2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems

# GasNet: Efficient Residential Building Gas Leak Monitoring via Opportunistic Networking

Zenghua Zhao[1,2], Xin Zhang[1], Xuanxuan Wu[1], Xiang-Yang Li[3], Junze Han[3]
[1]School of Computer Science and Technology, Tianjin University, China
[2]Tianjin Key Laboratory of Cognitive Computing and Application, China
[3]Department of Computer Science, Illinois Institute of Technology, USA

***Abstract*—GasNet is a sensor network for real-timely monitoring the gas leak in residential buildings, in order to efficiently protect the buildings from gas leak explosion. GasNet provides low-latency and reliable gas leak ALARM message delivery in complicated RF environments. It does so through ROAP, a novel opportunistic forwarding mechanism considering link correlation. ROAP leverages a hop tree in opportunistic forwarder list selection to ensure the shortest hops to the sink. Furthermore, link correlation is used to obtain more accurate forwarding probability and to achieve fast retransmission. We implement the prototype of GasNet in campus testbed and evaluate its performance via extensive experiments. The experiment results show that ROAP achieves $100\%$ gas leak ALARM message delivery in low latency.**

## I. Introduction

Natural gas is the primary fuel for domestic cooking and heating in many cities around the world. Unfortunately, explosion caused by gas leak takes place every year [1]. The explosion leads to a lot of injures and even death since it usually results in the partial collapse of the building. This is even more fatal in China, since a building usually houses hundreds of families, and high-rise apartment buildings over 15 stories are very common, especially in metropolis. An automated approach for detecting the gas leak is thus of the utmost importance to keep the residents safe.

In current solutions, gas leak detectors are installed in the kitchen near the gas pipe individually [2]. If the gas concentration is above a safe threshold, the detector gives an alarm, so that the householder can shut down the gas valve and go out for help. However some explosion happened when there were no people in the apartment. In this paper,we present the design of GasNet, a wireless sensor network system for gas leak monitoring. The aim of the system is to detect the gas leak in every apartment in a building, and deliver the emergency event (*i.e.*, gas leak concentration is over a safety threshold) to the remote control center, *e.g.*, the emergency service. If gas leak occurs, the control center can provide help as soon as possible to avoid violent explosion.

Low latency and reliability are the primary requirements of GasNet. It is intuitive to achieve low latency by traditional shortest path routing if links are reliable [3]. However, the RF environment in kitchens are challenging according to our in-situ measurements shown in Section II-B. Packet losses are frequent due to interference of cross technologies (*e.g.*, WiFi, microwave oven and cordless phone) [4], [5]. Moreover, the neighborhood size is small limited to the construction structure, which will result in route hole problem when some nodes fail to work [6]. All the above make the shortest path routing inappropriate for GasNet.

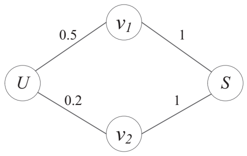

Fig. 1: A simple example

Opportunistic routing was used to improve network performance under link unreliability, leveraging the broadcast nature of wireless radio [7], [8]. We illustrate this through a simple example. In Fig 1, node $u$ sends packets to node $s$ via node $v_1$ and $v_2$. The packet reception rate of each link is labeled above the link. For the shortest path $u \to v_1 \to s$, the expected transmission cost from $u$ to $s$ is then $1/0.5+1 = 3$ [9]. While using opportunistic routing, both $v_1$ and $v_2$ have opportunities to forward the packet if any of them receives the packet. In this case, the expected transmission cost from $u$ to $s$ becomes $1/(1-(1-0.5)(1-0.2))+1 \simeq 2.6$, smaller than that of the shortest path routing. This results are obtained supposing independent links, however in fact links are sometimes correlated [10]. We re-examine the example assuming links $(u, v_1)$ and $(u, v_2)$ are positively correlated, *i.e.*, node $v_2$ receives the packet under condition that $v_1$ receives the same one. Under this complete positive link correlation situation, the expected transmission cost from $u$ to $s$ is thus the same as that of the shortest path, no benefits of opportunistic routing at all. While if the two links are negatively correlated, *i.e.*, node $v_2$ receives the packet under condition that $v_1$ does not receive the same packet, then the expected transmission cost is hence $1/(0.5+0.2)+1 \simeq 2.4$, the smallest one. In brief, link correlation has impact on the performance of opportunistic routing. We observe from experiment results (refer to Sec. II-B) that link correlation does exist in GasNet scenarios, therefore it is more accurate to take link correlation into considerations while estimating expected transmission cost.

Motivated by this, we propose ROAP (Reliable Opportunistic Acquisition Protocol), a novel hop-tree based opportunistic forwarding protocol leveraging link correlation. The basic idea is that forwarders with smaller expected transmission cost have higher priority to forward the packets, where the expected transmission cost is calculated considering link correlation. To meet the application requirements on low latency and

978-1-4799-6036-1/14 $31.00 © 2014 IEEE
DOI 10.1109/MASS.2014.44

reliability, we have to tackle the following challenges:

**How to select forwarders and calculate the expected transmission cost in presence of link correlation**. Unlike [11], in ROAP, forwarders of a node $u$ must be $u$'s neighbors with smaller hop count than $u$. This rule ensures that the packet is forwarded along a shortest-hop path. The expected transmission cost is calculated considering link correlation. The estimation in practice is also given.

**How to schedule forwarders**. In contrast to EXOR [7] and MORE [8], we schedule forwarders in a distributed manner by estimating the time for each forwarder to forward a packet.

**How to balance low latency and reliability**. In case that none of forwarders receive the packet, we adopt retransmission to improve reliability. However, retransmission gains reliability at the cost of delay. To trade reliability off latency, we design a fast retransmission mechanism leveraging link correlation.

In addition, we address the route hole problem and node failures by dynamic hop tree maintenance.

In summary, our contributions are two-folds:

Firstly, to meet the application requirements, ROAP presents a novel opportunistic forwarding mechanism considering link correlation based on the hop tree. In contrast to [11], ROAP leverages the hop tree in forwarder list selection to ensure shortest hops to the sink. Moreover, link correlation is leveraged to obtain more accurate forwarding probability and to achieve fast retransmission.

Secondly, We implement the prototype of GasNet in 20-node testbed on campus and evaluate its performance via extensive experiments.

Experiment results show that ROAP offers $100\%$ ALARM message delivery in highly dynamic environments. It is highly robust to topology changes and failures.

The rest of the paper is organized as follows. We first present system requirements for gas leak monitoring and outline the challenges of using IEEE 802.15.4 wireless communications in kitchen environments in Section II. Section III elaborates on the design of ROAP. We present our implementation and evaluation on testbed in Section IV. Section V reviews related work and we conclude in Section VI with a summary and discussion about future work.

## II. System Design Rationale

A gas leak monitoring system requires a low-cost emergency event delivery system that offers reliability and low latency. In this section we introduce the system requirements and describes the challenges of reliable and real-time data acquisition in this environment.

### *A. Application Requirements*

There are usually hundreds of apartments in a residential apartment building. To monitor the gas leak in the building, a wireless gas leak sensor is deployed in each apartment. The sensors in one building comprise a sensor network with one sink. The sink is also a gateway to access the Internet via cellular communication networks such as 3G/4G, in order to deliver messages to the control center. We focus on the sensor network consisting of the gas leak sensors inside one building.

Gas leak must be sensed in real time, say at 20Hz. However, the sensing data are reported to the control center (the sink) only when the gas concentration is more than a safety threshold, *i.e.*, gas leak is detected. Gas leak event should be delivered to the control center as fast as possible in order to protect the apartment from potential gas explosion. On the other hand, GasNet must provide data yielding of $100\%$ to support effective protection mechanism.

Therefore, the requirements of gas leak detection networking are as follows:

**Low latency**: The network should be able to deliver the emergency event as fast as possible to save time for the control center to take reaction against the gas leak.

**Reliability**: The network should be reliable to ensure that the emergency event can reach the control center.

**Robustness**: The network should be robust to node failures and external interference due to the small neighborhood size and many cross technology interference sources.

We aim to design GasNet to meet all the requirements above. Before we delve into the detailed system design, let us check the challenges we have to address.

### *B. RF Environment*

Characterizing GasNet environment is crucial to understand the challenges it poses to reliable and low-latency data collection protocols. Therefore we carry out a set of experiments. The nodes are placed near the gas range in kitchens in different apartments. Any two of nodes are separated by walls or floors. During the experiment, one node broadcasts 10,000 100-byte packets back-to-back, the others receive. All nodes operate at the 802.15.4 frequency channel 26. The link-layer ACK is disabled and no retransmission either, in order to obtain accurate loss rate. Upon receiving a packet, each receiver logs the LQI (Link Quality Indicator), and the packet sequence number. The results are listed in Table I.

TABLE I: Loss rate through walls/floors

| what across | loss rate% | LQI |
|---|---|---|
| 1 wall | 0.24 | 194.4 |
| 2 walls | 14.45 | 111 |
| 3 walls | 28.83 | 79.2 |
| 4 walls | 100 | - |
| 1 floor | 10.07 | 127.8 |
| 2 floors | 45.93 | 68.4 |
| 3 floors | 100 | - |

**Link reliability**. As shown in Table I, the packet loss rate increases to over $45\%$ across two floors and achieves $100\%$ while across four walls or three floors. This is primarily due to construction structure and cross technology interference from WiFi, Microwave ovens, and cordless phones [4], [5]. The link of GasNet is unreliable, and the link quality is dynamic.

**Neighborhood size**. Although the network is large scale with hundreds of nodes, the neighborhood size is small. Building construction impacts on the transmission range, therefore there are only limited number of neighbors across floors and

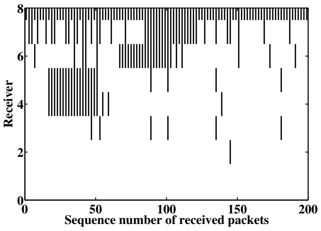

Fig. 2: Packet reception at 8 receivers

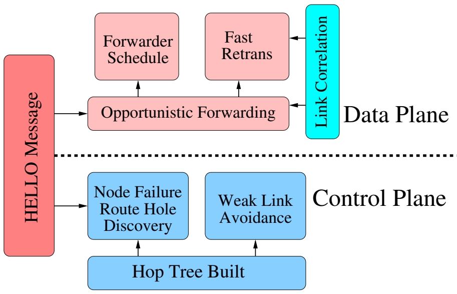

Fig. 3: The framework of ROAP

walls. Small neighborhood size makes node failure at the risk of "route hole" [6], *i.e.*, a node has no forwarders, so it cannot forward packets received from other nodes to the sink. Route hole leads to packet loss and retransmission in vain.

**Link correlation**. To show there exists link correlation, we carry out a small experiment on our testbed. One gas leak detection sensor broadcasts 200 packets on channel 26. 8 nodes are deployed randomly on the other one or two floors. All the nodes that hear the packets report to a centralized server. Fig. 2 shows packet loss at different receivers from empirical measurements. Packet losses are marked by black overlines. Long vertical overlines indicate that packets are lost at multiple receivers. It is obvious that packet losses on different links in Fig. 2 are correlated, since the vertical overlines are randomly distributed for independent links. This experiment reaffirms the empirical study reported in recent works [10], [12]. Therefore link estimation should take link correlation into considerations for accuracy.

## III. Reliable Opportunistic Acquisition Protocol

ROAP lies at the center of GasNet. As shown in Fig. 3, ROAP has a control plane to control network topology and a data plane for data retrieval. This division of responsibilities ensures timely adaptation to link quality variations and reliable data delivery with low latency. Specifically, the control plane constructs a hop tree rooted at the sink, and maintains the tree against link quality degradation and node failures. On the other hand, the data plane relies on opportunistic forwarding leveraging link correlation to deliver emergency data to the sink. ROAP is designed in a fully distributed manner, which makes it scalable to the network size.

### A. Protocol Design Overview

In the control plane, a hop tree rooted at the sink is established via flooding HOP messages right after the network starts up. Each node holds a hop value which means the hop distance to the sink. The hop count is also used to select forwarders in the data plane. The hop tree is maintained via broadcasting HELLO messages among neighborhood periodically. HELLO messages contribute to (a) finding malfunction nodes in time to tackle a "route hole", (b) adapting to link quality dynamics to avoid weak links (LQI $< 100$) and (c) exchanging information needed by the data plane.

In the data plane, when a sensor node detects an emergency event (the gas leak concentration over a safety threshold), it sends ALARM messages periodically until the gas leak concentration is under the safety threshold. The ALARM message is forwarded along the hop tree in reverse direction (from leaves to the root) in an opportunistic way. In contrast to the traditional routing protocols, all the neighbors who receive the ALARM message have opportunities to forward it. Therefore it is robust to node failures, since there are usually more than one forwarders at each hop. Note that only the neighbors with smaller hop count can be the forwarders, which ensures a shortest hop to the sink.

On the other hand, when many forwarders receive the ALARM message, they will contend the channel and thus cause collision and backoff. We therefore devise a mechanism to make a schedule, thereby only one of forwarders transmits the message. In order to optimize the performance in terms of reliability and low latency, we sort the forwarders in an ascending order by their expected transmission cost to the sink. We define the expected transmission cost as the expectation of how many transmissions a node needs to successfully transmit a packet to the sink through its forwarder list. The expected transmission cost is calculated considering link correlation to make it more accurate. A forwarder transmits the ALARM message only if other forwarders with smaller expected transmission cost do not receive the packet.

Even there are more than one forwarders, there might be the case that none of them receives the ALARM message. In this case, we present a fast retransmission mechanism. After a node forwards an ALARM message, it starts a timer. If the node does not overhear the message from its forwarders for a period of time, it retransmits the message. In order to shorten the wait time, once again we leverage link correlation to infer the loss of the ALARM message and retransmit the packet before timeout, thus called fast retransmission.

Now we turn to describe each component in detail. We first introduce the network model before presenting opportunistic forwarding in the data plane, for the sake of description.

### B. Network Model

We model the network as a directed graph $G(V, E)$ where $V$ is the set of wireless nodes, and $E$ is the set of links. For each pair of nodes $e_i = u, v_i \in V$, we use $P(e_i) \in (0, 1]$ to denote the probability that $u$ can directly deliver a packet to $v_i$. We use $P(\bar{e_i})$ to denote the probability that the packet is lost along link $e_i$. We define $P(e_i \cap e_j), i \neq j$ as the probability that both $v_i$ and $v_j$ receive the packet. Similarly, $P(e_i \cup e_j), i \neq j$ is the probability that at least one of them receives the packet. For a given node $u$, we use $Fwd(u)$ to denote the forwarder list of $u$, and $|Fwd(u)|$ is the size of the forwarder list.

### C. Opportunistic Forwarding with link correlation

When an emergency event is detected, ALARM messages are generated and forwarded to the sink reversely along the hop tree opportunistically. ALARM messages are forwarded according to the forwarding RULEs as follows.

**RULE 1**: For a node $u$, only its neighbors with smaller hop count than it can be in its forwarder list.

**RULE 2**: When multiple nodes receive the ALARM message from node $u$, only the nodes in its forwarder list (*i.e.*, the forwarders) have opportunities to forward the message.

**RULE 3**: Upon receiving the ALARM message from node $u$, a forwarder transmits the message only if other forwarders with smaller expected transmission cost than it did not receive the message.

RULE 1 and RULE 2 ensure that the ALARM message is forwarded with the shortest hops to the sink. RULE 3 allows the forwarder with the smallest expected transmission cost to have the highest priority to forward the message. The above three rules make the forwarding an optimal one, *i.e.* the ALARM message is forwarded with minimum hops and minimum expected transmission cost. We will show how to calculate the expected transmission cost in presence of link correlation according to these forwarding rules.

Consider a node $u$ and its neighbors $v \in N(u)$. We will compute the expected transmission cost of node $u$. To understand our approach better, we introduce some definitions first. Given a set of nodes $S$, let $S^*$ denote the ascendingly sorted list of $S$ based on the expected cost by each node in $S$ to send data (via possible relay) to the sink. Let $Fwd(u)$ denote the forwarder list of node $u$. According to RULE 3, $Fwd(u)$ consists of the neighbors of $u$ with the smaller hop count than $u$.

To find the expected transmission cost at node $u$, we first sort the forwarder list $Fwd^*(u)$ in ascending oder by the expected transmission cost, *i.e.*, $Fwd^*(u) = \{v_1, v_2, \ldots, v_M\}$, where $i < j \Rightarrow C_{v_i} < C_{v_j}$, and $M = |Fwd^*|$. Let $\rho_u$ denote the probability that a packet sent by node $u$ is received by at least one node in $Fwd^*(u)$, then

$$\rho_u = P\left(\bigcup_{i\in[1,M]} e_i\right) = \sum_{k=1}^{M}(-1)^{k-1} \sum_{J\subseteq[1,M],|J|=k} P\left(\bigcap_{j\in J} e_j\right). \quad (1)$$

Let $C_u^l$ denote the expected transmission cost that node $u$ sends a packet to at least one node in the forwarder list $Fwd^*(u)$. $C_u^l$ can be calculated as follows:

$$C_u^l = \frac{1}{\rho_u}. \quad (2)$$

When at least one node in the forwarder list received the packet sent by node $u$ successfully, we need to calculate the expected transmission cost to forward the packet via the forwarder list. We first assume that the expected transmission cost of node in the forwarder list of $u$ has already been computed, and then present how to compute them later. We assume that only one node from the forwarder list that receive the packet will forward the packet. Moreover, the node forwards the packet only if all the nodes with smaller expected transmission cost do not receive it. This is essential to avoid channel contention and achieve optimal performance. We will show later how this can be implemented via a distributed scheduling.

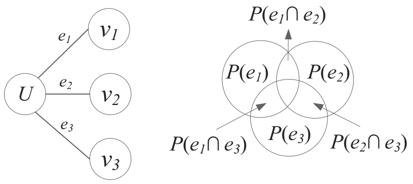

Fig. 4: Link correlation

We now calculate the expected transmission cost, $C_u^f$, for $u$ to forward the packet to the sink by its forwarders. The expected transmission cost of $v_i, (i = 1, \ldots, v_M)$ is $C_{v_i}$, which are supposed to be known. The probability that node $v_1$ forwards the packet is $P(e_1)$. Node $v_2$ will forward the packet if it receives the packet and node $v_1$ does not receive the packet. Therefore the probability that $v_2$ forwards the packet is $(P(e_2) - P(e_1 \cap e_2))(1 - P(e_1))$, in presence of link correlation. Similarly, the probability that $v_i, (i \in [2, M])$ forwards the packet is $P_{v_i}^f$:

$$P_{v_i}^f = \left(P(e_i) - P((\bigcup_{k\in[1,i-1]} e_k)\bigcap e_i)\right) \times \left(1 - P(\bigcup_{k\in[1,i-1]} e_k)\right).$$

Hence, the expected transmission cost of node $u$ to transmit a packet using one of its forwarders is

$$\beta = P(e_1)C_{v_1} + \sum_{i=2}^{M} P_{v_i}^f \times C_{v_i}. \quad (3)$$

Since $\beta$ is computed under condition that one forwarder node receives the packet, then we have

$$C_u^f = \frac{\beta}{\rho_u}. \quad (4)$$

Let $C_u$ denotes the expected transmission cost of node $u$ to broadcast a packet through forwarder list $Fwd^*(u)$, $C_u$ is calculated as,

$$C_u = C_u^l + C_u^f. \quad (5)$$

The cost consists of two parts. The first part is the expect cost for node $u$ to successfully transmit a packet to at least one forwarder in $Fwd^*$. The second part is the expected cost that there is one node in the forwarder list $Fwd^*$ to help to relay the packet to the sink.

**Three-forwarder case**. To illustrate this, we start by considering a simple case in Figure 4 [12], assuming $|Fwd^*(u)| = 3$. Suppose a node $u$ has three one-hop neighbors $v_1, v_2$ and $v_3$, which are sorted at ascending order by their expected transmission cost $C_{v_i}, (i = 1, 2, 3)$. $C_{v_i}, (i = 1, 2, 3)$ has been known. The three links are $e_i = (u, v_i), (i = 1, 2, 3)$ respectively. Figure 4 depicts the Venn diagram of the successful packet reception in this example. In presence of link correlation, the probability $\rho_u$ that a packet sent by node $u$ is received by at least one of the three nodes is:

$$\begin{aligned}\rho_u = P(e_1) + P(e_2) + P(e_3)\\ - P(e_1 \cap e_2) - P(e_2 \cap e_3) - P(e_1 \cap e_3)\\ + P(e_1 \cap e_2 \cap e_3) \quad (6)\end{aligned}$$

$C_u$ can be calculated as follows. The probability that node $v_1$ forwards the packet is $P(e_1)$. Node $v_2$ will forward the packet if it receives the packet and node $v_1$ does not receive the packet. Therefore the probability that $v_2$ forwards the packet is $(P(e_2)-P(e_1 \cap e_2))(1-P(e_1))$, in presence of link correlation. Similarly, the probability that $v_3$ forwards the packet is

$$\left(P(e_3) - \sum_{j=1}^{2} P(e_3 \cap e_j) + P(\bigcap_{j=1}^{3} e_j)\right) \times (1 - P(e_1) - P(e_2) + P(e_1 \cap e_2)).$$

Hence, the expected transmission cost of node $u$ to transmit a packet using one of its forwarders, $C_u^f$, is

$$\begin{aligned}C_u^f = \frac{1}{\rho_u}(P(e_1)C_{v_1}\\ + (P(e_2) - P(e_1 \cap e_2))(1 - P(e_1))C_{v_2}\\ + \left(P(e_3) - \sum_{j=1}^{2} P(e_3 \cap e_j) + P(\bigcap_{j=1}^{3} e_j)\right)\\ \times (1 - P(e_1) - P(e_2) + P(e_1 \cap e_2))\, C_{v_3}). \quad (7)\end{aligned}$$

To get $C_u$ with three forwarders, we need to compute $\binom{1}{3} + \binom{2}{3} + \binom{3}{3} = 7$ polynomial terms where $\binom{b}{a}$ is the number of selecting $b$ items from $a$ ones. Indeed, for more general cases of $m$ forwarders, the computational complexity of $C_u$ is on the order to obtaining all possible logical combinations, *i.e.*, $2^m - 1$. Although in our wireless network, the number of forwarders is relatively small due to limited neighborhood, the exponential growth of complexity with $m$ shall be avoided when possible. Moreover, it is not easy to calculate all these conditional probabilities in practice. Therefore we present an approximate approach to estimate conditional probability in concurrent receptions to simply the calculation in practice [12], [13].

**Expected transmission cost estimation with link correlation in practice**. To estimate $C_u$, we need to get the probability $P_{v_i}^f$ that each node $v_i \in Fwd^*(u)$ receives a packet from node $u$ successfully, under the condition that none of the nodes in the forwarder list with smaller expected transmission cost receives it. Suppose each node maintains a packet reception bitmap recording the reception status of a fixed number of most recent packets. We leverage HELLO message in our design.

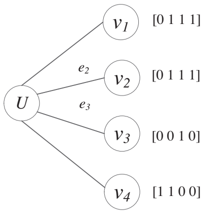

Fig. 5: Probability estimation in practice

Every node broadcasts a HELLO message to its neighbors periodically as shown in Section III-F. Each HELLO message has a sequence number which has a maximum value and can be used round robin. Each node exchanges its reception bitmap with its neighbors. Assume the bitmap length is $N$, and $M = |Fwd^*(u)|$, we then have

$$\hat{\rho}_u = \frac{1}{N} \sum_{k=1}^{N} B_{v_1}(k)||B_{v_2}(k)|| \dots ||B_{v_M}(k),$$

$$\hat{P}(e_i) = \frac{1}{N} \sum_{k=1}^{N} B_{v_i}(k),$$

$$\begin{aligned}\hat{P}_{v_i}^f = \hat{P}(e_i)\\ \times \frac{\sum_{k=1}^{N} [1 \oplus (B_{v_1}(k)||B_{v_2}(k)|| \dots ||B_{v_{i-1}}(k))] \& B_{v_i}(k)}{\sum_{k=1}^{N} B_{v_i}(k)},\\ i = 2, \dots, M. \quad (8)\end{aligned}$$

where $B_{v_i}(k)$ is a bit representing the reception status of the $k$th HELLO message of neighbor $v_i$. $B_{v_i}(k) = 1$ if node $v_i$ receives the $k$th HELLO message, otherwise $B_{v_i}(k) = 0$.

For example, in Figure 5 [12], node $u$ has 4 forwarders, $v_1, v_2, v_3$, and $v_4$. We calculate $\hat{P}_{v_4}^f$. Suppose the bitmap of node $v_1$ is [0111], which indicates that $v_1$ misses the 1st message and receives the 2nd, 3rd and 4th ones. When node $u$ receives the bitmaps from all its forwarders, it can use Eq. (8) to calculate

$$\begin{aligned}\rho_u = \frac{1}{4}(0||0||0||1 + 1||1||1||0||0 + 1||1||1||0 + 1||1||0||0)\\ = 100\%,\end{aligned}$$

$$\hat{P}(e_4) = (1 + 1 + 0 + 0)/4 = 50\%,$$

and

$$\begin{aligned}\hat{P}_{v_4}^f = \hat{P}(e_4)((1 \oplus (0||0||0)) \& 1) + ((1 \oplus (1||1||0)) \& 1)\\ + ((1 \oplus (1||1||1)) \& 0) + ((1 \oplus (1||1||0)) \& 0))\\ /(1 + 1 + 0 + 0) = 25\%.\end{aligned}$$

$C_u$ then can be estimated as $\hat{C}_u$:

$$\hat{C}_u = \frac{1}{\hat{\rho_u}} + \frac{1}{\hat{\rho_u}}(\hat{P}(e_1)\hat{C}_{v_1} + \sum_{i=2}^{M} \hat{P}_{v_i}^f \hat{C}_{v_i} \quad (9)$$

**Calculate expected transmission cost Distributedly**. In the above, we assume that we have known the expected transmission cost of the forwarders, in order to calculate the expected transmission cost of node $u$. Here we present how to calculate the expected transmission cost of the forwarders. Once again, we design a distributed approach for them. We leverage HELLO message to exchange the expected transmission cost among the node and its forwarders.

For the nodes with hop count $h = 1$, *i.e.*, one-hop neighbors of the sink, the expected transmission cost of them is calculated as $1/\rho_u$ [9]. For other nodes with hop count higher than 1, their $C_u$s are initialized to 0, and calculated as follows.

Upon receiving an HELLO message from a node $v$ with smaller hop count (*i.e.*, its forwarders), node $u$ with $h > 1$ checks the expected transmission cost $C_v$ carried by the message. If $C_v \neq 0$, and $C_v$ is different from that in its local neighbor list, which means the forwarder updated its expected transmission cost, then node $u$ calculates its own expectation transmission cost $C_u$ according to Eq. (9).

### *D. Forwarding Schedule*

From the approach to calculate the expected transmission cost, we know that the forwarder list of node $u$ is prioritized. A forwarder of $u$ forwards a packet only when none of the forwarders with higher priority receives the packet. Therefore an schedule is needed to make sure that all the forwarders know when to forward the packet without contention with others.

We now present such a forwarding schedule. A packet sent by node $u$ carries the sequence of its forwarders, therefore the forwarder receiving the packet knows its own order to forward. Upon receiving an ALARM packet, a node checks the forwarder sequence in the packet. If the node is the first forwarder, it forwards the packet immediately. Otherwise, it has to wait for a period of time $T_{wait}$. If it overhears the packet forwarded by others during $T_{wait}$, it then drops the packet. But how to decide $T_{wait}$?

Before we delve into the detail, let us firstly illustrate how the nodes contend the channel. Suppose the popular IEEE 802.15.4 standard is adopted as the MAC layer technique. IEEE 802.15.4 uses CSMA/CA to contend the channel [14]. Before a node sends a packet, it senses the channel for DIFS (DCF InterFrame Space) and backoffs randomly for a period of time. If the channel is idle during this period of time, then it sends the packet. Regardless of the propagation time, the time for a node to finish sending a packet is:

$$T_{send} = T_{DIFS} + T_{backoff} + T_{trans},$$

where $T_{DIFS}$ is the time of DIFS, $T_{backoff}$ and $T_{trans}$ are the time to backoff and to transmit a packet respectively.

Waiting time $T_{wait}(m)$ of the $m$th forwarder is thus estimated as:

$$T_{wait}(m) = (m-1) \times T_{send}, m \in [1, |Fwd^*(u)|]. \quad (10)$$

By using the scheduling approach, we reduce not only collisions but also the chance that a packet is forwarded via a high-cost link, since the winer always has the smallest expected transmission cost among the forwarders who receive the packet.

### *E. Fast retransmission leveraging link correlation*

Opportunistic forwarding makes all the forwarders have opportunities to transmit the ALARM message, which improves the probability for $u$ to succeed sending the message. However what if all the forwarders of $u$ miss the message? Even though this is an event with low probability, we still have to address it to achieve high delivery ratio. An retransmission scheme is needed. Since the packet is broadcasted to make all the forwarders receive them, there is no link-layer ACK to notify node $u$ that the packet has been received successfully. The intuitive one is that node $u$ retransmits the packet if it does not overhear the packet sent by its forwarders after a time period of $T_{wait}(u)$. $T_{wait}(u) = |Fwd^*(u)| \times T_{send}$, since in the worst case, only the last one in the forwarder list receives the packet. However, if the last one does not receive the packet either, node $u$ has to wait for so long a time to start retransmissions.

We thus devise a fast retransmission leveraging link correlation. The idea is based on the observation: if links are correlated, then the probability that other nodes receive the packet under the condition of the first node missing it is small. The retransmission algorithm is thus intuitive as follows.

Node $u$ calculates the link correlation factors of the first forwarder and every other forwarders, denoted by $K_i, i = 2, \ldots, |Fwd^*(u)|$ [10]. If there are $m$ links who are highly positively correlated with link $e_1$, then node $u$ infers that if the first forwarder does not receive the packet, then other $m$ links do not either, therefore it will retransmit the packet after time $T_{wait}(u)$,

$$T_{wait}(u) = (|Fwd^*(u)| - m + 1) \times T_{send}.$$

Specifically, if all the links are highly positive correlated, the wait time is shorten to $T_{send}$. Since node $u$ can retransmit the packet without waiting for all the forwarders in presence of positive link correlation, we call it fast retransmission. On the other hand, if all the links of $u$ are positively correlated, it is the best to use traditional routing. Fortunately, this it not true.

**Over-transmission avoidance**. Over-transmission will cause network congestion and thus waste network resources. There are two cases that might cause over-transmission. (a) One of the forwarder did send the packet, but node $u$ does not overhear it. This happens when link is asymmetric. In this case, node $u$ retransmits the packet. (b) One of the forwarder did send the packet, but other forwarders with lower priority do not overhear it. This happens when the forwarders are hidden terminals. In this case, forwarders with lower priority will transmit the packet after their wait time expire.

To address over-transmission problem, we make the following rules:

(1) if a forwarder receives a packet from node $u$, and it has forwarded the packet, then drops it.

(2) if a forwarder receives a packet from other forwarders, then drops it, and if it is waiting for its turn then stop.

### *F. The control plane*

After the network starts up, the sink floods a HOP message to build a hop tree [15]. The HOP message includes hop value in it. The HOP message sent by the sink has the hop value of 1. Every node in the network has a local hop count which is initialized to be a very large value compared to the depth of the tree, say 255. Upon receiving a HOP message, a node checks the hop value $h$ in the message against the local one. If the local hop count is more than $h$, then the node updates the local hop count to $h$, and increments the hop value in the message and floods the message again. Otherwise, the node does not update the local hop count or forward the message to avoid flooding storm. In this way, every node in the network will have a hop count eventually.

The hop tree is maintained by broadcasting HELLO messages periodically among neighbors. HELLO message has a node ID and a corresponding hop value. Each node maintains a local neighbor list. The neighbor list has node ID, its hop count, and TTL (Time To Live), and other information needed by opportunistic forwarding. TTL indicates the latest time receiving the neighbors HELLO message.

**Weak link avoidance**. Upon receiving a HELLO message, a node checks its node ID against the record in its neighbor list. If this is a new neighbor, then the node makes a new record and inserts it to the neighbor list. Otherwise, the node updates the corresponding record in the neighbor list. It also checks the hop value $h$. If the local hop value is larger than $h+1$ and the LQI of this message is more than a threshold $\beta$, it means the node has a smaller hop count to the sink through this neighbor and the link quality is good enough, so the node changes its hop count to $h+1$. The LQI threshold $\beta$ is set to 100 according to our measurements in Section II-B. Weak link can be avoided to be selected as a forwarder in this way. The hop count is also updated dynamically.

**Node failure and route hole detection**. Each node $u$ checks TTLs in its neighbor list periodically. If TTL of a record is expired, the corresponding neighbor maybe fails to work. In this case, node $u$ reports a NodeFailure message to the sink and deletes the corresponding record from its neighbor list. Since HELLO messages are sent in a fixed period, and the period cannot be set too short in order to avoid heavy overhead. Therefore the node failure might not be found in time. However, this problem can be addressed using opportunistic forwarding.

If there are no neighbors with hop count smaller than node $u$, then it has no forwarders, *i.e.*, there is a route hole. Suppose the smallest hop count among the neighbors is $h$, node $u$ then updates its hop count to $h+1$, thereby it can find new forwarders.

**Congestion avoidance**. To avoid congestion, HELLO message is sent at a fixed period plus a small random time. Actually, it is a tough task to synchronize sensor noses since they have no hard clock built-in [16]. Furthermore, HELLO messages are limited in one-hop neighbors, that is to say, when a node receive a HELLO message, it does not forward it. Therefore HELLO message would not lead to network congestion.

## IV. Implementation and Evaluation

We design the wireless gas leak sensor node and implement its prototype as shown in Figure 6. It mainly consists of a gas leak detection sensor, lights, a buzzer, a ZigBee module and a power module. When the gas concentration is above a threshold, it will alarm with sound and flash, and send ALARM messages to the sink. The ZigBee module adopts Jennic 5139 [17], a low-power high-performance transceiver module. It is compatible to IEEE 802.15.4 and ZigBee standard. The protocols proposed are also implemented using Jennic API.

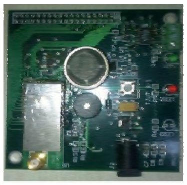

Fig. 6: The prototype of the wireless gas leak detection sensor node

### A. Experiment setup

We built a test-bed consisting of 20 gas leak detection sensors and 1 sink. They are deployed in the toilets on multiple floors in a Building at Tianjin University campus, since the toilet has the similar structure with that of kitchen in terms of water pipes and WiFi interference. We carried experiments under 4 scenarios, where 20 gas leak detection sensors are deployed on multiple floors to construct 4,5,6 and 7-hop trees. In each scenario, one gas leak detection sensor is at one end of the floors, and the sink is at the other end. Gas leak detection sensor sends ALARM messages when it detects a high-level gas concentration. We use a lighter to simulate gas leak during experiments. ALARM messages are generated at a speed of 1 every 2 seconds. 150 ALARM messages are sent to the sink in total.

### B. Performance Metrics

We consider the following performance metrics in our experiments:

**ED (Earlist end-to-end Delay)**: it describes how long it will take for the sink receiving an ALARM message after an emergency event happened. ED is an important metric to indicate the system performance. Assume $t_1$ is the time the first ALARM message sent by a gas leak detection sensor, and $t_2$ is the time the sink receiving an ALARM message from the sensor for the first time. ED is calculated as $ED = t_2 - t_1$.

**AD (Average end-to-end Delay)**: the average end-to-end delay of ALARM messages transmitted from the source sensor to the sink.

**Delivery ratio**: defined as the ratio of the number of ALARM messages received by the sink to the total number of ALARM messages sent by a source sensor. It indicates the reliability of the system.

**Delivery cost**: the total number of the ALARM messages forwarded by nodes in the network divided by the number of ALARM messages sent by a source sensor.

As a comparison, we use the conventional flooding routing protocol (FR) and the shortest path routing protocol (SPR) as benchmarks. The flooding routing protocol broadcasts packets along the direction of the sink, each node involved only

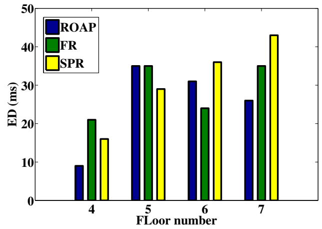

Fig. 7: The earliest end-to-end delay

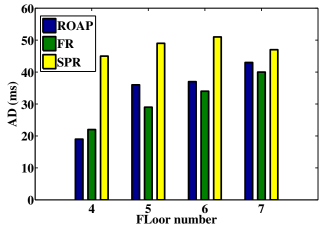

Fig. 8: The average end-to-end delay

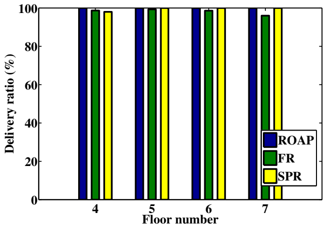

Fig. 9: The delivery raio

forwards the same packet once to reduce redundancy. In the shortest path routing protocol, packets are forwarded through a path with minimum cost. Retransmission is adopted in SPR, and the path is maintained dynamically.

### C. Performance

In this section, we present the experiment results. Fig. 7 shows the earliest end-to-end delay at four scenarios for ROAP, flooding and shortest path routing protocols. We can see that at 4-hop and 7-hop scenarios, ROAP has the shortest ED values, and at other scenarios it is comparable to FR or SPR. The average end-to-end delay is illustrated in Fig. 8, we can see that ROAP has the shortest AD, the AD of the SPR is the longest. That is because that ROAP selects the forwarders opportunistically, thus has more opportunity to deliver the packets. We present their delivery ratios in Fig. 9. In each scenario, ROAP achieves 100% delivery ratio, better than the other two protocols.

TABLE II: Delivery cost at normal and node failure

| | ROAP | FR | SPR |
|---|---|---|---|
| Normal | 1.22 | 10.1 | 1.73 |
| Node failure | 2.19 | 11.57 | 2.37 |

### D. Robustness to node failures

To measure how effectively ROAP can adapt to node failures, we removed two nodes that were forwarding the most packets in the network during the experiments. ROAP uses multiple forwarders and fast-retransmission mechanism. In addition, the failed nodes will be removed from the hop tree and the forwarder list. This causes rapid route recovery around the failure.

TABLE III: Performance at node failure

| | ROAP | FR | SPR |
|---|---|---|---|
| DR (ms) | 100 | 94.0 | 93.7 |
| AD (ms) | 25 | 21 | 226 |

Table II lists the delivery cost for ROAP, FR and SPR at normal and node failure cases at 4-hop scenario. Compared with the delivery cost at normal case, all the three protocols increase the number of packets forwarded in the network.

Table III lists the DR and AD for ROAP, FR and SPR at normal and node failure cases at 4-hop scenario. ROAP achieves 100% delivery ratio at the case of node failure, the other two has different decrease. The average delay of ROAP is compatible at normal and node failure cases, however, AD of SPR increases to 226ms, much longer than that at normal case.

## V. Related Work

**Data collection/traditional routing.** Previous sensor networks that did not implement end-to-end reliability exhibited data yield of 20 $\sim$ 60% [18], [19] and thus fail to meet the requirements of gas leak monitoring. Reliable collection protocols [20]–[22] employ local data caching and end-to-end retransmissions to improve data yields. Most of the reliable collection protocols aim to energy efficiency rather than low latency [21]–[23], whereas SPEED [24] focuses on real time without considering reliability. ROAP is designed to meet the requirements of GasNet, taking both reliability and low latency into considerations.

**Opportunistic routing.** Opportunistic routing has been proved efficient in unreliable scenarios. As pioneering work, ExOR [7] and MORE [8] are designed for large file transferring in wireless static mesh networks. Guo *et al.* [25] propose Opportunistic Flooding tailored for low-duty-cycle networks with unreliable wireless links. The key idea is to make probabilistic forwarding decisions at a sender based on the delay distribution of next-hop nodes. ORTR (Opportunistic Real Time Routing) [26] is presented to guarantee delivery of data under time constraints with efficient power consumption. Lu *et al.* [27] present PRO (Protocol for Retransmitting Opportunistically) to improve the performance of IEEE 802.11 WLANs. PRO is a link-layer protocol that allows overhearing nodes to function as relays that retransmit on behalf of the source after they learn about a failed transmission. Mao *et al.* [11] propose opportunistic routing for wireless sensor networks targeting energy efficiency. They focus on selecting and prioritizing forwarder list to minimize energy consumption by all nodes. In contrast to these work, ROAP selects forwarders based on a hop tree and taking link correlation into considerations.

**Link correlation.** link correlation has not been considered until recently [10], [12], [13]. Zhu *et al.* [13] propose Collective Flooding (CF), which exploits the link correlation to achieve flooding reliability using the concept of collective

ACKs. In [10], Srinivasan *et al.* introduce a metric $K$ to capture the degree of packet reception correlation on different links. $K$ can be a metric for quantifying what kind of a network is present and help decide which protocols to use for that network. Wang *et al.* [12] present CorLayer to exploit link correlation to improve the energy efficiency of reliable broadcasts.

**Surveillance system based on sensor networks.** Sensor networks have been applied in surveillance systems [15], [28]–[30]. Since each surveillance system has its own requirements and environments, they cannot be used in our gas leak monitoring system. In contrast, GasNet is designed specified for such a system.

## VI. Conclusion

The GasNet system presented in this paper is among the first attempts to provide a systematic solution for detection of gas leak in residential high-rise buildings. To meet the application challenging requirements in terms of reliability and low latency, we propose ROAP, an opportunistic forwarding protocol leveraging link correlation. It selects forwarders and calculate the expected transmission cost in presence of link correlation. Extensive experiment results show that ROAP is highly robust to topology changes and failures. In the future work, we plan to further evaluate the performance of GasNet in a large scale real network, and compare it with other well-known protocols such as CTP [23] and CF [13].

## Acknowlogement

The research is partially supported by NSFC (National Natural Science Foundation of China) under Grant No. 61172063. The research of Li is partially supported by NSF CNS-1035894, NSF ECCS-1247944, NSF CMMI 1436786, and NSFC under Grant No. 61228202.

## References

[1] "Gas explosion," http://en.wikipedia.org/wiki/Gas_explosion.

[2] "Gas detector," http://en.wikipedia.org/wiki/Gas_detector.

[3] S. Kwon and N. Shroff, "Analysis of shortest path routing for large multi-hop wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 3, pp. 857–869, June 2009.

[4] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving wi-fi interference in low power zigbee networks," in *SenSys '10*. ACM, 2010, pp. 309–322.

[5] S. Gollakota, F. Adib, D. Katabi, and S. Seshan, "Clearing the rf smog: Making 802.11n robust to cross-technology interference," in *SIGCOMM '11*. New York, NY, USA: ACM, 2011, pp. 170–181.

[6] Q. Fang, J. Gao, and L. Guibas, "Locating and bypassing routing holes in sensor networks," in *IEEE INFOCOM '04*, vol. 4, 2004, pp. 2458–2468 vol.4.

[7] S. Biswas and R. Morris, "Exor: opportunistic multi-hop routing for wireless networks," in *SIGCOMM '05*. New York, NY, USA: ACM, 2005, pp. 133–144.

[8] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *SIGCOMM '07*. New York, NY, USA: ACM, 2007, pp. 169–180.

[9] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *MobiCom '03*. New York, NY, USA: ACM, 2003, pp. 134–146.

[10] K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari, "The k factor: inferring protocol performance using inter-link reception correlation," in *MobiCom '10*. New York, NY, USA: ACM, 2010, pp. 317–328.

[11] X. M. S. T. . X. X. . X. yang Li, "Energy-efficient opportunistic routing in wireless sensor networks," in *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 11. IEEE Computer Society, Nov 2011, pp. 1934 – 1942.

[12] S. Wang, S. M. Kim, Y. Liu, G. Tan, and T. He, "Corlayer: a transparent link correlation layer for energy efficient broadcast," in *MobiCom '13*. New York, NY, USA: ACM, 2013, pp. 51–62.

[13] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang, "Exploring link correlation for efficient flooding in wireless sensor networks," in *NSDI'10*. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–15.

[14] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," Tech. Rep., 2006.

[15] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo, "A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications," in *MSWiM '04*. New York, NY, USA: ACM, 2004, pp. 157–164.

[16] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *Network, IEEE*, vol. 18, no. 4, pp. 45–50, 2004.

[17] "Jennic5139 datasheet," http://www.jennic.com/support/datasheets/jn5139_module_datasheet.

[18] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, "Firewxnet: A multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments," in *MobiSys '06*. New York, NY, USA: ACM, 2006, pp. 28–41.

[19] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *SenSys '04*. New York, NY, USA: ACM, 2004, pp. 214–226.

[20] M. Krogmann, M. Heidrich, and et. al., "Reliable, real-time routing in wireless sensor and actuator networks," *ISRN Communications and Networking*, 2011.

[21] J. Paek and R. Govindan, "Rcrt: Rate-controlled reliable transport protocol for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 7, no. 3, pp. 20:1–20:45, Oct. 2010.

[22] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica, "Flush: A reliable bulk transport protocol for multihop wireless networks," in *SenSys '07*. New York, NY, USA: ACM, 2007, pp. 351–365.

[23] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *SenSys '09*. New York, NY, USA: ACM, 2009, pp. 1–14.

[24] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "Speed: a stateless protocol for real-time communication in sensor networks," in *ICDCS '03*, 2003, pp. 46–55.

[25] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *MobiCom '09*. New York, NY, USA: ACM, 2009, pp. 133–144.

[26] J. Kim and B. Ravindran, "Opportunistic real-time routing in multi-hop wireless sensor networks," in *SAC '09*. New York, NY, USA: ACM, 2009, pp. 2197–2201.

[27] M.-H. Lu, P. Steenkiste, and T. Chen, "Design, implementation and evaluation of an efficient opportunistic retransmission protocol," in *MSWiM '04*. New York, NY, USA: ACM, 2009, pp. 73–84.

[28] O. Chipara, C. Lu, T. C. Bailey, and G.-C. Roman, "Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit," in *SenSys '10*. New York, NY, USA: ACM, 2010, pp. 155–168.

[29] Y. Zeng, C. J. Sreenan, L. Sitanayah, N. Xiong, J. H. Park, and G. Zheng, "An emergency-adaptive routing scheme for wireless sensor networks for building fire hazard monitoring," *Sensors*, vol. 11, no. 3, pp. 2899–2919, 2011.

[30] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao, "Racnet: A high-fidelity data center sensing network," in *SenSys '09*. New York, NY, USA: ACM, 2009, pp. 15–28.