2010

A Metric for Routing in Delay-Sensitive Wireless Sensor Networks

Zhen Jiang
West Chester University of Pennsylvania, zjiang@wcupa.edu

Jie Wu
Temple University

Risa Ito
West Chester University of Pennsylvania

Recommended Citation
Jiang, Z., Wu, J., & Ito, R. (2010). A Metric for Routing in Delay-Sensitive Wireless Sensor Networks. *7th IEEE Conference on Mobile Ad Hoc Sensor Systems*, 272-281. http://dx.doi.org/10.1109/MASS.2010.5664010

A Metric for Routing in Delay-Sensitive Wireless Sensor Networks

Zhen Jiang
Dept. of Computer Sci.
West Chester University
West Chester, PA 19382

Jie Wu
Dept. of Computer & Info. Sciences
Temple University
Philadelphia, PA 19122

Risa Ito
Dept. of Computer Sci.
West Chester University
West Chester, PA 19382

***Abstract*—We propose a new scheme to reduce the end-to-end routing delay in the mission-critical applications of the wireless sensor networks (WSNs) under the duty cycle model. While greedy routing in the synchronized MAC model has been studied extensively, efficient routing in an asynchronous MAC model is considerably different because the wake-up time and availability of a node along the pre-decided path are not synchronized and can be changed by many dynamic factors. The challenge is to catch this dynamic change in time and furthermore, to minimize its impact on routing decisions. We proposed a normalized evaluation value $\in [0, 1]$ at each node under the proactive model for all different paths passing through, saving the cost and delay of the reactive information model. Its measurement interprets the existence of the fastest path to the edge of the networks in a certain direction, directing any local advance greedy in the same direction. We provide a new strategy for greedy routing. First, it waits for the appearance of the expected forwarding successor; if this fails, then it will select the backup by the "first-wake-up, first use" policy to avoid a dead wait. We focus on an "everyone" model, in which each node will apply the same generic process in a fully distributed manner in order to achieve a reliable solution. Applying our approach in the networks with a uniform wake-up schedule, we illustrate the substantial improvement of our approach in both analytical and experimental results compared with those best known to date.**

Keywords: Delay, distributed algorithms, routing, wireless sensor networks.

I. Introduction

Wireless sensor networks (WSNs) have great long-term economic potential and the ability to transform our lives. In many mission-critical applications, it is very important to send surveillance results without any unnecessary delay. Affected by the unstable nature of the wireless signals and the complex terrain of the deployment area, surveillance reports cannot, in many cases, be sent to sink directly and require a multi-hop relay path. In traditional multi-hop routing schemes, the path is built by the independent decision at each intermediate node where a designated next-hop relay node is selected from all available 1-hop neighbors. A neighbor closer to the destination is preferred to avoid any unnecessary hop [8] in use. Such a node selection is also called *localized greedy forwarding* (or simply greedy forwarding). Otherwise, the routing takes a detour.

Recent systems [3], [12] have adopted the asynchronous sleep-wake scheme [9], [14] to reduce the overhead of neighbor synchronization. In this duty cycle system, the sleep-wake schedule at each node uses a predictable pseudo-random

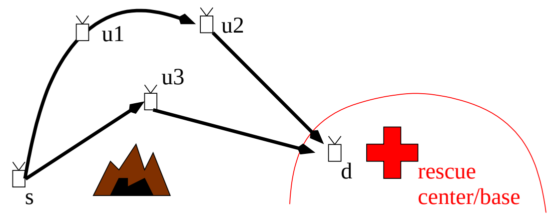


Fig. 1. Multiple-hop unicasting in the disaster recovery application.

sequence, but is independent of those of other nodes. No synchronization is required. Prior work of *anycast* [2], [4], [9], [15] has proposed the use of the "first-wake-up, first use" policy (FWFU), where each node forwards the packet to the first candidate node that wakes up. However, as indicated in [7], that candidate node may be in the path with more of a delay to the base/sink. The reduction in 1-hop delay, also called *cycle waiting time*, may not necessarily lead to the optimization on the end-to-end delay. Consider the scenario in Figure 1. Node s wishes to send a report to the base. Blocked by the mountainous terrain, its signal cannot directly reach the sink d and requires a relay path. Among its neighbors, u_1 is the first node that wakes up. When the routing reaches it, another relay node u_2 is needed for the packet sent to the destination. If s can hold the packet and wait until u_3 wakes up, the path $s - u_3 - d$ has less hops and the routing takes less time.

Each greedy forwarding needs to determine which neighbor to choose on the behalf of the entire path. This relies on accurate information of the elapsed time along each possible path, not only the cycle waiting time, but also the transmission time. By exchanging the pseudo-random seed and the last wake-up time, a node can easily forecast its neighbor's next appearance. However, that neighbor's connection can be deferred to the next cycle in the schedule sequence due to signal fading, interference, and any purposed re-schedule for performance optimization in the beaconing process. In many cases, a neighbor can become unreachable due to mobility, failures, communication jams, and power exhaustion. To achieve a path without unnecessary delay, it is important to catch each change in time and allow the routing to adjust to a better path.

Our work aims to provide the required information in a new metric for the duty cycle systems. The minimum cost and the effectiveness of reducing routing delay, especially under unpredictable schedule changes, distinguishes our solution from others. We face three challenges of the variation of node/connection availability in duty cycle systems.

- First, how does each node collect its information and then control the cost? Without using any global control, the information will be accumulated by exchanges among 1-hop neighbors. The transmission time is considered as well as the cycle waiting time. In order to complete the collection quickly, we need to control the scalability within a limited area (i.e., region) for the search of the routing path with the minimum delay, even when many nodes and connections change their availability.
- Second, how can the granularity of such a region be determined? The neighborhood connections at each node are of irregular structure in WSNs. A relay node will change the scope of the neighborhood watch as well as its availability in the least-time routing path. The concern region for metric evaluation is also irregular and may change for each different routing request. We need a relatively stable region in metric evaluation to avoid changing its value too often and too quickly in the proactive model.
- Third, how does the designated metric information reflect the quality of a routing? We need to study the effectiveness of the localized processes in both metric evaluation and greedy forwarding. We focus on a practical routing solution under frequent changes of asynchronous neighbor schedules and node availability.

In our approach the greedy forwarding is limited within the request zone in [8] so that all possible paths can be controlled in a quadrant. Such a forwarding, also called LF routing, has a simple structure for easy information construction. A simple value $M \in [0, 1]$ is provided at each node. "0" indicates that the LF routing and its succeeding paths from this node will be blocked by local minima. Accessing such a node will incur detours, which require extra neighbor synchronization and data transmission. "1" indicates a permanently awakened node or sink that is ready for data transmission at any time. Otherwise, $\frac{1}{M}$ implies the minimal transmission time of a non-detour LF path built from this node to a nearby permanently awakened node, such as the sink or edge nodes of the networks. As usual, these nodes always remain active to provide a complete, constant coverage. That path is also called the *reference path* of this M and will be used to guide the routing in the same direction. The larger the value of M, the less delay along that reference path there will have. M's construction reuses the beacon message at no extra cost. Its update is dependent on the duty cycles of all 1-hop neighbors, not just one single neighbor connection. It can remain stable even when many nodes change their duty cycles or availability.

Like a lighthouse guiding boats to the harbor at night, but not necessarily illuminating everywhere, this metric value guides our LF routing to select a neighbor with a relatively higher priority (i.e., less delay), approaching the destination greedily in the same direction that its reference path does. When the dynamics incur a change of metric value, any in-progress routing heading into the update propagation can make an alternative selection to avoid the dead wait for a single neighbor. Strictly speaking, the approach supports segmented routing: the routing changes its indirect referee while its forwarding region switches from one quadrant to the other. In a sample deployment model, our analytical and experimental results show the effectiveness of our metric in achieving a reduction of end-to-end delay in greedy forwarding, even when many nodes change their duty cycles and availability.

Our contributions are threefold:

1) The detour and its unnecessary relay have been ignored in existing routings in duty cycle systems. They are considered in our metric. A balanced, comprehensive measurement is provided for each localized routing decision to achieve better end-to-end performance.
2) Unlike reactive methods requiring a probing process to fetch the information, our metric evaluation is conducted under a proactive model, saving the cost and delay for routing decision. The implementation problems are addressed. A balance point of the tradeoff between precision and construction cost is proposed.
3) We provide both analytical and experimental results to illustrate the effectiveness of our balanced measurement in achieving less delay in data transmission, even in a highly dynamic network within which many nodes change their duty cycles and availability.

The remainder of the paper is organized as follows: Section 2 introduces some necessary notations and preliminaries. We provide details of the network model, the LF routing, and the data transmission with duty cycles for each 1-hop advance. Section 3 highlights the challenge to provide delay information in the proactive mode. We then discuss the idea of our approach. Section 4 presents our delay metric and its detailed evaluation processes. Section 5 provides our greedy forwarding based on this measurement. In Section 6, we prove the bound of delay in our routing when each node wakes up in a true schedule or when they can change their schedules dynamically. In Section 7, both the analysis results and the simulation results are illustrated to prove the improvement of our routing in delay and completion time compared with the best results known to date. The simulation results also show an acceptable construction cost in the metric evaluation. This proves the cost-effectiveness of our approach. Section 8 discusses the existing issue in related work. Section 9 concludes this paper and provides ideas for future research.

II. Preliminary

A. *WSNs with a guided schedule (GS)*

A WSN under the duty cycle model can be represented by a simple undirected graph $G = (V, E)$, where V is a set of vertices (nodes) and E is a set of undirected edges. $N(u)$ denotes the set of neighbors within the radius of node u. $n(u)$ ($\subseteq N(u)$) denotes the set of neighbors that are currently

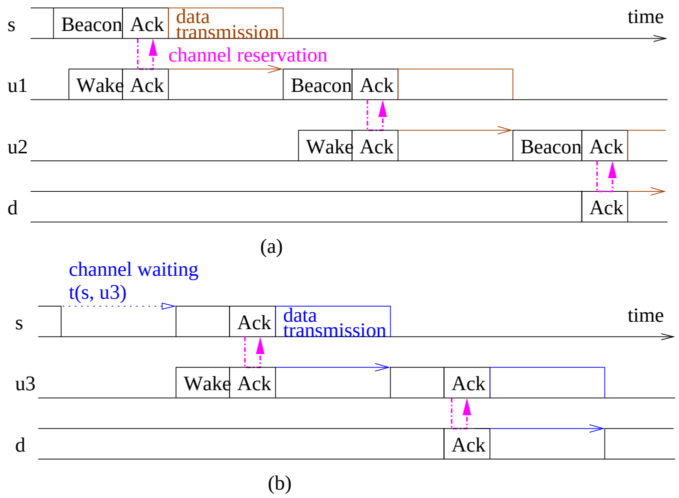


Fig. 2. Time sequence for the sample routing in Figure 1. (a) Path $s - u_1 - u_2 - d$ uses the FWFU policy for the successor selection, and (b) Path $s - u_3 - d$ uses selection with an appropriate wait, which requires an accurate prediction of wake-up time.

awakened with u. Each node u has the location (x_u, y_u), simply denoted by $L(u)$. $\mid L(u) - L(v) \mid$ is the distance between two nodes u and v. $s(x_s, y_s)$ and $d(x_d, y_d)$ are the source and destination nodes.

Our networks are deployed in a 2-D plane. Data can report to sinks with satellite signals or mesh nodes along the edge with Internet access. They are set in safe areas and do not have power inefficiency. We keep them awakened to provide complete coverage constantly. Other nodes inside the deployed area will periodically go to sleep in a cycle in order to save energy and extend lifetime. The schedule is determined by a pseudo random sequence with a preset seed in the uniform distribution. Each time a node u wakes up, it initiates a beaconing process to connect nodes within its communication range. When a neighbor v receives this beacon message ($v \in n(u)$), it will respond to u and share the information, including the location, seed of pseudo random sequence, last wake-up time, metric values, etc. Each node can predict the next appearance of its neighbors.

A short message system with the FWFU waiting schedule is adopted in our networks. The packet will advance one-hop in each cycle until it is delivered to the destination d. When an active node u needs to communicate, it will start from the beaconing process. Whenever a neighbor wakes up during this period (i.e., $v \in n(u)$), it will respond to u. After that, u can forward the packet to v. An example of this non-delay transmission is shown in Figure 2 (a), along the routing path $s - u_1 - u_2 - d$ in Figure 1. A node u will keep beaconing its neighbors cycle by cycle until a neighbor becomes available for forwarding. This scheme has been used in existing *anycasting* [2], [4], [9], [15]. Simply, this mode is denoted by FW.

The system also supports guided schedule changes, denoted by GS, that are required for performance optimization in [7], [13]. A node u can select one of its neighbors v and expect it to wake up after a certain time, following its own schedule. u will hold the packet and switch to sleep mode, allowing other nodes in its neighborhood to communicate. After time $t(u, v)$, u will wake up to continue communicating with v. $t(u, v)$, also called the cycle waiting time, is the time difference between u's appearance and v's coming appearance. After the message is sent, u will schedule back to its original sleep-wake sequence. In our approach after the target neighbor is selected, the corresponding waiting time is set with metric evaluation. The example of data transmission with an appropriate wait is shown in Figure 2 (b), along the routing path $s - u_3 - d$ in Figure 1. However, when v misses its schedule or is no longer available at the expected time, u will switch back to the FW mode.

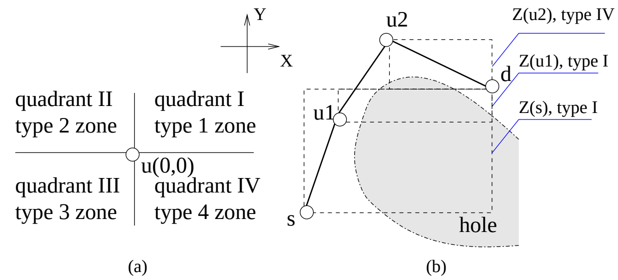


Fig. 3. Definition of (a) forwarding zones and (b) request zones.

To study the impact of a dynamic change, the schedule sequence is randomly changed in the Poisson process, with which each node u wakes up. We preset the intensity λ so that the time $t(u, v)$ can be controlled within a uniform range 2β with an average of β. Note that for each pair of neighboring nodes u and v, $t(u, v)$ is directional and independent. Each node u needs a local clock to maintain $t(u, v)$. However in this paper we use global time in the slots to simplify the discussion.

B. Limited greedy forwarding (LF)

As described in LAR scheme 1 in [8], the selection of the forwarding successor can be limited within the request zone in order to achieve a simple regularity structure. The request zone is a rectangle in the corresponding quadrant (see Figure 3 (a)) with both u and d at opposite corners (see Figure 3 (b)). The request zones, with respect to d, in quadrants I, II, III, and IV are of types 1, 2, 3, and 4, denoted by $Z_i(u, d)$ $(1 \leq i \leq 4)$. Each corresponding quadrant is called type-i *forwarding zone*, denoted by $Q_i(u)$. A greedy advance in $Z_i(u, d)$ is called type-i forwarding. The discussion in this paper focuses on type-1 forwarding and the corresponding information collection. The rest of the results can easily be derived by rotating the plane.

Algorithm 1 shows the details. At each hop, a successor is selected in the request zone. Indicated in [5], a single LP routing path may experience different types of forwardings when the relative position of d to the intermediate node changes and d is located in different types of request zones.

Compared with the region that contains all possible successors in anycasting, the forwarding zone has a limited area and

Algorithm 1 (LF routing): Determine the successor of node u (including node s) with respect to $N(u)$ [8].

1) If $d \in N(u)$, $v = d$.
2) Determine the request zone $Z_k(u,d)$ ($1 \leq k \leq 4$), according to $L(u)$ and $L(d)$.
3) Select $v \in N(u) \cap Z_k(u,d)$.

reduces the flexibility of LF. However, it has a simple structural regularity and each of its successful advances is a greedy forwarding. Next, we will present our metric information under the GS model for LF routing. The information-based routing can achieve better performance than anycasting in terms of delay (i.e., the speed of routing). In this way, we show the value of our metric. Table I summarizes all of the notions used in this paper.

ρ	node density in deployment
s / d	source / destination
u	the current node of the routing from s to d
$L(u)$	location of node u, i.e., (x_u, y_u) in the 2-D plane
$N(u)$	1-hop neighbor set of u
$n(u)$	set of u's neighbors currently awakened
$t(u,v)$	cycle waiting time that u waits for $v \in n(u)$
β	length of duty cycle, maximum value of $\frac{t(u,v)}{2}$
$T(u,v)$	total time of a one-hop transmission from u to v
$Q_i(u)$	type-i forwarding zone ($1 \leq i \leq 4$)
$Z_i(u,d)$	type-i request zone with respect to $Q_i(u)$ and d
η	the average number of neighbors in Q_i
τ	the average number of different key paths in Q_i
$M_i(u)$	delay estimation for forwarding inside $Q_i(u)$
$M(u)$	delay estimation array, tuple ($M_i(u) : 1 \leq i \leq 4$)

TABLE I
LIST OF NOTIONS USED.

III. Problem and the Proposed Idea

Our goal is to achieve the optimization of delay for a single routing, instead of the mean time of the delay. Unlike those methods determining the wake-up time to fit the subsequent path, our approach selects the path with the best schedule to reduce end-to-end delay. We focus on an "everyone" model, in which each node will apply the same generic process in a fully distributed manner, in order to achieve a reliable solution. More specifically, for each node u along the routing path to the destination d, we provide the information of each 1-hop neighbor $v \in N(u)$, interpreting the elapsed time of its subsequent path in a global view. This evaluation will be used for the decision at u to achieve the quickest path.

The larger evaluation value, the less delay the path likely has. That is, for any two paths $\{u, u_1, u_2, \cdots, u_k, u_{k+1} = d\}$ and $\{v, v_1, v_2, \cdots, v_{\mathbb{k}}, v_{\mathbb{k}+1} = d\}$, we have constraint 1:

$$\min \sum_{i=1}^{i=k} T(u_i, u_{i+1}) < \min \sum_{j=1}^{j=\mathbb{k}} T(v_j, v_{j+1}),$$

iff

$$M(u,d) \geq M(v,d).$$

Note that k and $\mathbb{k}$ are not necessarily the same. Thus, M is an evaluation function that includes (1) the delay caused by cycle waiting time $t(u,v)$, (2) other delay costs in message transmission $\in T(u,v)$, and (3) the number of hops along the entire path k and $\mathbb{k}$. As usual, a shorter path ($k < \mathbb{k}$) takes less transmission time. It will be selected with a larger evaluation value.

The information must be derived from a generic construction process "$\circ$" by exchanges among 1-hop neighbors. That is, we have constraint 2:

$$M(u,d) = \min\{M(v,d) \circ T(u,v) \mid v \in F(u,d)\}$$

where $F(u,d)$ is the set of forwarding candidates of u in the routing to d. Most existing routings use the definition in LAR scheme 2 [8], i.e., $F(u,d) =$

$$\{v \mid v \in N(u) \wedge \mid L(u) - L(d) \mid > \mid L(v) - L(d) \mid\}.$$

To store and exchange information easily, information $M(u,d)$ must be normalized in $\in [0,1]$, fitting the resource constraint of WSN. Note that the normalized value will possibly cause a round-off error and cannot represent the exact delay time. Indeed, a relatively high value is selected in the routing decision, regardless of its numerical value.

We focus on a practical solution under the proactive model because the delay and cost of the information collection under the reactive model (on-demand, e.g. [6], [7], [10]) are the problems that cannot be ignored. Our information at each node is constituted before any routing is initiated. Because the destination d is unknown, the number of evaluation records maintained at each node must be reduced to 1. That is,

$$\mid M(u) \mid = O(C) \rightarrow 1$$

where $M(u) = \{M(u,d) \mid d \in V\}$. This is the implementation problem because d is relevant in constraint 2.

In our approach, we replace $F(u,d)$ with $Z_i(u,d) \cap N(u)$. Because $Z_i(u,d) \cap N(u) = Q_i(u) \cap N(u)$ and $1 \leq i \leq 4$, we can achieve

$$\mid M(u) \mid = 4.$$

Because $Z_i(u,d) \subset F(u,d)$, the new $M(u)$ may have a loss of precision when it does not have the exact delay measurement. However, this will reduce the complexity of the decision algorithm and the cost of information construction. By sacrificing few opportunities of taking the best path, our approach aims to guarantee that the result path has a performance very close to the optimal one, especially in dynamic networks where the nodes change their availability or schedule frequently. For each region divided at u, $Q_i(u)$, we use one designated $M(u)$ evaluation value. Usually, it is the delay measured from u to the closest permanently awakened node, say v. Like the lighthouse guiding boats to the harbor at night, $M(u)$ guides the routing to advance greedily in the same direction from u to v. The routing will not miss any path of greedy forwarding because the path from u to d likely shares the most selections with the path to v. When the routing changes the relative position to the destination, it changes the forwarding direction

Algorithm 2 (Metric evaluation under the GS model).

1) Each permanently awakened node u sets $M(u)$ to a fixed $(1, 1, 1, 1)$. If the node u is unavailable for a routing relay, it sets a fixed $(0, 0, 0, 0)$, until this unavailable node is recovered. Every other node v sets $M(v)$ to a changeable $(0, 0, 0, 0)$.
2) Then, each node will have a stable status by applying Eqs. (1) and (2) with a beaconing scheme.
3) In case any node changes its schedule, the above process with Eq. (2) will be applied.

and the referee v. The details will be discussed in the next section.

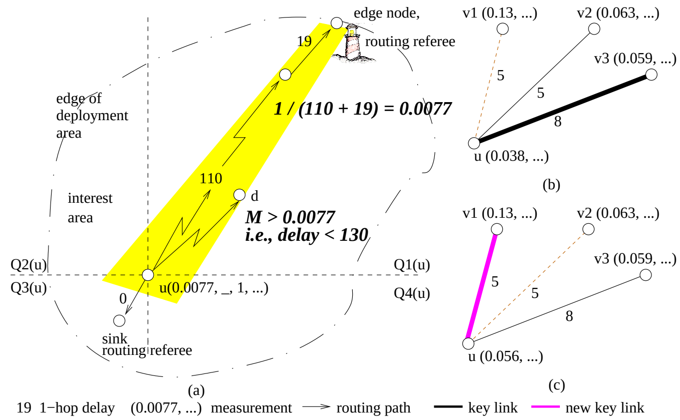


Fig. 4. Illustration of the definition of $M(u)$ and its updates.

IV. Metric Evaluation

Our new metric describes the minimal elapsed time of a successful routing from the current node to the closest permanently awakened node, under the GS model. As shown in Figure 4 (a), the larger the value, the less delay the path likely has. Such a value also implies a larger value (i.e., less delay) of a successful routing to reach a closer destination. In the following, we will discuss this metric and its details in Algorithm 2. The metric is used by each node u to determine greedy forwarding.

According to different types of forwarding zones, our metric is a 4-tuple ($C = 4$). Permanently awakened nodes set their fixed values to $(1, 1, 1, 1)$, in which "1" indicates that there is no delay for any of them to receive messages. If any of them is unavailable for a routing relay, it sets a fixed $(0, 0, 0, 0)$, until this unavailable node is recovered. Other nodes set a changeable $(0, 0, 0, 0)$, in which "0" indicates an initial value of unknown delay or endless delay ($= \infty = \frac{1}{0}$). After this, u will update $M_i(u)$ once with:

$$M_i(u) = \max\{\frac{1}{t(u,v)+\beta+\frac{1}{M_i(v)}}\}, \quad 1 \le i \le 4 \tag{1}$$

where $v \in n(u) \cap Q_i(u)$, and the selected link $\{u, v\}$ is called the *key link* of u for $M_i(u)$. It builds up the reference path from u to the permanent nodes, with the minimum delay. After this, $M_i(u)$ will stabilize by repeating:

$$M_i(u) = \max\{M_i'(u), \max\{\frac{1}{t(u,v)+\beta+\frac{1}{M_i(v)}}\}\}, \quad 1 \le i \le 4 \tag{2}$$

where $M_i'(u)$ is the original value before the update of $M_i(u)$, and $v \in n(u) \cap Q_i(u)$. Note that $n(u)$ is predictably changeable due to the value of $t(u, v)$ ($v \in n(u)$). Eq. (1) initiates the update. Eq. (2) will catch the maximum overall value for the stable status after all available $N(u)$ neighbors have been contacted. If any node changes its schedule, the above process with Eq. (2) will be applied until all nodes have stable information. Starting from the permanently awakened nodes of the networks with a fixed status, the whole phase converges quickly, as we will show in the experimental results later.

An example of the evaluation for $M_1(u)$ is shown in Figures 4 (b) and (c) when $\beta = 6$. At first, among all $N(u)$ neighbors $\in Q_1(u)$, v_2 and v_3 wake up first ($v_2, v_3 \in n(u)$) and exchange their M values with u. Therefore, u will use $t(u, v_2) = 6$ and $t(u, v_3) = 8$ to calculate $M_1(u) =$

$$1/(t(u,v_2) + \beta + \frac{1}{M(v_2)}) = 1/(5 + 6 + \frac{1}{0.063}) = 0.038.$$

The link (u, v_2) is set as the key link. When node v_1 appears in $n(u)$ (see Figure 4 (c)), the link $\{u, v_1\}$ will be selected as the key link. By using Eq. (2), we have $M_1(u) =$

$$1/(t(u,v_1) + \beta + \frac{1}{M(v_1)}) = 1/(5 + 6 + \frac{1}{0.13}) = 0.056.$$

It is the final stable value for $N(u) = \{v_1, v_2, v_3\}$ when no node changes its schedule.

Theorem 1: *$M_i(u)$ is a required evaluation function.*

Proof: Obviously, $1 \le i \le 4$ and $0 \le M_i(u) \le 1$. Due to the definition of $M_i(u)$ in Eqs. (1) and (2), $M_i(u)$ satisfies the second constraint for localized information construction. According to Eqs. (1) and (2),

$$\frac{1}{M_i(u)} = t(u,v) + \beta + \frac{1}{M_i(v)}$$

when (u, v) is the key link. That is, $\frac{1}{M_i(u)}$ is the minimal elapsed time from u to the closest permanently awakened node v. This satisfies the first constraint. Therefore, the statement is proven. ■

Any sink available to receive the message will be active and keep its "1" status. When $Q_i(u) \cap N(u) = \phi$, a local minimum occurs. $M_i(u)$ will set its "0" status. Otherwise, when every node $v \in Q_i(u) \cap N(u)$ has $M_i(v) = 0$, $M_i(u) = 0$ and u will be identified as one of those nodes whose succeeding routings will all be blocked; That is, detour and extra relay are needed.

V. Greedy Forwarding with Metric Information under the GS model (MR)

Basically, greedy forwarding under the GS model will first select a neighbor $v \in N(u)$ (instead of $n(u)$ in anycasting) along the key link in $Z_k(u, d)$ if it has the largest M value. When $M_k(u) > 0$, the path is achieved by greedy forwarding

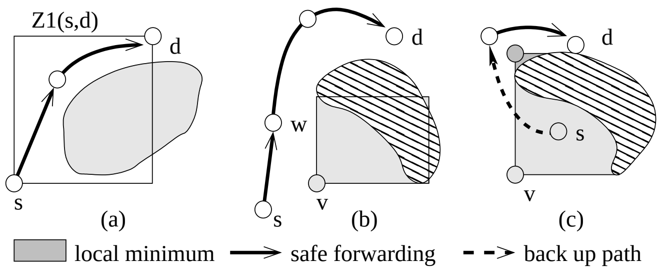


Fig. 5. Samples of the MR routing.

only (with an appropriate wait at each intermediate node), as we can prove in the following theorem. Samples can be seen in Figures 5 (a) and (b).

Theorem 2: *For a type-k forwarding, when $M_k(u) > 0$, the path from u to d can be conducted without any detour.*

Proof: Since $M_k(u) > 0$, there is always a neighbor $v \in N(u)$ that $M_k(v) > 0$, according to the definition in Eq. (1) and Eq. (2). The greedy forwarding can select v as the successor and such a process will continue. If it is blocked by a local minimum at a node w, we have $M_k(w) = 0$. However, $M_k(w) > 0$ has been confirmed at u's preceding node, which leads to a contradiction. ■

For type-k forwarding, when $M_k(u) = 0$, but $M_{\Bbbk}(u) > 0 \wedge k \neq \Bbbk$, the routing from u can use the type-$\Bbbk$ forwarding to leave such an unsafe area, until the type-k forwarding can continue. An example of the MR routing with a guided backup path can be seen in Figure 5 (c).

After v is selected, u will wait $t(u, v)$ until v wakes up. Due to many dynamic factors, v can be unavailable at that time. Then, u switches to an FW mode. It will keep waiting until $n(u) \neq \phi$. A node $v \in n(u)$ with less of detour, indicated by

$$t(u, v) + \beta + \frac{1}{M_k(v)},$$

will be selected. It is a backup phase after the failure of the guided waiting phase.

When the source has the tuple $(0, 0, 0, 0)$, the network may be disconnected. Our MR routing will then stop and wait until a better network configuration emerges. The details of the MR routing are shown in Algorithm 3.

VI. Performance Evaluation

In this section, we provide an analysis on the average time that a node needs to wait for the successor in a successful MR forwarding. In terms of the number of hops along the entire path, the total cycle waiting time can be determined, which is the major difference between our routing and traditional anycasting. To simplify the analysis, we assume that each node has the same transmission radius r under the well-known unit disc graphs (UDG) communication model in this paper. The results will be used to compare with the experimental results in the next section. They will provide an estimation of sacrifice

---

Algorithm 3 (MR routing): Determine the successor v at node u (including node s) with respect to $N(u)$.

1) Apply steps 1) and 2) of Algorithm 1.
2) **Safe forwarding**. If $M_k(u) > 0$, select $v \in N(u) \cap Z_k(u, d)$, where (u, v) is the key link of $M_k(u)$.
3) **Backup path forwarding**. Otherwise, for any $M_{\Bbbk}(u) > 0$ ($k \neq \Bbbk$), conduct a type-$\Bbbk$ safe forwarding.
4) **Guided waiting phase**. After v is selected, wait $t(u, v)$ until it wakes up.
5) **FW backup phase**. If v misses the contact at that expected time, u switches to an FW mode; that is, u waits until $n(u) \neq \phi$ and selects $v \in n(u) \cap Z_k(u, d)$ indicated by $t(u, v) + \beta + \frac{1}{M_k(v)}$, preferred to the selection in Q_k.

---

in our tradeoff for less hops and less transmission time of the path, which will be proven to be acceptable and worthy.

First, we will study the ideal case. No node changes its channel schedule so that each $t(u, v)$ is not only predictable, but also truly occurs. Note that in the duty cycle systems with a uniform distribution in the schedule sequence, it has been well known (e.g., [9]) that a node will take on the average time $\frac{t}{k+1}$ to get in contact with the next-hop node, where k is the number of forwarding options and t is the maximum waiting time. Instead of using all neighbors at each intermediate node, our MR routing always follows the path with key links. The analysis is built on the number of 1-hop neighbors of node s that can impact the only key path to d in an h-hop MR routing, and the maximum waiting time along such a key path.

Corollary 1: *When each node u has a true schedule, the average cycle waiting time for each packet sent along an h-hop path that is built in the MR routing is*

$$\bar{E}(\tau)h = h\frac{2\beta}{\tau + 1},$$

where $\tau = n/3$, $n = \frac{\rho r^2}{h}\sum_{i=2}^{h} \arccos(1/i)$, and r is the radius of communication range.

Proof: $t(u, v) \in [0, 2\beta]$. For the path with total h hops, the cycle waiting time is $\in [0, 2\beta h]$. For each node u along the path that is i-hops away from d, its physical distance to d, ξ, is in the range $[0, i \times r]$, where r is the radius of node u. On average, $\xi = \frac{i \times r}{2}$. Shown by [9], [15], the region for greedy forwarding is the overlap area of two discs: the first disc has a radius r and the center u; and the second one has a radius δ and the center d. The region area is

$$\frac{2\arccos(\frac{r/2}{\xi})}{2\pi} \times (\pi r^2)$$

and can be estimated by $\frac{\arccos(1/i) \times r^2}{h}$. After we introduce the deployment density ρ, we can determine the value of n in terms of h:

$$n = \frac{\rho r^2}{h}\sum_{i=2}^{h} \arccos(1/i).$$

Since four forwarding zones are used at each node u, on average, u will have $4n$ 1-hop neighbors. For any two that are neighboring with each other, one of them cannot be on the key path. Node u can have 6 different neighbors that are not neighboring with each other; i.e., 6 different key paths. Since the forwarding is unidirectional and may not share the key path in the opposite direction, s will have

$$\tau = \frac{4n}{6 \times 2} = \frac{n}{3}$$

1-hop neighbors for the routing decision, and their subsequent paths impact the only key path to d in the MR routing.

Therefore, in terms of the duty cycle length $2\beta h$ and the forwarding set size $\tau = n/3$, the average cycle waiting time in an MR routing is proven. ■

Next we will study the dynamic situation which is when δ out of Δ nodes change their schedules unpredictably in a Poisson process under our network model. The following corollary proves that a bounded cycle waiting time can be achieved. Note that when the routing uses stable metric information, it can ensure the path due to the use of fixed key links along the reference path. The cycle waiting time along such a stabilized subsequent path can be determined by Corollary 1. The result shows that our MR routing will not wait too long if it misses the contact. Actually, the MR routing speeds up in a highly dynamic situation because of the use of an FW mode after the miss. Note that without an appropriate wait, directly applying the FW mode at step 5 in Algorithm 3 will be a special case of anycasting, causing worse end-to-end delay.

Corollary 2: *In a network with total Δ nodes, when δ nodes change their schedule, a message sent along the success path, built by our MR routing, has the average delay of*

$$h(p\bar{E}(\tau) + \bar{p}(\frac{q}{2} + \bar{q})(\bar{E}(\tau) + \bar{E}(n-1) - \bar{E}(n))),$$

where $p = 1 - (1 - \frac{1}{\Delta})^{\delta}$, $\bar{p} = 1 - p$, $q = \frac{\bar{E}(\tau)+\bar{E}(n-1)-\bar{E}(n)}{2\beta}$, *and* $\bar{q} = 1 - q$.

Proof: The cycle waiting time changes only when the schedule of nodes along the key links changes. Note that if the last relay node u does not change its schedule, no matter how fast the routing has been conducted before, the routing will wait until the wake-up of u occurs. The average cycle waiting time $\bar{E}(\tau)$ for each hop is the same, with the probability of p. We have the expected waiting time of $p \times h \times \bar{E}(\tau)$ for the whole path without being changed. Otherwise, with a probability of $\bar{p}$, the waiting time per hop can be either shortened or prolonged. For such a change at each hop, with a probability of q, a node can wake up earlier than the expected waiting time of the rest. We have $q = \frac{\bar{E}(\tau)+\bar{E}(n-1)-\bar{E}(n)}{2\beta}$. On average, the expected waiting time per hop is

$$q \times \frac{\bar{E}(\tau) + \bar{E}(n-1) - \bar{E}(n)}{2}.$$

Otherwise, with a probability of $\bar{p} \times \bar{q}$ for each hop, the routing will switch to an FW mode and wait for the next-hop node. In our MR routing, on average, after waiting $\bar{E}(\tau)$ and missing the target successor, there is another node available in time $\bar{E}(\tau) + \bar{E}(n-1)$. Thus, the delay for the entire path is:

$$ph\bar{E}(\tau) + h \times \bar{p} \times (q\frac{\bar{E}(\tau) + \bar{E}(n-1) - \bar{E}(n)}{2} + \bar{q}(\bar{E}(\tau) + \bar{E}(n-1) - \bar{E}(n)).$$

The statement is proven. ■

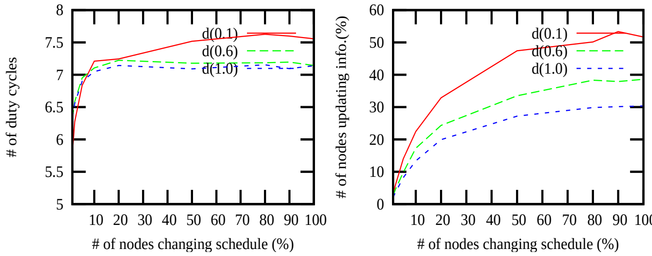
VII. SIMULATION

In this section, we will provide experimental results to show the substantial improvement of our MR routing (with the metric information under the GS schedule model), in achieving a path with less delay. We use a custom simulator built in C++. We use the results of the number of rounds in construction convergence and the number of nodes involved in an information update to illustrate the scalability of our metric evaluation. We also show that setting an appropriate cycle waiting time can substantially reduce the transmission time and the number of hops, improving the end-to-end performance. The results are compared with those of anycasting (denoted by FW) [2], [4], [9], [15] and dynamic programming (DP) [7] − the best solutions known to date for delay sensitive WSN applications. FW reduces 1-hop cycle waiting time only. DP is a solution in the reactive model, but has the cost and delay problem in the on-demand probing process. The above analytical results are also displayed here, in order to verify the losses and gains of our trade-off in developing a localized, scalable, and effective metric evaluation.

A. Simulation environment

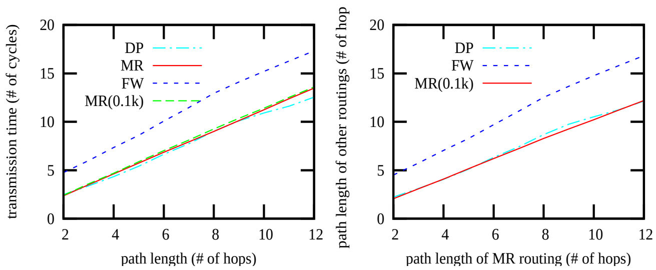
In the simulations, nodes with a communication radius of 10 meters are deployed to cover an “interest area” of 200m × 200m in the center, under different density models. We implemented the network model in section II. We deploy enough sinks in the center of the interest area so that each initiated communication has an available receiver. We keep the edge nodes alive to provide a complete, constant coverage in order to simulate the use of wireless mesh nodes in reality. In a real application, the sinks are distributed more sparsely so that the length of the path for each surveillance report is shorter, creating better performance in both information construction and routing process. We implement the information models for the FW, DP, and MR routings, respectively. The local minima are created by randomly turning off 1∼10% of the nodes and disconnecting their links. This also simulates the cases when the nodes fail or are affected by traffic. In the information construction for the MR model, we only collect 1-hop neighbor information at each cycle. For the DP model, each node collects the information from all nodes in the entire networks. This is a model retrieving global information of delay. Then, our MR routing, anycasting, and greedy forwarding under the DP model are applied, denoted by MR, FW, and DP, respectively.

We test all routings in the networks with different node densities ρ = 0.1, 0.6, and 1.0 nodes-per-m^2, denoted by



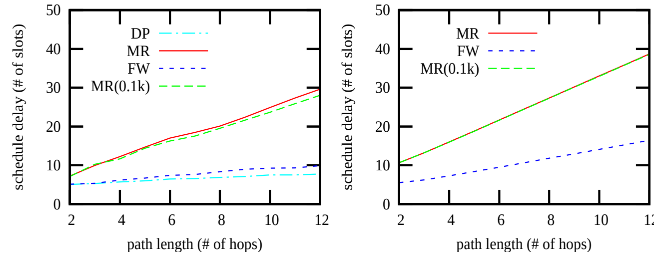
(a) # of duty cycles needed to converge (b) # of nodes updating metric information

Fig. 6. Construction cost in different models



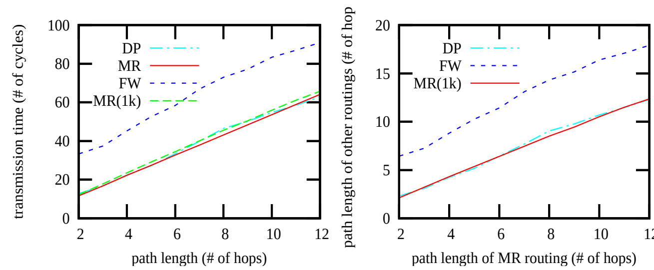
(a) Transmission time (# of cycles) (b) # of hops

Fig. 7. Transmission time in heavy duty networks (in cycles)



(a) Experimental results (b) Analytical results

Fig. 8. Scheduling delay in heavy duty networks (in slots)



(a) Transmission time (# of cycles) (b) # of hops

Fig. 9. Transmission time in light duty networks (in cycles)

d(0.1), d(0.6), d(1.0), respectively. We also test two kinds of networks, each with a different weight of duty cycles: one uses a 20% duty cycle (i.e., $\beta = 5$), which is relatively heavy, and the other uses a 4% duty cycle (i.e., $\beta = 25$), which is relatively light. In the heavy duty networks, denoted by MR(0.1k), we also change the schedule of 100 nodes to verify the impact of dynamic factors on the use of metric information in routing. For the light duty networks, denoted by MR(1k), we have more idle nodes, so we change the schedule of 1,000 nodes. Note that the schedule change will not affect the FW information model and its routing much, but it will force the DP model method to renew all information.

Based on our study, the routing does not need more than 12 hops unless a tremendous situation occurs, in which the network is usually disconnected. To compare MR, FW, and DP fairly, we only record the experimental results when each path is no longer than 12 hops long. For each case, more than 200 samples are tested. We collect and display results in terms of the number of hops that are made in the MR routing for the same pair of source and destination.

B. Scalability of information construction

Figure 6 (a) shows the converging speed of our metric evaluation. Figure 6 (b) shows the average number of nodes (in percentage of total deployed nodes) involved in the type-1 information updates for the MR routing. The results in the networks with different deployed density: d(0.1), d(0.6), and d(1.0), respectively, are displayed. Note that each type of status has similar results for the updates. The results show that increasing the scale of networks will not reduce the converging speed of information construction and will not incur more updates. The evaluation will still be affordable when many nodes change their schedule and need information updates. This proves the scalability of our metric evaluation compared with the information collection needed for the DP model.

C. Routing Performance

Figures 7 and 8 show the results of routing performance in the network with $\rho = 0.1$. Figure 7 (a) compares the transmission time of the DP, MR, and FW routings. The data is collected from heavy duty networks ($\beta = 5$), which have a high volume of traffic. It shows that our MR routing can achieve the same performance as the DP routing, even when some nodes change their sleep-wake schedule dynamically (MR(0.1k)). Both MR and DP have better performance than FW. Figure 7 (b) shows the number of hops achieved in the DP and FW routings compared with those in MR. As a result, the FW takes more hops. It proves the effectiveness of our strategy to reduce the transmission delay by achieving a path with less hops. Both Figures 7 (a) and (b) show the results of our MR routing in the dynamic situation MR(0.1k). The results confirm our expectation on the scalability of the MR routing. Figure 8 (a) shows the elapsed cycle waiting time along the entire path in different routings: DP, MR, and FW. Due to the use of the GS schedule, our MR routing will wait for longer time in each advance to achieve a better end-to-end performance. By using tuple of the cycle waiting time of FW, our MR routing can achieve a quicker message delivery. The results in dynamic situation MR(0.1k) show that the MR routing does not increase waiting time by introducing the FW mode to balance the dynamic changes, while the DP model completely fails to apply due to the cost

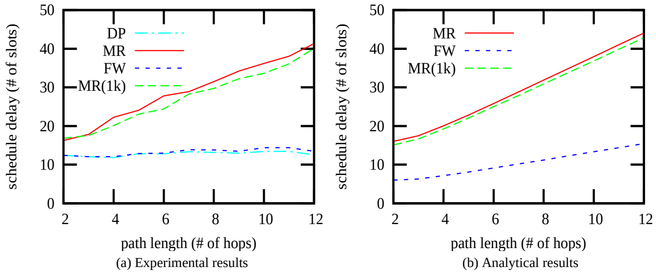


Fig. 10. Scheduling delay in light duty networks (in slots)

of information reconstruction. Using the network parameter $\rho = 0.1$, $\beta = 5$, and $\delta = 0.1k$, the analytical results of FW [15] and MR can be derived (see Figure 8 (b)). Compared with experimental results, the correctness and effectiveness of our metric information can be confirmed. Figures 9 and 10 show the routing performance results in the light duty network with node density of 0.6 node(s)-per-m^2, where $\rho = 0.6$, $\beta = 25$, and $\delta = 1k$.

VIII. Related Work

The existing delay-sensitive routings applicable to duty cycle systems have mainly focused on anycasting. In the routing schemes in [2], [4], a node simply drops the message when it has more than two detours and resorts to separated retrials. In many cases, the reporting process could fail to reach the sink while having too many nodes involved, disabling those nodes' ability to deliver any packet for other communications. Thus, the quality of routing cannot be guaranteed. The opportunity routing proposed in [1] adopts a random walk technique. Although the delay is bounded, it is too long for a real application. In [9], [15], the author assumes that the node density is high enough to have an awakened neighbor available for a greedy forwarding.

Such an assumption is too strong for our application, in which a sparse deployment is usually required due to limited rescue forces. When these methods are applied in our system, as indicated in [11], the local minima will occur. The hull routing (or perimeter routing) can be applied to determine the detours, making the subsequent hops to progress to the destination with a higher probability. Our early work [5] on local minima indicates that such detours can be avoided. A smart decision is made early to avoid using those nodes if their succeeding greedy forwardings are blocked. However, the routing requires accurate neighborhood information when such a decision should be made.

In [7], the dynamic programming (DP) is applied to determine the minimal delay in a routing from one node to its destination. However, it requires the information of each node for any possible forwarding path. For any possible destination selected by a source, the information of the entire network needs to be collected. Moreover, when any node changes its schedule due to interference or other dynamic factors, the information needs reconstruction, which requires a long time to converge in a system without global control [13]. This approach is impractical to real delay-sensitive applications. Therefore, a more accurate and more effective description of dynamic variation is needed.

IX. Conclusion

A guided cycle waiting model, GS, is provided to optimize the successor selection in greedy forwarding. A new metric is repetitive under the GS model to build a path with less delay, even when many nodes are changing their schedules. In our future work, we will study the throughput and energy cost of our approach, since they are directly related to transmission time and number of hops in routing, and provide more comprehensive experimental results. We will also study other performance factors, implement them in our metric evaluation, and provide a more systemic study of routing performance in delay-sensitive applications. The results will be expected to implement in a real testbed.

Acknowledgment

This work was supported in part by NSF grants CNS 0626240, CCF 0840891, and CCF 0936942. Contact E-mail: zjiang@wcupa.edu.

References

[1] P. Basu and C. Chau. Opportunistic forwarding in wireless networks with duty cycling. *Proc. of ACM workshop on Challenged Networks*, 2008.

[2] R. Choudhury and N. Vaidya. Mac-layer anycasting in ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 34(1):75–80, 2004.

[3] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. *Proc. of the 4^{th} International Symposium on Information Processing in Sensor Networks*, 2005.

[4] S. Jain and S. Das. Exploiting path diversity in the link layer in wireless ad hoc networks. *Proc. of WoWMoM*, pages 22–30, 2007.

[5] Z. Jiang, J. Ma, W. Lou, and J. Wu. An information model for geographic greedy forwarding in wireless ad-hoc sensor networks. *Proc. of IEEE INFOCOM*, pages 825–833, 2008.

[6] C. Joo, X. Lin, and N. Shroff. Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks. *Proc. of IEEE INFOCOM*, pages 1103–1111, 2008.

[7] J. Kim, X. Lin, N. Shroff, and P. Sinha. Optimal anycast technique for delay-sensitive energy-constrained asynchronous sensor networks. *Proc. of IEEE INFOCOM*, 2009.

[8] Y. Ko and N. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *Proc. of ACM/IEEE MobiCom*, pages 66–75, 1998.

[9] S. Liu, K. Fan, and P. Sinha. CMAC: An energy efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks. *Proc. of IEEE SECON*, pages 11–20, 2007.

[10] M. Lukic, B. Pavkovic, N. Mitton, and I. Stojmenovic. Greedy geographic routing algorithms in a real environment. *Proc. of the Fifth International Conference on Mobile Ad-Hoc and Sensor Networks*, 2009.

[11] S. Nath and P. Gibbons. Communicating via fireflies: Geographic routing on duty-cycled sensors. *Proc. of the ACM IPSN'07*, pages 440–449, 2007.

[12] J. Redi, S. Kolek, K. Manning, C. Partridge, R. Rosales-Hain, R. Ramanathan, and I. Castineyra. JAVeLEN - an ultra-low energy ad hoc wireless network. *Ad Hoc Networks*, 6(1):108–126, 2008.

[13] J. Tate and J. Bate. An improved lightweight synchronisation primitive for sensornets. *Technique report of University of York*, 2009.

[14] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.

[15] M. Zorzi and R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance. *IEEE Transactions on Mobile Computing*, 2(4):337–348, 2003.