A Metric for Routing in Delay-Sensitive Applications of 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 an enhanced forwarding algorithm for delay-sensitive applications of wireless sensor networks, under the duty cycle model. A comprehensive study of message transmission delay with asynchronous link connections (on propagation detour, channel access waiting, etc.) brings an insight to a new metric for routing selection in the multi-hop system; in contrast to the one that is optimized on the Dijkstra's shortest path algorithm. We provide a simple local description of such a metric with the consideration of the cost in information construction, which is impacted by source and destination's relative positions and other dynamic factors in the networks. The measurement at each node interprets a global optimization and can be shared simultaneously in many different routing cases. Applying our approach deployed with a uniform distribution in the networks, we illustrate the substantial improvement of our information-based routing in reducing delay in both analytical and experimental results. Our approach focuses on a reliable, scalable solution in highly dynamic situations. The results are compared with those best known to date.**

***Index Terms*—Delay, forwarding path, distributed algorithms, wireless sensor networks.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have great long-term economic potential and the ability to transform our lives. Consider the WSN application for emergency disaster recovery. Before delivering food, water, and medicine, and even dispatching doctors to the survivors, we need to know where and how many of these things are needed. The most efficient way is to send trained animals and robots carrying portable equipment to search for victims. The environment will be sensed by the equipment and the information will be collected by the wireless communication in order to estimate the amount of need at the base. It is life-critical to send surveillance results without any unnecessary delay. Affected by the unstable nature of wireless signal and the complex terrain of deployment area, the surveillance reports cannot be sent to sink directly in many cases and require a multi-hop relay path. In the 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 will be 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], [11] have adopted the asynchronous sleep-wake scheme [9], [13] to reduce the overhead of neighbor synchronization. Each node is allowed to go to sleep periodically to save energy and therefore prolongs its lifetime. In such a duty cycle system, the sleep-wake schedule at each node uses a predictable pseudo-random sequence, but is independent of those of other nodes. Thus, no synchronization is required. Prior work in the literature has proposed the use of *anycast*, where each node forwards the packet to the first candidate node that wakes up [2], [9], [14]. However, as indicated in [7], the first candidate node that wakes up may not have a small 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.

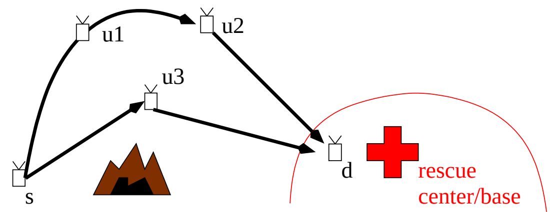

Fig. 1. Multiple-hop unicasting in the disaster recovery application.

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. Therefore, we need to determine what neighbor to choose for any routing decision. This requires more information than just cycle waiting time.

Our work proposes a new metric in the duty cycle systems. The transmission time from the designated neighbor to the destination can be estimated in its evaluation, alternatively called information construction, so that a smart selection in the greedy forwarding can be made (in a localized method) to achieve the global optimization of the end-to-end delay. Unlike existing routings [1], [10], [14] in duty cycle systems that ignore the transmission time wasted in detours, a more comprehensive, balanced measurement is proposed. We illustrate three challenges of asynchronous neighbor connection, which distinguish our solution from others.

First, how does each node collect the information and then control the cost of its collection process? Due to many dynamic factors such as signal fading, communication jamming, and interference, the access of an available channel may be deferred to the next cycle in the predictable schedule sequence. Without using any global information/control, the cycle waiting time will be accumulated for the estimation of transmission time. In order to complete the information collection quickly, we need to control the scalability of information collection within a limited area (i.e., region), even when many links change their availability dynamically.

Second, how can the granularity of such a region be determined? The metric value of a node changes when the relative locations of the source and destination update. We need a relatively stable region in metric evaluation to avoid changing the metric value too often and too quickly.

Third, how does the designated metric information reflect the quality of a routing? We need to study the effectiveness of the localized processes under our metric model, and optimal greedy forwarding that achieves the least end-to-end delay. The proposed information-based routing must still be applicable when many nodes have changed their access time, but have not yet updated their information in other nodes.

Our metric provides a simple value $M \in [0, 1]$ at each node. "0" indicates a stuck node where the greedy forwarding and its succeeding routing will be blocked by local minima. "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 path built from this node to a nearby permanently awakened node, such as the sink or edge nodes of the networks. As usual, such nodes always remain active to provide a complete, constant coverage. They also play the role of relay ends. The larger the value of M, the less delay in routing there will be. Its construction reuses the beacon message that is exchanged among neighbors, at no extra cost. M's update is dependent on the predictable duty cycles of all 1-hop neighbors, not just a single neighbor connection. It can remain stable even when many nodes change their duty cycles. Such a construction process follows the gradient, starting from any region in the networks that contains permanently awakened nodes. The region size is a trade-off between precision and construction cost.

Like a lighthouse guiding boats to the harbor at night, but not necessarily illuminating everywhere, such a metric value guides the routing to select a direction with relatively less delay, approaching the destination greedily. Our metric model is constituted in a reactive model, in which the cost and delay in the probing process of the proactive model (e.g., [7]) can be saved. When the dynamics incur a metric value change, any in-progress routing heading into the propagation can make an alternative selection to avoid a deadly wait for a neighbor that did not wake up as scheduled. Strictly speaking, the approach supports segmented routing in a dynamic situation, while the routing changes its indirect referee after the relative position of the destination to the current node changes. Our analytical and experimental results show the effectiveness of our metric in achieving reduction on the end-to-end delay in greedy forwarding routing.

Our contributions are threefold: **(a)** The detour delay that has been ignored in the existing routings in duty cycle systems is considered in our metric. A balanced, comprehensive measurement is provided for each localized routing decision to achieve better end-to-end performance. **(b)** Unlike proactive methods requiring a probing process to fetch the global information, our metric evaluation is conducted under a reactive model. The problems of such implementation are addressed. A balance point of the tradeoff between precision and construction cost is proposed. **(c)** 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.

II. PRELIMINARY

A. *WSNs with 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 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, respectively.

Our networks are deployed to cover a disaster area. Assume that nodes are deployed on a 2-D plane. Nodes can report to the sink with satellite signals or wireless mesh nodes along the edge with an access to the Internet. Usually, they are set in safe areas and do not have power inefficiency. We keep these sites/nodes 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 the pseudo random sequence with a preset seed. Each time a node u wakes up, it will initiate 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, metric values, etc. Thus, each node can predict the next appearance of the other.

A short message system is adopted in our networks. The packet will advance 1-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 such a 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), for the routing path $s - u_1 - u_2 - d$ achieved in Figure 1. A node u will keep beaconing its neighbors cycle by cycle until there is an available neighbor for forwarding. Such a scheme has been used in existing *anycasting* [4], [9]. It is called a "first-awaken" strategy, simply denoted by FA.

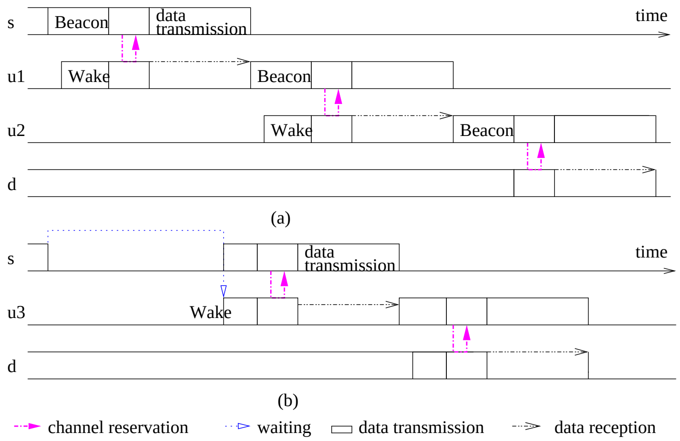

Fig. 2. Time sequence for the sample routing in Figure 1. (a) Path $s-u_1-u_2-d$ uses successor selection without waiting time, and (b) Path $s-u_3-d$ uses selection with an appropriate wait, which requires accurate prediction of wake-up time.

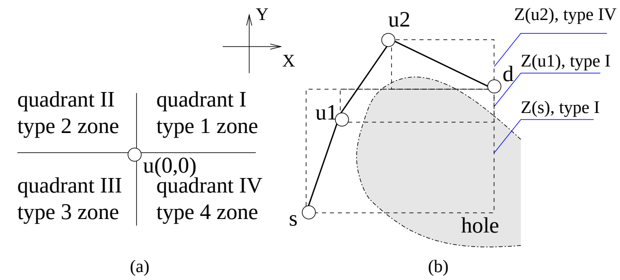

Fig. 3. Definition of (a) forwarding zones, and (b) request zones.

Algorithm 1 (LAR1 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 $v \in N(u) \cap Z_k(u, d)$.

The system also supports the schedule change that is required for the performance optimization in [6], [12]. A node u can select one of its neighbors v and expect such a node to wake up after a certain time, following its own predictable schedule. The node 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 the contact with v. Such a waiting period, also called the cycle waiting time, is the time difference between u's appearance and v's coming appearance. It is denoted by $t(u, v)$. After the message is sent, u will schedule back to its original sleep-wake sequence. Such a guided schedule is denoted by GS. In our networks, after the target neighbor is selected, the corresponding waiting time is set with our metric evaluation. The example of data transmission with an appropriate wait is shown in Figure 2 (b), for the routing path $s-u_3-d$ built in Figure 1. However, when v misses its schedule or is no longer available at the expected time, u will switch to an FA mode.

In the network deployed in the uniform distribution, the random sequence is determined by the rate λ of the Poisson process, with which each node u wakes up. We preset the rate 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 only to maintain $t(u, v)$. However, in this paper, we use a global time in slots to simplify the discussion.

B. Limited greedy forwarding

As described in LAR scheme 1 in [8], the selection of a 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 the 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 $Z_i(u, d)$ $(1 \leq i \leq 4)$. Respectively, each corresponding quadrant is called a type-i *forwarding zone*, denoted by $Q_i(u)$. A greedy advance with $Z_i(u, d)$ is called the 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 of such a routing, which is denoted by LAR1. At each hop, a successor is selected with the request zone, by the rectangle area with two opposing corners being the current and destination nodes. As indicated in [5], a single path of LAR1 may experience different types when the relative position of d to the intermediate node changes and d is located in different types of request zones.

Note that LAR1 is not conducted under the GS model. Compared with the region that contains all possible successors in anycasting, the forwarding zone in LAR1 has a limited area and reduces the routing flexibility. However, it has a simple structural regularity and each of its successful advances is a greedy forwarding. In this paper, we will present our metric information for LAR1 routing so it can be applied under the GS model. The information-based routing can achieve better performance than anycasting, in terms of the transmission delay (i.e., the speed of routing). In this way, we show the impact value of our metric. Table I summarizes all of the notions used in this paper.

III. Problem and the Proposed Idea

Note that our goal is to achieve the optimization on delay for a single routing, instead of the mean time of delay. Unlike those methods determining the wake-up time to fit the subsequent path, our approach develops the information for the current node u to select the best subsequent path from its 1-hop neighbors. The validation of such information relies on the existence of an evaluation function f satisfying the following constraints:

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 1-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.

- $0 \leq f(u, d) \leq 1$ and $f(u, u) = 1$.
- $f(u, d) = 0$ when keeping greedy forwarding along the path from u to d is impossible; that is, detours are needed.
- For any two paths $\{u, u_1, u_2, \cdots, u_k, u_{k+1} = d\}$ and $\{v, v_1, v_2, \cdots, v_{\Bbbk}, v_{\Bbbk+1} = d\}$, we have

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

iff

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

The function f is normalized to a value $\in [0, 1]$. The larger the value of f, the less delay in routing there will be. A failure of routing implies an ∞ ($= \frac{1}{0}$) delay. Obviously, $f(u, u) = 1$ implies no delay for u to send itself a message. Note that k and $\Bbbk$ are not necessarily the same. Thus, f is an evaluation function that includes (1) the delay caused by cycle waiting time, (2) other delay costs in message propagation (i.e., time of data transmission at each hop $T(u, v)$), and (3) the number of hops along the entire path k and $\Bbbk$. As usual, a shorter path ($k < \Bbbk$) takes less transmission time. Such a path will be selected with a larger evaluation value. Note that the normalization of evaluation $\in [0, 1]$ will possibly cause a round-off error, and cannot catch the exact delay time. Indeed, the evaluation is a neighbor's selection with a relatively high value, regardless of its numerical value. More importantly, such normalization reduces the cost in information storage and exchange, fitting a resource constrained WSN application.

Theorem 1: *$f(u) = f(u, *)$ has multiple values and the number of different values $| f(u) |$ satisfies*

$$| f(u) |= O(k \bullet \varpi) \gg O(1),$$

where "$*$" stands for any possible destination d, k is the number of local minima existing in the networks, and ϖ is the number of nodes along the longest boundary of a local minimum, i.e., the minimum set of nodes that encloses a local minima and its stuck nodes.

Proof: First, considering a convex hull H that encloses one local minimum and has a perimeter length b, $b = O(\varpi)$. Accessing any node contained inside a hull will definitely cause a block in the greedy forwarding. For any H, we can always find a node u outside so that a set $\bar{A}$ of $O(\frac{b}{4})$ nodes along the hull will have $1 > f(u, v) > 0$ (for any $v \in \bar{A}$). Thus, $O(\frac{b}{4}) = O(\varpi)$. Since the path from u to each v is progressive and can be conducted in a greedy forwarding, the delay will be different. Otherwise, all nodes along a hull will have the same distance to any node existing in the networks; that is, such a hull is a single node. This leads to a contradiction in the definition of the hull. Considering k different local minima located in different places, their $\bar{A}$ sets will be different. The delay of the paths from u to $O(k \bullet \varpi)$ nodes is different. Therefore, the statement is proved. ■

As we proved in Theorem 1, $f(u) = f(u, *)$ is a multiple value function. The evaluation of f is still not applicable to a reactive model due to the unknown d, causing the maintenance cost for a large number ($O(k \bullet \varpi) \gg O(1)$) of records at each node. Our task is to find a feasible implementation of f under the GS model: $f_L(u)$, which satisfies the following constraints:

- $f_L(u) = f_L(v) \circ T(u, v)$ where $v \in n(u) \subseteq N(u)$.
- $| f_L(u) |= C \rightarrow 1$.
- $cod(f_L) \subset cod(f)$.

The first constraint requires each node to constitute its evaluation value with 1-hop neighbor information. To achieve a practical solution, the number of evaluation records maintained at each node must be reduced; that is, $| f_L(u) | \simeq C \rightarrow 1$. This will reduce the complexity of the decision algorithm and the cost in information construction. Note that a set of nodes may have the same f_L value. Changing the location of destinations within a region of these nodes will not incur a new evaluation. In other words, there is no need for information reconstruction, in contrast to the information evaluation in [7] based on Dijkstra's shortest path algorithm. Due to the use of such stable region, $cod(f_L) \subset cod(f)$. f_L may have a loss of precision when it does not have the exact delay measurement as the global information f does.

In our approach, the precision remains, while $cod(f_L)$ and $| f_L(u) |$ are minimized. The idea is illustrated in Figure 4 (a). For each region divided at u, we use one designated f evaluation value only. Usually, it is the delay measured from u to the closest permanently awakened node, say v. Like the lighthouse guiding boats, such a value guides the routing to advance to the destination greedily with 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 and the referee v. By applying the forwarding zone of LAR1 routing, we have $C = 4$. The details of this f_L function for the duty cycle networks (under the GS model) will be discussed in the metric in the next section.

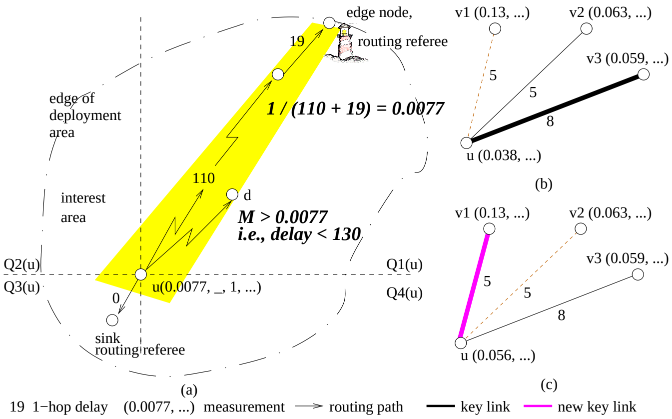

Fig. 4. Illustration of (a) the definition of $M(u)$, and (b) and (c) its updates.

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. Each other node v sets $M(v)$ to a changeable $(0,\ 0,\ 0,\ 0)$.
2) Then, each node will have stable status by applying Eqs. (1) and (2) with a beaconing scheme.
3) In case any node changes its schedule, the above process with Eqs. (2) will be applied.

IV. Metric Evaluation

Our new metric describes the minimal transmission 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 will likely have. Such a value also implies a higher value (i.e., less delay) of a successful routing to reach a closer destination. In the following, we will discuss a metric and its details in Algorithm 2. The metric is used by each node u to determine its evaluation value.

According to different types of forwarding zones, our metric is a 4-tuple ($C = 4$). Permanently awakened nodes set their fixed delay ratio 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 timeless delay ($= \infty = \frac{1}{0}$). After this, u will update $M_i(u)$ once with:

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

where $v \in n(u) \cap Q_i(u)$, and the selected link $\{u, v\}$ is called the *key link* of u for $M_i(u)$. After this, $M_i(u)$ will stabilize by repeating:

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

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 2: *$M_i(u)$ is a required evaluation function $f_L(u)$.*

Proof: Due to the definition of $M_i(u)$ in Eq. (1) and Eq. (2), $0 \leq M_i(u) \leq 1$ and $M_i(u)$ satisfies the first constraint of the localized implementation. 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; i.e., a detour is needed. According to Eq. (1) and Eq. (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 transmission time from u to the closest permanently awakened node v. This satisfies the last constraint of function f. Due to the use of different forwarding zones ($1 \leq i \leq 4$), we are left with $C = 4$. Therefore, the statement is proven. ■

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

Basically, the 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 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).

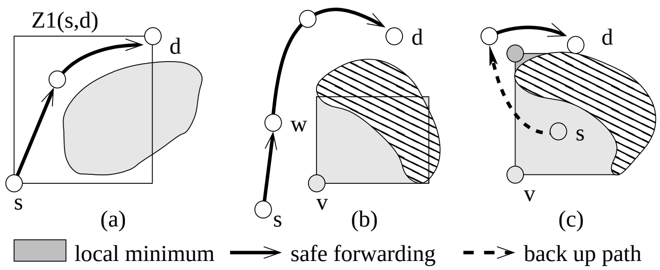

Fig. 5. Samples of the MR routing.

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) After v is selected, wait $t(u,v)$ until it wakes up.
5) If v misses the contact at that expected time, u switches to an FA 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.

Theorem 3: *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 in the above process at u's preceding node. This leads to a contradiction. ∎

For a 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 away from 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 node v is selected, u will wait $t(u,v)$ until v wakes up. Due to many dynamic factors, v could be unavailable at that time. Then, u switches to an FA mode. It will keep waiting until $n(u) \neq \phi$. A node $\in n(u)$ with less detour, indicated by $t(u,v) + \beta + \frac{1}{M_k(v)}$, will be selected. It is implemented in a superseding rule on the candidates selected in Q_k.

When the source has the tuple $(0,0,0,0)$, the network may be disconnected. In a cautious way, our MR routing will stop and wait to send out a message 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 any casting. 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 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 when 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 schedule sequence, it has been well known (e.g., [9]) that a node will take on 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*

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

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 $\in [0, h \bullet 2\beta]$. For each node u along the path that is i-hops away from d, its physical distance to d, ξ, is in the range $[0, i \bullet r]$, where r is the radius of node u. On average, $\xi = \frac{i \bullet r}{2}$. Shown by [9], [14], 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 $\frac{2\arccos(\frac{r/2}{\xi})}{2\pi} \bullet (\pi r^2)$ and can be estimated by $\frac{\arccos(1/i) \bullet r^2}{h}$. 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

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

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 $h \bullet 2\beta$ and the forwarding set size $\tau = n/3$, the average cycle waiting time in an MR routing is

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

. ■

Next, we will study the dynamic situation when δ out of Δ nodes in the networks are affected by dynamic factors and change their schedule randomly. The following corollary proves a balanced cycle waiting time is achieved when we introduce an FA mode to avoid deadly waiting for any expected node that cannot appear at its scheduled time. Note that when the routing uses stable metric information, it can ensure the path due to the fixed key links in use. The cycle waiting time along such a stabilized subsequent path can be determined by Corollary 1. The following corollary focuses on the path where, at each hop, the information is out of synchronization and inconsistent. The result shows that our MR routing will not wait too long when it misses the contact because the outdated information is used in prediction. Actually, the MR routing speeds up in such a highly dynamic situation because of the use of an FA mode after the miss. Note that without an appropriate wait, directly applying step 5 in Algorithm 3 with our metric information will be a special case of anycasting, causing even worse delay.

Corollary 2: *In a dynamic network with total Δ nodes, when δ nodes change their schedule, a message sent along the success path built in 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. Considering the probability of such case, p, we have the expected waiting time of $p \bullet h \bullet \bar{E}(\tau)$ for the whole path. 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

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

Otherwise, with a probability of $\bar{p} \bullet \bar{q}$ for each hop, the routing will switch to an FA 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 $\bar{E}(\tau) + \bar{E}(n-1)$. Therefore, the delay for the entire path is:

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

The statement is proven. ■

VII. Simulation

In this section, we will provide the 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 information updates 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 (FA) [9], [14] and dynamic programming (DP) [7] $-$ the best solutions known to date for delay sensitive WSN applications. The above analytical results are also displayed here, in order to verify the loss and gain of our trade-off in developing a localized, scalable, and effective metric evaluation.

Simulation environment. In the simulations, nodes with a communication radius of 10 meters are deployed to cover an interest area of 200m $\times$ 200m in the center, under different density models. We realized the network model in section II and the corresponding GS model (with asynchronous node actions). We deploy enough sinks in the center of 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 real application, the sinks are distributed more sparsely, so that the length of path for each surveillance report is shorter, creating better performance in both information construction and routing process. We implement the information models for the FA, DP, and MR routings, respectively. The local minima are created by randomly turning off 1$\sim$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 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, FA, and DP, respectively.

We test all routings in the networks with different node density (ρ = 0.1, 0.6, and 1.0 nodes-per-m^2, denoted by d(0.1), d(0.6), d(1.0), respectively). We also test two kinds of networks, each with different duty cycle weights: 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

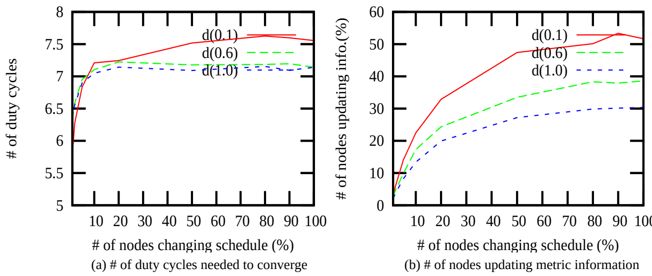

Fig. 6. Construction cost in different models

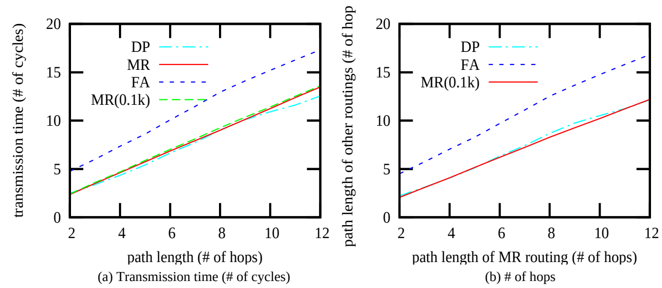

Fig. 7. Transmission time in heavy duty networks (in cycles)

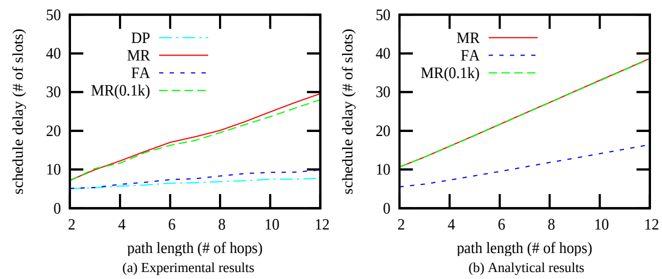

Fig. 8. Scheduling delay in heavy duty networks (in slots)

is relatively light. The performance results are compared to show the substantial improvement of our new routing, and the effectiveness of our metric. 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 FA 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 until a local minimum occurs under a tremendous situation, in which the network is disconnected. To compare MR, FA, and DP fairly, we only record the experimental results when each path is no longer than 12 hops. For each case, > 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.

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 of the number of 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 cost is related to the complexity of local minimum configuration, not to the network density or scale. It will still be affordable when many nodes change their schedule and need information updates in other nodes. This proves the scalability of our metric evaluation, compared with the information collection needed for the DP model.

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 FA 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 FA. Figure 7 (b) shows the number of hops achieved in the DP and FA routings, compared with those in MR. As a result, the FA 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 FA. Due to the use of the GS schedule, our MR routing will wait for an expected longer time to achieve a better end-to-end performance. By using triple the cycle waiting time of FA, 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 FA mode to balance the dynamic changes, while the DP model completely fails to apply due to the cost in information reconstruction. Using the network parameter $\rho = 0.1$, $\beta = 5$, and $\delta = 0.1k$, the analytical results of FA [14] 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

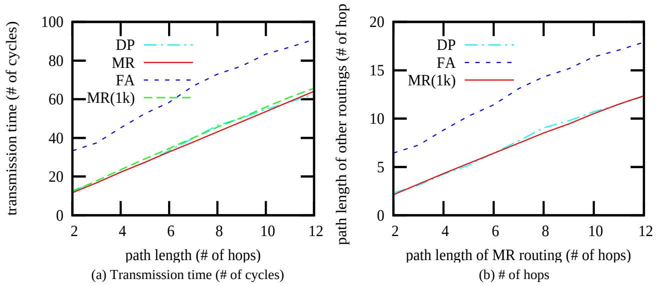

Fig. 9. Transmission time in light duty networks (in cycles)

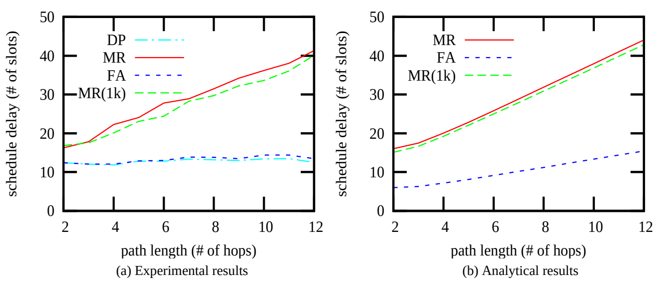

Fig. 10. Scheduling delay in light duty networks (in slots)

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 and is not suitable for our application. In [9], [14], the author assumes that the node density is high enough to ensure there is always 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 [10], 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 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 [12]. Such a collection of delay information makes this approach impractical to real delay-sensitive applications. Therefore, a more accurate description of dynamic variation is needed so that the metric information can be relatively stable when many nodes change their schedule in a highly dynamic network.

IX. Conclusion

A localized cycle waiting model, GS, is provided to optimize the successor selection in greedy forwarding. A new metric is provided 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 systematic study of routing performance in delay-sensitive applications.

Acknowledgment

This work was supported in part by NSF grants CNS 0422762, CNS 0434533, CNS 0531410, 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 dtecting 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] J. Kim, X. Lin, N. Shroff, and P. Sinha. On maximizing the lifetime of delay-sensitive wireless sensor networks with anycast. *Proc. of IEEE INFOCOM*, 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] S. Nath and P. Gibbons. Communicating via fireflies: Geographic routing on duty-cycled sensors. *Proc. of the ACM IPSN'07*, pages 440–449, 2007.

[11] 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.

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

[13] 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.

[14] 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.