HybridCast: A Hybrid Probabilistic/Deterministic Approach for Adjustable Broadcast Reliability in Mobile Wireless Ad Hoc Networks

Thadpong Pongthawornkamol, Klara Nahrstedt
University of Illinois at Urbana-Champaign
{tpongth2,klara}@cs.uiuc.edu

Guijun Wang
Boeing Phantom Works
guijun.wang@boeing.com

Abstract

Broadcast is a crucial yet expensive building block for many applications in bandwidth-scarce mobile wireless ad hoc networks. We propose a hybrid deterministic/probabilistic, decentralized broadcast protocol with adjustable broadcast reliability and overhead. The paper first proposes a purely probabilistic, topology-aware broadcast algorithm. The probabilistic broadcast adjusts each node's broadcast forwarding probability locally such that the average broadcast reliability requirement is met. An extension of the probabilistic broadcast to tolerate node mobility and packet loss is then presented. Furthermore, the paper augments the proposed *probabilistic* broadcast scheme with an existing *deterministic* broadcast protocol in order to reduce excessive broadcast overhead. The proposed hybrid protocol, called *HybridCast*, combines good characteristics of probabilistic broadcasts, such as adjustable reliability and resilience to mobility, with good characteristics of deterministic broadcasts, such as few retransmissions and low packet collisions. The simulation results show that the proposed protocol can achieve the system's reliability requirement with good tolerance to mobility and packet losses while incurring low broadcast overhead.

I. Introduction

Broadcast operation is a process of delivering certain information from one node to all nodes in the system. In wireless ad hoc networks, broadcast is a basic yet crucial operation that serves as a building block for several operations such as route discovery, data dissemination, and data aggregation. While broadcast is important and useful, it is also considered expensive in bandwidth-scarce wireless ad hoc network.

Being considered as a classical problem in wireless ad hoc network domain, the reliable broadcast problem has been addressed by a plethora of works [1]–[6]. The main goal of such works is to deliver all messages to all the nodes in the system with as smallest forwarding overhead as possible. In the other words, the existing broadcast protocols try to find the smallest set of relay nodes (i.e. retransmitting nodes) in order to deliver a packet to all or almost all nodes. It is commonly known that the reliable broadcast problem in static wireless ad hoc networks can be reduced to the problem of finding a minimum connected dominating set (MCDS) in a graph, which has been proved to be an NP-hard problem [7]. However, in the context of mobile ad hoc networks where each node moves constantly with non-negligible speed and different mobility patterns, finding an optimal forwarding set based on the complete graph topology is not feasible due to rapid topology changes [5]. Instead, an ideal broadcast protocol for mobile ad hoc networks must sustain frequent topology changes while providing good broadcasting reliability with near-optimal overhead.

The reliable broadcast protocols proposed so far can be categorized into two groups based the way the forwarding node set is determined. The first approach is *deterministic broadcast* [2]–[4]. In a deterministic broadcast protocol, the set of relay nodes is chosen deterministically to cover the entire graph. When a node receives a broadcast message, it will decide deterministically whether to forward the message or not. The advantages of deterministic broadcast schemes are low broadcast overhead and low potential packet collisions. However, deterministic broadcast schemes are prone to node mobility, as the deterministic rules rely tightly on membership information, which becomes stale as node speed increases.

Another category of reliable broadcast protocols is called *probabilistic broadcast* [6], [8]. In a probabilistic broadcast protocol, the forwarding set is chosen in a probabilistic manner. When a node receives a broadcast message, it will forward the message with some probability. The forwarding probability is either statically or dynamically computed locally at each node based on each node's environment (i.e. topology or channel condition). In contrast to deterministic broadcast, a probabilistic broadcast protocol usually causes redundant retransmissions, which incurs relatively more overhead and potential packet collisions. On the other hand, the broadcast redundancy makes probabilistic broadcast resilience to node mobility and node failures.

In order to combine the benefits of probabilistic broadcast and deterministic broadcast altogether, this paper proposes *HybridCast*, a hybrid probabilistic/deterministic broadcast protocol with adjustable broadcast reliability for mobile ad hoc networks. Under low node mobility, HybridCast operates in deterministic mode to save overhead. However, when node mobility

or the packet loss rate is high, HybridCast operates in probabilistic mode for better resilience. Each node in HybridCast locally adjusts the forwarding scheme and forwarding probability such that the system reliability requirement is met while minimizing broadcast overhead. Hence, Hybridcast can be considered as a generalization of the existing deterministic broadcast approach to handle with mobility and variable channel conditions more efficiently.

This paper has three main contributions. First, it proposes a topology-aware, probabilistic broadcast protocol with *adjustable reliability and overhead*. The protocol is also able to tolerate high mobility and packet loss. Second, the paper proposes the hybrid probabilistic/deterministic broadcast scheme that augments our proposed probabilistic broadcast protocol with existing deterministic broadcast protocols [4] to further reduce overhead. Third, the paper presents extensive simulations on different scenarios. The results have shown that the proposed HybridCast protocol can achieve system's reliability requirements with less overhead compared to pure probabilistic broadcast.

The organization of this paper is as follows. Section II presents the system model and the design goals of our broadcast protocol. Section III describes the detail of the proposed probabilistic broadcast protocol. Section IV discusses the adopted deterministic broadcast protocol and the proposed hybrid probabilistic/deterministic protocol (HybridCast). Section V presents simulation results of performance evaluation and comparison between the proposed schemes. Finally, Section VI concludes the paper.

II. Reliable Broadcast Problem

In this section, we describe the formulation of the reliable broadcast problem, along with models and assumptions used in this paper.

A. Problem Formulation

For each node x in the system, we define *broadcast reliability* at node x, denoted by $r_x \in [0, 1]$, as a probability that x will receive a message m from the broadcast operation. Hence, assuming independence between messages in the system, r_x can be calculated as the fraction of overall distinct messages that x receives. We define the *average system reliability*, denoted by R, as the average of broadcast reliability values of all nodes in the system. That is, let V denote the set of all nodes in the system, $R = \frac{\sum_{v \in V} r_v}{|V|}$.

With such definitions, the reliable broadcast problem is the problem of finding a broadcast protocol such that each node x in the system receives its broadcast reliability r_x no less the reliability requirement R^*, where R^* is a predetermined value based on the application and priority of each message, allowing per-message reliability differentiation. For example, messages containing periodic temperature sensor readings may require $R^* = 0.75$ while critical emergency messages may require $R^* = 0.95$

B. System Model

The system operates on an arbitrary mobile wireless ad hoc network using standard CSMA/CA MAC layer protocols such as 802.11 DCF. Any pair of nodes can communicate to each other when the distance between them is less than the transmission range of the network interface. However, collision may occur when a node receives broadcast signals from multiple senders at the same time. Each node can move at arbitrary speed. Despite node mobility, we assume that network density is high enough to ensure network connectivity. Each node is cooperative and can be the broadcast source.

C. Design Goals

In this paper, we design a reliable broadcast protocol for mobile ad hoc networks in order to satisfy the following goals.

Adjustable Broadcast Reliability: As mentioned in Section II-A, the primary goal of the our broadcast protocol is to provide a broadcast service to the system such that each mobile node x receives broadcast r_x no less than to the reliability requirement R^*.

Minimized Broadcast Overhead: The overhead of broadcasting a message m is equal to the fraction of forwarding nodes in the system that forward the message m. Our reliable broadcast protocol is aimed to minimize the broadcast overhead as much as possible while achieving the required level of broadcast reliability R^*.

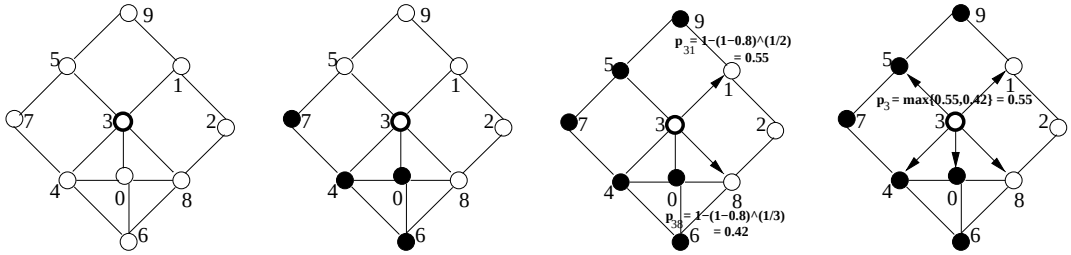
Resilience to Mobility and Packet Loss: In mobile ad hoc networks, each node can move arbitrarily with variable speed, ranging from typical human speed (i.e. walking human) to high speed (i.e. running vehicles). Hence, the topology of the network can change frequently over time. Moreover, packet loss due to collision and contention can be another factor to degrade performance of the broadcast system. Another goal of our reliable broadcast protocol is to sustain the reliability requirement R^* regardless of node mobility and packet loss.

In order to satisfy all goals mentioned, we propose a hybrid probabilistic/deterministic reliable broadcast protocol. In Section III, we first propose a novel pure probabilistic broadcast protocol that can achieve the specified reliability requirement R^* even under high mobility. In Section IV-B, we then augment our pure probabilistic broadcast with a well-known deterministic broadcast protocol to reduce overhead.

III. Mobility-aware Probabilistic Broadcast Protocol

This section describes the detail of the proposed topology-aware probabilistic broadcast protocol. First, the basic concept of the proposed broadcast protocol in order to achieve the application-specified reliability requirement, denoted by R^*, is described in Section III-A with the assumption of no stale topology information and no packet loss. An extension to the basic protocol to address node mobility and packet loss is then described in Section III-B.

A. Basic Protocol



(a) Node 3's neighbor information

(b) Node 3 hears a message from node 4

(c) Node 3 hears the duplicated message from node 5, and calculates rebroadcast probability for each of the remaining nodes

(d) Node 3 computes the overall rebroadcast probability

Fig. 1. Example of the topology-aware probabilistic broadcast when the reliability requirement $R^* = 0.8$

The basic idea of the protocol is as follows. Each node in the system maintains the information of its 2-hop neighbors. Such information is obtained by periodically exchanging beacon messages among neighbors. The 2-hop neighbor information provides each node sufficient topology information while sustaining node mobility. The overhead from the beacon message can be further reduced as follows. Under high traffic, the periodic beacon messages can be piggybacked with data messages. Under low traffic, each node can increase the length of the beacon interval to reduce excessive overhead. The detail of such optimization, however, is considered as a future work and beyond the scope of this paper.

Whenever a source node wants to broadcast a new message to the other nodes in the system, it broadcasts that packet *once*. When a node x hears a new message m, it delivers m to the upper-layer application. At the same time, x decides to retransmit the message m with certain probability p_x or not retransmit m with probability $1 - p_x$. The rebroadcast probability p_x is calculated locally at node x such that x can satisfy reliability requirement of all x's 1-hop neighbors. The calculation of p_x relies on x's 2-hop neighbor information as follows.

When a node x receives a new message m for the first time from one of its 1-hop neighbors y, x starts a timer with random delay to avoid collisions.[1] Node x also *marks* that y has already received packet m (since x received m from y). Node x also marks all y's 1-hop neighbors, since those y's 1-hop neighbors must have already received the broadcast message m from y's already, assuming no collision and up-to-date neighborhood information[2]. Since x has 2-hop neighbor information and y is a 1-hop neighbor of x, x will know the set of y's 1-hop neighbors as well. While the timer is running, if x receives more duplicate copies of m from other 1-hop neighbors, it also marks all subsequent senders along with their 1-hop neighbors. When the timer expires, for each x's *unmarked* 1-hop neighbor z, x then calculates p_{xz}, which is the probability that x should rebroadcast in order to provide enough broadcast reliability to z. Let r_z denote the reliability at node z (i.e. the probability that z receives m), which must be at least equal to the reliability requirement level R^*. Let N_{xz} denote the number of z's 1-hop neighbor that *have already been marked* by x during the timer phase, plus x itself. The calculation of p_{xz} is done as follows.

[1]The maximum delay of the timer can be set to be large enough to avoid collision, but small enough to ensure timely delivery of the message. One possibility is to set the maximum delay to the delay requirement of the message, divided by the network diameter.

[2]The effect of collision and stale membership information will be addressed in Section III-B

$$
\begin{aligned}
r_z &= P[z \text{ receives at least one copy of } m] \\
&= 1 - P[z \text{ receives no copy}] \\
&= 1 - (1 - p_{xz})^{N_{xz}} \\
&\geq R^* \\
p_{xz} &\geq 1 - (1 - R^*)^{\frac{1}{N_{xz}}} \qquad (1)
\end{aligned}
$$

That is, p_{xz} must be set, in conjunction with other potential forwarding nodes, to satisfy the reliability requirement of z, which is equal to the application-specified requirement R^*. The number of z's 1-hop neighbors that are already marked, denoted by N_{xz} is known to x based on x's 2-hop neighbor information and the marking process during the timer phase. Hence, p_{xz} can be calculated at x for each of its *unmarked* 1-hop neighbors z. Note that if $N_{xz} \leq 1$, then $p_{xz} = R^*$ due to the fact that x is the only potential forwarding node for z. Finally, node x's overall rebroadcast probability p_x is then calculated as the maximum value of p_{xz}. That is, for each $z \in x$'s unmarked 1-hop neighbor set,

$$p_x = \max_{z \in \{x\text{'s unmarked 1-hop neighbors}\}} p_{xz}$$

With this calculated rebroadcast probability p_x, x will rebroadcast the packet m such that it can satisfy reliability requirement of all remaining 1-hop neighbors that have not received the message m. Hence, x will rebroadcast the message m with probability p_x or drop the message m with probability $1 - p_x$. A node x will make forwarding decision for each distinct message m *only once*. That is, after making the decision, if x receives more duplicate copies of m, x will drop the subsequent copies of m automatically.

Figure 1 shows an example of the topology-aware probabilistic broadcast at node 3. First, node 3 maintains 2-hop neighbor information as illustrated in Figure 1(a). When node 3 receives a message m from node 4 (Figure 1(b)), it marks node 4, along with node 4's 1-hop neighbors (node 0,6,7) as they have received m from node 4. At the same time, node 3 also starts a counter with random delay for the message m. While the counter is running, node 3 just happens to hear another copy of m from node 5 (Figure 1(c)). Node 3 then marks node 5 and all of node 5's neighbors as well. When the counter expires, node 3 then calculates the corresponding rebroadcast probability for each of node 3's 1-hop neighbor that remains unmarked (node 1 and node 8). For node 1, there are 2 potential forwarders ($N_{31} = 2$), which are node 3 and node 9. Hence, each of node 1's potential forwarders need to forward m with probability $p_{31} = 1 - (1 - 0.8)^{\frac{1}{2}} \approx 0.55$ in order to satisfy reliability requirement $R^* = 0.8$ at node 1. For node 8, there are 3 potential forwarders ($N_{38} = 3$), which are node 0, node 3, and node 6. With the same calculation, each of node 8's potential forwarders, including node 3, need to reforward m with probability $p_{38} \approx 0.42$. Finally, node 3 then calculates its overall rebroadcast probability p_3 to be the maximum value of all individual rebroadcast probability values (Figure 1(d)). That is, $p_3 = \max(0.42, 0.55) = 0.55$, which means node 3 will rebroadcast m with probability 0.55 in order to satisfy reliability requirement at all of node 3's 1-hop neighbors.

With the rebroadcast probability calculation algorithm presented above, the probabilistic broadcast protocol can dynamically and locally adjust the broadcast probability such that each node in the system achieves the reliability requirement R^*, which is specified by the application. However, the basic topology-aware probabilistic broadcast does not address packet collision and high mobility. Also, it incurs comparatively larger overhead than deterministic broadcast protocols.

B. Handling Mobility and Packet Loss

To address packet losses and high mobility that may decrease the broadcast reliability in the proposed probabilistic protocol, we extends our basic probabilistic broadcast protocol as follows. The extended protocol is almost the same as the basic protocol described in Section III-A. However, when a node x calculates the retransmit probability in order to satisfy reliability requirement of its 1-hop neighbor node z (i.e. Equation (1)), the extended protocol incorporates two adjustment variables into Equation (1), α and β as follows.

$$
\begin{aligned}
r_z &= 1 - (1 - \alpha_z p_{xz})^{\beta_z N_{xz}} \\
&\geq R^* \\
p_{xz} &\geq \frac{1 - (1 - R^*)^{\frac{1}{\beta_z N_{xz}}}}{\alpha_z} \qquad (2)
\end{aligned}
$$

The variable $\alpha_z \in [0, 1]$ and $\beta_z \in [0, 1]$ are *channel quality* and *neighbor stability* at node z respectively. The channel quality at node z, denoted by α_z, is defined as the probability that z will receive a packet transmitted by its neighbors successfully.

The lower α_z is, the more lossy the channel at z is. The neighbor stability at node z, denoted by β_z, is defined as the fraction of z's 1-hop neighbor information that is still valid over time Δt[3]. That is, the lower β_z is, the more quickly the neighbor information at node z becomes stale. Both α_z and β_z variables are added into Equation (1), resulting in Equation (2), which gives better adjustment to imperfect channel condition and node mobility. Each node x in the system will periodically estimate its local channel quality and neighbor stability value α_x and β_x, and piggybacks the two values in its beacon message. With such approach, each node in the system will know the channel quality and neighbor stability of itself and its neighbors as well so that it can calculate the forwarding probability accurately.

Estimating Channel Quality (α): Each node x can simply estimate its channel quality (α_x) by having each node assigning a sequence number to each data message it transmits. By looking at the sequence number piggybacked in each received data message, a node x can calculate the total of number of data messages each of its neighbors has transmitted over a period of time. Each node x also keeps the record of number of data messages it successfully received from each of its neighbors over time as well. With such information, node x can estimate its local channel quality over time as follows.

$$\begin{aligned} \alpha_x &= \frac{\text{total \#messages } x \text{ received from its neighbors}}{\text{total \#messages transmitted by } x\text{'s neighbors}} \\ &= \frac{\sum_z \text{\#messages } x \text{ received from } z}{\sum_z \text{\#messages } z \text{ has transmitted}} \end{aligned} \tag{3}$$

With equation (3), each node x can periodically calculate and report its α_x to its neighbor via its beacon messages. The period each node x calculates α_x can be adjusted to suit channel quality fluctuation and node mobility. In the experiment, each x re-calculates its α_x whenever x receives more than k new distinct messages and the time since the last calculation of α_x is more than the beacon interval.

Estimating Neighbor Stability (β): The neighbor stability at each node x, denoted by β_x, represents the changing rate of x's neighbor set. Such value can be calculated periodically and locally at each node x by calculating the fraction of its remaining 1-hop neighbors over time as follows.

$$\begin{aligned} \beta_x &= \frac{\text{fraction of } x\text{'s remaining neighbors over time } \triangle t}{\triangle t} \\ &= \frac{|\{x\text{'s remaining neighbors at } t+\triangle t\}|}{|\{x\text{'s neighbors at } t\}|.\triangle t} \\ &= \frac{|\{x\text{'s neighbors at } t\} \cap \{x\text{'s neighbors at } t+\triangle t\}|}{|\{x\text{'s neighbors at } t\}|.\triangle t} \end{aligned}$$

The period $\triangle t$ can be set, depending on how reactive each node would be to system mobility. In the experiment in this paper, the period $\triangle t$ is set to the beacon interval.

With the proposed extension, the system can adjust its forwarding probability based on the estimated channel condition and node mobility and satisfy reliability requirement R^* of any message m. However, the probabilistic nature of the approach tends to cause redundant retransmission and incurs more overhead as compared to pure deterministic broadcast schemes.

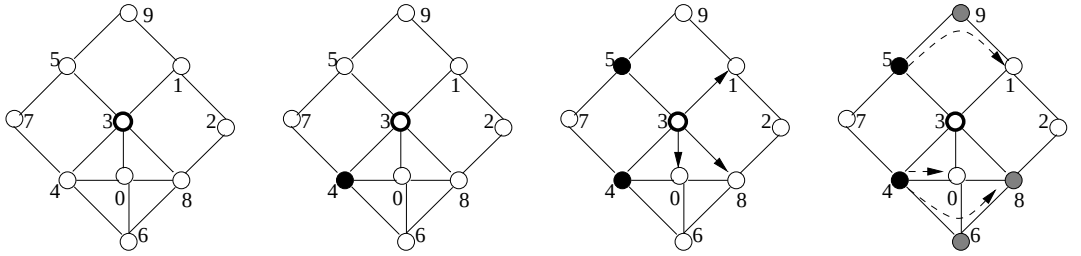
IV. HybridCast

In this section, we discuss the hybrid probabilistic/deterministic broadcast protocol called HybridCast. First, we present in Section IV-A an existing deterministic broadcast algorithm to be combined with our proposed probabilistic broadcast protocol from Section III-A. The details of HybridCast, which is the combination between the two schemes, is then presented in Section IV-B.

A. Deterministic Broadcast Algorithm

This section describes the existing deterministic broadcast protocol to be combined with our proposed probabilistic broadcast. We choose to use Dai and Wu's deterministic broadcast algorithm with the self-pruning rule [4], since the protocol can operate on 2-hop neighbor information and hence can share the data structure with the proposed probabilistic protocol.

[3]In this paper, Δt is equal to the node beacon interval.



(a) Node 3's 2-hop topology

(b) Node 3 after receiving a message from node 4

(c) Node 3 after receiving a duplicate message from node 5

(d) Node 3 does not need to rebroadcast

Fig. 2. Example of Dai and Wu's deterministic, self-pruning broadcast algorithm

Like our proposed probabilistic broadcast protocol, Dai and Wu's deterministic broadcast protocol assumes each node to have 2-hop neighbor information[4] obtained via periodic beacon messages. When a node x receives a new broadcast message m from node y, it starts the timer with a random delay for packet m and marks the sender node y as *BLACK node*. While the timer is running, if x receives any duplicate copy of m from another node z, x will also mark z as BLACK node as well. When the timer expires, x will decide whether it should rebroadcast m or not based on the following condition. x will rebroadcast m if and only if all x's non-BLACK 1-hop neighbors are connected to at least one BLACK node via a path consisting of only nodes with id higher than x (i.e. via a path consisting of only GRAY nodes). Figure 2 shows an example of Dai and Wu's algorithm where node 3 receives two copies of message m from node 4 and node 5 before the timer expires. With node 3's 2-hop neighbor information, node 3 does not need to retransmit since node 3's remaining non-BLACK 1-hop neighbors (i.e. node 0,1,8) are connected to at least one BLACK nodes via paths containing only GRAY nodes (i.e. "5-9-1","4-0","4-6-8" respectively).

Under no mobility and perfect channel condition, Dai and Wu's deterministic broadcast protocol has been proved to deliver each message to all nodes in the system with only small number of forwarding nodes [4]. However, the reliability performance is affected significantly as mobility increases.

B. Hybrid Deterministic/Probabilistic Broadcast Protocol

To achieve controllable broadcast reliability with low overhead and high resilience with mobility, this section describes HybridCast protocol that combines the mobility-resistance property of the proposed probabilistic protocol (Section III-A and III-B) with the low overhead property of Dai-Wu's deterministic protocol (Section IV-A). The basic concept of HybridCast protocol is to use the deterministic broadcast if possible to minimize overhead. However, if the deterministic scheme cannot achieve the application-specified reliability R^* due to packet loss or high mobility, HybridCast will instead use the probabilistic broadcast to fill the reliability gap.

In HybridCast protocol, when a source node would like to broadcast a new message m to the system, the source node have two possible modes.

- with probability γ, the source node initiates the broadcast of the message m in *deterministic mode*. Any node that receives m will follow the deterministic protocol described in Section IV-A.
- with probability $1-\gamma$, the source node initiates the broadcast of the message m in *probabilistic mode* with the message reliability requirement R_p^*. Any subsequent node that receives m will follow the probabilistic protocol described in Section III in order to achieve the reliability requirement R_p^* of the message m.

Note that switching probability γ and the probabilistic-mode reliability requirement R_p^* must be chosen such that overall broadcast reliability of the combined schemes is at least equal to the original requirement R^* while trying to reduce broadcast overhead as much as possible. Specifically, γ and R_p^* can be obtained from the following optimization problem.

$$\text{minimize } O = \gamma O_d + (1-\gamma)O_p$$

$$\text{subject to } R = \gamma R_d + (1-\gamma)R_p^* \geq R^*$$

where O and R are average system broadcast overhead and reliability achieved by HybridCast protocol respectively. The variable O_d and R_d denote the average system broadcast overhead and reliability achieved by using the deterministic mode.

[4]Dai and Wu's protocol also works with any k-hop neighbor information. In our paper, we use $k=2$ as it gives good pruning results with acceptable background overhead.

Parameters	Value
Area size	1000m x 1000m
Transmission range	150m
Bandwidth Capacity	1 Mbps
#Nodes (Avg. #1-hop neighbors)	60 - 200 (4 - 14)
#Sources	5
Speed	1 m/s - 20 m/s
Load	5 kbps
Message size	64 bytes
Beacon interval	5 seconds
Simulation Time	900 seconds
#Runs	5

TABLE I
SIMULATION PARAMETERS

The variable O_p is the broadcast overhead incurred by using the probabilistic mode *in order to guarantee* reliability requirement R_p^*. Note that O_p is a function of R_p^* estimated from Equation (2). The variable O_d and R_d can also be estimated locally at each node x based on its 2-hop neighbor information in the same way with the calculation of the channel quality α_x as follows. To estimate R_d, each node keeps track of the *deterministic-mode* messages it has received so far. To estimate O_d, each node keeps track of the *deterministic-mode* messages it has *transmitted* so far. Let N_d be the total *deterministic-mode* messages transmitted in the system so far. Hence, R_d and O_d can be calculated locally as the fraction of the deterministic-mode messages it has received and transmitted so far out of N_d.

V. EXPERIMENTAL RESULTS

We evaluate the performance of HybridCast via simulations using NS-2 network simulator version 2.30. The parameters used in the simulations are shown in Table I. Unless otherwise specified, the value in each parenthesis in Table I is used as the default parameter value. The experiment consists of 60 to 200 nodes ($\approx$ 4 to 14 neighbors per node). In the experiment, we compare the performance of three broadcast protocols–the purely probabilistic broadcast proposed in Section III, Dai-Wu's deterministic broadcast protocol mentioned in Section IV-A, and the proposed HybridCast protocol in Section IV-B. We also presents the result of the Null-MAC flooding scheme as the upper-bound of achievable broadcast reliability. All protocols (except the Null-MAC flooding) rely on 2-hop information between neighbor nodes. For all protocols, we use *exactly* the same general settings such as maximum packet timer delay and beacon message exchange frequency.

A. Effect of Reliability Requirement (R^*)

Figure 3(a) shows the average broadcast reliability of the system with 100 nodes moving with 5 m/s maximum speed. According to the figure, almost all protocols can satisfy the reliability requirement (R^*) in most scenarios. However, the purely deterministic protocol cannot achieve reliability requirement $R^* > 0.9$, while the probabilistic can perform above the baseline requirement and hybrid schemes performs slightly under the baseline requirement.

Figure 4(a) shows the broadcast overhead in terms of the fraction of broadcast forwarding nodes in the system with 100 nodes moving with 5 m/s maximum speed. It can be seen that the purely probabilistic protocol incurs larger overhead, especially when the reliability requirement approaches 1.0. On the other hand, the purely deterministic approach incurs small but constant overhead in all scenarios since there is no dynamic adaptation scheme. Finally, HybridCast incurs the same level overhead as the deterministic protocol in most scenarios except when the deterministic protocol cannot achieve the system requirement $R*$. In such scenario, HybridCast incurs slightly more overhead than the deterministic protocol due to its adaptation towards the probabilistic protocol.

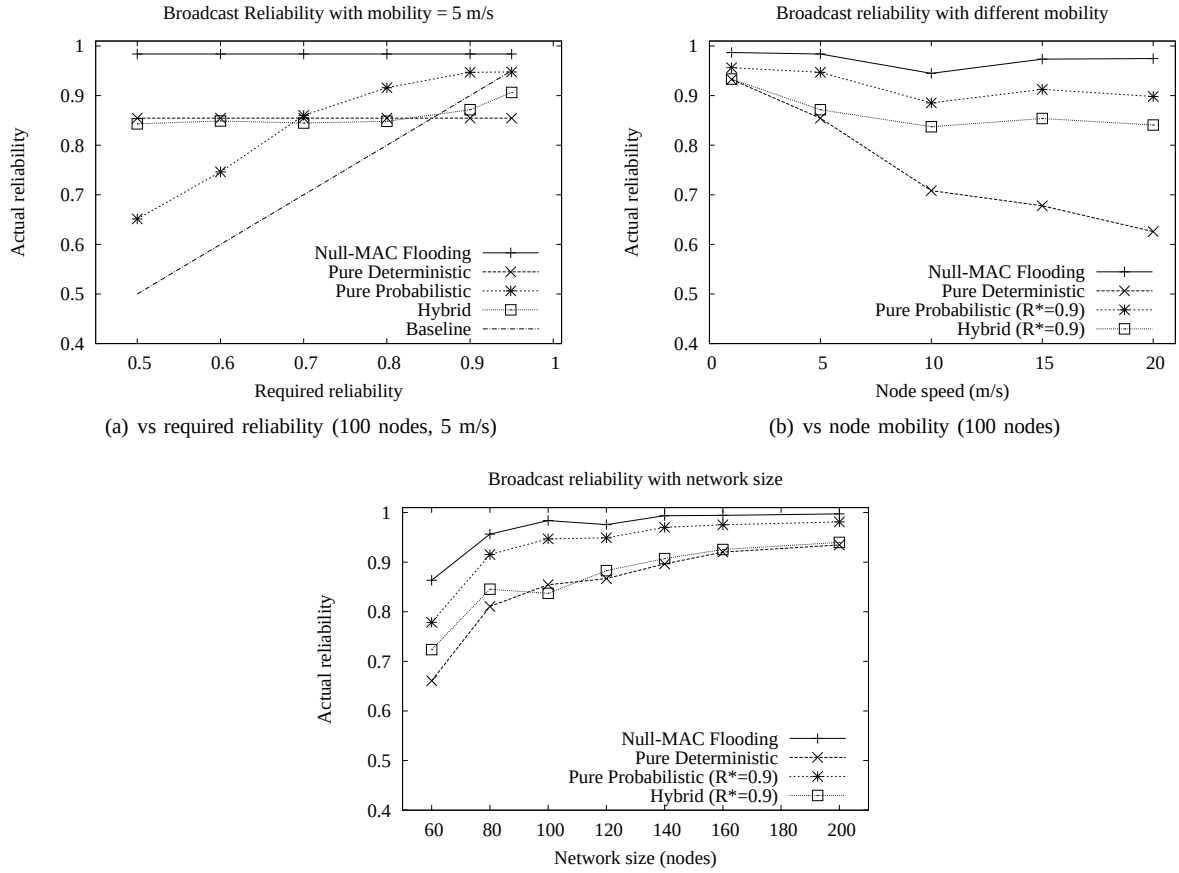
B. Effect of Mobility

Figure 3(b) and Figure 4(b) presents the achieved reliability and overhead of different protocols under different mobility levels respectively. It can be seen that the reliability of the purely probabilistic protocol does not drop when mobility increases. On the other hand, the purely deterministic scheme suffers from mobility as expected. HybridCast also suffers from high mobility to a smaller degree due to the fact that the probabilistic part helps alleviate the mobility problem.

Figure 4(b) shows the fraction of forwarding nodes of each protocol under different mobility. The result shows that HybridCast adapts from the deterministic scheme towards the probabilistic scheme as mobility level increases.

C. Effect of Network Size

Figure 3(b) and Figure 4(b) presents the achieved reliability and overhead of different protocols under different system size with maximum speed 5 m/s. As system size grows, the reliability of all schemes increases and the overhead of all schemes



(a) vs required reliability (100 nodes, 5 m/s)

(b) vs node mobility (100 nodes)

(c) vs system size (5 m/s)

Fig. 3. Average system reliability

decrease due to higher network connectivity and more spatial reuse. Again, HybridCast tries to adapt itself to reduce overhead while maintaining the broadcast reliability to the level required by the system.

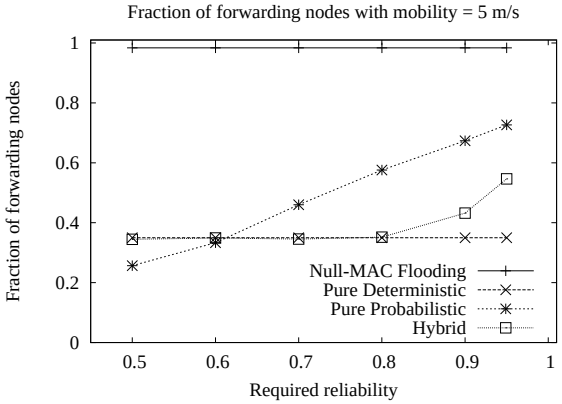
D. Effect of Traffic Intensity

Figure 5(a) and Figure 5(b) shows average node reliability and fraction of forwarding nodes respectively in the system of 100 nodes with maximum speed 10 m/s under different traffic load. As seen from the figures, no protocol is able to achieve the reliability requirement when the system is overloaded due to packet contention and collision.

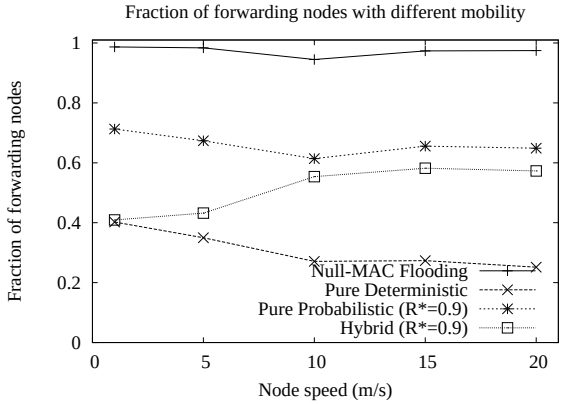
Another observation is that all protocols, especially the pure probabilistic protocol and Hybridcast protocol, converge to the same performance in overloaded traffic. This is due to the fact that all protocols do not have the capability to detect packet collision. Instead, they consider collided packets as lost packets, which in turn trigger all protocols to forward more packets in order to boost reliability. However, such adjustments further increase load and hence reduce the performance of the system. One solution to achieve reliability in overloaded system is to incorporate packet collision and contention due to system overload into the forwarding probability calculation, which is considered as the future work of this paper.

E. Effect of Mobility Compensation

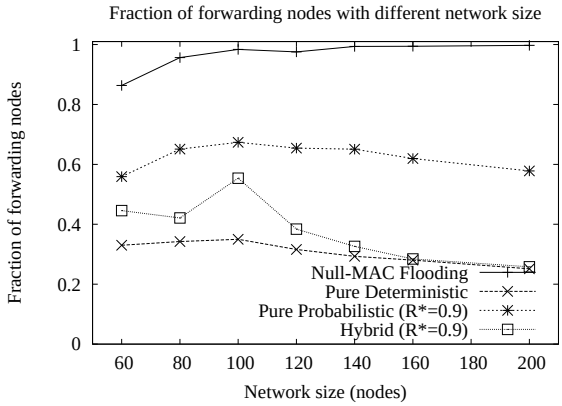
To measure the effectiveness of the compensation mechanism presented in Section III-B, the comparison between the protocols without compensation mechanism and the protocols with compensation mechanism is shown in Figure 6(a) and 6(b). As seen from Figure 6(a), pure probabilistic broadcast and Hybridcast protocol without compensation mechanism suffer from mobility at higher level than the ones with compensation mechanism. This is expected because the protocols without compensation mechanism do not account node mobility and channel condition when calculating forwarding probability. However, as shown in Figure 6(b), mobility resilience comes with the price of additional overhead, as it requires more packet forwarders in order to achieve reliability requirement when node mobility increases.



(a) vs required reliability (100 nodes, 5 m/s)

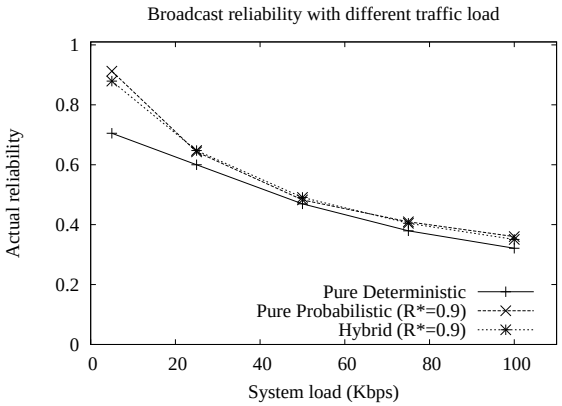


(b) vs node mobility (100 nodes)

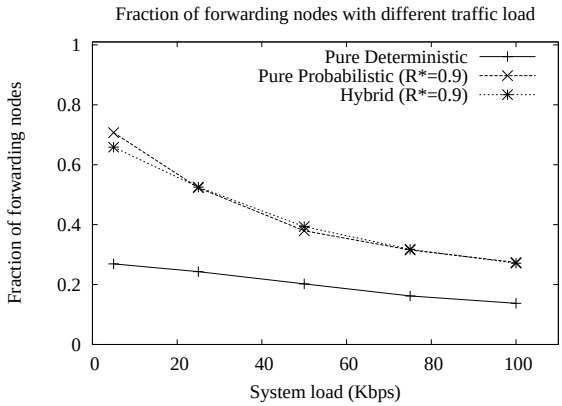


(c) vs system size (5 m/s)

Fig. 4. Fraction of forwarding nodes

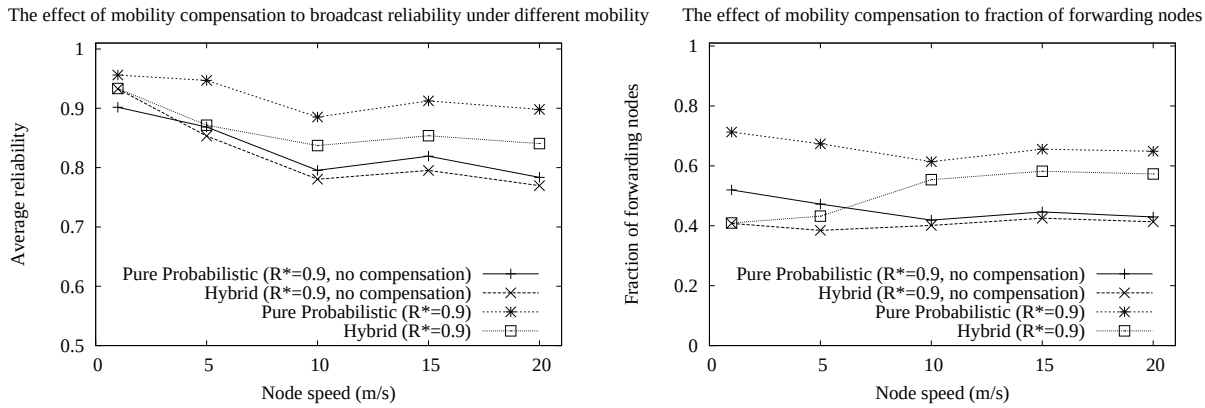


(a) Average system reliability



(b) Fraction of forwarding nodes

Fig. 5. Effect of traffic intensity to system performance (100 nodes at 10 m/s speed)



(a) Average system reliability

(b) Fraction of forwarding nodes

Fig. 6. Effect of mobility compensation to system performance (100 nodes)

VI. Conclusions

This paper discussed the problem of adjustable broadcast reliability in mobile wireless ad hoc networks. The paper first proposed a topology-aware, mobility-resistant probabilistic broadcast protocol with automatic adaptation to satisfy broadcast reliability level required by the application even under high mobility or packet loss. In order to reduce broadcast overhead, the paper combined the proposed probabilistic broadcast protocol with an existing deterministic broadcast protocol, resulting in the hybrid probabilistic/deterministic broadcast called HybridCast. Finally, the simulation results showed that the HybridCast protocol can achieve reliability requirement under different mobility and generally incurs less overhead than the pure probabilistic protocol.

References

[1] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Wirel. Netw.*, vol. 8, no. 2/3, pp. 153–167, 2002.

[2] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," *Mob. Netw. Appl.*, vol. 9, no. 2, pp. 141–149, 2004.

[3] Y. Wang, W. Wang, and X.-Y. Li, "Distributed Low-cost Backbone Formation for Wireless Ad Hoc Networks," in *Proc ACM MobiHoc '05*. New York, NY, USA: ACM, 2005, pp. 2–13.

[4] F. Dai and J. Wu, "Performance Analysis of Broadcast Protocols in Ad Hoc Networks Based on Self-Pruning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 1027–1040, 2004.

[5] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," in *Proc. ACM MobiHoc*, 2002, pp. 194–205.

[6] P. Kyasanur, R. R. Choudhury, and I. Gupta, "Smart Gossip: an Adaptive Gossip-based Broadcasting Service for Sensor Networks," in *Proc IEEE MASS'06*, 2006, pp. 91–100.

[7] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit Disk Graphs," *Discrete Mathematics*, vol. 86, no. 1-3, pp. 165–177, December 1990.

[8] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic Broadcast for Flooding in Wireless Mobile Ad Hoc Networks," in *Proc IEEE WCNC'03*, 2003, pp. 1124–1130.