RRBS: Reliable Broadcasting over Unreliable Wireless Links in Resilient Ad-Hoc Networks

Fu-Wen Chen and Jung-Chun Kao
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.

***Abstract*—This paper addresses the minimum transmission broadcast problem in resilient ad-hoc networks and presents efficient solutions, including centralized heuristic algorithms and a distributed algorithm. In disaster-resilient networks, wireless links are unreliable due to harsh environments. Distinct from related work in the literature which typically assumes wireless links are reliable, we address the issue of broadcasting over both reliable wireless links and unreliable wireless links. Our main contributions are as follows: First, we develop heuristic algorithms for both reliable- and unreliable-link model. Second, we propose a distributed algorithm based on 1-hop neighbor information. Using simulation, we confirm that the proposed heuristic algorithms can reduces the number of transmissions significantly and the proposed distributed algorithm performs comparably well to the centralized heuristic algorithms in terms of both delivery ratio and the number of transmissions.**

***Keywords*—broadcast, flooding, ad-hoc networks, heuristic algorithm, resilience**

I. INTRODUCTION

Broadcasting, in which a node sends a message to all other nodes in the network, is a common and vital operation in disaster-resilient ad-hoc networks. Broadcasting is required by many on-demand routing protocols such as AODV [1] in their route discovery processes. Besides, broadcasting is widely used for sending safety messages to nodes over the entire network or a certain region in vehicular ad-hoc networks and wireless sensor networks. In disaster-resilient networks, wireless links are likely unreliable due to harsh environments. To provide efficient and reliable broadcast service over unreliable wireless links is of paramount importance.

Naive broadcast schemes are inefficient in wireless networks [2]. A representative example is flooding, in which each node rebroadcasts a message when receiving that message for the first time. Pure flooding often causes too many unnecessary packet transmissions and may lead to broadcasting storm [3]. To avoid the broadcasting storm problem, a crucial issue is to develop a broadcast scheme with the minimum number of transmissions. This problem is referred to as the minimum transmission broadcast (*MTB*) problem [2].

In the MTB problem, network models, particularly link models, play an important role and may affect performance significantly. There are two fundamental types of link models—the *reliable-link model* and the *unreliable-link model*. Under the reliable-link model, packets transmitted over any link are always delivered provided that there is no collision. On the contrary, under the unreliable-link model, packets transmitted from one end of a link reach the other end with a probability. Since wireless links are inherently error-prone (due to a number of dynamic factors such as noise, fading and interference), the unreliable-link model is more practical than the reliable-link model. This paper studies the MTB problem under *both* of the models; however, the focus is on studying reliable broadcasting over *unreliable* wireless links.

Most of related work in the literature instead assumes the reliable-link model. Under the reliable-link model, the MTB problem is equivalent to the maximum leaf spanning tree (MLST) problem [4] and the minimum connected dominating set (MCDS) problem. Packets can be optimally broadcast along the constructed spanning tree or connected dominating set. Unfortunately, the MLST and MCDS problems have been proven NP-hard [5]. Therefore, a number of approximation algorithms [6], [7], [8] and sub-optimal broadcast schemes [9], [10], [11] have been proposed.

Among the sub-optimal broadcast schemes, Khabbazian and Bhargava in [11] proposed the Responsibility-Based Scheme (RBS) to reduce the number of transmissions while guaranteeing full delivery under the assumption of the reliable-link model. In RBS, each node is responsible for transmitting the message to the closest nodes which have not received the message. A node that is not responsible for any other node does not need to rebroadcast the message, thus reducing the occurrences of two close neighbors broadcasting the same message. Although RBS is very simple and does not require knowing 2-hop (or higher-hop) neighbor information, it is very effective in reducing the number of transmissions.

From a practical point of view, although the simplicity and effectiveness makes RBS very attractive under the reliable-link model, wireless links are inherently unreliable but fast algorithms/heuristics designed for the unreliable-link model has not been well investigated. We study these issues in this paper and our contributions include:

- Developing centralized heuristic algorithms for both reliable- and unreliable-link model, and
- Proposing a distributed yet near-optimal broadcast scheme, called RRBS, under the unreliable-link model.

What follows explains the contributions mentioned above.

Since the MTB problems are NP-hard, optimal solutions cannot be derived in polynomial time. The MTB problems have been formulated into mixed integer linear programming (MILP) problems in [12], which also proposed fully-distributed, game-based approaches. In this work, we develop two heuristic algorithms, which are centralized, for the two models, respectively. Indeed, the broadcast problem under each model

This research was supported in part by National Science Council, Taiwan, under grants NSC 98-18-E-007-007, NSC 98-2219-E-007-012, NSC 99-2219-E-007-005, and NSC 100-2219-E-007-003.

is transformed into a problem of graph theory. And the heuristic algorithms actually attempt to find a maximum leaf spanning tree and a *special* minimum spanning tree which we name "the maximum leaf minimum spanning tree".

Because RBS does not perform well in terms of delivery ratio under the unreliable-link model (as shown later in Section V), we propose a distributed scheme called the Reliable Responsibility-Based Scheme (RRBS). RRBS leverages 1-hop neighbor knowledge for not causing high overhead and uses the notion of reliable zone and unreliable zone to improve the delivery ratio. As depicted in Fig. 1, transmission coverage is divided into reliable zone and unreliable zone, whose ranges can be adjusted adaptively according to the application's reliability requirement and the number of neighbor nodes.

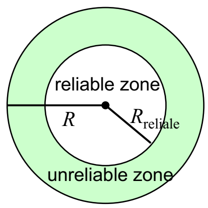

Fig. 1. Transmission coverage is divided into two zones. The inner region is reliable zone, and the outer shadowing region is unreliable zone.

With the notation of reliable/unreliable zones, RRBS makes the forwarding decision as follows. After receiving the broadcast message for the first time, a node overhears transmissions for a defer period and determines whether or not to be responsible for some neighbor node. To avoid two potential forwarders from broadcasting the same message, only the potential forwarder closest to a neighbor node which has not received the message is responsible for that neighbor. To not erroneously infer other nodes as (qualified) potential forwarders, only the transmissions within the reliable zone are considered reliable and nodes outside the reliable zone are not presumed (qualified) potential forwarders. Simulation results show that RRBS achieves comparable performance to our proposed centralized heuristic algorithm under the unreliable-link model.

The remainder of this paper is organized as follows. Section II describes the network model in detail. For both network models, our heuristic algorithms are described in section III. In section IV, we introduce RRBS and compare RRBS with RBS. Section V presents our simulation results which show performance improvement of our proposed schemes. Finally, we present some concluding remarks in section VI.

II. Network Model

We consider a wireless network in which there are many nodes. The set of all the nodes in the network is denoted by V. For simplicity of exposition, we assume that all nodes have equal transmission range R, although the algorithms proposed in this paper do not require this assumption. The distance between any two nodes, node u and node v, is denoted by d_{uv}.

Node v is a neighbor of node u if $d_{uv} \leq R$. $N(v)$ is defined as the set of all the neighbor nodes of node v. For a set of nodes S, $N(S)$ is defined as the set of all neighbor nodes of every node $v \in S$. Same as [11], we assume that each node knows its own location (obtained using GPS or other localization techniques) and the locations of its 1-hop neighbors (achieved by periodically broadcasting hello messages to neighbors).

For any two nodes u and v, the link reception probability that node v can successfully receive a packet sent from node u is denoted by p_{uv}. In general, the shorter d_{uv} is, the greater p_{uv} is. Depending on the value of p_{uv}, we define two link models—the reliable-link model and the unreliable-link model as follows.

- *The reliable-link model*: Any pair of nodes within the transmission range has a reliable link connecting each other. More precisely, for any two nodes u and v, $p_{uv} = 1$ if $d_{uv} \leq R$ and $p_{uv} = 0$ otherwise. (Typically $p_{uv} = 1$ for all pairs within the transmission range, but in general it can be a single constant between 0 and 1.) Under this model, provided that there is no collision, packets transmitted within the transmission range are always delivered.
- *The unreliable-link model*: Links connecting two nodes through wireless channels are not necessarily reliable. For any two nodes u and v, $0 < p_{uv} \leq 1$ if $d_{uv} \leq R$ and $p_{uv} = 0$ otherwise. Under this model, provided that there is no collision, packets transmitted from one end of a link reach the other end with a reception probability of p_{uv}.

A major difference between the above two models is that the link reception probabilities of all node pairs in the unreliable model vary according to the pair distance and other factors (such as fading, shadowing, interference, and noise), but their values in the reliable-link model are either one (within transmission range) or zero (out of transmission range). Undoubtedly, the unreliable-link model matches the reality better than the reliable-link model. Whereas related work in the literature assumes the reliable-link model, this paper presents techniques designed for the two models.

III. Heuristic Algorithms

Since the optimal solutions of MTB problems cannot be derived in polynomial time, we develop two heuristic algorithms which construct near-optimal broadcast trees. What follows introduces the two heuristic algorithms—one for the reliable-link model and the other for the unreliable-link model.

A. Heuristic Algorithm for the Reliable-Link Model

Under the reliable-link model, the MTB problem is equivalent to the maximum leaf spanning tree (MLST) problem. Because the MLST problem has been proven NP-hard, Lu proposed a 3-approximation algorithm in [6]. However, our study through simulation shows that Lu's original algorithm could not achieve near-optimal performance at high node density and therefore we modify Lu's algorithm.

We outline Lu's algorithm in this section; readers are suggested to see the details in [6]. Lu's algorithm consists of two stages. In the first stage, a leafy forest composed of leafy trees is constructed. A tree is leafy if *a)* it has one or more nodes with degree of at least 3, and *b)* each node with degree two in the tree connects to exactly two nodes with degree of at least 3 in the tree. In the second stage, each leafy tree is considered as an individual node and a spanning tree connecting these nodes is constructed using any spanning tree algorithm. The authors of [6] suggest to grow the leafy forest in the descending order of node degree for performance improvement, but we find out that it still cannot achieve near-optimality at high node density.

Our modification of the order of growing the forest improves the performance of Lu's algorithm especially at high node density. We define a few terms of terminology:

- The *current degree* of node u is the number of edges that connect node u and have been added to the forest so far.
- The *potential degree* of node u is the number of edges that connect node u but have not been added to the forest.

In the first stage of Lu's algorithm, a node could be newly added to the leafy forest as an internal node only if the sum of its current degree and potential degree is greater or equal to three. In a sense, a node with greater potential degree is more suitable to be an internal node. Thus, the order in which the forest grows is modified to be the descending order of the potential degree of nodes. Current and potential degrees are updated whenever the forest is updated. With such a dynamic construction of the leafy forest, the performance of the heuristic algorithm is better than that of Lu's algorithm especially at high node density.

B. Heuristic Algorithm for the Unreliable-Link Model

A single transmission over an unreliable link may fail. To guarantee delivery, a sender (*i.e.,* internal node) needs to take a number of retransmissions until a message successfully reaches all of the intended recipients (*i.e.,* children of the internal node). Taking retransmissions into account, the MTB problem under the unreliable-link model is defined to find the broadcast tree with the minimum *expected* number of transmissions until all the nodes have received the message successfully.

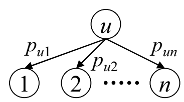

Fig. 2. Node u has n children. The message reception probability from node u to node i is p_{ui}.

Due to linearity, the expected number of transmissions until all nodes in a broadcast tree have received a message is equal to the sum of the expected numbers of transmissions for each internal node to send the message to its children. Suppose node u is an internal node. As depicted in Fig. 2, we denote the number of node u's children by n and the reception probabilities at its children by p_{ui}, $i = 1, 2, \ldots, n$. Let T_u be the random variable representing the number of transmissions by node u until all its children have received the message. $\overline{T_u}$ can be modeled as follows:

$$\begin{aligned}\overline{T_u} &= \sum_{t=0}^{\infty} P(T_u > t) \\ &= 1 + \sum_{t=1}^{\infty}\left(1 - P(T_u \le t)\right) \\ &= 1 + \sum_{t=1}^{\infty}\left(1 - \prod_{i=1}^{n}(1-(1-p_{ui})^t)\right) \\ &\doteq 1 + \sum_{t=1}^{\infty}\sum_{i=1}^{n}(1-p_{ui})^t \\ &= 1 + \sum_{i=1}^{n}\frac{1-p_{ui}}{p_{ui}}. \end{aligned} \quad (1)$$

Having the above approximation, we define the *edge cost* of edge (u, v) as $(1 - p_{uv}) / p_u$ if node u is an internal node and node v is its child; otherwise, the edge cost is defined to be zero. The MTB problem under the unreliable-link model is equivalent to find a spanning tree with the minimum sum of the number of internal nodes and the total edge cost. We call this problem as the maximum leaf minimum spanning tree (MLMST) problem.

The MLMST problem is NP-hard since the NP-hard MLST problem is its special case when the edge cost of each edge is zero. Based on the modified Lu's algorithm described in the previous subsection, we design a heuristic algorithm for the MLMST problem.

The heuristic algorithm also consists of two stages. In the first stage, *mediocre links* are temporarily removed to avoid including high-cost edges, and after that, a leafy forest is constructed in the same way as the heuristic algorithm for the reliable-link model described in Section III.A. An edge (u, v) is a mediocre link if $p_{uv} < p_{th}$ and is a decent link if $p_{uv} \ge p_{th}$, where p_{th} is the threshold of the reception probability that tells apart decent links and mediocre links. p_{th} is set to 0.8 in our simulation unless otherwise specified, and the reason will be given in section V.B. In the second stage, mediocre links are put back, and after that, we consider all leafy sub-trees as individual nodes and construct a minimum spanning tree using Prim's algorithm. The Prim's algorithm starts with the sub-tree consisting of the source node and then spans the network by greedily adding, one at a time, an edge with the minimum cost. Adding edge (u, v) brings a cost of $(1 - p_{uv}) / p_{uv}$ if node u is an internal node before adding the edge. Otherwise, it increases the total cost by $1 + (1 - p_{uv}) / p_{uv}$, since adding edge (u, v) makes node u switch to an internal node from a leaf node and such a switch additionally increases the total cost by 1. The edge cost of each node has to be updated whenever any node becomes an internal node. As shown later in Section V, this heuristic algorithm is highly efficient at any node density.

IV. Reliable Responsibility-Based Scheme

Khabbazian and Bhargava proposed the Responsibility-Based Scheme (RBS) [11] to minimize the number of transmissions. RBS performs very well under the reliable-link model, but its performance in terms of delivery ratio degrades seriously under the unreliable-link model. To solve this problem, we modify RBS and propose the Reliable Responsibility-Based Scheme (RRBS) whose goal is to minimize the number of transmissions while maximizing the delivery ratio.

Same as RBS, RRBS assumes that each node knows the relative locations of its 1-hop neighbors. RRBS also assumes that the packet reception probability can be estimated from the pair distance. In this section, we first outline the key idea and major disadvantage of RBS and then describe RRBS in detail.

A. RBS

RBS [11] is a receiver-based broadcasting scheme: Senders do not instruct receivers whether or not to rebroadcast the received message; instead, receivers themselves decide whether or not to rebroadcast the message. When a node, say node u, receives a broadcast message for the first time, it will set a defer timer. During the defer period, it may hear other nodes broadcasting the same message and can guess which neighbors may have already received the message. (More precisely, when node *u* hears a neighbor, say node *v*, broadcasting a message, node *u* guesses that all of its neighbors within the transmission range of node *v* have successfully received the message.) When the defer timer expires, if node *u* perceives responsible for at least one neighbor node, it will rebroadcast the message. (Node *u* is responsible for a neighbor node, say node *v*, if node *u* guesses that node *v* has not received the message and node *u* itself is nearest to node *v* among the neighbor nodes that node *u* thinks have already received the message.)

RBS makes a guess based on received messages and 1-hop neighbor information. However, because a number of factors such as noise and interference are dynamic and wireless links are unreliable, the guess may be wrong. A wrong guess may prevent nodes from broadcasting the message to their neighbor nodes. As a result, delivery ratio degrades severely. We refer to this problem as the *wrong guess problem*.

B. RRBS

To solve the wrong guess problem, we propose the Reliable Responsibility-Based Scheme (RRBS). In the following, we explain the key ideas and describe the pseudo code of RRBS.

1) Key Ideas

Since a wrong guess degrades the performance of RBS severely, RRBS focuses on improving the guessing mechanism. As depicted in Fig. 1, the transmission range of a node is divided into two zones—reliable zone and unreliable zone. The radius of the transmission range is R, and the radius of the reliable zone is R_{reliable}. Only transmissions within the reliable zone would be considered reliable, but a node may still need to be responsible for its neighbor nodes in the unreliable zone. The guessing mechanism is modified as follows: When a node *u* hears a neighbor node *v* broadcasting a message, node *u* thinks its neighbor nodes within the reliable zone centered at node *v* can receive the message.

The ideas about reliable zone have been described above. The next issue is how to define the size of the reliable zone. If the reliable zone is too large, then wrong guess occurs frequently and the delivery ratio degrades severely. On the other hand, if the reliable zone is too small, then each node thinks most of its neighbor nodes cannot receive the message and consequently a lot of redundant transmissions take place.

Two important factors in determining the size of reliable zone are the reliability requirement and the number of neighbors. If the reliability requirement specified by an application is high, then the reliable zone would be smaller to prevent the wrong guess from happening, at a cost of causing more transmissions.

Besides the reliability requirement, the number of neighbors is another important factor in determining the size of reliable zone. If a node has a lot of neighbors, it has higher chance of hearing the message several times; thus, the wrong guess can be tolerated better. In this case, the reliable zone can be larger and the reliability can still keep high. On the other hand, if a node has few neighbors, then any wrong guess may degrade the reliability seriously. The relation between the number of nodes and p_{reliable} used in our simulation is shown in Table 3.

2) Pseudo Code

The pseudo code of the RRBS algorithm is given in Fig. 3. This algorithm runs distributedly on each node. In Fig. 3, L_{Nbr} is the neighbor list of node *u*, and L_{NbrRcv} is the set of neighbors that may have already received the message (guessed by node *u*). When node *u* receives a message for the first time from a neighbor node (say node *v*), it will set a defer timer. During the defer period, node *u* updates its L_{NbrRcv} every time upon hearing the same message from other neighbors. The update procedure follows the guessing mechanism described previously. When the defer timer expires, node *u* determines whether or not to broadcast the message. If node *u* finds that it is responsible for any neighbor, then it will broadcast the message (the algorithm returns true). Otherwise, it will keep silent (the algorithm returns false).

```
Algorithm: RRBS for node u
L_Nbr ← the set of neighbors of u
L_NbrRcv ← ϕ
if u receives a message for the first time (from v) then
  UpdateList(v)
  set a defer timer
  while the defer timer not expired do
    if u receives the message from any neighbor v' then
      UpdateList(v')
    end if
  end while
  for each node v in L_Nbr do
    if v ∉ L_NbrRcv then
      if u is closer to v than all nodes in L_NbrRcv then
        return true
    end if
  end for
  return false
end if

Procedure: UpdateList(v)
if v ∉ L_NbrRcv then
  add v to L_NbrRcv
end if
for each node i in L_Nbr do
  if i ∉ L_NbrRcv and d_vi ≤ R_reliable then
    add i to L_NbrRcv
  end if
end for
```

Fig. 3. Pseudo code of the RRBS algorithm.

V. Performance Evaluation

In this section, we evaluate the performance of all the broadcast schemes/algorithms presented in sections III and IV. The performance metrics of interest are transmission ratio and delivery ratio. The transmission ratio is defined as the number of transmissions divided by the number of nodes—assuming of having identical delivery ratio, a smaller value of transmission ratio implies higher message efficiency. Under the unreliable-link model, it is possible for a node to broadcast a message multiple times to ensure delivery to its neighbor(s). Therefore, the transmission ratio is not confined to one under the unreliable-link model. The delivery ratio is defined as the percentage of nodes which can receive the broadcast message—the larger the delivery ratio is, the more reliable the broadcast scheme is. The ultimate objective is to minimize the transmission ratio while maximizing the delivery ratio.

A. Simulations for the Reliable-Link Model

TABLE 1
SIMULATION PARAMETERS USED FOR THE RELIABLE-LINK MODEL

Parameter	Value
Size of area	$1000 \times 1000\text{m}^2$
Transmission range	200m
Number of nodes	10–200
Propagation model	two-ray ground
Collision model	collision-free

We first run simulations for the reliable-link model. In this simulation setup, important simulation parameters are shown in Table 1. We consider the collision-free environment where transmissions within the transmission range are always delivered. In each simulation run, we distribute a number of nodes, ranging from 10 to 200, uniformly over a square region of area $1000 \times 1000\text{m}^2$. A transmission ratio value is averaged over 500 instances. Note that because RRBS degenerates to RBS under the reliable-link model, RRBS and RBS have exactly the same performance in this simulation setup.

As can be seen in Fig. 4, the transmission ratio decreases as node density increases for all schemes, which is to be expected since a transmission is delivered to more nodes at higher node density. The simulation results also show that our heuristic modification of the order of growing the forest effectively improves the performance of Lu's algorithm especially at high node density. In addition, as a distributed scheme, RRBS/RBS achieves better performance at high node density compared to Lu's algorithm, which is a centralized algorithm. This shows the message efficiency of RRBS/RBS..

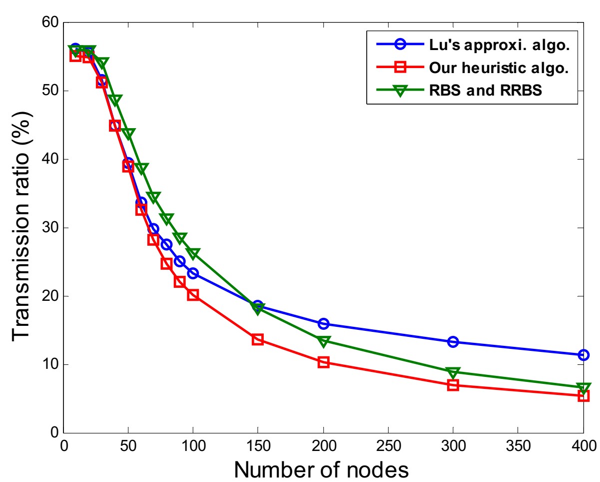

Fig. 4. The transmission ratio vs. the number of nodes, under the reliable-link model.

B. Simulations for the Unreliable-Link Model

For the unreliable-link model, as shown in Table 2, we run simulations by adding the additive white Gaussian noise (AWGN) to the wireless channel. The free space model is used for modeling the path loss. This implies that the longer distance, the smaller reception probability. Other simulation parameters are similar to what are used for the reliable-link model. In RRBS, each node sets p_{reliable} according to its number of neighbors as shown in Table 3. In this simulation setup, we evaluate both the delivery ratio and transmission ratio of several schemes.

TABLE 2
SIMULATION PARAMETERS USED FOR THE UNRELIABLE-LINK MODEL

Parameter	Value
Size of area	$1000 \times 1000\text{m}^2$
Transmission range	360m
Number of nodes	10–100
Propagation model	free space ground
Collision model	collision-free
Channel model	AWGN
Modulation	BPSK

TABLE 3
EACH NODE SETS DIFFERENT p_{reliable} ACCORDING TO ITS NUMBER OF NEIGHBORS

# of neighbors	at most 9	10 to 14	at least 15
p_{reliable}	0.9	0.5	0.3

To evaluate the performance of RRBS in terms of delivery ratio, we compare RRBS with pure flooding which achieves the highest delivery ratio in a collision-free environment, among all possible schemes without retransmission. As we can see from Fig. 5, RRBS can achieve nearly the same delivery ratio as pure flooding, but the transmission ratio of RRBS is much lower than that of pure flooding as shown in Fig. 6. Therefore, we claim RRBS achieves near-optimality in terms of delivery ratio. On the contrary, the delivery ratio of RBS degrades to some significant extent due to the wrong guess problem.

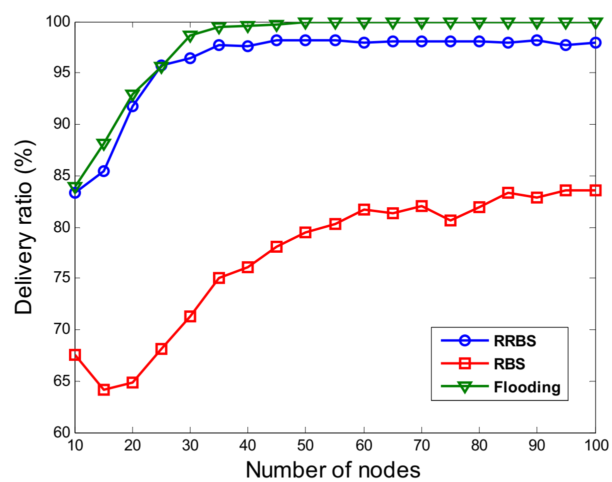

Fig. 5. The delivery ratio vs. the number of nodes under the unreliable-link model. (Nodes are uniformly distributed.) The results for the heuristic algorithm are not shown here because it achieves 100% delivery ratio as long as the network is connected.

To evaluate RRBS's performance in terms of transmission ratio, we compare RRBS with our proposed heuristic algorithm which is centralized and is regarded as a baseline. As shown in Fig. 6, the transmission ratio of RRBS and RBS are lower than the baseline because part of nodes can not receive the message and they have no chance to transmit the message. The lower the delivery ratio is, the lower the transmission ratio tends to be. Unless the number of nodes is small, RRBS can achieve nearly 100% delivery ratio and the transmission ratio is close to that of our centralized heuristic algorithm. This shows that RRBS can achieve high delivery ratio without much redundant transmissions in a distributed fashion.

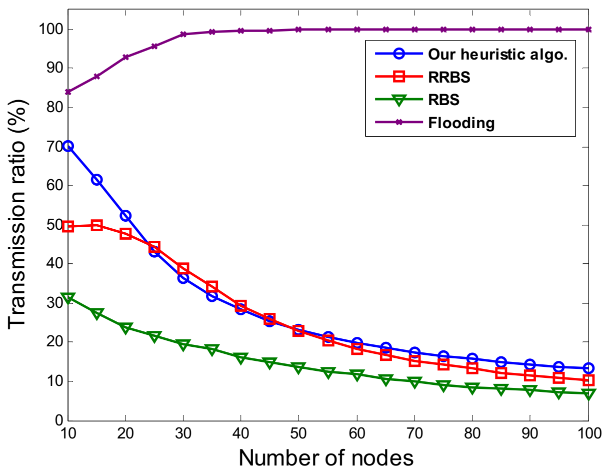

Fig. 6. The transmission ratio vs. the number of nodes, under the unreliable-link model. (Nodes are uniformly distributed.)

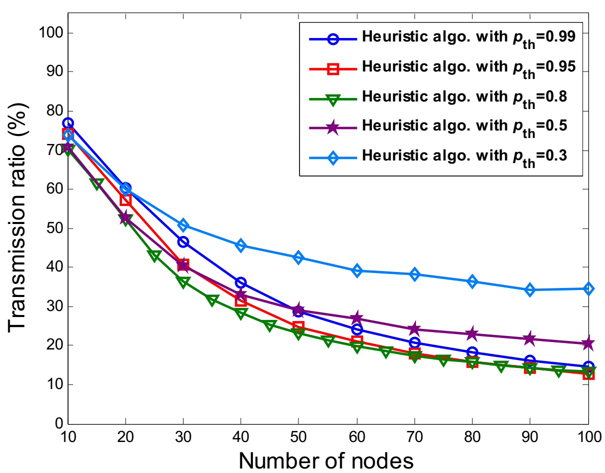

Fig. 7. The impact of different p_{th} on the performance of our heuristic algorithm. p_{th}=0.8 is the best and thus is used in other parts of our simulation.

We extend the simulation to evaluate the impact of p_{th} on the performance of our proposed heuristic algorithm. p_{th} is a threshold used to filter out mediocre links. To determine the best p_{th}, we run our heuristic algorithm with various p_{th}. As shown in Fig. 7, p_{th} of different value results in different performance. When p_{th} is too low, many mediocre links are considered in the first stage, which causes poor performance. On the contrary, when p_{th} is too high, only few links are considered in the first stage and the leafy forest could not be constructed in a good way. As we can see, the performance is best when p_{th} = 0.8; therefore, p_{th} is set to 0.8 in our simulation.

VI. Conclusion

In this paper, we have addressed the minimum transmission broadcast (MTB) problems under the reliable-link model and under the unreliable-link model. Under the reliable-link model, our proposed techniques outperform the counterparts of the state-of-art techniques: Our heuristic algorithm performs comparably well to Lu's 3-approximation algorithm at low to medium node density and performs better at high node density.

More importantly, this paper presents techniques exclusively designed for the MTB problem under the unreliable-link model. We have proposed a centralized heuristic algorithm as well as a distributed algorithm. Simulation results show our proposed distributed algorithm achieves near-optimality in terms of delivery ratio and has comparable performance to the centralized heuristic algorithm in terms of number of transmissions.

As can be seen from the simulation results, any broadcast scheme without retransmissions cannot achieve high delivery ratio when nodes are distributed sparsely in a wireless network. This is because nodes are far away from each other and a single transmission over wireless channels cannot deliver with a high probability. As part of our future work, we will investigate a broadcast scheme to ensure a high delivery ratio and a small number of transmissions in this sparsely-distributed condition.

References

[1] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. IEEE WMCSA*, New Orleans, Feb. 1999.

[2] J. Hong, W. Li, S. Lu, J. Cao, and D. Chen, "Sleeping Schedule Aware Minimum Transmission Broadcast in Wireless Ad Hoc Networks," in *Proc. IEEE ICPADS*, Melbourne, Dec. 2008.

[3] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. ACM MobiCom*, Seattle, Aug. 1999.

[4] T. Fujie, "An exact algorithm for the maximum leaf spanning tree problem," *Comput. Oper. Res.,* vol. 30, pp. 1931-1944, Nov. 2003.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. San Francisco: Freeman, 1979.

[6] H.-I. Lu and R. Ravi, "Approximating Maximum Leaf Spanning Trees in Almost Linear Time," *Journal of Algorithms,* vol. 29, pp. 132--141, Oct. 1998.

[7] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, New York, June 2002.

[8] S. Funke, A. Kesselman, U. Meyer, and M. Segal, "A simple improved distributed algorithm for minimum CDS in unit disk graphs," *ACM Transactions on Sensor Networks,* vol. 2, pp. 444-453, 2006.

[9] B. Bako, F. Kargl, E. Schoch, and M. Weber, "Advanced Adaptive Gossiping Using 2-Hop Neighborhood Information," in *Proc. IEEE GLOBECOM*, New Orleans, LA., Dec. 2008.

[10] H. Liu, P. Wan, X. Jia, X. Liu, and F. Yao, "Efficient Flooding Scheme Based on 1-Hop Information in Mobile Ad Hoc Networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, April 2006.

[11] M. Khabbazian and V. K. Bhargava, "Efficient Broadcasting in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing,* vol. 8, pp. 231-245, Feb. 2009.

[12] F.-W. Chen and J.-C. Kao, "Game-based Broadcast over Reliable and Unreliable Wireless Links in Multihop Networks," *IEEE Transactions on Mobile Computing*, preprint available online at http://dx.doi.org/10.1109/TMC.2012.133.