2012

A Quick and Reliable Routing for Infrastructure Surveillance with Wireless Sensor Networks

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

Jie Wu
Temple University

Minyi Guo
Shanghai Jiaotong University

Zhenping Zhao
GreatBit, Inc., Nanjing

Donghong Wu
Shanghai Jiaotong University

Recommended Citation

Jiang, Z., Wu, J., Guo, M., Zhao, Z., & Wu, D. (2012). A Quick and Reliable Routing for Infrastructure Surveillance with Wireless Sensor Networks. *31st IEEE International Symposium on Reliable Distributed Systems*, 255-260. http://dx.doi.org/10.1109/SRDS.2012.83

A Quick and Reliable Routing for Infrastructure Surveillance with Wireless Sensor Networks

Zhen Jiang*, Jie Wu†, Minyi Guo‡, Zhenping Zhao§, and Donghong Wu‡
*CS Dept., West Chester University, West Chester, PA 19382, USA
†CIS Dept., Temple University, Philadelphia, PA 19122, USA
‡CS Dept., Shanghai Jiaotong University, Shanghai, P.R.China
§GreatBit Inc., Nanjing, P.R.China

***Abstract*—Recent work has addressed the impact of link burstiness as a local phenomenon. To achieve better end-to-end performance in a mission-critical application of wireless sensor networks, such as the infrastructure surveillance system, the routing faces three issues in message delivery: whether this link burstiness has any mutual impact with other delay factors, whether a failed transmission is worthy to retry, how the number of retrials is determined so that the delay cost can be calculated for the use of an alternative path. In this paper, we provide a fully distributed solution with a new measurement. Our routing uses this reference information to select a relatively better successor candidate in each 1-hop advance. Our analytical and experimental results show the effectiveness of this approach in reducing the end-to-end delay, even though many unexpected changes of link quality or node availability occur.**

I. INTRODUCTION

In many applications, WSNs are deployed to monitor the impact of the forces of nature on the infrastructure safety, e.g. bridge collapse detection [6]. It is very important for the routing to send surveillance results without any unnecessary delay, which can be caused by extra transmissions in a detour or an unexpected wait for the availability of the relay successor and the corresponding link connection.

To reduce the waiting time in the neighbor synchronization, recent systems have adopted the asynchronous sleep-wake scheme [12]. By obtaining the pseudo-random seed and the last active slot of the receiver, a node sending the message can easily forecast the waiting time until the next active time of the receiver, i.e., the *cycle waiting time* (CWT). The delay accumulated along any selected path is determined by not only the hop distance, but also the CWT at each hop.

However, to precisely measure the accumulated delay of all possible succeeding paths for the routing decision in the real deployed detection system is not easy. The structure and materials (of the bridge) also affect the signal transmission and its quality. The fading, reflection, and interference become more complicated and cause link burstiness [13]. The unpredictable link quality can affect any possible path for a given routing and force a new decision to be made. The ETX-like accumulation in the heuristic manner (e.g., [1], [5]) will have cost and consistency problems in the reconstructions. If inconsistent information is used in the selection, the routing may easily be trapped in a dead zone (sometimes called a "dead spot") along its relay path. Maintaining the information for any unexpected change for each possible path is costly.

Our target is to measure not only the delay cost of retransmissions needed along any bursty link, but also its impact with other factors on the end-to-end delay performance. Unlike anycasting [2], [4] to focus on the local 1-hop solution, our work aims to the accurate information for the entire path and its use to reduce the end-to-end delay. Unlike [1], [5] that relies on a tree-based global information model, we focus on the generic solution in a fully distributed manner, in order to achieve a reliable solution for the global optimization in reducing the end-to-end delay. It is required to determine **a)** whether this transmission block has mutual impact with other factors, such as the CWT and deployed holes (also called local minima), to defer the routing, **b)** whether a failed transmission is worthy enough to retry in terms of the end-to-end delay performance, and **c)** how many retransmissions are needed. The challenge to implement the global optimization on end-to-end performance in an "everyone model" that *every* node needs the delay information of *all* possible paths to *any* destination in the entire network, after *each* dynamic change of link quality or node availability occurs. Our approach breaks through the above measurement barriers of cost and inconsistency, and achieve reliable and quick routing in the networks that are prone to unpredictable errors and interrupts.

Our contributions are fourfold: First, we conduct a comprehensive **study of delay** to avoid wasting any unnecessary time on the detour by the deployment holes, the CWT in the asynchronous sleep-wake scheme, and retransmissions by the link burstiness. This is the first attempt to study the local description of the global impact of link burstiness as well as other local phenomena on the end-to-end performance. Second, we provide **a new measurement** at each node, inferring the minimum delay-cost of any path passing through it. The information update is irrelevant to the position of the source and the destination in routing, so the information can be relatively stable, avoiding **the cost** in existing models. We achieve the the simplicity of structure regularity and the precision of minimum-delay-path description by balancing out the routing flexibility. Third, we provide **our new routings** based on our delay measurement. One uses a light-weight information collection process for the retransmission reservation, and the other is based on the complete measurement of link quality. The routing forwarding prefers to a neighbor with a relatively

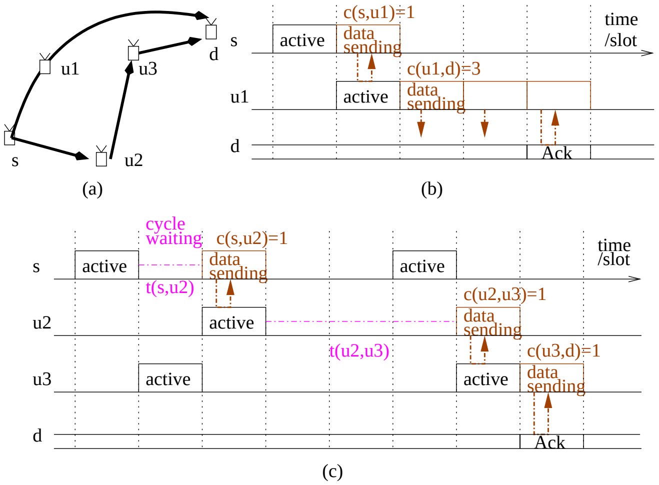


Fig. 1. Samples of emergency reporting in our duty cycle system. (a) Link connections, (b) data retransmission, and (c) cycle waiting and transmission time in hop advances.

higher measurement value and will adopt its succeeding path. Such a routing has the ability of self-adjustment and can still be effective, when the information in many nodes is not up-to-date and the corresponding update propagates head-to-head with the routing forwarding. This approach mitigate **the impact of inconsistency** in existing models against the link burstiness and other dynamics in the networks. Fourth, we focus on an "everyone" model, in which each node applies the same generic process in a fully distributed manner, in order to achieve **a reliable solution** of end-to-end performance vs. dynamic link burstiness.

Due to the limited space here, all the proofs are omitted but the details can be found in [7]

II. PRELIMINARY

A. *Network model*

A WSN 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 that can communicate with node u. $n(u)$ ($\subseteq N(u)$) denotes the set of neighbors that are currently awakened with u under the duty cycle model. Each node u has the location (x_u, y_u), simply denoted by $L(u)$. $s(x_s, y_s)$ and $d(x_d, y_d)$ are the source node and the destination sink, respectively.

Data can report to sinks which are set in safe areas and do not have power inefficiency issues. We keep them awakened to provide complete coverage constantly. Other nodes inside the deployed area will periodically go to sleep in a cycle with an average length of β. Each time a node u wakes up, it initiates a beaconing process to connect nodes within its communication range, in order to monitor any unexpected change in the topology. When a neighbor v receives this beacon message ($v \in n(u)$), it will respond to u and share the information, including the seed of the pseudo random sequence, the last wake-up time, and metric values. Each node can predict the next appearance of its neighbors.

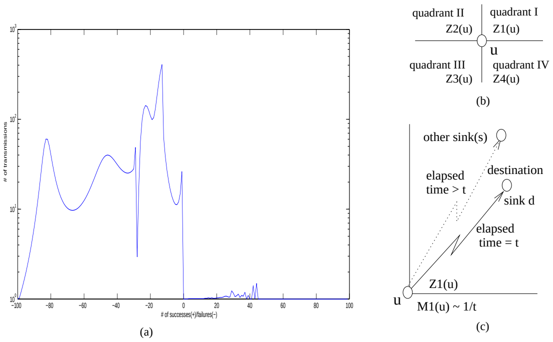


Fig. 2. (a) Link quality measured by the extended CPDF, (b) definition of request zones, and (c) its use to determine measurement M.

Our alert message is attached with the beacon message and will advance one-hop in each cycle. When a node u confirms the availability of every neighbor $v \in n(u)$, the quality of link $u - v$ can also be ensured for message delivery. A link with a string of consecutive successes or failures has a relatively stable quality for the successor selection in the upcoming relay. This selection will be attached with that routing message. When the current node u can receive the acknowledgement from that selected successor, u has accomplished the message relay and will schedule back to its original sleep-wake sequence. Otherwise, a retransmission is needed at the next active time of that selected successor, unless the quality of the corresponding link is too poor indicated by our metric evaluation. In such a case, a backup successor of a relatively high quality will be selected in the "first awaken" mode, in order to avoid waiting too long.

Samples are shown in Figure 1. Along the path $s - u_1 - d$, the source s prepares the report during its active slot and sends it out immediately in the next slot. In the well known "first-awaken" strategy that is adopted in anycasting [2], this node will keep beaconing its neighbors slot by slot until a neighbor, say u_1 in Figure 1 (b), successfully receives the information and becomes the successor for forwarding. Due to the burstiness of link $u_1 - d$, the relay of u_1 requires 3 trials. After the link $s - u_1$ becomes unstable, the source s in Figure 1 (c) can select the path $s - u_2 - u_3 - d$ and can expect u_2 to wake up in its own schedule after a number of slots $t(s, u_2)$. This node s will hold the report and switch to sleep mode, allowing other nodes in its neighborhood to communicate. After $t(s, u_2)$, s will wake up to continue the communication with u_2. When the expected successor misses its scheduled time or the transmission fails because of the link burstiness, the relay will be tried in our retransmission phase first, and then the anycasting backup phase will be applied.

B. *Measurement and prediction of link quality*

To estimate the delay cost in the retransmission, we need to concisely describe the link behavior and predict the success of its upcoming utilization. The conditional packet delivery function (CPDF) [11], [14] provides a succinct way to describe the durations of packet delivery correlations. In our measurement, the CPDF $c(u,v)$ is extended to the description of the number of transmissions needed to send a packet along the directional link $u \rightarrow v$, given k consecutive successes (for $k > 0$) or failures (for $k < 0$), simply denoted by $c_{u,v}(k)$. Sample of our link quality measurement from the real deployment environment is shown in Figure 2 (a). After 45 consecutive successes of d's appearance in $n(u_1)$, we have $c(u_1,d) = c_{u_1,d}(45) = 3$, which helps the routing to reserve a total of 3 transmissions (2 are for retransmissions) to deliver the data along the link $u_1 \rightarrow d$. Since sink d stays awake, the transmission is expected to finish in 3 slots. When an intermediate relay node is the target of a retransmission, its next wake-up is expected after β slots on average. The total delay cost in the relay from u to v is $t(u,v) \times c(u,v)$ slots (i.e., $\beta \times c(u,v)$ on average).

C. Anycasting

As described in LAR scheme 1 in [10], the selection of the forwarding successor can be limited within the quadrant (see Figure 2 (b)) in order to achieve a simple structure regularity. Our routing delivers the alert message to the nearest available sink. Thus, a neighbor with the best end-to-end performance in the request zone will be used as the forwarding successor in our relay (see Figure 2 (c)). Quadrants I, II, III, and IV, are of types 1, 2, 3, and 4 request zones, denoted by $Z_i(u)$ ($1 \leq i \leq 4$). Algorithm 1 shows the details.

Algorithm 1 (NR - Routing under LAR scheme-1 [10] without using any guide information): Determine the successor of node u (including node s) with respect to $n(u) \subseteq N(u)$ under anycasting scheme [2].

1) If $d \in n(u)$, $v = d$.
2) Determine all four request zones $Z_k(u)$ ($1 \leq k \leq 4$).
3) Determine the target sink and the corresponding request zone $Z_i(u)$ at the source s and keep the forwarding in type-i zones in the rest of routing.
4) Select $v \in n(u) \cap Z_k(u)$ whenever $n(u) \cap Z_k(u) \neq \phi$.

III. MEASUREMENT OF END-TO-END DELAY

Our M information describes the minimal delay of a type-i forwarding and its succeeding path from a node u to its nearest sink in $Z_i(u)$. Such reference information can be normalized into a single value $\in [0,1]$. The higher value it has, the faster the routing process will be to the target sink. The information constitution has two phases: the *initialization phase* is applied during the network initialization of the deployment, the *update phase* is applied when any node and/or link malfunction occurs. In the following, we will discuss the details of this measurement in Algorithm 2. The resultant value is used by each node to determine the successor in the forwarding.

Algorithm 2 (End-to-end delay measurement).

1) **Initialization phase**: Each permanently awakened sink u sets $M(u)$ to a fixed (1, 1, 1, 1). Every other nodes sets a changeable (0, 0, 0, 0). Then, each node will have a stable status by applying Eq. (1) with the beaconing scheme.
2) **Update phase**: In case any node u observes a change of schedule, link quality, or availability of neighbor node in its record (t, c, or n respectively), Eq. (2) is applied to update the status of u. Any change of $M(u)$ will trigger the update phase of its neighbors. Then, Eq. (1) will be applied to achieve a stable value.

A. Initialization phase

According to different types of forwarding zones, our metric is a 4-tuple. Permanently awakened sinks set their fixed values to (1, 1, 1, 1), in which "1" indicates that they can receive the message immediately without any delay. Other nodes set a changeable (0, 0, 0, 0), in which "0" (or $\frac{1}{\infty}$) indicates the initial value of an unknown delay or endless delay ($= \infty$). Then, node u will update its $M_i(u)$ ($\in [0,1], 1 \leq i \leq 4$) with:

$$\begin{aligned} M_i(u) = \max\{&M'_i(u), \\ &\max_{v \in N(u) \cap Z_i(u)}\{\tfrac{1}{t(u,v) \times c(u,v) + \frac{1}{M_i(v)}}\}\} \end{aligned} \tag{1}$$

where $M'_i(u)$ is the original value before the update of $M_i(u)$, and the selected link $u{-}v$ is called the *key link* of u for $M_i(u)$. Note that v is the joint node to connect the key links of u and v. $M_i(u)$ is constituted along a sequence of connected key links from the sink. Such a sequence of links is also called the *reference path* of u, which is the best path from u to the sink in $Z_i(u)$, according to the definition in Eq. (1). The following theorems prove the properties of our M-information.

Theorem 1. *If any type-i forwarding from node u cannot avoid the detour(s), u has a constant value $M_i(u) = 0$.*

Theorem 2. *The information updated by Eq. (1) will converge.*

Theorem 3. $M_i(u) > M_j(v)$ ($1 \leq i,j \leq 4$) *indicates a better path from u to the sink in $Z_i(u)$ (rather than the one from v to the sink in $Z_j(v)$) in terms of the end-to-end delay.*

In the sample case in Figure 1, d is the sink and $M_1(d) = 1$. As shown in Figure 1 (b), $t(u_1,d) = 1$ and $c(u_1,d) = 3$. We have: $M_1(u_1) = \frac{1}{1 \times 3 + 1} = \frac{1}{4} = 0.25$. Because $t(u_3,d) = 1$ and $c(u_3,d) = 1$, we have: $M_1(u_3) = \frac{1}{1 \times 1 + 1} = \frac{1}{2} = 0.5$. Because $t(u_2,u_3) = 4$ and $c(u_2,u_3) = 1$, we have: $M_1(u_2) = \frac{1}{4 \times 1 + \frac{1}{0.5}} = \frac{1}{6}$. Source s has two choices: u_1 and its succeeding path $s{-}u_1{-}d$, and u_2 and its succeeding path $s{-}u_2{-}u_3{-}d$. In our measurement, the elapsed time along the path $s - u_1 - d$ can be estimated as 5 slots: $t(s,u_1) \times c(s,u_1) + \frac{1}{M_1(u_1)} = 1 \times 1 + \frac{1}{0.25} = 5$ which is better than 8 slots along path $s - u_2 - u_3 - d$: $t(s,u_2) \times c(s,u_2) + \frac{1}{M_1(u_2)} = 2 \times 1 + \frac{1}{\frac{1}{6}} = 8$. Although $u_1 - d$ has link burstiness, our information indicates that this link and the corresponding path is worthy to try. When link $s - u_1$ becomes unstable $c(s,u_1) = 5$, the path $s - u_1 - d$ requires more elapsed time (i.e., $5 \times 1 + \frac{1}{0.25} = 9$ slots). It indicates a new path $s - u_2 - u_3 - d$.

B. Update phase

The following explains the effectiveness of our information when it converges after any of network dynamics occurs.

Firstly, when any node or its link changes the status (caused by node failure, signal fading, energy depletion, etc) and fails to relay the message, this will incur the failure in the beaconing process. This node will disappear from the corresponding n set in its neighbors. Secondly, using a link that has changed its quality (by reflection, interference, etc) may change the number of required retransmissions and affects the end-to-end delay performance. Such a change can be detected by the beaconing process and be described with the CPDF measurement c. Thirdly, when any node u changes its schedule of active time, every 1-hop neighbor will also change its CWT $t(v, u)$.

When a node u detects any of the above changes in n, c, or t, it will re-calculate its $M_i(u)$ $(1 \leq i \leq 4)$ with:

$$M_i(u) \quad = \quad \max_{v \in N(u) \cap Z_i(u)} \{ \frac{1}{t(u,v) \times c(u,v) + \frac{1}{M_i(v)}} \} \qquad (2)$$

If the new value is different ($M_i(u) \neq M'_i(u)$), a notification will be sent to all of its neighbors the next time u appears in their n set. This will trigger the re-calculation among them until no node in the networks needs to update the M value. Then, each node will continue to apply Eq. (1) until the evaluation process converges. The following corollaries prove the properties of this information in any unstable environment where updates are always needed.

Corollary 1: Scalable information update for link burstiness. *When no node changes its wake-up schedule, the probability of the update of M_i at each node u, say $P_l(i)$, is proportional to the number of nodes that are affected by link burstiness.*

Corollary 2: Scalable information update for schedule optimization. *When k nodes need to change their wake-up schedule to mitigate the impact of interference or reflection, the probability of the update of M_i at each node u, say $P_n(i)$, is proportional to $\sqrt{k}$.*

IV. Routing Protocol

Based on the M information, our proposed routing will have two phases: first, select the successor with the best performance estimation according to the M value, wait for its wake-up, and transmit/retransmit the report; second, when the quality of links along the selected path becomes poorer than the expectation after M is newly updated, a backup successor in current n set with a relatively larger M value is selected in the anycasting strategy to avoid waiting too long.

The optimization of the end-to-end delay in the global view is achieved by gradually approaching the sink, guided by the reference path of each intermediate node (via selecting the successor with the largest M value). Note that the reference path that was selected early does not necessarily contain the one selected later due to the information update by those ongoing changes of the node schedule and link quality during the routing process. We adopt an optimistic manner: the closer the routing approaches the destination, the fewer links have burstiness to defer the message delivery. That is, our evaluation M value referred to in the latter routing phase will more likely be consistent to the network configuration and can infer the neighbor selection more precisely.

Firstly, our routing, simply denoted by FR, adopts the M-information that is collected in a light-weight evaluation process. By Monte Calro experiments in the real deployment system, we observed that up to 30% of the retransmissions are needed for a single link on average. Therefore, we set each $c(u, v)$ a fixed value ($= 1 + 30\% = 1.3$). Such information mainly interprets the delay impact of deployment holes and CWTs only (by values of n and t respectively). The retransmission reservation in the M evaluation does not infer the exact link quality of each hop, causing the selection of a successor incorrect. Therefore, a more accurate reservation is needed for our information-based routing.

Secondly, we extend FR with the CPDF measurement c that is maintained up to date. This routing is simply denoted by DR. The integration of our CPDF measurement (of link quality) with the measurement of other delay factors in DR provides an accurate end-to-end delay evaluation. Compared with FR, we expect DR to achieve a better performance and to illustrate the value of our consideration of link burstiness in the routing with the critical time constraint.

The samples of successor selection that have been demonstrated in the previous section can easily be implemented in our DR routing with the complete M-information. The details of our M-information-based routing are shown in Algorithm 3. The followings prove the effectiveness of M information for our routings in the unstable deployment environment.

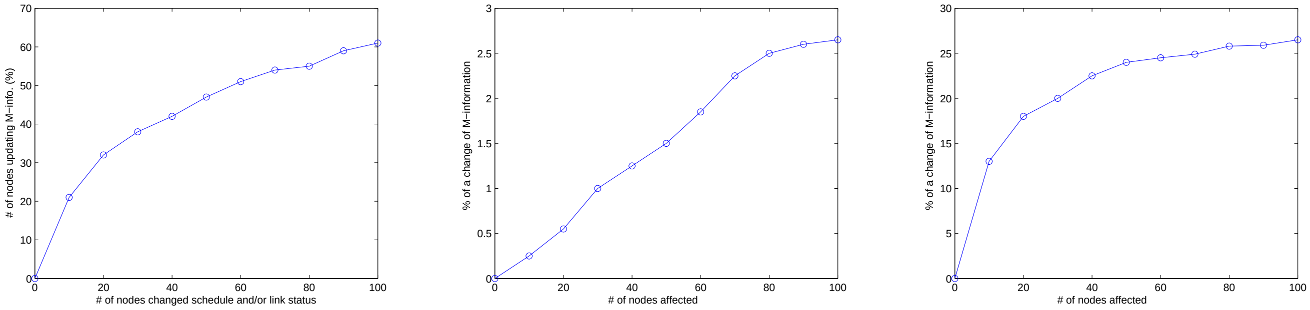
Algorithm 3 (Routing base on M-information): Determine the successor of node u (including node s) with respect to $N(u)$ [10].

1) Apply steps 1) and 2) of Algorithm 1.
2) **Transmission phase**. Select $v \in N(u)$ where v has the best end-to-end performance, indicated by $t(u, v) \times c(u, v) + \frac{1}{M_k(v)}$ $(1 \leq k \leq 4)$ and determine the corresponding type of greedy forwarding, say type-i. After v is selected, wait $t(u, v) \times c(u, v)$ until it appears in $n(u)$ (with $c(u, v) - 1$ retransmissions as reserved).
3) **Backup phase**. If v miss the contact at the expected time, u switches to anycasting mode; that is, u waits until $n(u) \neq \phi$ and selects backup successor $v \in n(u) \cap Z_k(u)$ with the best performance indicated by $t(u, v) \times c(u, v) + \frac{1}{M_k(v)}$, preferred to the selection in $Z_i(u)$.

Corollary 3: Scalability of DR and FR routings. *Regardless how the sinks can be deployed, the probability that the update of M can cause the change of successor selection, say P_c, is proportional to the impact of the link burstiness (P_l) and/or that of the schedule change (P_n).*

V. Simulation

In this paper, we use the experiment results from a custom simulator to demonstrate the substantial improvement of our information-based routings (FR & DR) in terms of the number of required slots (i.e., end-to-end delay performance), compared with the hop-distance-based greedy forwarding routing with retransmission (denoted by NR in this paper). We also



(a) M update in the entire network

(b) update for the link burstiness only (P_l)

(c) Update for the schedule changes only (P_n)

Fig. 3. Cost of our M information updates.

show that the cost and the impact of inconsistency under the network dynamics can be controlled in an acceptable range. In this way, we show that our information model is cost-effective to achieve the routing with the minimum delay.

This simulator emulates the network topology and link quality of the real deployment for bridge collapse detection. The bridge is as long as 600 yards and requires a total of 200 sensors (e.g., DEVISER/MEMSIC LOTUS equipped with 802.15.4 wireless communication) to cover the required surveillance. Each sensor node is bound on the surface of the bridge. The sinks are settled at a safe place near to the bridge abutment in order to permanently provide a complete and constant coverage so that any incoming safety alert message can be received. After a sensor detects a vibration in the dangerous range, a report will be initiated and be sent to any of the sinks. However, due to the bridge structure and materials, the neighborhood of each relay node is irregular and unstable, which increases the difficulty of analyzing the impact of link burstiness on the end-to-end performance.

We implement the network model with $\beta = 10$ (i.e., 10% duty cycle at each node) and simulate events to cause the change of link quality and/or wakeup schedule in up to 50% of the nodes (i.e., 100 nodes is the maximum). Those changes will be recorded by the CPDF measurement and furthermore affect our M-information. Each node u maintains the latest 100 $n(u)$ records at its regular wake-up time and determines the latest sequence of consecutive successes/failures of every neighbor $v \in N(u)$. u also maintains a CPDF array $c_{u,v}[-100..100]$ for v. Then, $c(u,v)$ can be calculated and be used to predict the success of the next beaconing process between these two nodes for our DR routing. The actual result, i.e., the success or failure, will trigger the update of both the sequence record and the CPDF array. In FR routing, $c(u,v)$ is set by a fixed value 1.3, i.e., the average number of retransmissions per link in the system. It may not be accurate to describe the character of each link, but it has a lightweight information collection process that lowers the cost of maintaining the CPDF data (i.e., sequence and array).

We simulate the vibration of the bridge to initiate the reporting process. From the source node, our routings, FR and DR, are conducted, and are compared to NR. For each routing, more than 100 cases are tested vs. dynamic changes of node availability, node schedule, and link quality.

A. Information construction

Our M-information, due to its structural simplicity, can be constituted by reusing the beaconing process. Other computational overhead, such as the maintenance of the CPDF data, can be controlled into each node's off-line activity. Thus, the cost of our M-information model can be ignored. Next, we show how we mitigate the impact of inconsistency in information updates. In our simulation, we test the probability that a node needs to update its measurement evaluation by a number of changes in the wakeup schedule and/or link quality in the network. This result is the only data that we collected from the deployment and can directly be related to the possibility of routing decision change.

Figure 3 (a) shows that upon the change of $0 - 100$ nodes, no more than 40 additional nodes (20% of the total) need to update their M-evaluation. It also shows that the more nodes change, the less additional updates occur.

Figure 3 (b) shows the probability that any intermediate node along a selected path needs to change its M value when certain numbers of nodes ($0 - 100$ nodes) change their link quality. From Figure 3 (b), we observe that the effect of link burstiness on routing can be controlled as we have shown in Corollary 1. Figure 3 (c) shows the changes by the wakeup schedule to verify the result in Corollary 2. We observed that most routing cases can use consistent M-information.

From the above experimental results, the chance of propagation to change the forwarding type in a routing can be ignored. That is, the delay of the information update and the impact of the inconsistency problem can be ignored.

B. Routing performance

For each NR reporting case that will eventually end with a different number of hops from 5 to 28, we count the elapsed time of the corresponding FR and DR routings in terms of the number of slots. Figure 4 shows the comparison of the end-to-end delay performance between all routings NR, FR, and DR. Compared with the NR routing, our M-information-based

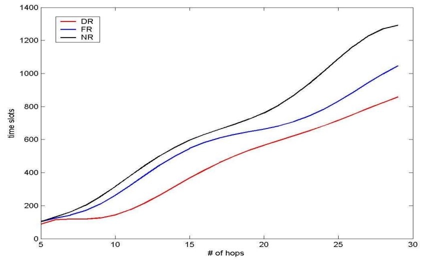


Fig. 4. Elapsed time of different routings: NR, FR, and DR.

routing, even when it uses a light-weight evaluation process in the FR routing, will save 20% of the time when delivering the message. When the complete M-information is used in DR routing, another 50% reduction can be achieved.

VI. RELATED WORK

The existing delay-sensitive routings applicable to duty cycle systems have mainly focused on anycasting [2]: Each node uses a broadcast address to send the packet. All active (listening) 1-hop neighbors that successfully received this packet will assess their own priority to act as the successor of the relay, based on how close they are to the destination. However, most of the successor candidates neighboring in the geographic topology may not wake up and such a non-waiting forwarding will experience the local minimum, causing the detours in the parameter routing phase. The success in [2], [12] cannot be repeated because their assumption that the node density is high enough to have an awakened neighbor in greedy forwarding is too ideal for duty cycle systems.

In [9], the dynamic programming (DP) is applied to determine the minimal end-to-end delay. To achieve a scalable solution in the networks where each node can change its wake-up schedule and the CWT for neighbors, a localized reference information model is provided in [8] to guide each 1-hop advance in a direction with relatively less delay.

All existing routing schemes ignore the temporal properties of wireless links in real environments [3]. An unstable Link and its burstiness can be caused by many factors such as attenuation, interference, noise, and motion. Without any accurate information of link quality, the geographic greedy routing may select a node whose succeeding links are all too bursty to successfully relay any message. A local minima occurs even when the nodes are deployed in a high density of distribution in the duty cycle system.

Many measurements (e.g., [5], [13]) of link quality rely on the simple Monte Carlo experiments and cannot predict the transmission successes for the routing selection. The conditional packet delivery function (CPDF) [11], [14] provides a succinct way to describe the durations of packet delivery correlations. When the transmission consistently fails or is successful in a certain period of time as in the past, this model can be effective to estimate the link quality for the routing. However, as indicated in [3], this reception rate is the percentage of a received packet over a period of time and ignores the *underlying distribution of the losses*. Short periods of reception failure will trigger the need for more retransmissions.

VII. CONCLUSION

In this paper, we provide a cost-effective method to deliver the safety alert message quickly in the WSN application for the bridge collapse detection, which has a critical time constraint. The key is to predict the elapsed time of retransmissions that are caused by link burstiness as well as other delay factors, such as local minima and CWT in the duty cycle systems, and to analyze its impact on the end-to-end delay of the routing process in the generic mode in a fully distributed manner. Our routing is implemented in the decisions at each intermediate node and the information is constituted by exchanges among 1-hop neighborhood in the local. This provides a reliable solution to mitigate the impact of any sudden change of node availability, CWT, or link quality, in a global view. The substantial improvement in our approach on the end-to-end delay performance has been analyzed and demonstrated by our experimental results in a custom simulation.

ACKNOWLEDGMENT

This research work was supported in part by NSF grants ECCS 1128209, CNS 10655444, CCF 1028167, CNS 0948184, CCF 0936942, CCF 0830289, and CCF 0840891.

REFERENCES

[1] M. Alizai, O. Landsiedel, J. Link, S. Gotz, and K. Wehrle. Bursty traffic over bursty links. *Proc. of the* 7^{th} *ACM SenSys*, 2009.
[2] P. Basu and C. Chau. Opportunistic forwarding in wireless networks with duty cycling. *Proc. of ACM CHANTS workshop*, 2008.
[3] A. Cerpa, J. Wong, M. Potkonjak, and D. Estrin. Temporal properties of low power wirelss links: Modeling and implications on multi-hop routing. *Proc. of ACM MobiHoc*, pages 414–425, 2005.
[4] R. Choudhury and N. Vaidya. MAC-layer anycasting in ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 34(1):75–80, 2004.
[5] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. *Proc. of the* 7^{th} *ACM SenSys*, 2009.
[6] C. Grosse and H. Reinhardt. A new concept for bridge monitoring using a wireless sensor network. *Proc. of the International Conference Concrete Platform*, pages 271–284, 2007.
[7] Z. Jiang, J. Wu, M. Guo, Z. Zhao, and D. Wu. A quick and reliable routing for infrastructure surveillance with wireless sensor networks. *Technical Report, CS Dept. at West Chester University*, 2012. Document also available at http://www.cs.wcupa.edu/zjiang/bursty2012.pdf.
[8] Z. Jiang, J. Wu, and R. Ito. A metric for routing in delay-sensitive wireless sensor networks. *Proc. of the* 7^{th} *IEEE International Conference on Mobile Ad Hoc and Sensor Systems (IEEE MASS)*, 2010.
[9] 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.
[10] Y. Ko and N. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *Proc. of ACM/IEEE MobiCom*, pages 66–75, 1998.
[11] H. Lee, A. Cerpa, and P. Levis. Improving wireless simulation through noise modeling. *Proc. of the ACM IPSN'07*, 2007.
[12] S. Liu, K. Fan, and P. Sinha. CMAC: An energy efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks. *Proc. of the* 4^{th} *IEEE SECON*, pages 11–20, 2007.
[13] S. Munir, S. Lin, E. Hoque, S. Nirjon, J. Stankovic, and K. Whitehouse. Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks. *Proc. of the ACM IPSN'10*.
[14] K. Srinivasan, M. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: Measuring wireless link burstiness. *Proc. of the* 6^{th} *Annual ACM Conference on Embedded Network Sensor Systems*, pages 29–42, 2008.