

A Fault Tolerant and Energy Efficient Routing Protocol for Urban Sensor Networks

Cao Liang, Xinming Huang
Department of Electrical and Computer Engineering
Worcester Polytechnic Institute
Worcester, MA 01609, USA
{cliang, xhuang}@ece.wpi.edu

Jing Deng
Department of Computer Science
University of New Orleans
New Orleans, LA, 70148, USA
jing@cs.uno.edu

ABSTRACT

Wireless sensor network technology provides an opportunity for innovation. Their applications in urban environment have attracted many attentions recently. Due to the dynamic operational environment, the reliability and energy efficiency are two important criteria for the routing protocol designs of wireless sensor networks. This paper presents a heuristic study on fault tolerant and energy efficient routing algorithm for wireless sensor networks in street-based urban environment. The leveling algorithm, combined with the energy extension scheme, is presented as an effective routing protocol in the presence of malfunctioned nodes. A local optimization scheme is further proposed for energy efficiency, which is implemented only based on the neighboring nodes information. The simulation results show that the proposed technique achieves optimal fault tolerant performance provided a limited transmission range. The average energy consumption and the packet latency is also simulated and compared with the theoretical optimum solution generated based on the global routing information.

Keywords

Urban wireless sensor networks, fault tolerance, energy efficiency, routing protocol

1. INTRODUCTION

Wireless sensor networks (WSNs) represent an emerging technology that has become very appealing to researchers. It is considered as a next generation technology to bridge between the Internet and the physical world. Recently the interests of urban environment wireless sensor networks (UWSNs) have been growing in both academia and industry. The applications of such networks include deploying the sensors in hazardous environments, such as urban search and rescue [1], flood monitoring and forecasting [2], and sewage system monitoring [3]. Other applications include city environment monitoring, such as traffic monitoring and air pollution detection [4, 5]. Due to the rapid population growth in urban regions, these applications have shown great social impact on our daily life and have attracted more and more attentions.

Many routing protocols have been successfully developed for traditional WSNs. Direct diffusion is a data-centric routing

protocol that is developed for information dissemination in sensor networks [6]. Low-energy adaptive clustering hierarchy (LEACH) is a well-known cluster-based algorithm [7]. Greedy perimeter stateless routing (GPSR) is a geographic routing protocol [8]. Temporally-ordered routing algorithm (TORA) is a distributed routing protocol based on “link reversal” algorithm [9]. A comparison of many other existing protocols is available in [10]. Most of the above routing protocols were focused on the effective information retrieval with the emphasis on energy efficiency for sensor networks. Unlike the randomly distributed sensor networks in rural areas, the sensor nodes in UWSNs are manually deployed in urban environment to form certain communication topology. To the best of authors’ knowledge, little work has been done to develop efficient routing algorithms for UWSNs in the existing literature.

In this paper, we present a heuristic study on fault tolerant and energy efficient routing protocol for UWSNs. We focus on a single network flow with one sink node attempting to collect information from a number of data sources. The data source nodes and the sink node are distributed along streets and avenues as in a geographic road map of an urban city. We begin by presenting the leveling algorithm from our previous work [11], which is an efficient approach with self-configuration and automatic routing path update in unpartitioned network. However, the leveling algorithm is limited to the unpartitioned networks. Thus a power extension scheme is added to the leveling algorithm to overcome the partitioned network problem, this achieves an optimum fault tolerant performance within the dedicated transmission range. Furthermore, because of the variety of the deploying environment, the power supply is not always available to the sensor nodes, which makes the power efficiency one of the key challenges in UWSNs. Data aggregation [12, 13] scheme is usually used in the data-centric networks to improve the power efficiency. Besides this, a local optimization algorithm is developed in this paper to discover power efficient paths from source nodes to sink node based on local information. By these two means, the power consumption is optimized for the whole network. A unique feature in our study is that different power levels can be assigned to different sensor nodes at the very beginning through manual deployment based on the workloads. The original power level for each node can be resulted from the software simulation of the data traffic and loading in the network. The system performances, in terms of network coverage efficiency, system latency and power consumption are simulated and compared between these algorithms. The results demonstrate that the local optimization algorithm can improve the fault tolerant performance and can achieve near optimum power consumption performance.

The main contributions of this paper are as follows: to the best of our knowledge, this is among the first few researches that present and analyze the routing problems for urban wireless

sensor networks; and a fault tolerant and power efficient routing algorithm is proposed and presented with simulation results.

The rest of the paper is organized as follows. Section 2 introduces the network topology of UWSNs and the evaluation metrics for the routing protocols in it. The details of the proposed routing algorithms are presented in Section 3. The theoretical analysis of the proposed algorithms is discussed in Section 4. Section 5 compares the performances of the proposed algorithms, which is followed by the conclusion Section 6.

2. NETWORK TOPOLOGY AND PERFORMANCE METRICS

Urban wireless sensor networks have been studied in a variety of city environment applications. A large number of sensor nodes can be distributed in various topologies. For security or environmental monitoring applications in urban area, one of the most convenient deployments is to distribute the sensor nodes along the roadways, for instance attaching them to the manhole cover [3], or the streetlights [4]. The sensor nodes are distributed along streets and avenues as in a geographic road map. Therefore, the network becomes a grid topology. To simplify the study case, we assume a regular grid topology as shown in Figure 1 as an example of this kind of UWSNs. We randomly pick a node to serve as the data sink and assume that the data processing center is located at that node. The distances between two adjacent nodes on the same row or column are the same. The objective is to achieve best network coverage meanwhile maximize the system lifetime under the power constraints.

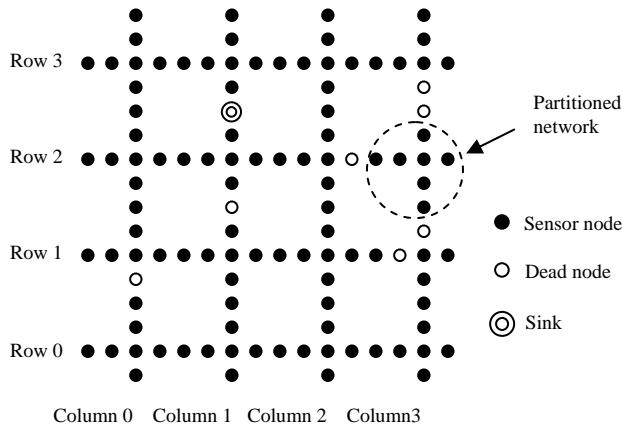


Figure 1. An example of the urban wireless sensor network.

While there are many ways in which the performance of a routing protocol can be evaluated, we use the following metrics.

- **Fault tolerance**

In dynamic environment of sensor networks, some nodes may fail due to a variety of reasons including power depletion, circuit malfunction, processor failure, unreliable radio links, and etc. For fault tolerance study, sensor nodes are randomly selected as malfunctioned nodes with a certain probability. The aim of fault tolerance routing is to maintain the effective network coverage and uninterrupted information retrieval from the field. In fault tolerant routing, the absolute node coverage, as the percentage of live nodes over the total number of nodes, is of less interest. It is

more important to consider the actual coverage efficiency, the definition of which is given below.

Coverage efficiency of a network is defined as the number of active nodes over the number of live nodes. If there is no routing path from a live node to the sink, the live node is considered inactive.

$$C_{eff} = \frac{N - N_d - N_p}{N - N_d} \quad (1)$$

where N is the total number of nodes, N_d is the number of malfunctioned nodes, and N_p is the average number of partitioned live nodes in the network.

- **Energy efficiency**

Because of the power constraint in the UWSNs, an energy efficient routing protocol is required to prolong the system lifetime. In our study, we aim at minimizing the total power consumption for packet delivery during one data collection period. The power consumption involved in the data collection procedure can be divided into three categories: data transmission power, data reception power and data aggregation power. Because it is shown in [14] that the data reception power is much less than the data transmission power, in this paper we neglect the energy needed during data reception procedure, as is done in other routing algorithms [15]. Also in practice, the energy consumption due to compression is significantly less than the data transmission power [16, 17]. We used packet overheads to represent the costs brought by data compressions. A simple power consumption model can be formularized on these assumptions for each data collection period:

$$E_{trans} = T_{trans} \cdot P_a \cdot d^2 \quad (2)$$

where T_{trans} is the total data transmission time, P_a is the antenna transmission power and d is the transmission distance. In our study, we assume a data sensing and collection task is conducted periodically for every node in the network. Thus the transmission time is proportional to the entire packets transferred during each data collection period. Considering a simple free-space path loss model, P_a is assumed to be proportional to the distance square between two communication nodes [7]. The theoretical analysis for power consumption is presented in Section 4. Because it is shown in [15] that the lifetime maximization problem in WSNs is NP-complete, the design of an on-line optimal routing protocol is not practical in reality. In Section 5, the energy consumptions of the proposed algorithms are compared with the global optimum solution obtained by the Dijkstra shortest path algorithm [18].

- **Latency**

Data sensed by the sensor nodes are typically time sensitive especially in some hazard monitoring applications. So it is important to receive data in a timely manner. In our study, we use the maximum path length to evaluate the data latency, which indicates the maximum number of hops for a message to travel toward the sink.

3. ROUTING ALGORITHMS

3.1 Leveling algorithm

We propose the leveling algorithm as an effective, low-complexity routing technique for unpartitioned network. In this algorithm, each node is assigned with a level value that indicates the distance hops to the data sink. During the initial setup phase, the level values of all nodes are set as infinity. The data sink first initiates an RREQ message to its neighbor and indicates a level

value of zero. The neighbor nodes receiving this message change their level value to 1 and send the RREQ message with a level value of 1. The same procedure quickly propagates through the entire network, similar to the initial route discovery as in AODV [19, 20]. The process is naturally analogous to the water flowing from the high elevation to lower level, thus we called it the leveling algorithm. Similar approach was proposed in the tree-based TAG algorithm [21]. In the TAG algorithm, the values associated with its entire sub-tree are lost if a node is out of function, since each node has exactly one parent node for data forwarding. In the leveling algorithm, multiple paths may exist for a node to route data back to the sink node. Thus the stability of the network is potentially better.

When some nodes become malfunctioned in the network, the leveling algorithm should be able to automatically update the routing paths by changing the level values of the affected sensors. The detailed procedure for the route maintenance in the adaptive leveling algorithm is listed below:

1. Each node broadcasts the level value to its neighbors periodically.
2. Each node keeps track of the validity of its own level. For any given node with level m , there should be at least one neighbor node whose level value is $m-1$ (except for the base station with zero value). If such a level broadcast is not overheard for a period of time, the node changes its level to infinity and enters the PANIC mode.
3. In the PANIC mode, a node sends out a broadcast with the PANIC flag set.
4. When a node overhears a broadcast with the PANIC flag set, it waits for a period of time before responding to it (this is necessary to make sure that its level is indeed valid). If the level of this node is valid, it broadcasts its level.
5. When a valid level message is overheard, nodes in PANIC mode sets its own level based on the level value heard. If multiple levels are heard, the smallest one is chosen to insure the shortest path to the sink node.
6. The same rule above is applied to every node repetitively until the network converges. The converging time and the route maintenance overhead are determined by the number of dead nodes as well as their locations in the network.

However, the leveling algorithm is limited to the unpartitioned networks. A partitioned network implies a certain area of the network that contains one or more live nodes but has no connection to the outside of the area as illustrated in Figure 1. When the leveling algorithm converges, the level values of all these isolated nodes turn into infinity that implies no valid path is available from these nodes to the data sink. It is imperative to develop an improved algorithm that can overcome such scenario.

3.2 Power extension algorithm

A heuristic approach to solve the partitioned network problem is to increase the transmission power and to extend the communication range of the nodes as needed. As in the urban grid network showed in Figure 1, a live node with level value m should have at least one immediate neighbor with the level value of $m-1$. If a node detects that there is no next hop node available due to node failure, a straightforward approach is to increase its transmission power to extend the communication range. Due to the concern of energy consumption, the power increase is limited to four times of the minimum transmission power in this paper that result in a doubled communication range according to the simple path loss model [7]. Equivalently, it provides the

additional ability for the active nodes to skip any single malfunctioned node along the routing path. After each power extension, the local path update is necessary to connect other nodes in the partitioned area to the sink node through active routing paths.

In practice, the route maintenance procedure is initiated from the data sink with a zero level value. If there is no next hop available among the directly connected neighbor nodes, power extension is requested. Consequently, local routing path update is performed immediately after each power extension. These two procedures are executed alternatively while the routing path maintenance propagates through the entire network. For validation purpose, the same process is repeated and the node level value is updated if a shorter path is available. It is worth to mention that this power extension technique is a greedy algorithm that converges quickly. It guarantees that all reachable nodes will be connected upon completion of the algorithm. However, the updated routing path does not necessarily reflect the lowest power consumption and minimum number of hops from each node to the data sink.

3.3 Local optimization algorithm

As mentioned in the previous subsection, the above routing algorithms may not necessarily lead to an optimum solution of the power consumption and latency. This is because during the routing discovery and maintenance procedures, these algorithms only aim at the network connectivity. Figure 2 (a) shows an example with the routing paths discovered by the leveling algorithm with numbers representing the level values of nodes. This network topology can be transformed into a tree-based structure. In this example, two paths are explored to route eight sensor nodes back to the data sink with a maximum latency of 7 hops. Intuitively, if we allow the diagonal power extension from node 7 to node 1, node 7 can be routed to data sink within 2 hops instead of 7 hops. Hence, the level value for it is changed to 2, as shown in Figure 2 (b). Similarly to previous algorithm, local routing path update is executed after each power extension. Both total power consumption and system latency are reduced.

In this optimization algorithm, we aim at minimizing the power consumption by breaking the long path into separate short paths. This is achieved by power extension in a fully connected network. However, the transmission power is proportional to the squared distance between two nodes. The criterion for power extension becomes extremely important. The optimum power efficient solution can be achieved based on the global information of the network, which requires the restoration of the link state and topology information of the entire network at each node, as proposed in [22]. In these table-driven routing protocols, the global information is updated periodically and the routing is updated by individual node based on Dijkstra shortest path algorithm. The high computation and communication costs of this scheme prevent it being a practical solution to our problem. In this paper, a simple criterion for power extension is presented as in equation (3), which is explained in details in Section 4.

$$1 + \sum_{j=L_i-\Delta L_i}^{L_i-1} r_c^{\min(k_c, j)} > r_d^2 \quad (3)$$

where L_i and ΔL_i is the current level value and the value difference after power extension for node i . r_c is the packet data compression ratio that is calculated as the compressed data size

over original data size. Each data can be compressed at most k_c times before data distortion happens. r_d is the distance ratio calculated as the power extension range over the original range of the examining node. If equation (3) is true, the transmission power is extended by this node to generate a shorter path to the sink node. The principle here is that each node checks its own power preservation status based on the level values of nearby nodes. If power preservation is observed by one node, it extends its power level to build a shorter path back to the data sink. After that, local path update is performed for all affected nodes. Because only the nodes within certain range (within two hops in our study) are involved in this optimization procedure, it is called a local optimization scheme. Without global information of entire network, this scheme may not guarantee the power preservation for the entire network. Interestingly, the results from the theoretical analysis in Section 4 and the software simulations in Section 5 demonstrate that our local optimization scheme can achieve a near optimum performance on power consumptions as well as better network latency performance at certain malfunctioned node ratios. Most importantly, the implementation of this scheme is much more cost efficient compared with the aforementioned table-driven routing protocols.

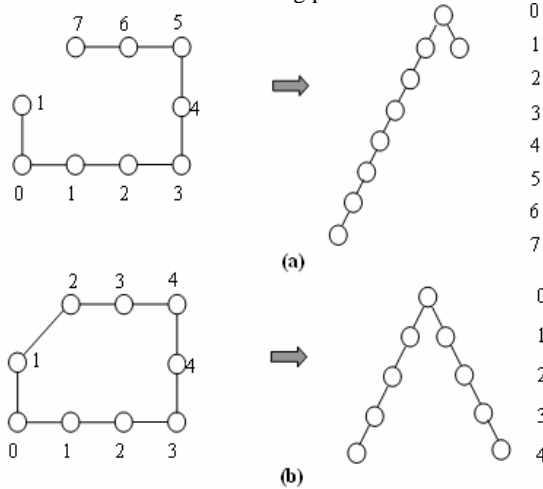


Figure 2. Local optimization algorithm and tree-based representation.

Although more power will be consumed by the power extended node, the total power consumption and latency performances of the entire network can benefit from this scheme as analyzed in Section 4. The initial power distribution of the sensor nodes can be differentiated based on the importance of individual node. This problem will be addressed in our future work.

4. ANALYSIS OF PROPOSED ROUTING ALGORITHMS

We analyze the power consumption during one data collection period of the proposed schemes in this Section. We begin by deriving the total transmission time for the leveling algorithm, which leads to the power consumption by the power model in equation (2). Then the analysis of the power extension and local optimization algorithms is discussed based on the leveling algorithm. We first state our assumptions and introduce some notations used throughout the analysis.

We assume each sensor carries out the same amount of sensing tasks and every node generates the same size of data at the same

rate. The total sensed data during one period is therefore $S_{data}N$. We also assume that every node only transmits once during each data collection period. It is worth to mention that the data transmission time T_{trans} is defined as the summation of the transmission time of every node. Some notations are defined in Table 1.

Table 1. List of analysis notations

Symbol	Definition
N	Total number of working sensors
L	Maximum level in the networks
L_i	Level value of node i , assuming the data sink has the level value of 0
N_i	Number of nodes at level i
S_{data}	Number of bits of information generated by each working sensor per sampling time period.
H_o	Number of bits of a packet header
S_{max}	Maximum payload length of a packet. Generally, $S_{max} \geq S_{data}$
r_c	Compression ratio due to data fusion. This is the ratio of aggregating similar results into a more compact packet. $r_c \leq 1$
k_c	Maximum number of compressions for one data unit
r_t	Data transfer rate for every node in the network (bits/sec)
M_i	Average number of packets at level i
M_{packet}	Total number of packets during one data collection period
T_o	Total transmission overhead in bits
t_{io}	Transmission overhead for node i
T_f	Total traffic in the network during one data collection period
T_{trans}	Total transmission time during one data collection period
E_{trans}	Total energy consumption during one data collection period
P_a	Antenna transmission power for one-hop minimum distance between two sensors

4.1 Energy consumption for leveling algorithm

In leveling algorithm, all nodes transmit data to their directly connected neighbor. Thus the transmission power for all nodes is the same. For a sensor at level L_i , the payload S_{data} is compressed $\min(k_c, L_i - 1)$ times along the path to the data sink. The first-level does not perform data compression, because no data aggregation is possible.

For the leveling method, the data transmission time is determined by the number of levels L , the information generation rate at every sensor node S_{data} , and the number of nodes on each level N_i , where

$$N = \sum_{i=1}^L N_i \quad (4)$$

The total traffic transferred in the network is,

$$T_f = \sum_{i=1}^L N_i S_{data} r_c^{\min(k_c, i-1)} + T_o \quad (5)$$

where T_o is total transmission overhead due to data aggregation and packet generation. No data aggregation occurs until the data are forwarded once. The average data traffic for a node at level i is,

$$T_i = \frac{1}{N_i} \sum_{j=i}^L S_{data} N_j r_c^{\min(k_c, j-i)} \quad (6)$$

Thus given the maximum packet payload length S_{max} , the number of packets per node at level i is,

$$M_i = \left\lceil \frac{1}{N_i S_{max}} \sum_{j=i}^L S_{data} N_j r_c^{\min(k_c, j-i)} \right\rceil \quad (7)$$

where the $\lceil x \rceil$ function finds the least integer greater than x .

The total number of packets transferred in the network is,

$$M_{packet} = \sum_{i=1}^L N_i \left\lceil \frac{\sum_{j=i}^L S_{data} N_j r_c^{\min(k_c, j-i)}}{N_i S_{max}} \right\rceil \quad (8)$$

Therefore, the total transmission overhead is,

$$T_o = H_o \cdot M_{packet} \quad (9)$$

By substituting it to equation (5), we can get the total data transmission time is,

$$T_{trans} = \frac{\sum_{i=1}^L \sum_{j=i}^L S_{data} N_j r_c^{\min(k_c, j-i)} + M_{packet} H_o}{r_t} \quad (10)$$

Therefore the energy consumption for the leveling algorithm is given as in equation (11):

$$E_{trans} = \frac{\sum_{i=1}^L \sum_{j=i}^L S_{data} N_j r_c^{\min(k_c, j-i)} + M_{packet} H_o}{r_t} P_a \cdot d^2 \quad (11)$$

4.2 Energy consumption for power extension algorithm

As discussed in Section 3, the power extension algorithm aims at improving the network connectivity by extending the transmission range. Thus the transmission distances for each node might be different. We use d_i to denote the transmission distance for node i . The total energy for this algorithm is calculated by summation the energy consumed by individual node.

The data traffic for node i can be derived as:

$$t_i = S_{data} + \sum_{j=L_i+1}^L n_{i,j} S_{data} r_c^{\min(k_c, j-L_i)} + t_{io} \quad (12)$$

where $n_{i,j}$ is the number of nodes with level j that transmit data through node i . The packet overhead for node i is calculated as:

$$t_{io} = m_i H_o = \left\lceil \frac{S_{data} + \sum_{j=L_i+1}^L n_{i,j} S_{data} r_c^{\min(k_c, j-L_i)}}{S_{max}} \right\rceil \cdot H_o \quad (13)$$

Thus the energy consumed by node i is:

$$e_i = \frac{t_i}{r_t} P_a d_i^2 \quad (14)$$

Here we use $P_a d_i^2$ to indicate the specific power level for different nodes. The total energy consumption for the power extension algorithm is:

$$E_{trans} = \sum_{i=1}^N e_i = \frac{\sum_{i=1}^N (S_{data} + \sum_{j=L_i+1}^L n_{i,j} S_{data} r_c^{\min(k_c, j-L_i)} + m_i H_o) \cdot d_i^2}{r_t} \cdot P_a \quad (15)$$

If for every node we have d_i equals to d , it can be proved that equations (11) and (15) are equivalent. This indicates that in an unpartitioned network, the energy consumed by the leveling and power extension algorithms are the same, which verifies the algorithm description in Section 3. For a partitioned network, the power extension scheme has a better network coverage performance at a cost of higher energy consumption.

4.3 Energy consumption for local optimization algorithm

To reduce the energy consumption and maximum latency, the local optimization algorithms is proposed based on the routing paths discovered by the power extension algorithm. Because the power extension level maintains the same (up to two hops), these two algorithms result in the same network coverage performance. The energy consumption for the local optimization algorithm can be derived the same as in the power extension algorithm shown in equation (15).

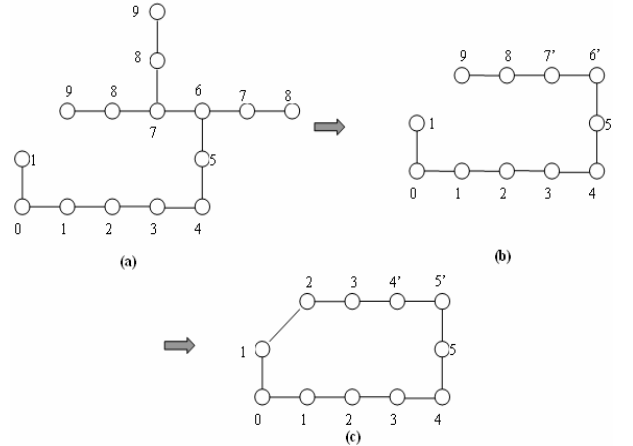


Figure 3. Local optimization algorithm analysis. (a) Local network topology after leveling algorithm. (b) Branch pruning topology. (c) Network topology after local optimization.

In order to compare the energy consumptions between these two algorithms, we notice that the network topology is able to be partitioned into different local networks based on the routing paths. Figure 3 (a) shows an example of this local network. We further assume that the level changing inside the local network won't affect any node outside of the local network, which can simplify the analysis without lost of generality. By this assumption, the branches in the local network can be pruned to the main path as shown in Figure 3 (b). This is valid because the

total data passing through the node 6 or 7 in (a) remain the same after the local optimization. Then the problem becomes to compare the energy consumptions of the network topology shown in Figure 3 (b) and (c) under the condition in equation (3).

Because of network convergence, the number of nodes that are affected by the local optimization is:

$$n = \left\lceil \frac{\Delta L}{2} \right\rceil \quad (16)$$

where ΔL is the change of level value at the power extension node.

The energy consumptions to send each sensed packet S_{data} back to the data sink are compared. Only the nodes affected by the local optimization are needed to be considered. The energy for the original topology shown in Figure 3 (b) is:

$$e_i = \frac{\sum_{j=0}^i S_{data} r_c^{\min(k_c, j)} + t_{io}}{r_t} \cdot d^2 \cdot P_a \quad (17)$$

where $i = L_i - n_k - 1$, $0 \leq n_k < n$ denotes the affected nodes in the local network.

After the local optimization, the level value for the affected nodes changes to:

$$i' = L_i - \Delta L + n_k - 1, \quad 0 \leq n_k < n \quad (18)$$

The energy consumption for each sensed data is derived as:

$$e_i' = \frac{\sum_{j=0}^{i'} d_j^2 S_{data} r_c^{\min(k_c, j)} + t_{i'o}}{r_t} \cdot P_a \quad (19)$$

Because the level value after local optimization for all affected nodes is smaller than the original level value, the packet overhead $t_{i'o}$ is always less than t_{io} . We first analysis the case when $r_c = 1$, which means no data aggregation is performed. Along with the condition in equation (3), it is straight forward to show that the energy consumed after local optimization is less than that of the original algorithm:

$$\sum_{i=L_i-n-2}^{L_i-1} e_i > \sum_{i=L_i-\Delta L-1}^{L_i-\Delta L+n_k-1} e_i \quad (20)$$

Thus the total energy consumption is reduced by the local optimization algorithm. It is observed that at high data compression ratios (experimentally $r_c \geq 0.9$), our local optimization algorithm can efficiently reduce the power consumption and the system latency. The simulation result in Figure 4 shows the percentage of energy preservation for the local optimization algorithm compared with the power extension algorithm, which supports the above analysis. It also shows that the maximum performance improvement can be achieved at about 15% malfunctioned nodes under the simulation parameters used in Section 5.

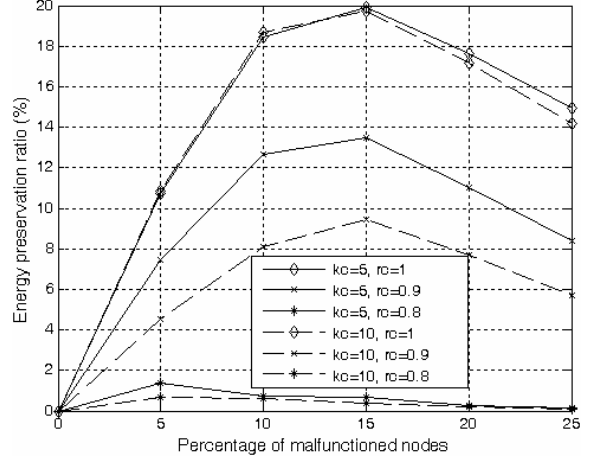


Figure 4. Energy preservation ratio for the local optimization algorithm compared with the power extension algorithm at different k_c and r_c .

5. SIMULATION RESULTS

We evaluate the performances of the proposed routing algorithms through software simulations. We investigate the fault tolerant, energy efficiency and network latency through the static simulations. In the static simulation, the percentage of malfunctioned nodes is given before simulation begins. The network coverage efficiency, average energy consumption and packet latency versus different percentages of malfunctioned nodes are compared among the proposed algorithms. The optimum energy solution obtained by Dijkstra algorithm is also simulated in comparison with the proposed algorithms.

The setup of our urban wireless sensor network is shown as in Figure 1, with 4 rows and 4 columns and a total of 112 sensor nodes. The case with a single data sink is considered and the location of the data sink is randomly selected. The transmission distance for the leveling algorithm is one hop, and the power extension range is up to two hops in the other algorithms. The simulation code for all algorithms is developed in Matlab. We assume that the data measured at each sensor is in the form of equal-sized packets. The system specifications throughout our simulations are listed in Table 2. From these system parameters, it can be calculated that the energy consumption for one package to be forwarded one hop is 0.1 mJ.

Table 2. Experiment setup parameters

Symbol	Value
N	112 sensor nodes
r_c	0.9
k_c	10
P_a	32 mW
r_t	38.4 kbps
S_{data}	15 bytes (120 bits)
H_o	12 bit

In the static simulations, the locations of the malfunctioned nodes are randomly selected. Figure 5 shows that the probability for network partitioning increases as the number of malfunctioned nodes grows for all three algorithm. Because the local optimization algorithm has the same power extension range as the power extension algorithm, the fault tolerant performances of these two algorithms are identical. Result of the leveling algorithm shows the probability of network partition reaches 99%

with only 10% malfunctioned nodes. While the other two algorithm reach the same probability at about 45% malfunctioned nodes. Thus the power extension-based algorithms show significant improvement to avoid the network partitioning.

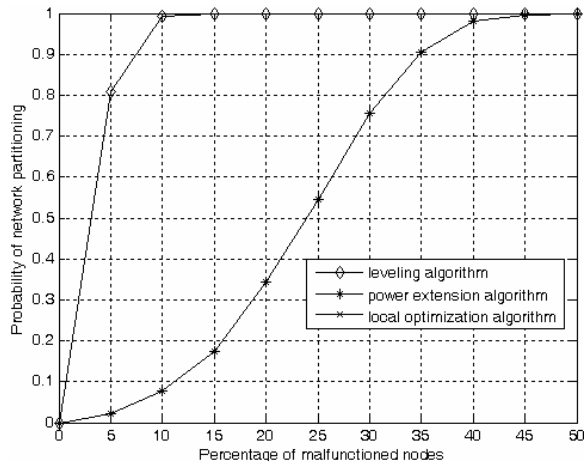


Figure 5. Probability of network partitioning VS. percentage of malfunctioned nodes.

The coverage efficiency of the network is also simulated in Figure 6 to illustrate the fault tolerant performances. It can be observed that at 25% dead nodes, the leveling algorithm only reaches 20% network coverage. While by extending the transmission power, the other two algorithms can achieve more than 95% network coverage.

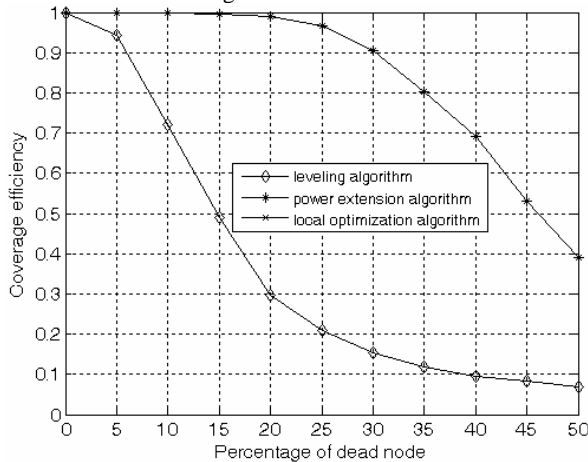


Figure 6. Coverage efficiency VS. percentage of malfunctioned nodes.

The energy efficiency performance for the proposed algorithms is also simulated in the static environment. Here we evaluate the average energy consumed by each connected node during one data collection period. The simulation results are shown in Figure 7. In this simulation, the optimum energy solution is calculated by the Dijkstra algorithm, which also allows up to 2 hops transmission range as in the power extension and local optimization algorithms. These three algorithms can, therefore, achieve the same fault tolerant performance.

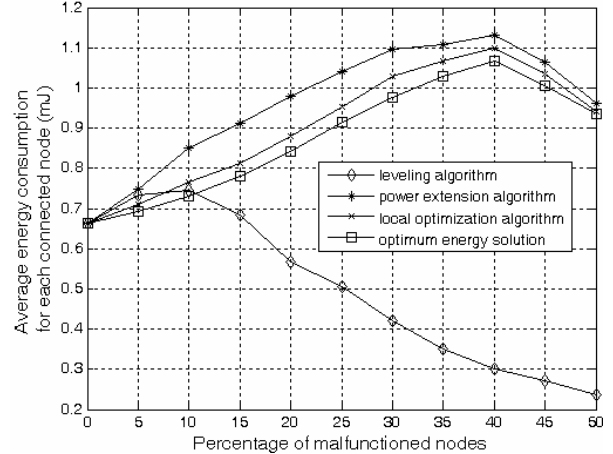


Figure 7. Average energy consumption for each connected node in one data collection period VS. percentage of malfunctioned nodes.

The results show that at low malfunctioned node ratios, the local optimization algorithm out performs both leveling and power extension algorithms in terms of energy efficiency. When the number of malfunctioned nodes increases (greater than 10%), the average energy consumption for the leveling algorithm decreases quickly. This is because as the network partitioning increases, the network coverage is reduced dramatically in the leveling algorithm. For the other three algorithms, more power is consumed to improve the fault tolerant performance. The optimum energy solution is most energy efficient among them. But the computational complexity and the requirement of the global network information prevents it as a practical solution for sensor networks. Figure 7 shows that the local optimization algorithm can achieve a near optimum solution with much lower computational complexity and no global information requirement.

Figure 8 compares the network latency of these routing algorithms. Because of network partitioning, at high percentage of malfunctioned nodes, latency for the leveling algorithm is much smaller than the other algorithms. The proposed local optimization algorithm is able to reduce the network latency at low malfunctioned node ratios when compared to the power extension algorithm.

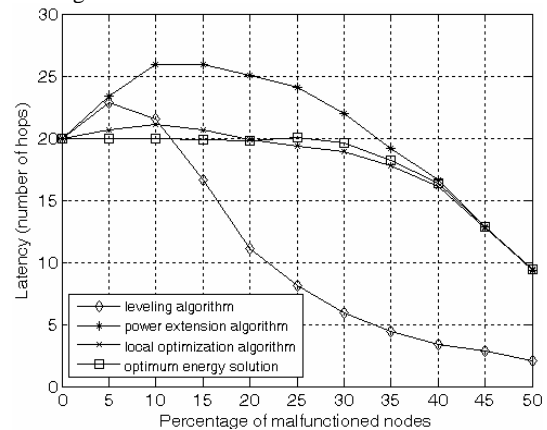


Figure 8. Network latency VS. percentage of malfunctioned nodes.

6. CONCLUSION

This paper presents a heuristic study on fault tolerant and energy efficient routing algorithms for wireless sensor networks in street-based urban environment. Firstly, the leveling algorithm is presented as an effective routing protocol for unpartitioned networks. Secondly, the energy extension scheme is provided to solve the network partitioning problems. Lastly, a local optimization algorithm is proposed to improve the energy efficiency for routing discovery and maintenance. The fault tolerance simulations show that the proposed algorithms can significantly improve the network coverage in partitioned networks and reduce the network partitioning rate. The average energy consumption and the packet latency are also simulated and compared between these algorithms. It shows that the local optimization algorithm can achieve near optimal energy consumption performance at low computational complexity and no global network information requirement. For future work, the initial energy distribution among the sensor nodes and the system lifetime will be investigated for the proposed algorithms.

7. REFERENCES

- [1] J. Reich, and E. Sklar, "Toward automatic reconfiguration of robot-sensor networks for urban search and rescue.," *In First International Workshop on Agent Technology for Disaster Management (ATDM): Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, ACM*, May 2006.
- [2] N.B. Chang, and D.H. Guo, "Urban Flash Flood Monitoring, Mapping and Forecasting via a Tailored Sensor Network System," *In Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control* pp. 757-761, Apr. 2006.
- [3] J.F. Mastarone, and W.J. Chappell, "Urban sensor networking using thick slots in manhole covers," *In Proceeding of the IEEE Antennas and Propagation Society International Symposium*, pp. 779 - 782, Jul. 2006.
- [4] M. Welsh, M. Ezzati, and J. Bers, "Collaborative Research: CRI: Large-Scale Open Sensor Network Testbed for Urban Monitoring," *Research proposal at Harvard University*.
- [5] E. Yoneki, "Evolution of Ubiquitous Computing with Sensor Networks in Urban Environments," *In Proceeding of the ACM International Conference on Ubiquitous Computing*, Sep. 2005.
- [6] C. Intanagonwivat, R. Govindan, and D. Estrin "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *In Proceeding of the ACM International Conference on Mobile Computing and Networking (MobiCom'00)*, pp. 56 - 67, Aug. 2000.
- [7] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660 - 670, Oct. 2002.
- [8] B. Karp, and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," *In Proceeding of the ACM International Conference on Mobile Computing and Networking (MobiCom'00)*, pp. 243 - 254, Aug. 2000.
- [9] V. Park, and M. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *In Proceedings of the IEEE Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97)*, vol. 3, pp. 1405-1413, Apr. 1997.
- [10] A. Ahmed, H. Shi, and Y. Shang,, "A survey on network protocols for wireless sensor networks," *In Proceedings of the International Conference on Information Technology*, pp. 301 - 305, Aug. 2003.
- [11] X.M. Huang, J. Deng. J. Ma, and Z. Wu,, "Fault tolerant routing for wireless sensor grid networks," *In Proceedings of the IEEE Sensors Applications Symposium*, pp. 66 - 70, 2006.
- [12] B. Krishnamachari, D. Estrin, and S. Wicker "The impact of data aggregation in wireless sensor networks," *In Proceedings of the IEEE 22nd International Conference on Distributed Computing Systems* pp. 575 - 578, Jul. 2002.
- [13] H.H. Yen, F.Y.-S. Lin, and S.P. Lin, "Efficient data-centric routing in wireless sensor networks," *In Proceedings of the IEEE International Conference on Communications (ICC'05)*, vol. 5, pp. 3025 - 3029, May 2005.
- [14] V. Shnayder, M. Hempstead, B. Chen, G. Werner-Allen, and M. Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications," *In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems* pp. 188 - 200, Nov. 2004.
- [15] J. Park, and S. Sahni, "An online heuristic for maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Computers*, vol. 55, pp. 1048 - 1056, Aug. 2006.
- [16] F. Zhao, and L. Guibas, "Wireless Sensor Networks: An Information Processing Approach," *San Mateo, CA: Morgan Kaufmann*, 2004.
- [17] Y.P. Chen, A.L. Liestman, and J. Liu, "A hierarchical energy-efficient framework for data aggregation in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 55, pp. 789 - 796, May 2006.
- [18] M. A. Weiss, "Data Structure and Algorithm Analysis in C++," *2nd Edition, Addison-Wesley*, 1999.
- [19] E.M. Royer, and C.K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, pp. 46 - 55, Apr. 1999.
- [20] C.E. Perkins, and E.M. Royer, "Ad-hoc on-demand distance vector routing," *In Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90 - 100, Feb. 1999.
- [21] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, "TAG a Tiny AGgregation Service for Ad Hoc Sensor Networks," *In Proceedings of the ACM Symposium on Operating System Design*, pp. 131-146, 2002.
- [22] T.H. Clausen, G. Hansen, L. Christensen, and G. Behrmann, "The optimized link state routing protocol, evaluation through experiments and simulation," *In Proceeding of IEEE Symposium on Wireless Personal Multimedia Communications*, Sep. 2001.