2010 24th IEEE International Conference on Advanced Information Networking and Applications

Iterative Route Discovery in AODV

Nashid Shahriar, Syed Ashker Ibne Mujib, Arup Raton Roy and Ashikur Rahman
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
Email: {nshahriar, ashker.mujib, arup.roy}@csebuet.org, ashikur@cse.buet.ac.bd

***Abstract*—Several protocols for ad hoc network try to reduce redundancy as an effective measure against broadcast problems. Though these protocols ensure good performance in a favorable environment, they perform poorly when node cooperation cannot be guaranteed due to intentional misbehavior or environmental hostility. As a result, the expected behavior of nodes to forward packets, which is the basic assumption of all broadcast approaches, cannot be achieved always. In this paper, we analyze the performance deterioration of these algorithms in hostile environment. As a remedy, we focus on the reverse direction and interestingly find that adding redundancy in a controlled manner can greatly compensate the performance loss due to node misbehavior. Here we propose a novel approach that tune the amount of redundancy so that reachability and routing load both remain at a satisfactory level. Comparing their relative performance we end up with the conclusion that though redundancy is undesired, controlled redundancy is effective in special situations like uncooperative environments.**

***Keywords:* Ad hoc Networks, Untrusted Environment, Cooperative Hosts, Controlled Redundancy, Dominant Pruning.**

I. Introduction

An *ad hoc wireless network* is a collection of wireless hosts forming a temporary network devoid of any centralized administration or supporting stationary infrastructure such as base stations, where hosts may communicate with one another. In such networks, each node operates not only as a host but also as a router by finding routes and forwarding data packets for other nodes. Ad hoc networks where hosts can move freely at will or at random, are called mobile ad hoc networks (MANET) [19]. Ad hoc wireless networks have a wide variety of applications ranging from military in battlefields, emergency disaster and rescue areas, to networks for interactive conferences. Moreover, such networks have gained mass interest recently due to the common availability of wireless cards, low cost laptops and palmtops with radio interfaces.

A major challenge in the design of ad hoc network is the development of dynamic routing protocols that can efficiently find routes between two communicating hosts. There are two basic data exchange modes-*unicasting* and *broadcasting*. Issues related to routing are reduction of routing load, radio power limitation, proper channel utilization, performance deterioration due to low bandwidth of wireless links, security concerns etc. Optimum solutions for these problems exist in a variety of approaches. But majority of these approaches rely on the assumption that they are operating on cooperative environment. That is, they trust each node by assuming that a node will obviously forward a packet when requested to do so. In reality, it is difficult to expect and maintain a favorable environment for an ad hoc network, as such networks are created on the fly to circumstance some sort of unexpected situation. Very few protocols [21], [23] consider the problems associated with an untrusted and hostile environment where a node might misbehave, thereby violating the assumption of mutual cooperation. In such environments there may be nodes which are *malicious, selfish* [13] or even intentionally *uncooperative* and *harmful* [18] or unreachable due to mobility.

Due to the host mobility and dynamic change of network topology in mobile ad hoc wireless networks, broadcast routing are performed more frequently and expected to be more efficient. Several routing protocols such as *Ad Hoc On-Demand Distance Vector routing* (AODV) [16], *Dynamic Source Routing* (DSR) [9] rely on broadcast to obtain routing information. Moreover, broadcasting is a common and fundamental operation in many applications e.g. graph related problem, distributed computing, multicast service in wired networks. One straightforward and obvious approach for broadcasting is *blind flooding*, in which each node will rebroadcast the packet whenever it receives it for the first time. Blind flooding generates many redundant transmissions and thus increases the routing load on the network. Uncontrolled flooding leads to a more serious *broadcast storm* [14] problem which is caused by serious redundancy, contention and collision in the network.

Therefore it is always the rational tendency of broadcast algorithm designers to cut down the redundancy by proposing efficient flooding algorithms [2], [4], [11], [14], [15], [17], [22]. The *Dominant Pruning* (DP) [11] is one of the promising approaches that utilizes neighborhood information to reduce redundant transmission. Though, DP is considered as the extreme counterpart of blind flooding, further improvement is possible which utilizes neighbor information more effectively. The *Total Dominant Pruning* (TDP) [12] and *Partial Dominant Pruning* (PDP) [12] are two such approaches that deal with the deficiency of DP and result to even more controlled broadcast.

Although eliminating redundant transmission is obvious in friendly, cooperative environment but may not be effective in untrusted, hostile environment. The reason is, controlled broadcasts rely heavily on some nodes of the connected dominating set [10] by trusting each node equally. If one such node somehow misbehaves, that may create a partition in the network and thus may deny to achieve the goal of the operation. As a remedy, another variant- *Multicover Dominant Pruning* (MDP), that relaxes the redundancy control of DP to compensate the performance loss caused by misbehaving

1550-445X/10 $26.00 © 2010 IEEE
DOI 10.1109/AINA.2010.128

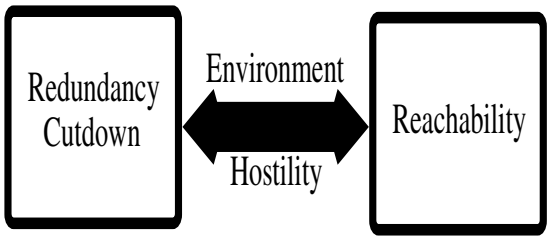


Fig. 1. Trade-off between redundancy and reachability in untrusted environment

nodes is proposed in [18]. One of the contributions of this paper is to analyze the performance of DP and its variants (i.e. PDP, TDP and MDP) in untrusted scenarios. For illustration, the broadcast component of a popular routing protocol *AODV* is modified to incorporate these broadcasting techniques.

In this paper, we also investigate a basic but important issue of broadcasting – the trade-off between redundancy and reachability aspects as illustrated in Figure 1. Through simulation we show that under the assumption of cooperative environment, redundancy degrades network performance. However, in a hostile environment where vicious nodes exists, redundancy cut-down causes a significant loss in the global reachability. Our analysis shows, adding redundancy in a controlled way in such situations upgrades the performance (i.e. increases reachability). These behavior urges us either to compromise between these two aspects or to go for a novel technique that integrates the advantage of both. Considering this trade-off, we propose an adaptive approach *Iterative Dominant Pruning* (IDP) that optimizes the above aspects irrespective of the environment. This scheme adapts with the situation by increasing the number of broadcasts only when needed, keeping it to a minimum value in friendly situation. IDP differs from the already established approaches applicable to MANET in the respect that the previous versions focus mainly on one direction of the broadcast problems that is either to attain high reachability or to cut down redundancy whereas IDP targets to achieve both goal optimally. We do not however analyze any killer application of IDP in this paper. Rather our vision is merely to illustrate a light-weight simplistic technique in hostile environment.

II. Efficient Broadcast approaches

Most of the previous works addressing node misbehavior has been focused on unicast [3], [6], [8], [9], [16] or multicast routing protocols [5]. In broadcasting, very few research aimed at this area [7], [20]. As our work focuses on broadcasting, to implement our approach we choose one of the mature algorithms for ad hoc network routing, AODV which exploits broadcasting. In AODV, when a node attempts to send a data packet to a destination, it uses *route discovery* process to find such a route. The discovery process starts by initiating a route request (RREQ) which is flooded blindly over the network. Each node upon receiving RREQ, rebroadcasts it, unless it is the destination or it has a route to the destination in its cache. The destination itself or any node in the path that contains the route then sends a route reply (RREP) to establish the route.

The blind flooding used in AODV gives rise to several problems which were mentioned previously. Some of the approaches against blind broadcasting are probabilistic [14] in nature, so they cannot guarantee all the nodes in the network receiving the broadcast packet. Another approach DP, gives this guarantee of reaching all the nodes while cutting down the number of broadcast transmission to a great extent. To achieve this, each node finds a subset of its one-hop neighbors which is called forward list. In the next hop, only the nodes in the forward list rebroadcast the packet to the two-hop nodes. Even more reduced broadcast techniques are PDP and TDP which utilize neighborhood information more effectively. PDP drops out the one-hop neighbors of common neighbor of both sender and receiver of a broadcast packet from the list of nodes to be covered. Similarly TDP drops the two-hop neighbors of the sender from that list. This requires extra three-hop neighbor information piggybacked in broadcast packet of TDP, which increases overhead. DP, TDP, PDP compute the forward node list in such a way that all two-hop neighbors are covered by the rebroadcast of at least one direct neighbor node.

Though DP and its variants perform well in normal case, they suffer in the untrusted situations. In these environments redundant broadcasts like MDP can be an effective solution. The idea behind MDP is to introduce redundancy in broadcasting to increase reachability without detecting or specifically identifying which nodes are misbehaving. To illustrate the idea behind MDP, let us assume that node v has just received a broadcast packet from node u and v is on u's forward list (F_u). Now node v has to compute its own forward list (F_v) to be inserted into the header of the rebroadcast copy.

The general *Multicover Dominant Pruning* presented in Algorithm 1 reformulates the approach of DP when computing forward list by ensuring that all two-hop neighbors ($N(N(v))$) are covered by at least m direct neighbor nodes ($N(v)$). This is done by iteratively selecting a node from the set $B(u,v) = N(v) - N(u)$ in such a way so that maximum number of nodes in the set U_v are covered. Here, $B(u,v)$ represents those neighbors of v which are possible candidates for the inclusion in F_v, and U_v denotes the set of uncovered two-hop neighbors of v. The element $mcounter(x)$ keeps track of how many times a node x is covered and is incremented after each time x is covered. The set of two-hop neighbors covered m times is denoted by Z and is initialized as a $NULL$ set. This algorithm terminates whenever Z equals U_v i.e. when all nodes in U_v are m covered or no further improvement is possible to make.

While DP, TDP, PDP which can be expressed as special case of MDP with $m = 1$, ensures single cover, MDP-2 maintains double cover, MDP-3 maintains triple cover for each two-hop neighbors. MDP-infinity maintains highest possible cover and is defined as $m = large\ number$ but terminates when no improvements can be achieved. The Algorithm 1 of MDP is presented in such a way that it can

Algorithm 1 MDP(m)

1: $F_v \leftarrow \phi$, $Z \leftarrow \phi$.
2: if $m = 0$ then
3: $\quad U_v \leftarrow N(N(v)) - N(u) - N(v) - N(N(u) \cap N(v))$
4: else
5: $\quad U_v \leftarrow N(N(v)) - N(u) - N(v)$
6: For each node $\omega \in U_v$ do
7: $\quad mcounter(\omega) \leftarrow 0$
8: For each node $\omega_i \in B(u, v)$ do
9: $\quad S_i \leftarrow N(\omega_i) \cap U_v$
10: Let $K = \{S_1, S_2, ..., S_n\}$.
11: Suppose S_k is the set such that $|S_k| = \max_{S_i \in K}\{|S_i|\}$
12: If $|S_k| = \phi$ then return F_v.
13: $F_v \leftarrow F_v \cup \{\omega_k\}$
14: For each node $x \in S_k$ do
15: $\quad mcounter(x) \leftarrow mcounter(x) + 1$
16: $\quad$ If $mcounter(x) = m$ then
17: $\quad\quad Z \leftarrow Z \cup \{x\}$
18: $\quad\quad$ For each $S_i \in K$ do
19: $\quad\quad\quad S_i \leftarrow S_i - \{x\}$
20: $K \leftarrow K - \{S_k\}$
21: If $Z = U_v$ then return F_v.
22: Otherwise go to step 11.

handle all the above cases except TDP because of its extra overhead requirement. The decision of which variation of MDP to use for broadcast depends on the value passed from outside of it through the parameter m. The special case of $m = 0$ computes U_v as required by PDP whereas other values of m designate the cover of MDP in the usual sense.

III. Reachability versus Redundancy aspects

While designing a broadcast protocol for ad-hoc networks, the primary goal is to ensure that all the desired nodes within the network receives the message which is measured as reachability. Another important goal, is to reduce the number of retransmissions, specially redundant retransmissions while reaching all the nodes in the network. The significant goal of reachability is not achieved by the above broadcast approaches in the untrusted environments due to lacking of effort and concentration imposed on this type of behavior. But such unwanted situations come into existence in most of the ad hoc networks. With the rapid advancement of ad hoc networks and wide variety of its usage, it is the high time to give more emphasis on the analysis of broadcast algorithms from this perspective.

All the aforementioned broadcasting approaches have been proved as complete and reliable. But they show expected performance only in the trusted cooperative environment. In that case, the reachability of the above techniques is close to 100%. Their order of redundancy with cooperative hosts is-

TDP <= PDP <= DP <= MDP-2 <= MDP-3 <= ... <= MDP-infinity

But in untrusted situation, the more rigid the protocol is, the more it suffers in case of reachability; because least redundant protocols have less option to cover all the nodes and thus fail to compensate the misbehavior of nodes. In this case, the performance loss in reachability is ordered as-

TDP >= PDP >= DP >= MDP-2 >= MDP-3 >= ... >= MDP-infinity

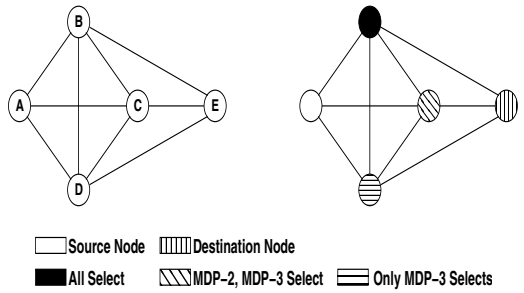


Fig. 2. Scenario differentiating the approaches of PDP, DP, MDP-2, MDP-3

This behavior can be explained by a sample scenario shown in Figure 2. Here node A tries to send a packet to node E. In blind flooding, all the intermediate nodes rebroadcast the packet which is redundant to reach to E. In DP, TDP, PDP the source selects only one of B, C, D in the forward list, MDP-2 selects two of them. Suppose DP, TDP, PDP choose B to rebroadcast. If node B drops the packet unconditionally, all of DP, TDP, PDP will never be able to succeed in reaching E. MDP-2 select both node B and C to rebroadcast and it will be successful through C even if B misbehaves. Now, suppose both node B and C misbehaves and so MDP-2 will also fail. To succeed in this case, MDP-3 which selects all of B, C, D, will be the appropriate choice. Thus MDP with $m >= 2$ decreases the probability of failure without having any information of the cause of failure.

With cooperative hosts there is no reason to argue for the effectiveness of the DP, PDP or TDP, as they are suitable from both reachability and redundancy perspective. But, in untrusted situation, variants of MDP are preferable as they show higher reachability than the variants of DP and also limits number of retransmissions caused by the multiple attempts compensating the failure to reach a node. The reason for better performance of MDP with $m >= 2$ is that, it introduces multiple paths for each two-hop neighbors so that if one node in the forward list misbehaves, others can discover the path which could not be obtained by single cover of DP, PDP or TDP. Our experimental results indicate that MDP with higher value of m increases reachability at the cost of increase in number of broadcasting nodes and routing overhead. Also for part of the topology with no misbehaving nodes, this increase puts a burden of unnecessary broadcast packets and extra CPU cycles to calculate multiple cover. So the implementation of MDP needs to control the redundancy by tuning the value of m either in application or routing layer. In our proposed *Iterative Dominant Pruning* (IDP), this tuning is performed in the routing layer where a node chooses the value of m intelligently at a particular time considering the number of failures in recent past.

Established approaches broadcast a packet only once, which

is the best option considering load on the network. Even if an approach makes multiple attempts, it retries with the same forward list. In hostile situation while broadcasting with a particular variant of MDP, we cannot be sure about successful reception after trying in this manner. So in our proposed algorithm IDP, we incorporate iteration by adding flexibility to the control of redundancy in subsequent attempts of broadcasting before announcing failure. After an attempt the source waits for the estimated amount of time to be sure about successful transmission and then upon detecting a failure it regenerates broadcast packet into network with more flexible version of MDP. IDP uses these subsequent attempts by computing forward list with least possible size in the first try; then expanding the forward list in following attempts. If there is no misbehaving node in the path and assuming no other loss, the first try is successful and we are done with the least possible redundancy. Otherwise, IDP chooses broadcasting techniques with increased redundancy iteratively and stops when succeeds in finding a path. So this scheme intelligently incorporates appropriate amount of redundancy at the right time.

Algorithm 2 ITERATIVE DOMINANT PRUNING

1: $forward\_list \leftarrow \phi$, $iteration \leftarrow 0$
2: $discovered \leftarrow false$, $\Delta \leftarrow NODE\_DEGREE$
3: while $discovered = false$ do
4: If $iteration = \Delta$ *or* $iteration > THRESHOLD$ then
5: $forward\_list \leftarrow MDP(infinity)$
6: Broadcast $RouteRequest$ with $forward\_list$
7: exit
8: $forward\_list \leftarrow MDP(iteration)$
9: Broadcast $RouteRequest$ with created $forward\_list$
10: Wait for $WAIT\_INTERVAL$
11: If $RouteReply$ is received while waiting then
12: $discovered \leftarrow true$
13: $iteration \leftarrow iteration + 1$

As a demonstration of our proposed idea, we present IDP in Algorithm 2 with MDP of Algorithm 1 as its subroutine for the case of route discovery by broadcasting. The basic strategy of IDP is obviously broadcasting with the most efficient version, but an adaptive incorporation of redundancy is done iteratively. Given up to two-hop neighborhood information, PDP incurs least possible redundancy while ensuring complete cover. Therefore, first attempt of broadcast should always be the most optimized one (i.e. PDP). Calling MDP with $m = 0$ from IDP computes forward list for PDP as a special case of single cover as shown in Algorithm 1. Here, with an objective to minimize the number of broadcast nodes, PDP subtracts the set of neighbors of each node in $N(N(u) \cap N(v))$ from the set of nodes to be covered as those nodes are assumed to be covered when computing forward list for source u. If first attempt fails to discover the path, second attempt involves a less conservative approach like DP which is achieved by a call to MDP with $m = 1$ and which drops out the restriction imposed by PDP. For subsequent attempts, in case of failure of the previous attempt, we need to deliberately augment redundancy for discovering a hidden trusted path, which MDP with $m = 2$ and $m = 3$ might do (being optimistic). This deliberate introduction of redundancy is incorporated in Algorithm 1 of MDP with increased $mcounter$ value that ensures greater cover. The iteration continues until number of attempts reach to $NODE\_DEGREE$, because greater cover than number of one-hop neighbors is not possible for a node. IDP should limit its number of attempts to a predefined $THRESHOLD$, as there may be some unreachable isolated hosts. In both cases, IDP terminates with MDP-infinity, because it is the best effort that can be employed.

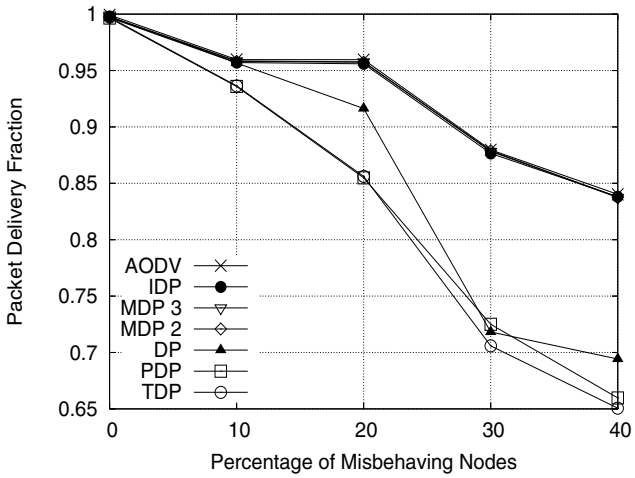
In IDP, we propose a simplistic but effective way to utilize the best features of both the controlled broadcast techniques and redundancy oriented methods, merged in a single approach assuming a MANET consisting of both cooperative and hostile hosts. While doing so, IDP does not try to detect the misbehaving nodes. This scheme is totally different from the security oriented approaches which first classify the untrusted nodes and then try to improve by skipping them. The motivation behind our idea is that in MANET it is not wise to rate the nodes based on some static criteria due to high mobility of nodes. Also to get the persistent knowledge of the node behavior in MANET either global control is needed which is not feasible or continuous monitoring is required which puts extra burden on the network and hosts. Avoiding these complexity, IDP presents an alternate way of efficient broadcast where controlled redundancy is exploited as the protective measure against misbehaving nodes. The decision of incorporating redundancy is distributed to each nodes in the network; the node surrounded by more vicious nodes adaptively employs more redundancy in broadcast. In IDP, it might be the case that some nodes selected for possible forwarding in the first iteration are untrusted and results in transmission failure. IDP does not concentrate on finding the suspects. Rather in the next iteration it just adds additional relay nodes to each two hop neighbors by ensuring higher cover, thus becoming less dependent on the previously selected nodes some of which are highly probable of being untrusted. In the worst case, the successive iterations may select nodes which all are misbehaving but on the average the heuristic nature of IDP's iteration performs well in spite of having some misbehaving nodes in its selection set. So in comparison with misbehaving node detection techniques, IDP attains the same high reachability but avoids the complexity and burden of detection methods.

IV. EXPERIMENTAL RESULTS

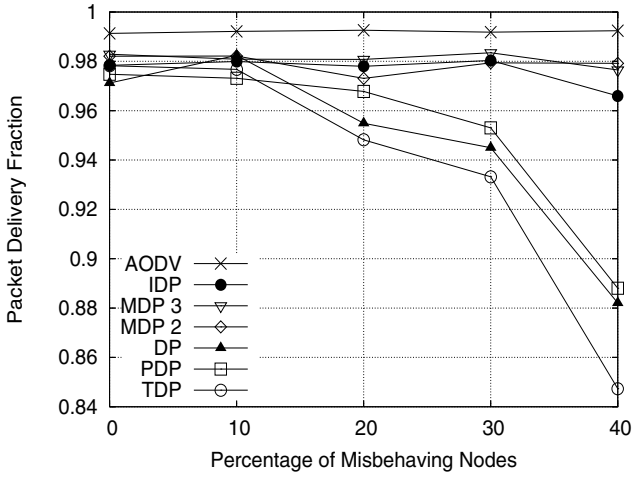
To evaluate the performance of dominant pruning and its variants, we build a detailed simulation model based on NS-2 [1] with wireless extensions. NS-2.31 is used for this purpose.

A. Scenario Generation

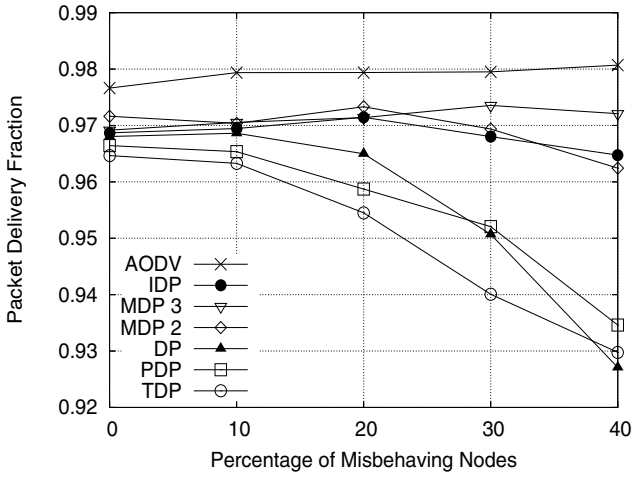
As standard scenario we use a 670m by 670m flat two-dimensional space with 50 nodes. The transmission range of



(a) Static Environment, Pause Time: 500



(b) Average Mobility, Pause Time: 250



(c) Highest Mobility, Pause Time: 0

Fig. 3. Packet Delivery Fraction with Varying Percentage of Misbehaving Node

each node is 250m which is roughly $\frac{2}{5}$ of each dimension. Pause times of the nodes are varied from 0 (maximum mobility) to 500 (static scenario) by step size 50. For each such case, 5 random scenarios are generated. Now, a certain percentage of misbehaving nodes is introduced during simulation. These nodes misbehave by dropping packets without rebroadcasting them. The number of misbehaving nodes varies between 0 and 20 with step size 5, which means that the percentage of misbehaving node is from 0% to 40%. As we increase the number of misbehaving nodes, the larger set includes the misbehaving nodes from the previous simulation. This ensures consistency between two scenarios. Similar approaches are mentioned in [13][18].

B. Traffic Generation

Traffic sources are CBR (constant bit rate) packets. Total 10 connections are used. Data sources generate unicast packet at regular 1 second intervals. Traffic generators on different sources start at time uniformly distributed between 0 and 50 seconds. The packet size is fixed at 512 bytes. Each simulation runs for 500 seconds of virtual time.

C. Performance Metrics

To see the effect of node misbehavior on variants of dominant pruning and on our proposed enhancements, we consider the following performance metrics-

- *Packet Delivery Fraction (pdf)*: is the ratio of number of successfully received CBR packets to number of sent CBR packets. If the route discovery of AODV explores the hidden trusted path from source to destination bypassing the misbehaving nodes efficiently, then obviously successful delivery of the data packet will increase. Therefore, the *pdf* of CBR packet is an useful measure to evaluate the reachability i.e. performance of the broadcast.
- *Routing Overhead*: is the total number of packets needed to exchange routing information among nodes.
- *Normalized Routing Load*: is the fraction of routing packets needed to successfully deliver one data packet.
- *Normalized Efficiency (N.E.)*: To scale the reachability measure in respect of redundancy, we propose this new metric, *N.E. pdf* is an indication of reachability; higher *pdf* means higher reachability. On the other hand, *Routing Overhead* is a measure of redundancy which is mainly composed of routing packets generated by network wide broadcasts. To increase reachability we may want to add redundant path that also increases the number of overhead packets. To rate a protocol we must consider both the issues. Therefore *Routing Overhead* contributing to high reachability should be penalized. Considering this notion *N.E* is defined as a function of *pdf* and *Routing Overhead* defined by the following equation:

$$N.E. = \frac{pdf}{pdf_{aodv}} - c * \frac{routingOverhead}{routingOverhead_{aodv}}$$

We incorporate DP and its variants in route discovery process of AODV, which is essentially blind flooding.

Thus, it is logical to assess the behavior of an approach with respect to pure AODV. Here, c is a measure of how routing overhead should be penalized comparing to successful delivery of a data packet. Different value of c can be chosen based on the proportion of overhead packet size with respect to data packet size. As the size of overhead packet is roughly one tenth of data packet size, 0.1 should be a reasonable assignment for c.

D. Performance Comparison

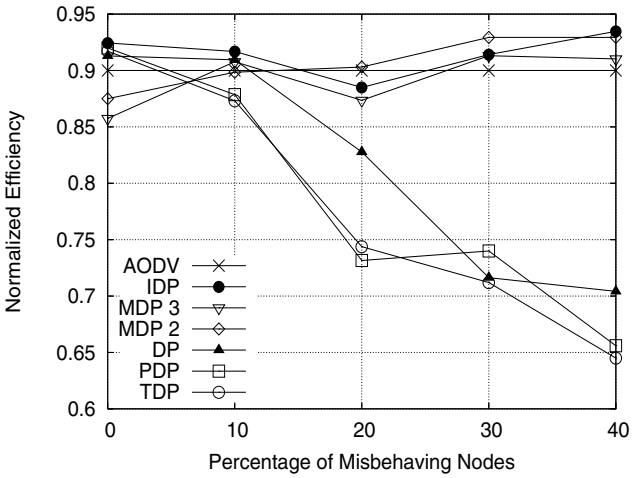
Let us begin by rationalizing our focus on IDP. Figure 3 illustrates the effect of node misbehavior on *pdf* observed for various approaches. For a static scenario (Fig:3(a)), versions with sufficient redundancy (AODV, MDP 2, MDP 3, IDP) make clusters and show high degree of reachability even with a considerable number of misbehaving nodes. Other techniques (PDP, TDP, DP) suffer a lot in the presence of misbehaving nodes and their performance degrades considerably. With the increase in mobility (Fig: 3(b), 3(c)) AODV performs the best, but still performance of IDP is very close to other redundant ones.

Figure 4 shows the effect of node misbehavior on $N.E$. In static scenario, performance of IDP lies above of most other variants. Not surprisingly, in mobile situations IDP is a clear winner. Though, with respect to successful delivery IDP may not be the best, its routing overhead is as low as other less redundant versions like TDP, PDP, DP. That is why $N.E$ is observed to be the best in IDP.

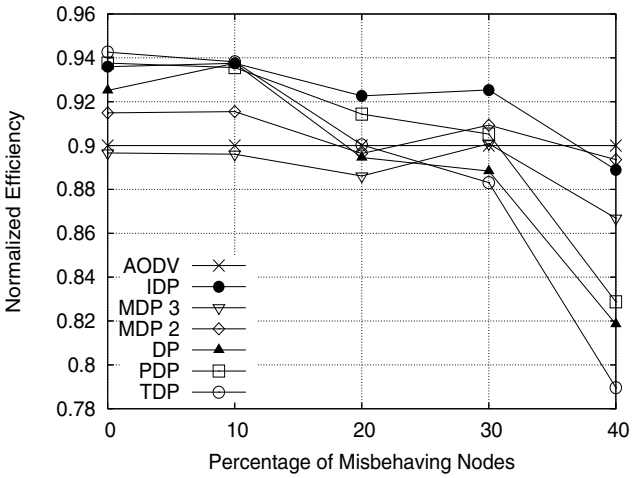
Figure 5 shows the effect of mobility on *N.E*. In all the cases, redundant approaches are almost oblivious to mobility and does not suffer much even in the presence of 40% misbehaving nodes. Performance of less redundant variants degrades largely in lower mobility, as unreachable destinations remain unreachable most of the times.

Next, Figure 6 shows the effect of node misbehavior on *Normalized Routing Load*. As expected, TDP and PDP incur the lowest amount of redundancy. IDP is also very close to these two and does not change much with the change in percentage of misbehaving nodes.

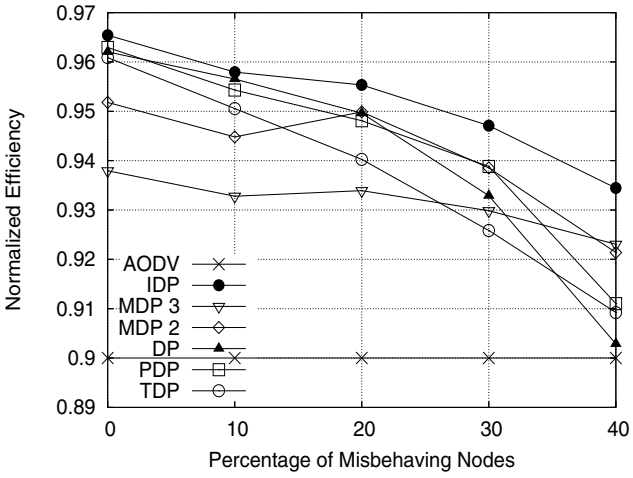
Table I shows the percentage of success in different iteration of IDP for the case of route establishment by broadcasting. For example, table I(a) shows that in static scenario and with 0% misbehaving nodes, all the routes are discovered by first two attempts, with PDP and DP. As percentage of misbehaving nodes increases, chances to find routes with these two variants become less probable. With the increase in mobility and misbehaving nodes, success in higher order iterations (MDP-2, 3, infinity) captures greater percentage. These are the cases where traditional approaches of trying with one variant would fail to succeed. Table I also stands for the redundancy control of IDP, because for all cases, least redundant version PDP (1st iteration) takes a major portion of obtaining success.



(a) Static Environment, Pause Time: 500

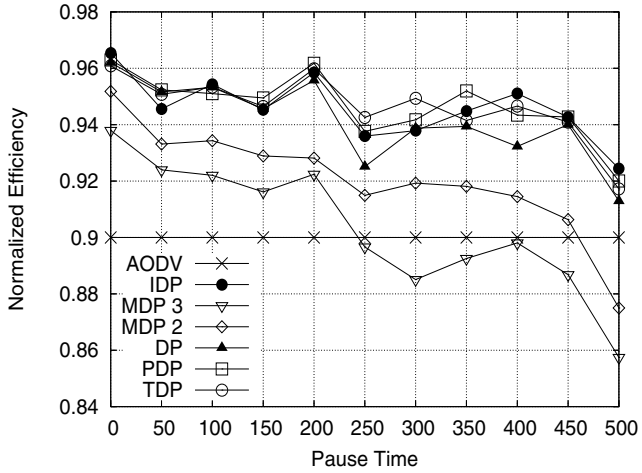


(b) Average Mobility, Pause Time: 250

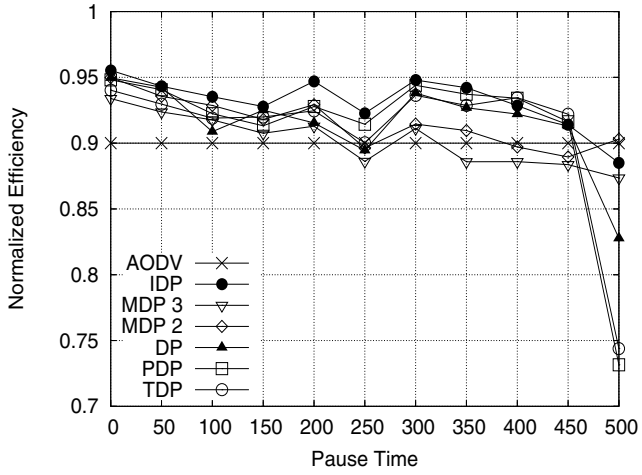


(c) Highest Mobility, Pause Time: 0

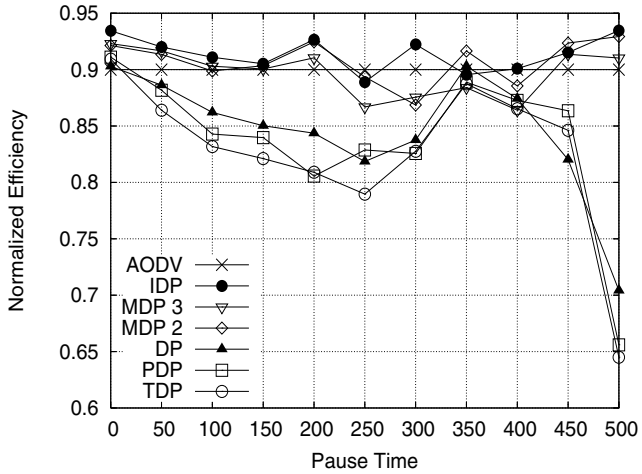
Fig. 4. Normalized Efficiency with Varying Percentage of Misbehaving Node



(a) 0% Misbehaving Node



(b) 20% Misbehaving Node



(c) 40% Misbehaving Node

Fig. 5. Normalized Efficiency with Varying Pause Time

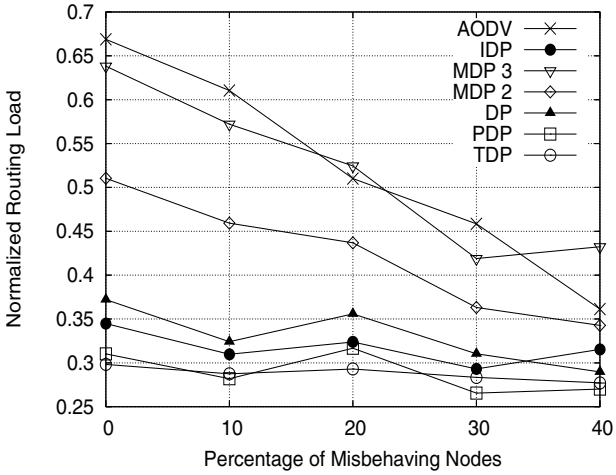


Fig. 6. Normalized Routing Load for Pause Time 250s

V. Conclusion and Future Works

From the above discussion and simulation results, it is clear that least redundant protocol like PDP, TDP, DP suffer badly in the presence of untrusted nodes. Adding redundancy with MDP can be a remedy but that redundancy is not desirable in the absence of untrusted nodes. As in the general case percentage of misbehaving nodes is not so high, blind deployment of MDP with $m >= 2$ is not effective from the perspective of network traffic. These issues demand a technique that lies somewhere in the middle of the above two. Our presented approach, IDP finds a dynamic way that cuts down redundancy for normal cases and incorporates controlled amount of redundancy only for the paths where any misbehaving nodes might be present. We can say that, though redundancy is undesired in trusted environment, controlled redundancy is effective for ad hoc networks where no assumption can be made about operational environment. This concluding remark is justified greatly in our simulation results.

IDP has similar reachability as AODV and at the same time has lower overhead as that of PDP and TDP. Therefore, it is desirable from both aspects. This approach will perform very well in trusted environment, though it is designed to be used in untrusted ones. This is because, when no misbehaving node is present, this algorithm will choose the least possible value of m and thus ensure lowest redundancy.

IDP is directly applicable to that class of broadcasts, where acknowledgment of successful transmission is returned back from the destination. Otherwise we need to incorporate this feature. The addition of multiple attempts may increase end to end delay which cannot be tolerated by some real time applications. Also, IDP assumes packet is dropped due to misbehaving nodes intentionally; when it is supposed to rebroadcast and exchange neighbor information packets. IDP does not consider packet dropping due to congestion in the network. In such transmission failure, IDP may intensify the problem by creating more congestion in the network.

TABLE I
DISTRIBUTION OF SUCCESSFUL ROUTE-SETUP IN ITERATIONS OF IDP

(a) STATIC ENVIRONMENT, PAUSE TIME: 500

Iteration	Percentage of Misbehaving Nodes				
	0%	**10%**	**20%**	**30%**	**40%**
1st	95.00%	80.00%	70.83%	57.14%	56.53%
2nd	5.00%	12.00%	12.50%	17.86%	13.04%
3rd	0%	4.00%	12.50%	21.43%	26.08%
4th	0%	4.00%	4.17%	3.57%	4.35%
5th	0%	0%	0%	0%	0%

(b) AVERAGE MOBILITY, PAUSE TIME: 250

Iteration	Percentage of Misbehaving Nodes				
	0%	**10%**	**20%**	**30%**	**40%**
1st	88.24%	89.23%	83.70%	78.81%	68.67%
2nd	8.09%	6.15%	7.41%	10.17%	12.00%
3rd	2.94%	4.62%	6.67%	6.78%	10.67%
4th	0%	0%	1.48%	2.55%	6.00%
5th	0.74%	0%	0.74%	1.69%	2.66%

(c) HIGHEST MOBILITY, PAUSE TIME: 0

Iteration	Percentage of Misbehaving Nodes				
	0%	**10%**	**20%**	**30%**	**40%**
1st	91.94%	90.68%	88.73%	83.57%	83.86%
2nd	4.03%	5.73%	5.88%	7.25%	4.93%
3rd	3.66%	2.51%	3.92%	6.28%	6.28%
4th	0.37%	0.72%	1.47%	1.93%	3.14%
5th	0%	0.36%	0%	0.97%	1.79%

We plan to investigate the effect of tuning the redundancy parameter m in some distinct scenarios. One assumption we adopt in favor of incrementing m is that - most of the packet drop is due to *route disassociation*, not due to *broadcast storm*. But in latter cases, we can tune m to be decremented whenever packet drop is due to collision or congestion and increment otherwise.

As our simulations were done with CBR packets, no reliability requirements were taken. Our next goal is to analyze how our proposed algorithm performs with TCP, which is common to most network applications.

VI. ACKNOWLEDGMENT

We would like to thank *Bangladesh University of Engineering & Technology* (BUET) for its generous support and research grant to make this work published. This paper is the outcome of the research conducted as part of the undergraduate thesis [24] under the supervision of Dr. A.K.M. Ashikur Rahman in CSE department, BUET.

REFERENCES

[1] The Network Simulator: NS-2: notes and documentation. http://www.isi.edu/nsnam/ns/.

[2] K. M. Alzoubi, P. J. Wan and O. Frieder. *New distributed algorithm for connected dominating set in wireless ad hoc networks.* In *Proc. HICSS-35*, 2002.

[3] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu and J. Jetcheva. *A performance comparison of multi-hop wireless ad hoc network routing protocols.* In *Proc. IEEE/ACM Intl. Conf. on Mobile Computing and Networking MOBICOM*, pages 85-97, 1998

[4] G. Calinescu, I. Mandoiu, P. J. Wan and A. Zelikovsky. *Selecting forwarding neighbors in wireless ad hoc networks.* In *Proc. ACM DIALM'2001*, pp. 34-43, Dec. 2001.

[5] G. Chelius, E. Fleury, and F. Valois. *Adaptive and Robust Adhoc Multicast Structure.* In *14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC 2003)*, September 2003.

[6] Z. J. Haas and M. R. Pearlman. *The zone routing protocol (ZRP) for ad hoc networks.* 1998, Internet Draft.

[7] F. Ingelrest, D. Simplot-Ryl and I. Stojmenovic. *Broadcasting in Hybrid Ad Hoc Networks.* In *Proc. 2nd Annual Conf. on Wireless On demand Network Systems and Services (WONS 2005)*, pages 131-138, January 2005

[8] M. Jiang, J. Li and Y. C. Tay. *Cluster based routing protocol (CBRP) functional specification.* 1998, Internet Draft.

[9] D. B. Johnson and D. A. Maltz. *Dynamic Source Routing in ad hoc wireless networks.* In Imielinski and Korth, editors, *Mobile Computing*, volume 353, Kluwer Academic Publishers, 1996

[10] D. Lichtenstein. *Planar formulae and their uses.* In *SIAM Journal on Computing*, 11(2): 329-343, 1982

[11] H. Lim and C. Kim. *Multicast tree construction and flooding in wireless ad hoc networks.* In *ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, 2000.

[12] W. Lou and J. Wu. *On reducing broadcast redundancy in ad-hoc wireless networks.* In *IEEE Transactions on Mobile Computing*, 1(2): 111-123, 2002.

[13] S. Marti, T. J. Giuli, Kevin Lai and M. Baker. *Mitigating routing misbehavior in mobile ad hoc networks.* In *Mobile Computing and Networking*, pages 255-265, 2000.

[14] S. Ni, Y. Tseng, Y. Chen and J. Sheu. *The broadcast storm problem in a mobile ad hoc network.* In *Proc. Mobicom'99*, pp. 151-162, Aug. 1999.

[15] W. Peng and X. C. Lu. *On the reduction of broadcast redundancy in mobile ad hoc networks.* In *Proc. First Annual Workshop on Mobile and Ad Hoc Networking and Computing, MOBIHOC*, pp. 129-130, Aug. 2000, Boston, USA.

[16] C. E. Perkins, E. M. Royers and S. R. Das. *Ad-hoc On-demand Distance Vector Routing (AODV), February 2003.* Internet Draft: draft-ietf-manet-aodv-13.txt.

[17] A. Qayyum, L. Viennot and A. Laouiti. *Multipoint relaying for flooding broadcast message in mobile wireless networks.* In *Proc. HICSS-35*, Jan. 2002.

[18] A. Rahman, P. Gburzynski and B. Kaminska. *Enhanced Dominant Pruning-based Broadcasting in Untrusted Ad-hoc Wireless Networks.* In *ICC*, 2007.

[19] E. M. Royer and C. K. Toh. *A review of current routing protocols for ad hoc mobile wireless networks.* In *IEEE Personal Communications,*, 6(2):46-55, 1999.

[20] I. Stojmenovic, M. Seddigh and J. Zunic. *Dominating sets and neighbor elimination based broadcasting algorithms in wireless networking protocol for wireless networks.* In *IEEE Transactions on Parallel and Distributed Systems*, 13(1): 14-25, January 2002.

[21] W. Wang, X. Y. Li and Y. Wang. *Truthful Multicast in Selfish Wireless Networks.* In *ACM MobiCom*, 2004

[22] J. Wu and F. Dai. *Broadcasting in ad hoc networks based on self-pruning.* In *Proc. of INFOCOM*, March 2003.

[23] S. Zhong, L. Li, Y. Liu, and Y. R. Yang. *On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks - an integrated approach using game theoretical and cryptographic techniques.* In *Proceedings of the Eleventh International Conference on Mobile Computing and Networking (Mobicom)*, Sept. 2005.

[24] *Effect of Redundancy on Broadcasting in Untrusted Ad hoc Wireless Network*, N. Shahriar, S. A. I. Mujib, A. R. Roy, Bangladesh University of Engineering & Technology, 2008.