Message Complexity (P_c=0.5, P_e=0.25)

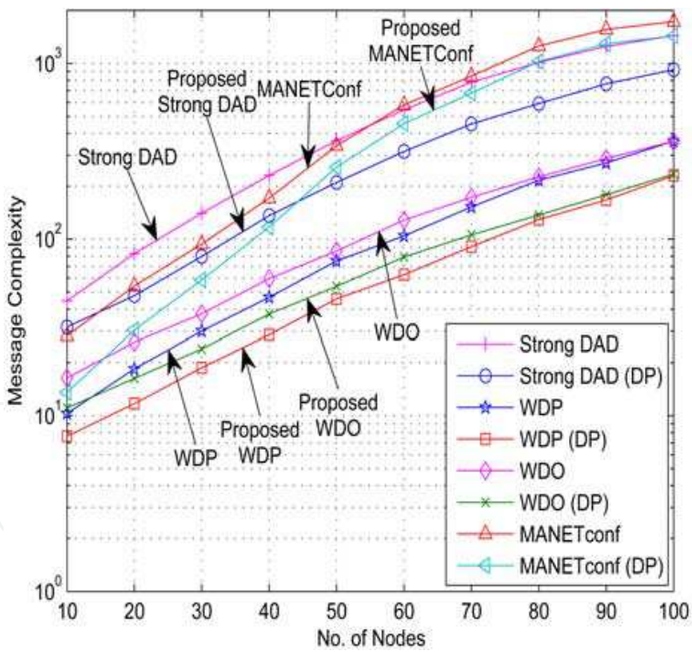

Fig. 6. Message Complexity (P_c=0.5, P_e=0.5)

As shown in Fig. 7, message complexity of 39.8%, 37.3%, 37.0% and 28.4% has been reduced respectively in comparison to the message complexity of the conventional Strong *DAD*, *WDP*, *WDO* and MANETconf when the Pe value equals zero. In the node range between 10 and 100, the reduced overhead percentage of the proposed algorithm is shown in Fig. 7.
It can be said that the reduction rate of Strong *DAD* is noticeably greater than the reduction rate of other *AAP*s, since Strong *DAD* uses more recursive broadcast mechanisms to resolve

duplicated IP addresses or for routing than other *AAP*s. In MANETconf, it is shown that as node number increases, message complexity rapidly decreases. Since in MANETconf all nodes unicast as the main operation, as node number increases, message complexity affected by broadcasting is reduced. In contrast, the reduction rate of message complexity affected by all nodes unicasting increased.

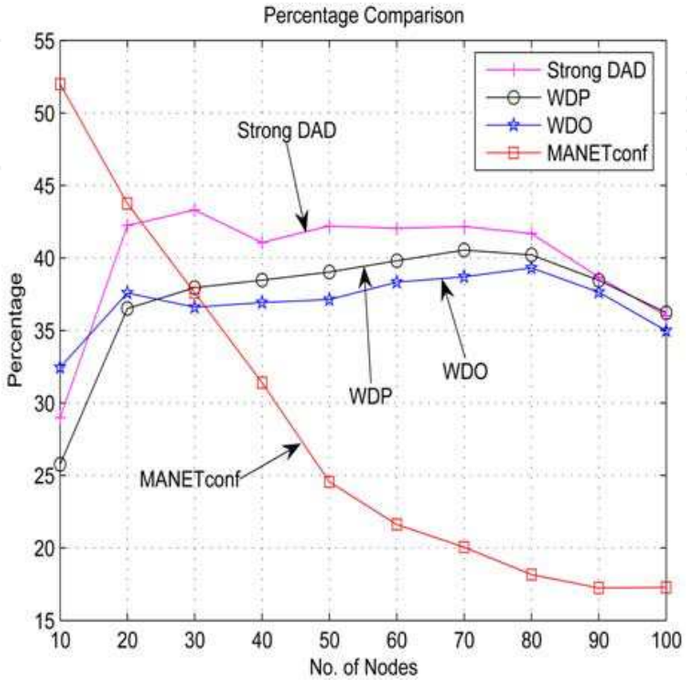

Fig. 7. Percentage Difference Comparison (P_c=0.5, P_e=0)

5. Conclusion

The wireless communication environment and the mobility of the nodes destabilize links. This results in link errors. Based on the link error probability, this chapter proposes two novel algorithms where the broadcasting redundancy was noticeably decreased using the *DP* algorithm and different messages used in *MANET AAP*s and routing algorithms are combined using *Hello* and *TU* messages.

The proposed algorithm can save the total number of control messages, compared to the conventional algorithm, due to the reduced number of *TU* messages generated in *AAP* and routing. The simulation shows the proposed algorithm saves 39.8%, 37.3%, 37.0% and 28.4% of message complexity compared to the conventional Strong *DAD*, *WDP*, *WDO* and MANETconf.

Several characteristics of *AAP*s are found. First, since Strong *DAD* uses more recursive broadcast mechanisms to resolve duplicated IP addresses compared to other *AAP*s, the reduction rate of Strong *DAD* is greater than the reduction rate of other *AAP*s. Second, it is shown in MANETconf that as node number increases, the reduction rate of message complexity rapidly decreases. Since in MANETconf in the main operation all nodes unicasts, as node number increases, the reduction rate of message complexity affected by broadcasting reduces, while the reduction rate of message complexity affected by unicasting by all nodes increases.

6. Acknowledgments

This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (KRF-2009-007128) and the Seoul R&BD Program (No. 10848).

7. References

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

[2] X.-Y. Li, Y. Wang, and W.-Z. Song, "Applications of k-local MST for topology control and broadcasting in wireless ad hoc networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 15, no. 12, pp. 1057-1069, Dec. 2004.

[3] C.-C. Shen, Z. Huang, and C. Jaikaeo, "Directional broadcast for mobile ad hoc networks with percolation theory," *IEEE Trans. Mobile Computing*, vol. 5, no. 4, pp. 317-332, Apr. 2006.

[4] J. Wu and F. Dai, "A Generic Distributed Broadcast Scheme in Ad Hoc Wireless Networks," *IEEE Trans. On Computers*, vol. 53, no. 10, pp. 1343-1354, Oct. 2004.

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

[6] K. Weniger, and M. Zitterbart, "Mobile ad hoc networks - current approaches and future directions," *IEEE Network*, vol. 18, issue 4, pp.6-11, July-Aug. 2004.

[7] F. Dai and J. Wu, "Efficient broadcasting in ad hoc wireless networks using directional antennas," *IEEE Trans. on Parallel and Distributed Systems*, vol. 17, no. 4, pp. 335-347, Apr. 2006.

[8] W. Lou and J. Wu, "On reducing broadcast redundancy in Ad hoc wireless networks," *IEEE Trans. on Mobile Computing*, vol. 1, no. 2, pp. 111-122, Apr.-June, 2002.

[9] H. Lim and C. Kim, "Flooding in wireless ad hoc networks," *Computer Comm. J.*, vol. 24, no.3-4, pp. 353- 363, 2001.

[10] C-. C. Shen, C. Srisathapornphat, R. L. Z. Huang, C. Jaikaeo, and E. L. Lloyd, "CLTC: A cluseter-based topology control framework for ad hoc networks," *IEEE Trans. Mobile Computing*, vol. 3, no.1, pp. 18-32, Jan.- Mar. 2004.

[11] S.-C. Kim and J.-M. Chung,"Message Complexity Analysis of Mobile Ad Hoc Network Address Autoconfiguration Protocols," *IEEE Trans. Mobile Computing*, vol. 7, no. 3, pp. 358-371, Mar. 2008.

Convergence and Hybrid Information Technologies

Edited by Marius Crisan

ISBN 978-953-307-068-1
Hard cover, 426 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

Starting a journey on the new path of converging information technologies is the aim of the present book. Extended on 27 chapters, the book provides the reader with some leading-edge research results regarding algorithms and information models, software frameworks, multimedia, information security, communication networks, and applications. Information technologies are only at the dawn of a massive transformation and adaptation to the complex demands of the new upcoming information society. It is not possible to achieve a thorough view of the field in one book. Nonetheless, the editor hopes that the book can at least offer the first step into the convergence domain of information technologies, and the reader will find it instructive and stimulating.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen