

Enhanced Termination Condition for Deterministic Broadcasting Protocols in Mobile Ad Hoc Networks

Wilson Woon and Kwan L. Yeung
Department of Electrical and Electronics Engineering,
The University of Hong Kong,
Pokfulam, Hong Kong
Email: {thwoon, kyeung}@eee.hku.hk

Abstract—Deterministic approach to broadcasting in Mobile Ad Hoc Networks (MANETs) is effective in reducing redundant broadcasting. In this approach, a transmitting node selects a subset of its immediate or 1-hop neighbors to rebroadcast the message such that all its 2-hop neighbors will receive the message, or being *covered*. In order to reduce redundant broadcasting, the set of 1-hop neighbors to be covered should be reduced as much as possible. Another important aspect that affects the effectiveness of a deterministic broadcasting protocol is the termination condition that inhibits a node from transmitting a particular message unnecessarily. However, existing termination conditions are not optimized. We propose a new *covered/uncovered* termination condition where each node is assigned with *covered/uncovered* status. In this paper, we show that our *covered/uncovered* termination condition ensures full network coverage, does not incur any control message overhead, and yet requires the least number of rebroadcasting nodes. When we apply the termination condition to some existing deterministic broadcasting protocols, the saving in the number of broadcasting nodes can be as significant as 45% when there are 100 nodes randomly distributed in an area of 1000 x 1000 m^2 .

I. INTRODUCTION

Broadcasting is a way of disseminating information to all nodes in a network. It also plays an important role in establishing routes for on demand routing protocols [7], building routing tables for table-driven routing protocols, and address assignment. In a large network, due to the limited transmission power, a single broadcasting or transmission by a source node is usually insufficient to reach all nodes. In this case, the source node requires the assistance of its immediate neighbors to forward or rebroadcast the message to the nodes within their coverage area that have not received the information. This process continues until all nodes in a network received a copy of the information. An effective broadcasting protocol minimizes the number of nodes involved in forwarding or broadcasting a message. Any saved broadcasting could reduce packet collisions in congested networks, prolonging the lifetime of nodes that are battery powered, and lower channel utilization.

There are two popular distributed approaches in designing broadcasting protocols for Mobile Ad Hoc Networks (MANETs), namely self-pruning and deterministic. The self-pruning approach gives each node the freedom to decide on its own whether or not to rebroadcast a message (usually after a “backoff” timer expires). The simplest example is

pure flooding, where each node rebroadcasts every unique message received exactly once (and as soon as the message is received). Obviously, pure flooding incurs a lot of redundant broadcasting. More efficient broadcasting protocols based on self-pruning can be found in [1-3], and some of them require neighborhood information for setting the “backoff” timer. In order to obtain updated neighborhood information, periodic HELLO message exchanges among neighboring nodes is required. The main advantages of the self-pruning approach are simple, easy to implement, and quite robust to changes in network topology.

On the other hand, the deterministic approach [4-6] requires a transmitting or broadcasting node to explicitly select a subset of its immediate neighbors (or 1-hop neighbors) to cover all its 2-hop neighbors. That is, before transmitting a message, a list of selected forwarding nodes’ IDs is attached with the message. A node that receives the message will check if it has been selected as a forwarder. If yes (and its broadcast termination condition specified in the next section is not triggered), it will in turn select a set of forwarding nodes before rebroadcasting the message. Otherwise, it will discard the message. As compared with self-pruning, the deterministic approach tends to be more complex as it requires additional message and processing overhead. However, only a small portion of nodes will perform the selection of forwarding nodes and participate in the broadcasting.

One common feature of both self-pruning and deterministic protocols is the broadcast termination condition. When a node is selected (using a deterministic protocol) or decides by itself (using a self-pruning protocol) to rebroadcast a message, it does not need to carry out the actual task of rebroadcasting if it deems such a broadcasting is unnecessary based on its termination condition. Obviously, termination condition also plays vital role in determining the performance of a broadcast protocol.

For a self-pruning protocol, the termination condition is straightforward. If all its 1-hop neighbors are covered (i.e. already received the message), the node terminates its broadcasting. To find out if all 1-hop neighbors are covered [1], the neighborhood knowledge is used together with a “backoff” timer. The purpose of a “backoff” timer is to defer the rebroadcasting action, thereby gaining sufficient time to infer the status of its 1-hop neighbors. When the timer expires and

there are still uncovered neighbors, the message is rebroadcast. Otherwise, the broadcasting is inhibited.

To the best of our knowledge, there are two termination conditions that have been proposed for deterministic broadcasting protocols [4]. The first one assigns a *marked/unmarked* status to each node. A node is *marked* if it has received a message. Once a node has received a message, it has to send a *marked* control message to inform all 1-hop neighbors about its status change. A node stops broadcasting if all its neighbors are *marked*. In the second termination condition, a *relayed/unrelayed* status is assigned to each node. Once a (*unrelayed*) node broadcasts a message, it becomes *relayed*. A *relayed* node inhibits itself from broadcasting the same message again. It is shown in [4] that the *marked/unmarked* termination condition is more effective than the *relayed/unrelayed* condition in suppressing the number of unnecessary rebroadcasting. But the need for sending *marked* control messages is costly.

Focusing on deterministic broadcasting protocols, we propose a very simple but extremely effective termination condition in this paper. To distinguish with the two existing conditions [4], we call ours *covered/uncovered* termination condition. Specifically, a node is assigned with *covered/uncovered* status. When an *uncovered* node receives a message and it is not selected as a forwarder by the sender, it becomes *covered*. When an *uncovered* node receives a message and it is selected as a forwarder, it becomes *covered* **after** rebroadcasting the message. When a *covered* node receives a message, the message is dropped upon arrival. In other words, broadcast terminates at a *covered* node. In this paper, we show that our *covered/uncovered* termination condition ensures full network coverage, does not incur any control message overhead, and yet requires the least number of rebroadcasting nodes.

This rest of this paper is organized as follows. Section II reviews existing broadcasting protocols based on the deterministic approach and the two termination conditions they adopted. In Section III, our proposed *covered/uncovered* termination condition is presented, and its effectiveness is quantitatively studied in Section IV by simulations. Finally, Section V concludes the paper.

II. EXISTING DETERMINISTIC BROADCASTING PROTOCOLS

Without loss of generality, we assume that all nodes have knowledge of their 1-hop (immediate) and 2-hop neighbors. This information can be obtained via periodic HELLO message exchanges [8]. A HELLO message contains the identity of the sender as well as its 1-hop neighbors. Upon receiving the message, a node will treat the 1-hop neighbors of the sender as its 2-hop neighbors. The set of 1-hop and 2-hop neighbors of a particular node u are denoted as $N(u)$ and $N(N(u))$ respectively.

In the following, we first focus on the mechanisms for selecting forwarding nodes, where three deterministic broadcasting protocols, DP, TDP and PDP are reviewed. Then the two broadcast termination conditions, *marked/unmarked* and *relayed/unrelayed*, are discussed. Note that termination

condition is an integral part of a broadcasting protocol even though we present them separately below for clarity.

A. Dominant Pruning (DP) Protocol

The DP algorithm [4] is one of the earliest deterministic broadcasting protocols. In this algorithm, a node v that receives a broadcast message from source node u selects a minimum number of forwarding nodes from $N(v)$ to cover all nodes in $N(N(v))$. Among the nodes in $N(N(v))$, nodes in $N(u)$ have already received the message while nodes in $N(v)$ will receive it when node v rebroadcasts the message. Therefore, node v just need to select its forwarding nodes from the set $B(u, v) = N(v) - N(u)$ to cover all 2-hop neighbors in the set $U(u, v) = N(N(v)) - N(u) - N(v)$. The selection of forwarding nodes can follow the greedy algorithm in [5]. Firstly, a node in the set $B(u, v)$ is selected as a forwarder if it is the only node that can cover a node in the set $U(u, v)$. The process continues by repeatedly selecting nodes in $B(u, v)$ that can cover the maximum number of uncovered nodes in $U(u, v)$. In case of a tie, the node with the smallest ID is selected. Here, uncovered nodes refer to nodes that are not-yet-covered by a node in $B(u, v)$.

Assume nodes u and v are the source and a selected forwarding node respectively. The DP algorithm is summarized below:

- 1) Node v establishes the set $B(u, v)$ and $U(u, v)$ using $N(N(v))$, $N(u)$, and $N(v)$:

$$U(u, v) = N(N(v)) - N(u) - N(v)$$

$$B(u, v) = N(v) - N(u)$$

- 2) Node v then executes the greedy algorithm in [5] to select forwarding nodes from $B(u, v)$ to cover all nodes in $U(u, v)$.

B. Total Dominant Pruning (TDP) Protocol

TDP [5] further reduces the size of $U(u, v)$ by allowing node v to receive a message piggybacked with $N(N(u))$ from node u . Therefore, $U(u, v) = N(N(v)) - N(N(u))$ and $B(u, v) = N(v) - N(u)$. Similar to DP, the greedy algorithm [5] is adopted to select forwarding nodes from the set $B(u, v)$ to cover all nodes in $U(u, v)$. TDP is more effective than DP in reducing redundant broadcasting but it incurs additional overhead in piggybacking each data message with a list of 2-hop neighbors of the senders.

The TDP algorithm is summarized below:

- 1) Node v establishes the set $B(u, v)$ and $U(u, v)$ using $N(N(v))$ and $N(N(u))$:

$$U(u, v) = N(N(v)) - N(N(u))$$

$$B(u, v) = N(v) - N(u)$$

- 2) Node v then executes the greedy algorithm in [5] to select forwarding nodes from $B(u, v)$ to cover all nodes in $U(u, v)$.

C. Partial Dominant Pruning (PDP) Protocol

The approach taken by PDP algorithm does not require additional overhead, like TDP. Instead of just excluding nodes in $N(u)$ and $N(v)$ from the set $U(u, v)$, nodes in the set $P(u, v) = N(N(u) \cap N(v))$ can be excluded as well. Therefore, the 2-hop neighbor set to be covered is now reduced to $U(u, v) = N(N(v)) - N(u) - N(v) - P(u, v)$.

The PDP algorithm is summarized below:

- 1) Node v establishes the set $B(u, v)$ and $U(u, v)$ using $N(N(v))$, $N(u)$, $N(v)$, and $N(N(u) \cap N(v))$:

$$U(u, v) = N(N(v)) - N(u) - N(v) - N(N(u) \cap N(v))$$

$$B(u, v) = N(v) - N(u)$$

- 2) Node v then executes the greedy algorithm in [5] to select forwarding nodes from $B(u, v)$ to cover all nodes in $U(u, v)$.

The correctness of DP is given in [6] while TDP and PDP are given in [4]. A previous study [4] concludes that TDP and PDP algorithms are more effective in reducing redundant broadcasting compared with DP algorithm. The difference in number of transmitting nodes between TDP and PDP is marginal in most cases.

D. Broadcast Termination Conditions

Thus far, the discussion focuses on how to select forwarding nodes. In this section, we describe the broadcast termination conditions that govern whether a forwarding node should indeed rebroadcast a message as it is being asked for. Even though the process of selecting forwarding nodes is deterministic, the decision of whether to actually rebroadcast or not is based on a given termination condition. The effectiveness of a broadcasting protocol in reducing redundant broadcasting is highly influenced by the termination condition used.

There are two termination conditions as proposed in [4]. The first one assigns a *marked/unmarked* status to each node. A forwarding node will rebroadcast a message only if there is at least one *unmarked* neighbor. Otherwise, the message is dropped. Initially, all nodes are *unmarked*. Upon receiving a broadcast message, a node will broadcast a control message “*marked*” to inform its neighbors regarding the change of its status. It is proven in [4] that if all the 1-hop neighbors of a selected forwarder have received the message (i.e. are *marked*), the selected forwarder does not need to rebroadcast the message because all its 2-hop neighbors have already been (or will be) covered by other broadcasting nodes. This *marked/unmarked* approach is very costly because it involves additional transmission of control messages. Moreover, this overhead could result in packet collisions in congested networks as well as additional energy consumption. Each node

will also need to keep track of the status of each neighbor, thus making it an expensive mechanism.

The second termination condition assigns a *relayed/unrelayed* status to each node. When a node has *relayed* or forwarded a message, its status is changed from *unrelayed* to *relayed*. Unlike the *marked/unmarked* termination condition, there is no need to inform all neighbors about the change of status. A selected forwarding node will be inhibited from transmitting if its status is *relayed*. The correctness of this termination condition is trivial because if a node has broadcast the message before, then all its 1-hop and 2-hop neighbors have already been covered and further rebroadcasting of the same message is redundant. Compared with the *marked/unmarked* termination condition, the *relayed/unrelayed* approach is more applicable in a real network but as shown in [4], it is not very effective in ensuring minimum number of broadcasting nodes.

As discussed previously, the *marked/unmarked* and *relayed/unrelayed* termination conditions determine whether a node selected for rebroadcasting should indeed rebroadcast a message or not. The DP protocol uses another “termination condition” when selecting forwarding nodes from the set $B(u, v)$. There is a possibility that a node in $U(u, v)$ may not be covered by any nodes in $B(u, v)$. Therefore, the DP protocol may get caught in a loop when executing step 2 of part A. In order to avoid this problem, a node should “terminate” the greedy algorithm [4] (or step 2 of part A) whenever no new forwarding node is selected from $B(u, v)$.

III. COVERED/UNCOVERED TERMINATION CONDITION

In this section, we propose a new broadcast termination condition. To distinguish with the two existing termination conditions [4], we call ours *covered/uncovered* termination condition. Specifically, a node is assigned with *covered/uncovered* status. Initially, all nodes are *uncovered*. When an *uncovered* node receives a message and it has not been selected as a forwarder by the sender, it becomes *covered*. When an *uncovered* node receives a message and it has been selected as a forwarder, it becomes *covered* **after** rebroadcasting the message. When a *covered* node receives a message, the message is dropped upon arrival. In other words, broadcast terminates at a *covered* node.

In the following, we show that our *covered/uncovered* termination condition ensures full network coverage, i.e. when the broadcast session completes, all nodes in the network receive the message and become *covered*.

Theorem 1: Using the proposed *covered/uncovered* termination condition, all nodes in the network will receive the message upon termination.

Proof: While an arbitrary node A is transmitting using a deterministic broadcasting protocol, in addition to all its 1-hop neighbors in $N(A)$, it must ensure all its 2-hop neighbors in $N(N(A))$ will receive the message. Assume node B receives the message from node A and is not selected as a forwarding node. Node B becomes *covered* using our *covered/uncovered* termination condition. Node B must be a member of $N(A)$,

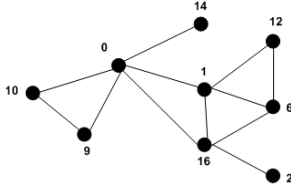


Fig. 1. An example that shows EDP outperforms DP

and $N(B)$ must be a subset of $N(N(A))$. From the basic operation of a deterministic broadcasting protocol, we know that all nodes in $N(B)$ are guaranteed to receive the message. It is likely that some nodes in $N(B)$ may have already received the message from other broadcasting nodes, some may receive the message together with node B , and some may receive the message later on from the forwarding nodes selected by node A .

We want to show that if all nodes in $N(B)$ are *guaranteed* to receive the message (without rebroadcasting by node B), node B itself does not need to broadcast. From the proof for the *marked/unmarked* termination condition, i.e. Theorem 4 in [4], we know that if all the 1-hop neighbors of a selected forwarder *have already received* the message (i.e. are *marked*), the selected forwarder does not need to rebroadcast because all its 2-hop neighbors are guaranteed to receive the message from *other* broadcasting nodes.

It is obvious that if all the 1-hop neighbors of node B are guaranteed to receive the message (instead of already received the message), all its 2-hop neighbors are still guaranteed to receive the message from *other* broadcasting nodes. ■

Examples: We now compare the performance of our *covered/uncovered* termination condition with the *marked/unmarked* condition in [4] using some designed examples. Note that termination condition must work with a broadcasting protocol. To this end, the three protocols reviewed in Section II are used. We use DP, TDP, and PDP to denote the protocols incorporated with the original *marked/unmarked* termination condition, while Enhanced DP (EDP), TDP (ETDP), and PDP (EPDP) are the protocols with our proposed termination condition.

Fig. 1 shows an example where EDP would outperform DP protocol in reducing unnecessary transmission. Using the original DP protocol, assume node 10 is the source node that selects node 0 as a forwarding node to reach nodes 14, 1, and 16. Node 9 is not selected to rebroadcast and will terminate its broadcasting by transmitting a *marked* message upon receiving the message from node 10. When node 0 receives the message, it first sends out a *marked* message and then rebroadcasts the message by selecting nodes 1 and 16 as forwarding nodes. Just like node 9, node 14 terminates once the message from node 0 arrives. While node 1 is rebroadcasting, no forwarding node is selected since it does not have any not-yet-covered 2-hop neighbors. On the other hand, node 16 rebroadcasts by selecting node 6 as a forwarder to cover node 12. Prior to rebroadcasting, nodes 1 and 16

TABLE I
AN ILLUSTRATIVE EXAMPLE FOR DP AND EDP

Protocol	Broadcasting Nodes
DP	0, 1, 6, 10, 16
EDP	0, 1, 10, 16

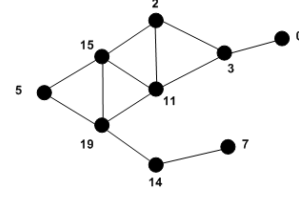


Fig. 2. An example that shows ETDP and EPDP outperform TDP and PDP respectively

TABLE II
AN ILLUSTRATIVE EXAMPLE FOR TDP, PDP, ETDP, AND EPDP

Protocol	Broadcasting Nodes
TDP	2, 3, 5, 11, 14, 15, 19
PDP	2, 3, 5, 11, 14, 15, 19
ETDP	2, 3, 5, 14, 15, 19
EPDP	2, 3, 5, 14, 15, 19

will transmit *marked* messages to indicate that their statuses have changed from *unmarked* to *marked*. Before the *marked* message from node 12 arrives at node 6, it rebroadcasts the message. Therefore, the transmission by node 6 is redundant. Node 2 terminates by transmitting a *marked* message. In EDP, node 6 is *covered* upon receiving the first message from node 1. Hence, it will not rebroadcast when receiving the message from node 16. Table I summarizes the set of nodes involved in the broadcasting as identified by DP and EDP.

Fig. 2 shows an example where EPDP and ETDP would outperform PDP and TDP respectively. Node 5 initiates the transmission by selecting nodes 15 and 19 as forwarding nodes using either PDP or TDP protocol. Nodes 15 and 19 rebroadcast by selecting node 2 and nodes 11 and 14 respectively as forwarders (after transmitting *marked* messages). Upon receiving the message, nodes 2, 11, and 14 will transmit *marked* messages. Node 2 will terminate thereafter. Node 7 terminates after receiving the message from node 14 and has transmitted a *marked* message. Here, node 11 should receive two messages. Assume the message from node 15 arrives first, followed immediately by the message from node 19. Node 11 rebroadcasts by selecting node 3 as a forwarding node to cover node 0. Finally, when node 0 receives the broadcast message from node 3, it terminates by sending a *marked* message. In EPDP and ETDP, node 11 is already *covered* when it receives the first message from node 15. Hence, it will not rebroadcast when receiving the message from node 19. Table II summarizes the set of nodes involved in the broadcasting as identified by TDP, PDP, ETDP, and EPDP.

IV. SIMULATION STUDIES AND RESULTS

In this section, simulation studies are performed to compare the performance of DP, PDP, and TDP against EDP, EPDP, and ETDP respectively in terms of the number of broadcasting nodes. Table III summarizes the general simulation settings.

The simulations are conducted in an ideal environment without channel contention, packet collisions, and nodes mobility that could change the overall network topology. In this environment, the packet size and channel bandwidth do not affect the simulation results. M number of nodes is randomly placed into an area of H m \times H m and M is increased from 20 to 100 to see the effect of different node density. The values for M and H are specified in Table III. For each value of M and H pair, 20 different topologies are simulated. Our network generator ensures that each simulated topology is connected. In each topology, we assume every node takes turn to become the source node of a broadcast session.

Fig. 3 shows the results of the number of broadcasting nodes incurred by DP, PDP, TDP, EDP, EPDP, and ETDP. In this figure, the number of broadcasting nodes increases with the number of nodes in the network for all protocols. This is because as the population increases and spread throughout the network, more nodes are needed to cover the entire network. In a sparse network of 20 nodes, the number of broadcasting nodes is more or less the same for all protocols. In such a network, the effectiveness of each protocol is not significant. The proposed enhancement shows significant improvement especially in dense networks such as 60 nodes and above. In networks of 100 nodes, the proposed EDP, EPDP, and ETDP incurred approximately 45%, 44%, and 42% saving in the number of broadcasting nodes, respectively. As illustrated in Figs. 1 and 2, the main culprit of unnecessary transmission in DP, PDP, and TDP is that a rebroadcasting node needs to rely on the status of its immediate neighbors when deciding to rebroadcast or not and the notification of a change in a neighbor's status from *unmarked* to *marked* may not be available during rebroadcasting. As a result, a node may rebroadcast a message even though all its neighbors are already "*marked*". In other words, a rebroadcasting node may decide to rebroadcast a message based on inaccurate information about the status of its immediate neighbors. On the other hand, the proposed *covered/uncovered* termination condition does not involve any transmission of control messages, thus does not rely on the status of immediate neighbors. Instead, it is purely based on the status of each individual node, whether they are *covered* or *uncovered*. As long as a node is *covered*, it does not need to rebroadcast the same message again (message that has been received before). This leads to significantly better performance gain especially in dense networks as shown in Fig. 3.

V. CONCLUSION

This paper begins with a thorough review of existing broadcasting protocols based on deterministic and their termination conditions. Existing deterministic protocols such as DP, PDP, and TDP are effective in selecting a minimum number of

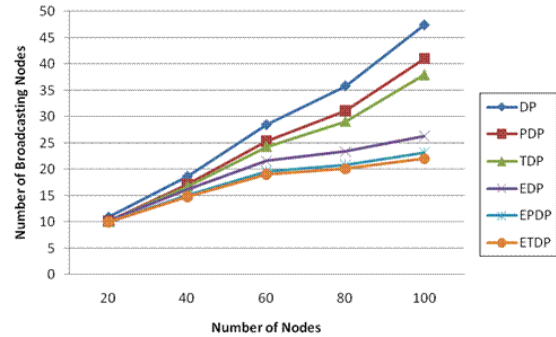


Fig. 3. Number of Broadcasting Nodes Incurred by DP, TDP, PDP, EDP, ETDP, and EPDP

TABLE III
COMMON SIMULATION PARAMETERS

Simulation Parameter	Value
Simulator	NS2.33
Network Size	1000 m \times 1000 m
Default Transmission Range	250 m
No. of Nodes	20, 40, 60, 80, and 100
Confidence Interval	95%

forwarding nodes. This is largely affected by the termination condition that further inhibits unnecessary transmission. There are two termination conditions that govern whether a node should rebroadcast or not. In the first approach, a node rebroadcasts a message if there is at least one *unmarked* neighbor or neighbor that has not received a message. The second approach prohibits a node that has *relayed* a message from transmitting.

The first termination condition involves additional control messages, thus not feasible for real-time networks. The major contribution of this paper is proposing a simple termination condition that eliminates the need for excessive overheads and is effective in reducing the number of transmitting nodes.

REFERENCES

- [1] W. Woon and K. Y. Yeung, "Self-Pruning Broadcasting for Mobile Ad Hoc Networks," *IEEE Globecom*, Hawaii, USA, Dec. 2009.
- [2] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *Proc. of MOBIHOC*, 2002.
- [3] Y. C. Tseng, S. Y. Ni, Y. S. Chen, and J. P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Wireless Networks*, vol. 8, no. 2-3, pp. 153-167, Mar. 2002.
- [4] 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 - 123, Apr-June 2002.
- [5] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying for Flooding Broadcast Message in Mobile Wireless Networks," *Proc. of HICSS-35*, pp. 3898-3907, Jan. 2002.
- [6] H. Kim and C. Kim, "Flooding in Wireless Ad Hoc Networks," *Computer Communications*, vol. 24, no. 3-4, pp. 353 - 363, Feb. 2001.
- [7] H. S. Chiu, Kwan L. Yeung, and K. S. Lui, "J-CAR: An Efficient Joint Channel Assignment and Routing Protocol for IEEE 802.11-based Multi-Channel Multi-Interface Mobile Ad Hoc Networks," *IEEE Trans. on Wireless Commun.*, vol. 8, no. 4, pp. 1706 - 1715, Apr. 2009.
- [8] I. D. Chakeres and E. M. Belding-Royer, "The Utility of Hello Messages for Determining Link Connectivity," in *Proc. of WPMC*, Hawaii, USA, Oct. 2002.