Computer Communications 35 (2012) 500–506

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A probabilistic and opportunistic flooding algorithm in wireless sensor networks

Dukhyun Chang [a], Kideok Cho [a], Nakjung Choi [b], Ted "Taekyoung" Kwon [a,*], Yanghee Choi [a]

[a] School of Computer Science and Engineering, Seoul National University, Seoul 151-742, Republic of Korea
[b] Alcatel-Lucent, Bell-Labs Seoul Ltd, Seoul 120-270, Republic of Korea

ARTICLE INFO

Article history:
Received 23 February 2011
Received in revised form 17 November 2011
Accepted 21 November 2011
Available online 28 November 2011

Keywords:
Flooding
Wireless sensor networks
Probabilistic
Opportunistic
Target reliability

ABSTRACT

In wireless sensor networks, many communication protocols and applications rely on flooding for various networking purposes. Prior efforts focus on how to design efficient flooding algorithms; that is, they seek to achieve full reliability while reducing the number of redundant broadcasting across the network. To achieve efficient flooding, most of the existing protocols try to reduce the number of transmissions, which is decided without considering any online transmission *result*. In this paper, we propose a probabilistic and opportunistic flooding algorithm that controls rebroadcasts and retransmissions opportunistically. It seeks to achieve a target reliability required by an application. For this purpose, it makes a given node select only the subset of its one-hop neighbors to rebroadcast the same message. It considers node relations such as link error rates among nodes in selecting eligible neighbors to rebroadcast. The sender controls the number of retransmissions opportunistically by tracking the current status of message reception at its neighbors. Simulation is carried out to reveal that our proposed scheme achieves the given target reliability with less overhead than other flooding algorithms in most cases, thus prolonging the network lifetime.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Flooding is one of key mechanisms that are widely used in various wireless networks. It propagates a message throughout a network for various purposes. Especially, flooding is usually leveraged to establish a route to the destination for unicast routing (e.g. AODV [1], DSR [2]). Similarly, when a node should inform other nodes of its link state, its latest link information is flooded across the network (e.g. OLSR [3]). Due to its viability, diverse flooding algorithms have been proposed in various wireless networks including wireless sensor networks (WSNs).

Since the objective of flooding is to make it sure that all the nodes in a network receive the same message, flooding is generally performed by making all the nodes rebroadcast the received message. However, this becomes inefficient as the node density increases, which is a typical case in WSNs. Another issue is that it is hard to achieve high reliability because wireless links generally suffer from high error rates. Thus, to achieve high reliability, retransmissions are often exploited. It is crucial to decide which node to rebroadcast and how many times to retransmit the message in a flooding mechanism, since the rebroadcasting of too many nodes and/or redundant retransmissions may cause traffic implosion [4], which leads to unreliability and energy inefficiency. Prior studies have proposed several flooding schemes that seek to achieve high reliability while reducing redundant traffic by controlling the number of broadcasts.[1] However, the existing approaches have not considered the effect of a transmission (or a retransmission) of a given node on the message reception by its neighbor nodes quantitatively.

Furthermore, in wireless sensor networks, the network-wide full reliability[2] may not always be required according to the application requirements. For example, many sensor network applications such as temperature monitoring or intrusion detection system deploy many sensors redundantly to cover the monitoring area for high reliability [5,6]. In this situation, a sink may want to disseminate a query with *partial reliability*. If the sink can achieve its own purpose only with *R*% of sensors, it may want to disseminate the query to only *R*% of sensors to reduce the number of rebroadcasts and thus energy consumption. Therefore, supporting flooding with partial reliability is another important technique to prolong lifetime of the

* Corresponding author. Tel.: +82 2 880 9105; fax: +82 2 872 2045.
E-mail addresses: dhchang@mmlab.snu.ac.kr (D. Chang), kdcho@mmlab.snu.ac.kr (K. Cho), nakjung.choi@alcatel-lucent.com (N. Choi), tkkwon@snu.ac.kr (T."Taekyoung" Kwon), yhchoi@snu.ac.kr (Y. Choi).

[1] For clarity purposes, we use the following definition in this paper. The number of 'broadcasts' is the total number of transmissions of the same message throughout the network. The number of 'rebroadcasts' means how many nodes in the network have rebroadcasted the same message; thus, a node who transmits the message multiple times is counted only once. The number of 'transmissions' of a given node is how many times the node has transmitted the same message.

[2] In reality, a network-wide 100% reliability is often infeasible; thus, we target a sufficiently high reliability (say 95%) that can satisfy the application requirements. In this paper, the full reliability refers to a sufficiently high reliability.

0140-3664/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2011.11.016

sensor network.

Some schemes aiming to provide partial reliability in WSNs have been proposed in various contexts. For example, MMSPEED [7] seeks to deliver unicast packets with partial reliability required by applications in a decentralized and probabilistic fashion. However, MMSPEED only deals with unicast flows and does not consider the reliability of flooding. In addition to the partial reliability, GARUDA [8] considers a few other semantics of reliability. For example, sensor network applications may necessitate reliable delivery to sensors such that the entire sensing field is covered, not to all the sensors in the field. ESRT [9] redefines "reliability" somewhat differently. ESRT first assumes that sensory data packets are periodically reported from sensors to the sink. During a session, depending on the level of network congestion, the sensors can adjust the reporting rate (or "reliability") to adapt to the network traffic load. However, the meaning of reliability in ESRT is different from our definition of partial reliability in this paper. To the best of our knowledge, how to support flooding with a target reliability has been missing in the literature.

In this paper, we seek to achieve a target reliability given by an application, ranging from full to partial reliability, while minimizing the number of broadcasts in a probabilistic and opportunistic manner. In other words, each node selects the subset of its one-hop neighbors, that will rebroadcast the same message by considering link error rates among the node itself, its one-hop neighbors and its two-hop neighbors for the target reliability. After a source node or a rebroadcasting node transmits a message once, it decides to retransmit or not by estimating the locally achieved reliability probabilistically and opportunistically. We note that OLSR tries to minimize the number of transmissions in flooding by making only selected nodes (called multi-point relays) rebroadcast the message. We extend the notion of multi-point relays (MPRs) to control flooding to achieve a target reliability required by applications. That is, depending on the target reliability and the link error rates[3] of neighbors, the set of MPRs of a sender will be dynamically adjusted.

The rest of this paper is organized as follows. Prior studies are discussed in Section 2. In Section 3, we propose a novel flooding algorithm, called POFA. Simulation results are shown in Section 4. Finally, Section 5 concludes this paper.

2. Related work

OLSR [3] is a proactive routing protocol for mobile ad hoc networks. OLSR relies on flooding to disseminate each node's local link information throughout the network to help other nodes build/update their routing tables. If every node participates in flooding, its signaling overhead would be substantial. Hence, OLSR seeks to minimize the number of broadcasts[4] by making only selected nodes (called as multipoint relays) rebroadcast the routing message. For this purpose, each node designates the subset of its one-hop neighbors as multipoint relays (MPRs), so that MPRs' rebroadcasting the message will reach all of its two-hop neighbors. However, in selecting MPRs, OLSR considers only coverage; link error rates and reliability level are not major concerns. One of reasons is that routing messages are periodically disseminated. Overall, OLSR is not adequate for broadcasting applications that require a target reliability.

As for flooding, numerous algorithms have been proposed to improve reliability or reduce redundancy or both. In gossip-based routing [10] and probabilistic broadcasting schemes [11], a node rebroadcasts messages with a certain probability, say r. By adjusting r, gossip-based routing tries to reduce the number of broadcasts in the network layer, while probabilistic broadcasting focuses on reducing both collisions and energy consumption in the MAC layer. Although these protocols effectively reduce the number of broadcasts, it is difficult to decide r to achieve the given target reliability.

RBP [12] improves reliability by controlling the number of retransmissions carefully. Each node rebroadcasts a received message at least once, and then decides whether to retransmit the received message or not by comparing the number of received (implicit or explicit) ACKs with some threshold, which is determined based on the number of its one-hop neighbors. If there are many one-hop neighbors, the probability of rebroadcasting by other nodes is high, which makes the threshold smaller. However, as the node density increases, making every node rebroadcast at least once will become inefficient.

RAFA [13] extends RBP in the sense that it takes the network topology further into account. RAFA notices that two nodes with the same number of one-hop neighbors can have distinct connectivity patterns among their respective one-hop neighbors. That is, rebroadcasting a message from a one-hop neighbor may effect other one-hop neighbors. In RAFA, connectivity among one-hop neighbors is taken into consideration to further reduce the number of retransmissions. In other words, it decides whether to retransmit the received message by comparing an expected reliability of 1-hop and 2-hop neighbors with a threshold. However, also in RAFA, every node rebroadcasts at least once, which is inappropriate to satisfy partial target reliability efficiently, although a threshold is adjusted by a target reliability. That is shown in the numerical results of this paper.

LAF [14] and BPS [15] are flooding protocols for WSNs, which leverage the locations of sensors to flood packets efficiently. LAF divides sensor nodes into virtual grids depending on their positions. In each grid, there is a gateway node responsible for forwarding messages across virtual grids. And if messages are relevant to a given grid, the gateway node will forward the messages to other sensor nodes within the given grid. In BPS, only a few nodes which cover all sensor nodes are selected to forward messages by exploiting their location information. Even though these protocols reduce the number of broadcasts, they require obtaining location information for every sensor node.

3. Probabilistic and opportunistic flooding algorithm (POFA)

In this section, we explain a probabilistic and opportunistic flooding algorithm (POFA) that reduces the number of broadcasts while satisfying the given target reliability. In OLSR, every link is presumed to be error-free and the subset of one-hop neighbors that cover all of the two-hop neighbors is selected as MPRs from the viewpoint of a sender. By contrast, our proposed scheme assumes each link has its own link error rate. Thus, the sender is aware of (i) link error rates between its one-hop neighbors and itself, and (ii) link error rates between its one-hop and two-hop neighbors. We assume that the link error rates can be calculated based on periodic message exchanges between sensor nodes for neighbor discovery and/or synchronization [16] or in a similar way as [13,17,18]. Our assumptions stand on some researches for these issues. Woo et al. [17], Woo and Culler [19] proposed the stable estimation methods of the link error rate using moving averages which can be used even when the link state is changed rapidly. With the stable estimation of link error rate, the exchanges of link information between neighbor nodes need not to be occurred frequently. This implies that piggybacked flooding packets or periodic messages are sufficient to exchange the link state in POFA. In addition, other works proposed estimation methods which used RSSI, LQI or distances between neighbor nodes

[3] The link error rate refers to the probability that a single transmission is not successful between a pair of nodes in communication range and is often called packet error rate in the literature.

[4] In this paper, a broadcast means a single transmission of a message to one-hop neighbors, while flooding refers to network wide dissemination of a message through hop-by-hop broadcasting.

[18,20]. With these schemes, the link status information can be obtained without overhead if some information such as geographical topology is known a priori.

Suppose that the required network-wide reliability is R. The key idea behind POFA is that, on the receipt of a message, a node who should relay the received message tries to deliver the message to R of its one-hop and two-hop neighbors probabilistically. To this end, the node will select reliability-aware MPRs (RA-MPRs), which constitute a subset of one-hop neighbors. Each RA-MPR should rebroadcast the message. For the sake of brevity, one-hop and two-hop neighbors are collectively called "close neighbors" of the sender hereafter.

3.1. Expected delivery probability (EDP)

For sake of exposition, suppose that all of one-hop neighbors are selected as RA-MPRs in this subsection. First, a sender who is to broadcast a message (or to rebroadcast a received message) will calculate an expected delivery probability (EDP). The EDP is the ratio of the expected number of "close neighbors" that will receive the message probabilistically to the number of all "close neighbors". In Fig. 1, the black circle is the sender, s, and the white circles indicate the "close neighbors" of s. s will calculate the EDP as follows.

First of all, s should calculate EDP_m for each "close neighbor m." which is a probability that a particular neighbor m will receive the message, while the EDP is a collective ratio as defined in the above. For a one-hop neighbor i whose link error rate from s to i is l_{si}, the EDP_i is given by $1 - l_{si}$. Even though there could be longer paths from s to i, we take into account only a one-hop path. For a two-hop neighbor j, we have to consider two-hop paths from s to j as illustrated in Fig. 2; one path through node i_1 and another path through node i_2. Even though there could be longer paths from s to j as illustrated dotted lines, we take into account only two-hop paths. Then, for a close neighbor m, EDP_m is given by

$$EDP_m = \begin{cases} 1 - l_{sm}, & \text{if } m \in N_1; \\ 1 - \prod_{k \in N_1} [1 - (1 - l_{sk}) \cdot (1 - l_{km})], & \text{if } m \in N_2. \end{cases} \quad (1)$$

where l_{ab} indicates a link error rate of the link from node a to node b, and N_1, N_2 are the sets of one-hop and two-hop neighbors, respectively.

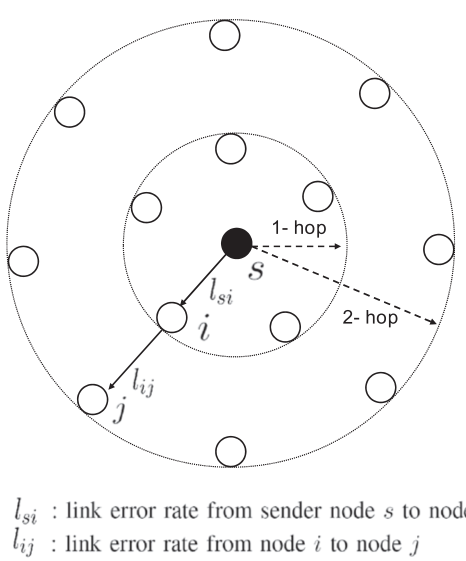

Fig. 1. Node s considers the link error rates of one-hop and two-hop neighbors in computing EDP.

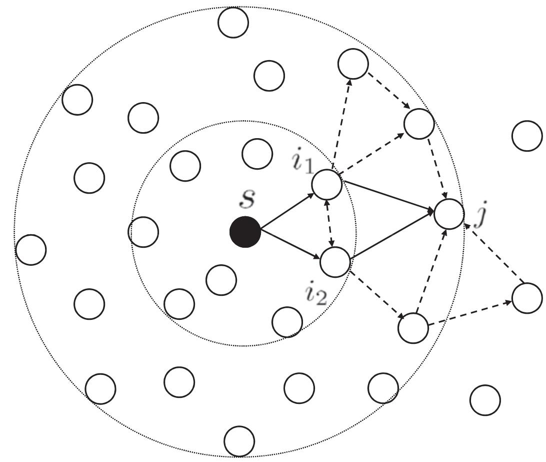

Fig. 2. Multiple paths can exist from s to a two-hop neighbor j.

If the EDP of s with a single broadcasting and all one-hop neighbors' broadcasting cannot make the EDP greater than target reliability, R, it should increment the number of transmissions for the same message. Suppose that s decides to transmit the message n times, then the EDP for one-hop neighbor i will be increased as n increases, which is given by $1 - l_{si}^n$. However, for a two-hop neighbor, s cannot guarantee that an intermediate one-hop neighbor will transmit the same message n times since intermediate node calculates its own EDP under its environments and decides the number of transmissions independently of node s. Thus, we make a conservative assumption that an intermediate one-hop neighbor will broadcast the message just once. Finally, when s transmits the message n times, the probability that a one-hop or two-hop neighbor m will receive the message is given by

$$EDP_m^n = \begin{cases} 1 - l_{sm}^n & \text{if } m \in N_1; \\ 1 - \prod_{k \in N_1} [1 - (1 - l_{sk}^n) \cdot (1 - l_{km})] & \text{if } m \in N_2. \end{cases} \quad (2)$$

3.2. Reliability-aware multipoint relay (RA-MPR) selection

This subsection describes how s selects its RA-MPRs among its one-hop neighbors. As mentioned in Section 2, a sender in OLSR selects rebroadcasting nodes, which are called MPRs. When s selects the MPRs among one-hop neighbors, it considers how to cover all of the two-hop neighbors. In POFA, a similar process is devised to select RA-MPRs. However, the goal of RA-MPR selection is to achieve a given target reliability, not to cover all of its two-hop neighbors. Each node selects its own RA-MPRs in a fully distributed manner with its local information like OLSR. When a node receives a message, it checks whether the node itself is an RA-MPR of the transmitter, and if so, it becomes the sender who should rebroadcast the message. Otherwise, it just receives the message and does not rebroadcast the message.

When selecting RA-MPRs, the EDP plays a key role. However, with RA-MPRs in mind, we should consider only RA-MPRs in one-hop neighbors (not all one-hop neighbors) in Eq. (2) since only RA-MPRs now rebroadcast the message.

$$EDP_m^n = \begin{cases} 1 - l_{sm}^n & \text{if } m \in N_1; \\ 1 - \prod_{k \in N_R} [1 - (1 - l_{sk}^n) \cdot (1 - l_{km})] & \text{if } m \in N_2. \end{cases} \quad (3)$$

where N_R is a set of the one-hop neighbors selected as RA-MPRs. Eq. (3) implies that, if all link error rates are the same, a one-hop neighbor covering more two-hop neighbors will have higher priority. When we count two-hop neighbors covered by a one-hop neighbor (or a RA-MPR candidate), we prefer two-hop neighbors which are not covered by the already-selected RA-MPRs.

Finally, the collective EDP with n transmissions from s (EDP^n) is calculated as follows:

$$EDP^n = \frac{\sum_{m \in N_1 \text{ or } N_2} EDP^n_m}{|N_1| + |N_2|} \quad (4)$$

where m is an arbitrary close neighbor and N_1 and N_2 are the set of all one-hop neighbors and the set of all two-hop neighbors, respectively.

Algorithm 1: RA-MPR selection

- N_1: a set of 1-hop neighbors
- N_R: a set of RA-MPRs
- R: a given target reliability
- n: the number of transmissions required by the sender for the target reliability
- EDP^n: an EDP of s with n transmissions
- EDP^n_{full}: an EDP of s with n transmissions when N_R is equal to N_1
- i^n_{max}: a node which increases the current EDP^n most among $N_1 - N_R$ when included in N_R
- EDP^n_{max}: an EDP^n when i^n_{max} is included in N_R

1: $N_R \leftarrow \phi, n \leftarrow 1, i^n_{max} \leftarrow \text{null}, EDP^n \leftarrow 0, EDP^n_{max} \leftarrow 0$
2: **while** $EDP^n < R$ **do**
3: **if** $EDP^n = EDP^n_{full}$ **then**
4: increase n by 1
5: initiate $N_R, i^n_{max}, EDP^n, EDP^n_{max}$
6: **end if**
7: **for all** $i \in N_1 - N_R$ **do**
8: $N_R \leftarrow N_R \cup \{i\}$, calculate EDP^n
9: **if** $(EDP^n > EDP^n_{max})$ or ($EDP^n = EDP^n_{max}$ and i covers more 2-hop neighbors than i^n_{max}) **then**
10: $i^n_{max} \leftarrow i, EDP^n_{max} \leftarrow EDP^n$
11: **end if**
12: $N_R \leftarrow N_R - \{i\}$
13: **end for**
14: $N_R \leftarrow N_R \cup \{i^n_{max}\}, EDP^n \leftarrow EDP^n_{max}$
15: **end while**

When a node selects RA-MPRs, the above EDP metric is used. s selects an RA-MPR among non-RA-MPR neighbors so as to increase the EDP most. And it selects RA-MPRs iteratively until its computed EDP becomes greater than the given target reliability. After RA-MPRs are selected, a list of RA-MPRs is known to its one-hop neighbors using piggybacked flooding (or any kinds of periodic control) messages.

Algorithm 1 shows a process determining the number of transmissions and RA-MPRs. Lines 12–20 in the pseudo-code show a process which finds a one-hop neighbor increasing EDP most iteratively among all non-RA-MPRs one-hop neighbors. And the result node is included in RA-MPRs as specified in lines 21–22. This process is repeated until the EDP satisfies the target reliability in line 5. Note that the transmission number n is initially set to 1. Thus, a node includes its one-hop neighbors to N_R until the target reliability is satisfied. However, if adding all one-hop neighbors into N_R cannot still satisfy the target reliability, s will increment its transmission number, n, to 2. Then it clears N_R and starts selecting RA-MPRs again. Lines 6–11 in the pseudo-code show this process.

For the sake of convenience, assume that the link error rates of all links are given by l and the target reliability is R. In Fig. 3, if l is 0.1 and R is 0.6, EDP^1_{max} is about 0.86 from Eq. (4), which is greater than R. Then s selects RA-MPRs one by one as it increases EDP^1. Node i_1 will be first selected as an RA-MPR because node i_1 covers three two-hop neighbors while other nodes cover less than that. Then EDP^1 is 0.53 when node i_1 is the only RA-MPR. Thus, it adds another node, which is node i_2, to N_R, because nodes i_3 and i_4 cover less two-hop neighbors than i_2 if we exclude the 2-hop neighbors already covered by i_1. After i_2 is selected as RA-MPR, EDP^1 becomes 0.65. Then sender s stops selecting RA-MPR nodes since its EDP exceeds R.

3.3. Retransmission policy

Algorithm 2: Retransmission policy

- N_R: a set of RA-MPRs
- N_2: a set of two-hop neighbors
- s: a sender
- R: a given target reliability

for all $i \in N_R$ **do**
if acknowledged from i **then**
$l_{si} \leftarrow 0$
else if unacknowledged from i **then**
$l_{si} \leftarrow 1 - P(M|A)$
end if
for all $j \in N_2$ connected to i **do**
$l_{ij} \leftarrow$ an estimated value
end for
end for
if online EDP $< R$ **then**
retransmit the message
end if

When flooding a message in the network, deciding the retransmission of a message is crucial: (i) to achieve the target reliability, and (ii) to reduce the overhead of flooding. Thus, the number of (re) transmissions should be carefully controlled so as not to incur too much redundant traffic. In POFA, the expected number of transmissions, n, can be different from the real number of transmissions since the result of a real transmission will be either a success or failure in contrast to the probabilistic calculation a priori. That is, the actual number of transmissions depends on the situation at the moment. For instance, if n of a given node is 3, the node is supposed to transmit the message 3 times to achieve the target

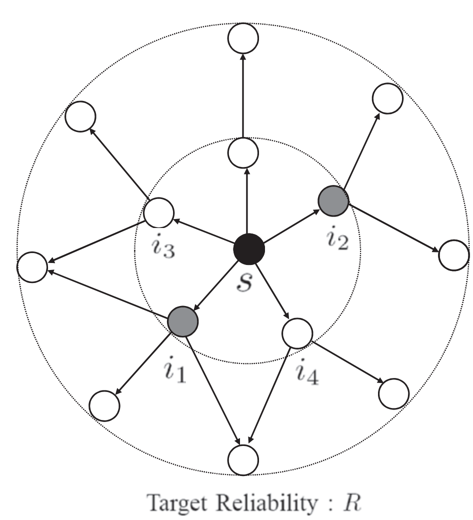

Fig. 3. RA-MPR selection: Two grey nodes are selected as RA-MPRs of s.

reliability. Suppose that the target reliability is already achieved after 2 transmissions. Then, the node can stop transmitting the message even if the message is transmitted less than n times.

To decide whether to retransmit the message or not opportunistically, it is crucial to know the transmission results of each broadcast. However, in broadcasting, acknowledgements (ACKs) should be handled carefully due to the ACK implosion problem [21]. More precisely, there are two kinds of ACKs, i.e., *explicit* and *implicit* ACKs. An explicit ACK refers to an ACK packet transmitted by a receiver to confirm successful reception (as explained in the above). On the other hand, an *implicit* ACK happens as follows. Suppose that node A sends a data packet to node B, and overhears B's forwarding the data packet to another node. In this case, node A can confirm that the packet is successfully received by B. The traffic overhead of a flooding scheme highly depends on how to combine these two kinds of ACKs, which will be discussed later.

To achieve the target reliability in POFA, it is significant whether RA-MPRs receive the message or not because only RA-MPRs rebroadcast the message. Thus, it is necessary to check RA-MPRs' receiving the message. Therefore, each RA-MPR gives an implicit ACK and/or an explicit ACK to the sender, s. If s overhears the rebroadcasted message of an RA-MPR, it is an implicit ACK. When an RA-MPR has received the same message from s twice or more, the RA-MPR sends an explicit ACK to s. With these two kinds of ACKs, s can opportunistically figure out whether it should retransmit the message or not to achieve the target reliability.

The key mechanism in the retransmission process is online EDP calculation. After sending the message, s calculates the online EDP from implicit and explicit ACKs. By contrast, the EDP described in Section 3.2 is called the offline EDP. The online EDP calculation is similar to the offline EDP calculation except that s can now confirm which RA-MPRs has received the message. If s receives an explicit or implicit ACK for a particular RA-MPR i, the corresponding l_{si} is regarded as 0. For an RA-MPR, r, whose ACK has not arrived at s, s infers EDP_r for the RA-MPR probabilistically by using a conditional probability $P(M|A)$ since the transmission error is applied to ACKs as well. Algorithm 2 shows how to decide retransmissions by using online EDP.

We define the conditional probability $P(M|A)$ where event A indicates that s has not received an ACK from a particular RA-MPR despite n transmissions, and event M means the RA-MPR actually receives the message during n transmissions from s. For the sake of exposition, we assume the link error rate is symmetric and given by l. Then $P(A)$ and $P(M \bigcap A)$ are given by:

$$P(A) = \sum_{k=0}^{n} \binom{n}{k} \cdot (1-l)^k \cdot l^n \tag{5}$$

$$P(M \bigcap A) = \sum_{k=1}^{n} \binom{n}{k} \cdot (1-l)^k \cdot l^n \tag{6}$$

where n is the transmission number of s and k indicates how many times the RA-MPR has received the message during n transmissions. Therefore, $P(M|A)$ is calculated by:

$$P(M|A) = \frac{P(M \bigcap A)}{P(A)} = 1 - \frac{l^n}{P(A)} \tag{7}$$

$P(M|A)$ means a probability that an RA-MPR has actually received the message despite the ACK from the RA-MPR has not received by s. The online EDP is calculated by replacing the EDP for an ACKed RA-MPR by 1 and the EDP for a non-ACKed RA-MPR by $P(M|A)$. $P(M|A)$ will be also used to calculate the EDP for a two-hop neighbor. If the online EDP is less than the target reliability, s retransmits the message.

4. Numerical results

We evaluate the performance of POFA by extensive simulation with NS-2 [22]. We compare POFA with other representative flooding algorithms such as RBP and RAFA in terms of reliability and flooding overhead. In all experiments, we evaluate the performance of the flooding algorithms in wireless multi-hop networking environments. The simulation parameters are described in Table 1. The simulation area is a $2000 \times 2000\ m^2$ rectangle space and 50, 100 or 150 nodes are randomly distributed in the simulation area. The payload of a packet is 64 bytes, which is about a half of general sensory packet size. A randomly chosen node floods packets to the whole network in each simulation run. The performance metric of flooding is averaged from 25 runs in each network topology.

We use three performance metrics: achieved (network-wide) reliability, achieved reliability to target reliability ratio (ATR) and the number of transmitted packets per node (NTP). The achieved reliability means the percentage of nodes that receive a flooding packet. The ATR is defined as the achieved reliability divided by the target reliability. If the ATR is 1, it indicates that the flooding algorithm achieves target reliability exactly. And if it is larger than 1 or smaller than 1, it overachieves or underachieves the target reliability, respectively. The NTP [13] is the total number of broadcasts divided by the number of nodes for each flooding to quantify the flooding overhead.

4.1. Flooding efficiency

To evaluate the efficiency, we measure the NTP of each flooding algorithm. In the case of POFA and RAFA, we choose two values for sufficiently high target reliability: 0.9 and 0.95. We do not choose 100% target reliability which is infeasible in most cases. Recall that RBP cannot choose a target reliability and it always seeks to achieve full reliability. Figs. 4–6 plot the achieved reliability and the NTP of each flooding algorithm as the link error rate increases. The line graphs indicate the achieved reliability and the bar graphs show the NTP.

Figs. 4–6 exhibits POFA always achieves high reliability near 1.0 like RAFA and RBP while flooding overhead is lower than those of RAFA and RBP in most cases. When the link error rate (LER) is small (0.05), the NTP of POFA is much smaller than that of RAFA or RBP as shown in Fig. 4 while achieving the comparable reliability as other algorithms. For instance, the NTP of POFA with the target reliability 0.9 is less than half of that of RAFA, while the achieved reliability of POFA is less than that of RAFA by 3%. This means that the energy consumption of transmissions in POFA could be about a half of RAFA's while both of them achieve the target reliability. However this performance gain is decreased as the LER becomes larger. As mentioned in Section 4.2, POFA increments the transmission number of the flooding packet aggressively since it seeks to accomplish the target reliability locally. In other words, in POFA, as the LER

Table 1
Simulation environments.

Parameters	Values
Network topology	Random topology
Size (m^2)	2000×2000
Number of nodes	50, 100, 150
Transmission range (m)	250
Message size – payload (bytes)	64
Transmission rate (Mbps)	1
MAC	IEEE 802.11
Link error rate	0.05, 0.1, 0.15, 0.2, 0.25
Target reliability	0.75, 0.8, 0.85, 0.9, 0.95

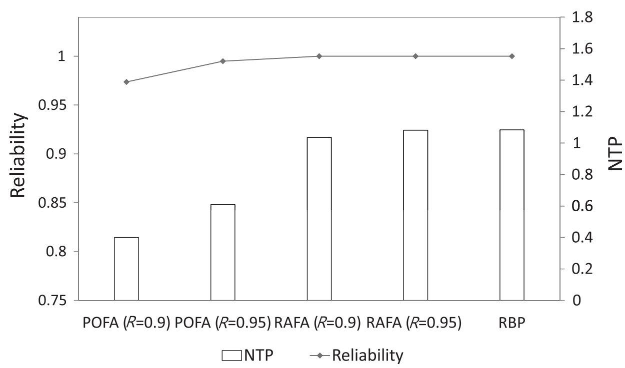

Fig. 4. Reliability and flooding overhead when link error rate is 0.05.

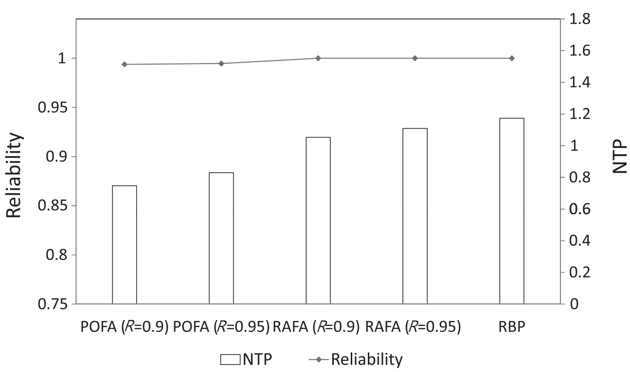

Fig. 5. Reliability and flooding overhead when link error rate is 0.15.

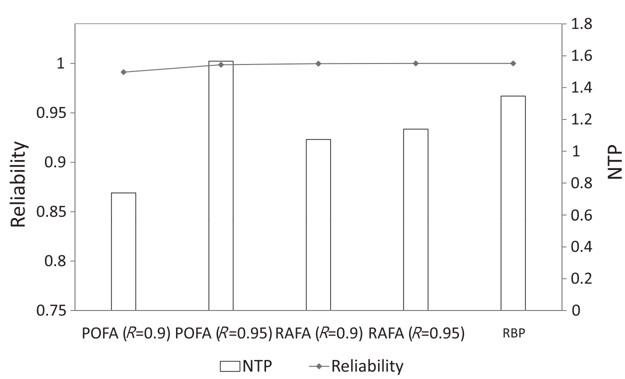

Fig. 6. Reliability and flooding overhead when link error rate is 0.25.

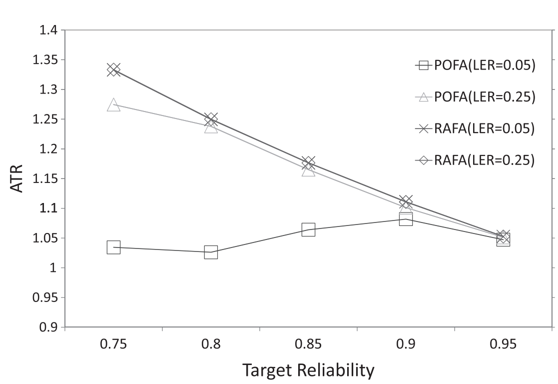

Fig. 7. Achieved reliability to target reliability ratio (ATR).

becomes higher, a node selects more RA-MPRs. Furthermore, each node makes a decision of retransmission based on online expected delivery probability locally. Therefore as the link error rate increases, not only a node but also its 1-hop and 2-hop neighbors, probably, retransmit the packet more times. *Recall* that, for the simplicity, we ignore the longer possible paths in calculating the EDP in Section 3.1 and the retransmissions of neighbor nodes in selecting RA-MPRs in Section 1. It implies that the ignored transmission number of the flooding packet becomes larger as the link error rate increases. Fig. 6 reveals that the overhead of POFA can be greater than even RBP if the link error rate is extremely high (0.25). However, with our simple model, POFA achieves the target reliability with lower overhead when the link error rate is low, *too*. *As shown* in Fig. 6, when the target reliability is given as 0.9, POFA achieves near full reliability with lower overhead. Even when the target reliability is given as 0.75, POFA achieves about 95% reliability, as shown in Fig. 7. It means that, although the link error rate is high, POFA can be still employed as an efficient flooding algorithm achieving high reliability if the target reliability is chosen properly.

4.2. Achieving the target reliability

Fig. 7 shows that POFA tends to achieve the higher reliability than the target probability as the LER increases. However, POFA still exhibits the ATR closer to 1 than RAFA, which implies that it reacts to target reliability more actively. Recall that RAFA performs flooding for the target reliability from the standpoint of the worst case neighbor (i.e., a one-hop neighbor with the least number of its own neighbors). Thus, when the LER is small (0.05), POFA achieves the target reliability more efficiently than RAFA. Even if the LER is high (0.25), POFA still performs better than RAFA.

Figs. 8–10 plot the achieved reliability of POFA versus target reliability with a range of the LER values in the case of 50, 100,

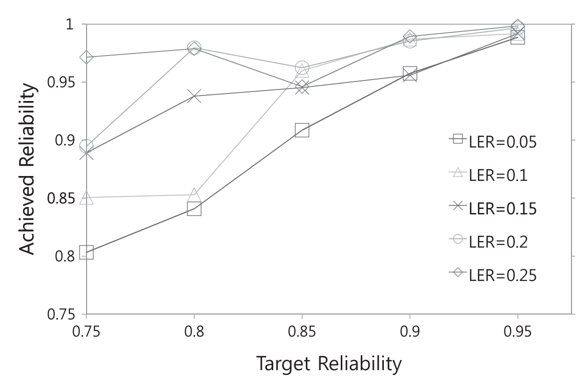

Fig. 8. Achieved reliability when the number of nodes is 50.

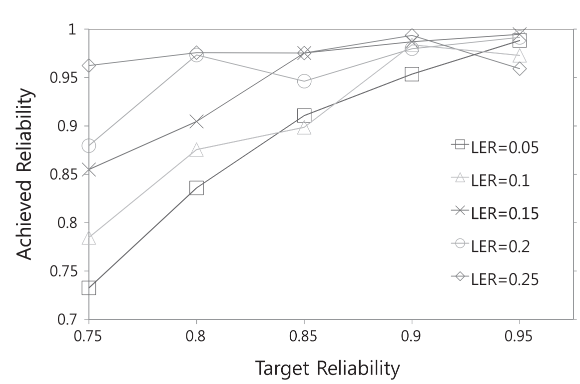

Fig. 9. Achieved reliability when the number of nodes is 100.

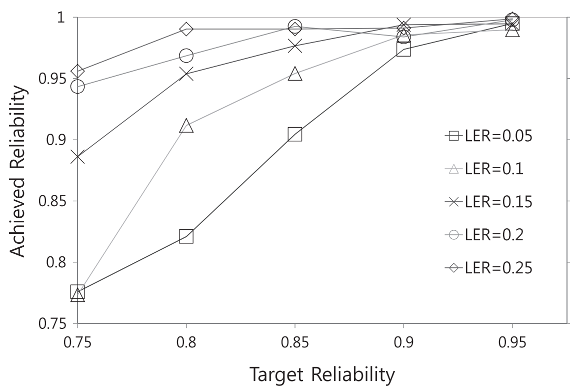

Fig. 10. Achieved reliability when the number of nodes is 150.

and 150 nodes, respectively. The achieved reliability of POFA is in general higher than the target reliability. Note that the relationship between the achieved reliability and the LER is quite counterintuitive. When the LER is low (say, 0.05), the achieved reliability is close to the target reliability, which demonstrates the effectiveness of EDP calculation. Meanwhile as the LER increases, the achieved reliability becomes higher than the target reliability. The reason is the coarse-grained impact of the number of transmissions. Recall the process of determining the number of transmissions. A sender increments the number of transmissions if including all one-hop neighbors as RA-MPRs cannot make the EDP greater than the target reliability. If the transmission number is incremented, the effect of broadcasting the packet one more time can be substantially high. That is, when the LER is high, the sender tries to broadcast the packet multiple times to satisfy the target reliability, which leads to the overachieved reliability. From Figs. 8–10, we observe that the network density does not have much impact on the achieved reliability.

5. Conclusions

In this paper, we propose a probabilistic and opportunistic flooding algorithm (POFA), which takes a probabilistic approach to achieve the network-wide target reliability by calculating the expected delivery probability (EDP) opportunistically in wireless sensor networks. A sender selects reliability-aware multi-point relays (RA-MPRs) and determines the number of transmissions to minimize the flooding overhead. Simulation results reveal that POFA achieves the target reliability with the less overhead than the previous approaches in most cases. In particular, when the link error rate is substantially high, POFA tends to achieves higher reliability than the given target reliability.

Acknowledgement

This work was supported by the IT R&D program of KCA (10913-05004: Study on Architecture of Future Internet to Support Mobile Environments and Network Diversity) and NAP of Korea Research Council of Fundamental Science & Technology.

References

[1] C. Perkins, E. Belding-Royer S. Das, Ad hoc On-demand Distance Vector (AODV) Routing, RFC 3561, July 2003.
[2] D. Johnson, D. Maltz, Dynamic Source Routing in Ad hoc Wireless Networks, Mobile Computing, Kluwer Academic Publishers, 1996. pp. 153–181.
[3] T. Clausen et al., Optimized link state routing protocol (OLSR), RFC 3626, October 2003.
[4] LAN MAN Standards committee of the IEEE computer society, Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE Std. 802.11, 1997.
[5] C. Intanagonwiwat, R. Govindan D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, ACM MobiCom, August 2000.
[6] Syed Muhammad, Khaliq-ur-Rahman Raazi, Sungyoung Lee, A survey on key management strategies for different applications of wireless sensor networks, Journal of Computing Science and Engineering 4 (1) (2010) 23–51.
[7] E. Felemban, C. Lee, E. Ekici, MMSPEED: multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks, IEEE Transactions on Mobile Computing 5 (6) (2006) 738–754.
[8] S.-J. Park, R. Vedantham, R. Sivakumar, I. Akyildiz, GARUDA: achieving effective reliability for downstream communication in wireless sensor networks, IEEE Transactions on Mobile Computing 7 (2) (2008) 214–230.
[9] Y. Sankarasubramaniam, O. Akan, I. Akyildiz, ESRT: event-to-sink reliable transport in wireless sensor networks, ACM MobiHoc, June 2003.
[10] Z. Has, J. Halpern, L. Li, Gossip-based ad hoc routing, IEEE/ACM Transactions on Networking 14 (3) (2006).
[11] Y. Sasson, D. Cavin, A. Schiper, Probabilistic broadcast for flooding in wireless mobile ad hoc networks, in: IEEE WCNC, March 2003.
[12] F. Stann et al., RBP: robust broadcast propagation in wireless networks, in: ACM Sensys, November 2006.
[13] Y. Lee et al., A reliability aware flooding algorithm (RAFA) in wireless multi-hop networks, in: IEEE ISCC, July 2008.
[14] H. Sabbineni, K. Chakrabarty, Location-aided flooding: an energy-efficient data dissemination protocol for wireless sensor networks, IEEE Transactions on Computers 54 (8) (2005) 36–46.
[15] A. Durresi et al., Optimized broadcast protocol for sensor networks, IEEE Transactions on Computers 54 (8) (2005) 1013–1024.
[16] Wei Ye, John Heidemann, Deborah Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking 12 (3) (2004).
[17] A. Woo, T. Tong, D. Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in: ACM SenSys, November 2003.
[18] Q. Cao, T. Abdelzaher, T. He, R. Kravets, Cluster-based forwarding for reliable end-to-end delivery in wireless sensor networks, in: IEEE INFOCOM, May 2007.
[19] A. Woo, D. Culler, Evaluation of efficient link reliability estimators for low-power wireless networks, Technical Report UCB/CSD-03-1270, U.C. Berkeley Computer Science Division, September 2003.
[20] Shan Lin, Jingbin Zhang, Gang Zhou, Lin Gu, Tian He, John A. Stankovic, ATPC: adaptive Transmission Power Control for Wireless Sensor Networks, ACM SenSys, October 2006.
[21] M. Impett, M. Corson, V. Park, A receiver-oriented approach to reliable broadcast in ad hoc networks, in: IEEE WCNC, September 2000.
[22] The network simulator – NS-2, <http://www.isi.edu/nsnam/ns/>.