



Title	A new phase for screening redundant broadcast nodes in source-independent broadcasting protocols
Author(s)	Woon, W; Yeung, KL
Citation	The 2011 IEEE International Conference on Communications (ICC 2011), Kyoto, Japan, 5-9 June 2011. In Proceedings of the IEEE ICC, 2011, p. 1-5
Issued Date	2011
URL	http://hdl.handle.net/10722/140254
Rights	Creative Commons: Attribution 3.0 Hong Kong License

A New Phase for Screening Redundant Broadcast Nodes in Source-Independent Broadcasting Protocols

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—Following the distributed approach, source-independent broadcasting protocols select a subset of nodes in a network as broadcasting nodes to cover the entire network. The selection of broadcasting nodes is performed *prior* to actual message transmission. These broadcasting nodes collectively form a connected dominating set or CDS. Aiming at finding a minimum CDS, existing source-independent broadcasting protocols consist of two phases. In this paper, we propose to add a third phase to eliminate unnecessary nodes in a CDS while ensuring all remaining nodes are still connected. We call it the *redundant node screening* phase. This paper shows that this new phase is a very important element that has been ignored by existing source-independent broadcasting protocols. When applying the new phase on existing broadcasting protocols, the savings in terms of number of nodes in the CDS could be as high as 21% in a 1000m x 1000m network of 20 nodes.

I. INTRODUCTION

Broadcasting is a way of disseminating information to all nodes in a network. During broadcasting the source node usually 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 had 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 a congested network, prolonging the lifetime of nodes that are battery powered, and lower channel utilization.

In this paper, we focus on designing distributed broadcasting protocols, where the protocol message exchange is between immediate neighbors. Many distributed broadcasting protocols [1-6] have been proposed for Mobile Ad Hoc Networks (MANETs) based on the deterministic approach, where a broadcasting node selects a subset of its immediate (i.e. 1-hop) neighbors to *cover* all its 2-hop neighbors. (Here “cover” means to reach a node by rebroadcasting the message.) Deterministic broadcasting protocols can be categorized

into source-dependent [1-3] and source-independent [4,5]. A source-dependent broadcasting protocol constructs a dedicated broadcast tree for each broadcast source/session, as initiated by the arrival of a broadcast message in real time.

On the other hand, source-independent protocols, such as IMPR (source-independent multipoint relay) [4] and enhanced IMPR (E-IMPR) [5], determine the set of broadcasting nodes (i.e. CDS nodes) *prior* to actual message arrival. Both IMPR and E-IMPR consist of two phases. In the first phase, every node is treated as a potential source of some broadcast message. Then every node selects its 1-hop neighbors (as its forwarding nodes) to cover all its 2-hop neighbors in parallel. In the second phase, the selection results are exchanged among *direct* neighbors and based on specific selection rules [4,5], some nodes (forwarding and non-forwarding) become CDS nodes. When the protocol ends, each node knows its status of being a CDS node or not. Subsequently, when a broadcast message is received by a CDS node, the message will be rebroadcast once. Otherwise, the message will be discarded.

In general, source-independent broadcasting protocol is more suitable for a static network or in a network where node mobility does not change the overall network topology (like wireless mesh networks). In this case, establishing a CDS is a one-time effort for all broadcast traffic. This paper focuses on enhancing the two existing source-independent broadcasting protocols, IMPR and E-IMPR, by adding a third phase to refine the solution. The objective of this refinement phase is simple: identifying and eliminating redundant CDS nodes while ensuring the (remaining) CDS nodes are still connected. To this end, the need for such a refinement phase is rather obvious but interestingly, it has been ignored by both IMPR and E-IMPR. This could be due to the misbelief that their CDS solutions are already good enough, or it is difficult to find a simple yet effective algorithm to eliminate the redundant CDS nodes.

The rest of this paper is organized as follows. Section II critically reviews the two existing two-phase protocols, IMPR and E-IMPR. In Section III, our proposed third phase for redundant CDS node screening is presented, and its effectiveness

This work is supported by Hong Kong Research Grant Council General Research Fund HKU 719108E

is quantitatively studied in Section IV by simulations. Finally, Section V concludes the paper.

II. EXISTING SOURCE-INDEPENDENT BROADCASTING PROTOCOLS

To the best of our knowledge, there are two source-independent broadcasting protocols based on the deterministic approach, IMPR (source-Independent Multipoint Relay) [4] and the Enhanced IMPR (E-IMPR) [5]. Both protocols consist of two phases, *forward node selection* and *CDS node selection*. (It should be noted that the original acronyms of IMPR are MPR-CDS [4] and MPR [5] while E-IMPR is EMPR [5].)

Without loss of generality, it is assumed that each node have local knowledge of the identities (IDs) of all their neighbors within 1-hop (immediate) and 2-hop range. The set of nodes within the 1-hop and 2-hop range of a particular node u are denoted as $N(u)$ and $N(N(u))$ respectively, where $N(u) \subseteq N(N(u))$. This information can be obtained via periodic HELLO message exchanges. A HELLO message contains unique ID of the sender as well as its 1-hop neighbor list. Upon receiving the message, a node u will treat the sender (v) as an immediate neighbor (in $N(u)$) while the nodes in $N(v)$ as within its 2-hop range (in $N(N(u))$). Let $H_1(u)$ and $H_2(u)$ denote the set of 1-hop and 2-hop neighbors of node u , respectively. We have $H_1(u) = N(u)$ and $H_2(u) = N(N(u)) - N(u)$

A. IMPR Protocol

1) *Phase 1: forwarding node selection*: Firstly, a node v selects nodes in $N(v)$ that are the only nodes that can reach those nodes in $H_2(v)$ as its forwarding nodes. Note that nodes that are not-yet-covered by any forwarding nodes are referred as uncovered nodes. For any uncovered nodes in $H_2(v)$, select an immediate neighbor from $N(v)$ as a forwarding node if it covers the most number of uncovered nodes until there is no more uncovered nodes. In case of a tie, the node with smallest ID is selected. The selection algorithm is summarized below:

- 1) Select a node u in $N(v)$ as a forwarding node if it is the only node that can cover a node in $H_2(v)$. (Priority is given to such a node u because it must be selected as a forwarding node.)
- 2) Select a node u in $N(v)$ as a forwarding node if it covers the most number of uncovered nodes in $H_2(v)$.

2) *Phase 2: CDS node selection*: When each node selects its set of forwarding nodes, the selection results are exchanged among immediate neighbors by a single broadcasting message. Then based on the selection results received from all the neighbors, each node decides *independently* whether it should become a CDS node or not. A node that fulfills *either* one of the following rules becomes a CDS node:

- **Rule 1** - The node has a smaller ID than all its immediate neighbors.
- **Rule 2** - The node is a forwarding node selected by its smallest ID neighbor.

Rules 1 and **2** play a very important role in restricting the number of nodes in a CDS while ensuring that the

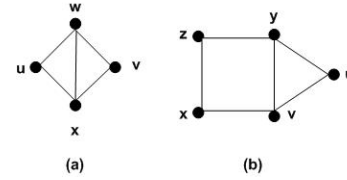


Fig. 1. Two sample networks to illustrate the IMPR and E-IMPR protocols

whole network is fully covered. The correctness proof of this approach can be found in [4].

Fig. 1(a) illustrates the two-phase operation of the IMPR protocol. In the first phase, both nodes u and v select node w as their forwarding node. From node v 's point of view, nodes w and x can cover its 2-hop neighbor, node u . Since both w and x cover the same number of uncovered nodes, node w is selected because it has smaller ID. The forwarding node selected by node v is similar to node u . Nodes w and x do not have any 2-hop neighbor. Therefore, there is no forwarding node being selected. In the second phase, the selection results are exchanged. Upon receiving the results from nodes u and v , node w is aware of being selected as a forwarding node. When nodes u , v , w , and x apply the two CDS node selection rules, nodes u and v fulfill **Rule 1** and node w fulfills **Rule 2**. In the latter case, node w is being selected as a forwarding node by node u , which is its smallest ID neighbor. Therefore, in this example, nodes u , w , and v form a CDS.

B. E-IMPR Protocol

1) *Phase 1: forwarding node selection*: E-IMPR [5] enhances the forwarding node selection algorithm of IMPR with the notion of “free neighbor”. Fig. 1(b) illustrates the “free neighbor” concept. Assume node v selects node x as its forwarding node to cover node z . When node x evaluates **Rule 2**, it will accept the decision made by node v and becomes a node in CDS. However, if node v selects node y instead, its selection will be ignored by node y because node v is not its smallest ID neighbor. Nevertheless, node v can select node y as one of its forwarding nodes because the inclusion of node y will not increase the number of nodes in CDS. That is, node y will fail **Rule 2** (and will not be in the CDS) when evaluating the selection made by node v . In this case, node y should be included in the CDS as a “free neighbor” of node v .

In E-IMPR, “free neighbors” are first selected as forwarding nodes. Then, the same selection algorithm used by IMPR is executed. The refined forwarding node selection algorithm is summarized below:

- 1) Add all “free neighbors” as forwarding nodes.
- 2) Select a node u in $N(v)$ as a forwarding node if it is the only node that can cover a node in $H_2(v)$.
- 3) Select a node u in $N(v)$ as a forwarding node if it covers the most number of uncovered nodes in $H_2(v)$.

2) *Phase 2: CDS node selection*: Similar to IMPR, upon receiving the selection results of forwarding nodes from all its neighbors, each node decides *independently* whether to

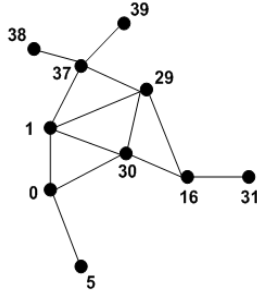


Fig. 2. An example network to show the effectiveness of "free neighbor"

become a CDS node or not based on the following rules. A node becomes a CDS node if it fulfills *either* one of the following rules:

- **Enhanced Rule 1** - The node has a smaller ID than all its immediate neighbors and has two or more unconnected neighbors.
- **Rule 2** - The node is a forwarding node selected by its smallest ID neighbor.

The correctness proof of the two selection rules above can be found in [6]. The only difference between **Enhanced Rule 1** and **Rule 1** (of IMPR) is that **Enhanced Rule 1** attempts to prohibit "leaf" nodes from becoming a CDS node. An example of "leaf" node is node u in Fig. 1(b). In this case, even though node u has smaller ID than nodes v and y , all its neighbors are connected. That is, nodes v and y are immediate neighbors of each other. Likewise, nodes u and v in Fig. 1(a) are also leaf nodes.

Recall the solution obtained using IMPR for Fig. 1(a), where the CDS consist of nodes u , w , and v . We can easily see that node w (or node x) alone is already sufficient to form a CDS. With E-IMPR, leaf nodes u and v will fail **Enhanced Rule 1** and thus the only (non-leaf) node in CDS is w . Similarly, in Fig. 1(b), nodes u , v , y , and x form a CDS according to IMPR while only nodes v , y , and x form a CDS based on E-IMPR.

Fig. 2 illustrates the effectiveness of the "free neighbor" concept. Using IMPR, nodes 0, 1, 29, 30, 16, and 37 form a CDS. Nodes 0 and 16 become CDS nodes due to **Rule 1**. Nodes 1 and 30 are selected as forwarding nodes by their smallest ID neighbor (**Rule 2**), node 0 while nodes 29 and 37 are selected by node 1. In the forwarding node selection phase of E-IMPR, node 1 treats node 30 as its "free neighbor". Therefore, node 30 is selected as a forwarding node. Then, node 1 selects node 37 as a forwarding node to cover its remaining uncovered 2-hop neighbors (nodes 38 and 39). Since node 29 is not selected (by node 1) as a forwarding node, it will not be a CDS node.

III. PHASE 3: REDUNDANT CDS NODE SCREENING

A. Background and Motivation

This section presents the proposed additional phase specifically for screening redundant nodes in a CDS of a source-

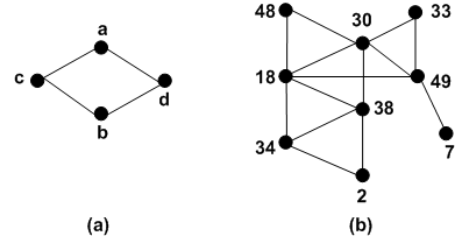


Fig. 3. Three sample networks to illustrate the proposed Phase 3

independent broadcasting protocol, let it be IMPR or E-IMPR. A CDS node is regarded as *coverage-redundant* if *all* its 1-hop neighbors are also covered by one or more other CDS nodes. If a CDS node is coverage-redundant, we proceed to check if it is also *connectivity-redundant*. A CDS node is connectivity-redundant if its removal will not cause network partitioning among the remaining CDS nodes. If a CDS node is both coverage- and connectivity-redundant, it can be removed from the CDS.

The general principle above is simple, but the checking for redundancy is non-trivial because each node only has local neighborhood knowledge, i.e. neighbors within its 1-hop and 2-hop range. Before we present our design, recall that at the end of **Phase 2**, each node only knows its own CDS status. Therefore, the first step in **Phase 3** is that every CDS node broadcasts a CDS notification message to let all its neighbors know its CDS status. Then based on the CDS notification messages received (from all neighboring CDS nodes), a CDS node (u) can determine if itself is coverage-redundant. Assume node u has k CDS neighbors, and denote the corresponding set by \mathbf{K} . Node u is coverage-redundant if $N(u) \subseteq \bigcup_{k \in \mathbf{K}} N(k)$.

Note that such a checking is not *comprehensive* because some 1-hop neighbors of a CDS node can be covered by a CDS node that is 2-hop away. Consider the example in Fig. 3(a), where non-CDS nodes c and d are covered by their CDS neighbors a and b . Since nodes a and b are 2-hops away from each other, they do not know each other's CDS status because the CDS notification message is confined to 1-hop. In this case, nodes a and b will (incorrectly) consider themselves as being not coverage-redundant. Although extending the CDS notification message beyond 1-hop helps, the associated complexity is high, thus it is not pursued in this paper.

If a CDS node is deemed coverage-redundant, we check if it is also connectivity-redundant. That is, if it is removed, will *all* its CDS neighbors still connected? It can be seen that node u is connectivity-redundant if

- ◊ There exists $v \in \mathbf{K}$ such that $N(u) \subseteq N(v)$ OR
- ◊ For *every* pair of $(v, w) \in \mathbf{K}$, we have $v \in N(w)$ (or $w \in N(v)$).

The first condition also implies that node u is coverage-redundant. Admittedly, the requirement of the second condition is stringent. It needs *every* pair of CDS neighbors to be pairwise connected. In fact, this condition can be relaxed (thus enhanced) without requiring additional neighborhood

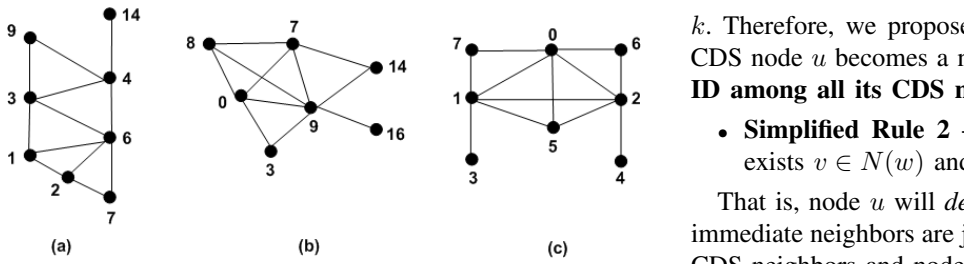


Fig. 4. More sample networks to illustrate the proposed Phase 3

information (i.e. still confined to 2-hop). We leave that for future consideration.

Nevertheless, the above connectivity-checking has a flaw. Consider Fig. 3(b). Nodes 18, 30, 38, and 49 are some of the nodes that form a CDS using either IMPR or E-IMPR. Using the above conditions, node 30 can *de-select* itself from the CDS because its CDS neighbors, nodes 18 and 49 can cover all its 1-hop neighbors. Next, node 18 can *de-select* itself from the CDS because it knows that CDS nodes 30 and 38 can jointly cover all its 1-hop neighbors. The removal of nodes 18 and 30 from the CDS will cause node 48 to be uncovered. To avoid this “chain” effect in departure, an additional constraint is added such that only the node with the *smallest* ID among all its CDS neighbors is allowed to leave.

Next, we combine the checking for both *coverage* and *connectivity* into a single set of rules. A CDS node u becomes a non-CDS node if **node u has the smallest ID** and

- **Rule 1** - There exists $v \in \mathbf{K}$ such that $N(u) \subseteq N(v)$ OR
- **Rule 2** - For every pair of $(v, w) \in \mathbf{K}$, there exists $v \in N(w)$ and $N(u) \subseteq \bigcup_{k \in \mathbf{K}} N(k)$.

Note that the two rules above should be carried out sequentially. Upon executing **Rule 1**, all remaining CDS nodes are required to broadcast a CDS notification message to update their immediate neighbors about their current CDS status before executing **Rule 2**. This message overhead is acceptable because only nodes (still) in the CDS will broadcast the message. More importantly, this message overhead is necessary to ensure more redundant CDS nodes can be eliminated by applying **Rule 2**. Fig. 4(a) shows an example. When IMPR and E-IMPR incorporate **Rules 1** and **2** (of **Phase 3**), nodes 3, 4, and 6 form a CDS if *no* CDS notification message is exchanged upon executing **Rule 1**. However, with CDS notification message exchanges, node 3 can be eliminated from the CDS. That is, when node 1 removes itself using **Rule 1**, it broadcasts a CDS notification message so that node 3 can *de-select* itself when executing **Rule 2**. Without the participation of node 1 in the CDS, node 3 will now have the smallest ID among all its CDS neighbors (nodes 4 and 6).

B. Implementation

Notice that there are ${}^k C_2$ pairs of CDS neighbors to be checked when implementing **Rule 2**, where k is the number of CDS neighbors a node has. Besides, the success rate of all ${}^k C_2$ pairs that are pairwise connected decreases quickly with

k . Therefore, we propose a simplified version of **Rule 2**. A CDS node u becomes a non-CDS node if **it has the smallest ID among all its CDS neighbors** and

- **Simplified Rule 2** - For any pair of $(v, w) \in \mathbf{K}$, there exists $v \in N(w)$ and $N(u) \subseteq (N(v) \cup N(w))$.

That is, node u will *de-select* itself from the CDS if all its immediate neighbors are jointly covered by *any two connected* CDS neighbors and node u 's ID is the smallest among all its CDS neighbors.

C. Examples

Phase 3 is effective in eliminating unnecessary nodes in the CDS as illustrated in the following examples. The examples also show that this phase is necessary for a source-independent broadcasting protocol to eliminate unnecessary CDS nodes. To the best of our knowledge, this mechanism is unavailable in existing protocols [5,6]. Nevertheless, we do not discard the possibility that there are other more sophisticated redundant CDS node screening mechanisms. A nice feature of **Phase 3** is its flexibility of having additional rules to screen out more redundant CDS nodes. One simple extension is to *de-select* a CDS node if *three connected* CDS neighbors jointly cover *all* its neighbors and its ID is the smallest among all its CDS neighbors. On the flip-side, such extension would increase the complexity of a source-independent broadcasting protocol and intuitively, the gain from it may not be very significant.

If **Rule 1** of **Phase 3** is used, the corresponding versions of IMPR and E-IMPR are called IMPR(1) and E-IMPR(1). On the other hand, IMPR(2) and E-IMPR(2) will incorporate **Rule 1** and **Simplified Rule 2** of **Phase 3**. In the previous section, nodes u , w , and v in Fig. 1(a) form a CDS based on IMPR. Using the proposed **Phase 3**, since all neighbors of u and v are covered by node w , nodes u and v will *de-select* themselves from the CDS. Therefore, only node w remains in the CDS. This CDS set is the same as the one formed using E-IMPR. In Fig. 1(b), nodes u , v , y , and x form a CDS according to IMPR. By applying **Phase 3**, node u will *de-select* itself from the CDS.

Fig. 4(b) shows an instance where IMPR(1) and E-IMPR(1) would reduce the size of the CDS formed using IMPR and E-IMPR. In this case, nodes 0 and 9 form a CDS using either IMPR or E-IMPR. Node 0 is selected as a CDS node because it has the smallest ID among all its neighbors (**Rule 1**) and it has at least two unconnected neighbors (**Enhanced Rule 1**). On the other hand, node 9 is a CDS node because it is being selected as a forwarding node by node 0 to cover nodes 14 and 16. Since the selection comes from a node with the smallest ID (**Rule 2** of **Phase 2**), node 9 must accept the selection made by node 0. By applying the proposed **Phase 3**, node 0 *de-selects* itself from the CDS because all its immediate neighbors (nodes 3, 7, 8, and 9) are covered by node 9.

Fig. 4(c) shows the effectiveness of **Simplified Rule 2** in eliminating redundant CDS nodes. In this sample network, nodes 0, 1, and 2 form a CDS using IMPR, E-IMPR, IMPR(1), or E-IMPR(1). IMPR(2) and E-IMPR(2) eliminate node 0 from

the CDS because nodes 1 and 2 jointly cover all its neighbors and it has the smallest ID among all its CDS neighbors.

IV. SIMULATION RESULTS

In this section, simulation studies based on NS2.34 [7] are performed to evaluate and compare the performance of IMPR, E-IMPR, IMPR(1), IMPR(2), E-IMPR(1), and E-IMPR(2) in terms of number of nodes in CDS. The simulations are conducted in an ideal environment without channel contention, packet collision, and node mobility. In this environment, the packet size and channel bandwidth do not affect the simulation results. M nodes is randomly placed into an area of $Hm \times Hm$ and M is increased from 20 to 200 to see the effect of different node density. Each node has a fixed transmission range of 250m. The value of H is 1000. For each value of M and H pair, 10 different topologies are simulated. The network generator used ensures that each simulated topology is connected. In each topology, it is assumed that every node takes turn to become the source node of a broadcast session.

Fig. 5 shows the number of nodes in CDS increases as the population increases for all protocols. This is because more CDS nodes are needed to cover all nodes as the number of nodes in the network increases and spreads throughout the network. When comparing E-IMPR and IMPR, the former clearly performs better due to the effectiveness of **Enhanced Rule 1** and “free neighbor” concept in selecting smaller number of CDS nodes. In networks of 20 to 80 nodes, the E-IMPR protocol outperforms IMPR only marginally. However, in networks of 100 nodes and above, the margin of gain becomes more prominent. The same is observed when comparing E-IMPR(1) with IMPR(1) and E-IMPR(2) with IMPR(2).

Fig. 5 confirms the effectiveness of the proposed protocols, IMPR(1), IMPR(2), E-IMPR(1), and E-IMPR(2), in eliminating redundant CDS nodes. IMPR(1) and IMPR(2) outperformed IMPR while E-IMPR(1) and E-IMPR(2) performed better than E-IMPR. This is a clear indication that the two rules in **Phase 3** are able to eliminate “leaf” nodes and unnecessary nodes from becoming CDS nodes as shown in Figs. 4(b) and 4(c). The results also show that IMPR(2) outperforms E-IMPR. However, IMPR(1) could not perform better than E-IMPR in a network of 200 nodes due to the effectiveness of the “free neighbor” concept. The effectiveness of “free neighbor” is also observed in all networks except for 140, 160, and 200 nodes where E-IMPR(1) outperforms IMPR(2). Nevertheless, the ability of IMPR(2) to outperform E-IMPR in all scenarios show that the combination of **Rule 1** and **Simplified Rule 2** of **Phase 3** can exceed the effectiveness of “free neighbor”. The performance gained achieved by E-IMPR(2) compared with E-IMPR and IMPR(2) against IMPR are approximately 14% and 21% respectively in a network of 20 nodes. Finally, it is not surprising to see E-IMPR(2) emerges as the most effective protocol. This is because it incorporates **Enhanced Rule 1**, **Rule 2**, and the two rules specified in **Phase 3** that are effective in screening out redundant CDS nodes.

As specified in the previous section, **Phase 3** has the flexibility of incorporating additional rules to screen out more

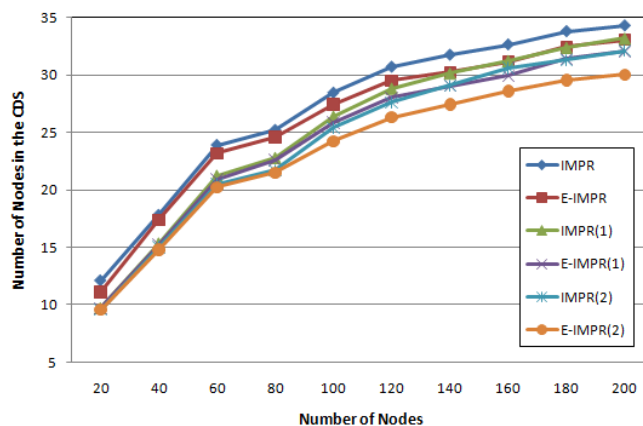


Fig. 5. Number of CDS Nodes Incurred by IMPR, E-IMPR, IMPR(1), E-IMPR(1), IMPR(2), and E-IMPR(2)

redundant CDS nodes. However, this would increase the complexity of a broadcasting protocol. Moreover, the performance gain will not be very significant as indicated in Fig. 5 when comparing IMPR(2) and E-IMPR(2) with IMPR(1) and E-IMPR(1) respectively. Even though the gain from our **Phase 3** is not very significant in some scenarios, our extended work, i.e. simulations on larger networks, show that our enhancement is indeed very effective in screening out redundant CDS nodes.

V. CONCLUSION

Existing source-independent broadcasting protocols such as IMPR and E-IMPR have specific rules that are effective in selecting a minimum number of nodes to form a CDS. However, the CDS may contain redundant nodes. Since IMPR and E-IMPR do not incorporate any mechanisms to remove redundant CDS nodes, this paper proposes an additional phase of operation to screen out redundant CDS nodes while ensuring the remaining CDS nodes are still connected. Simulation results show that the saving in the number of CDS nodes can be as significant as 21% for a network of 20 nodes.

REFERENCES

- [1] 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.
- [2] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying for Flooding Broadcast Message in Mobile Wireless Networks," in *Proc. of HICSS-35*, pp. 3898-3907, Jan. 2002.
- [3] H. Kim and C. Kim, "Flooding in Wireless Ad Hoc Networks," *Computer Communications*, vol. 24, no. 3-4, pp. 353 - 363, Feb. 2001.
- [4] C. Adjih, P. Jacquet, and L. Viennot, "Computing Connected Dominated Sets with Multipoint Relays," *Ad Hoc & Sensor Networks*, vol. 1, pp. 27-39, May 2005.
- [5] J. Wu, W. Lou, and F. Dai, "Extended Multipoint Relays to Determine Connected Dominating Sets in MANETs," *IEEE Trans. on Computers*, vol. 55, no. 3, Mar. 2006.
- [6] W. Woon and K. L. Yeung, "Enhanced Termination Condition for Deterministic Broadcasting Protocols in Mobile Ad Hoc Networks," in *Proc. of IEEE VTC*, Ottawa, Canada, Sept. 2010.
- [7] K. Fall and K. Varadhan, "The ns manual," The VINT Project, <http://www.isi.edu/nsnam/ns/doc/>, Nov. 2008.