Distributed Faulty Node Detection and Isolation in Delay-tolerant Vehicular Sensor Networks

Bing Zhu, Wenzhu Zhang, Wei Feng and Lin Zhang
Department of Electronic Engineering, Tsinghua University, Beijing, P.R.China
Email:{zhub10, zhwz, feng_wei, linzhang}@tsinghua.edu.cn

***Abstract*—Distributed faulty node detection and isolation in metropolitan-area delay-tolerant vehicular sensor networks are investigated in this paper. For the intermittently connected large-scale dynamic networks, we design a practical faulty node detection algorithm that vehicles can perform on board to detect faulty sensors by utilizing the local data's spatial correlation. Based on the distributed detection algorithm, a faulty node isolation scheme is proposed to control the diffusion of the faulty data. The parameters of the detection and isolation algorithms to achieve the best performance are investigated based on simulations over real-world data sets. The results show that our detection algorithm performs almost as well as other relative algorithms but demands much less on sensors' communication ability. More than 70 percent of faulty nodes can be identified, and more than half of the faulty data can be reduced in the network by the means of isolation while the total overhead decreases by 35%.**

***Index Terms*—faulty node detection, faulty node isolation, distributed algorithm, delay-tolerant vehicular sensor networks.**

I. Introduction

In recent years, Wireless Sensor Networks (WSNs) have been widely investigated and many applications have been implemented in industry control, environment surveillance, public security, and many other areas that benefit people's life. One obvious trend in the development of WSNs is the fast increasing system scale, which renders the reliability a significant factor in the design and deployment. In realistic large-scale WSNs deployment, sensors are normally made cheap but not sufficiently reliable to meet the requirements of the cost effectiveness. Sensors may suffer unnegligible error probability and cannot collect accurate data frequently. As a result, the network performance is deteriorated if the faulty nodes cannot be detected and isolated in time. Therefore, faulty node detection and isolation is playing a more and more important role to improve the network performance. Unfortunately, straightforward solutions to this task are costly and unrealistic because to check the numerous sensor nodes one by one periodically will consume too many network resources, especially for large-scale WSNs. To our best knowledge, several works on faulty node detection in WSNs have been done, but most of them are not dealing with Mobile Ad-hoc Networks (MANETs), particularly intermittently connected Delay Tolerant Networks (DTNs). Furthermore, faulty node isolation in MANETs is rarely investigated.

In this paper, one of such mobile sensor networks, namely, the Metropolitan-area Vehicular Sensor NETworks (MVSNETs), is selected as the application scenario to study the faulty node detection and isolation. In MVSNETs, sensors are deployed in metropolis for urban environment surveillance such as air quality, humidity, temperature, etc., and vehicles, which move around the whole city, act as carriers to collect and transmit data. Given that the network system is in large scale and sparse, intermittently connected, frequently changed in topology and limited in computation and storage capacity, a self-organizing, energy-efficient and delay-tolerant distributed faulty node detection and isolation algorithm is proposed. In our MVSNET model, there are three kinds of nodes, a large number of fixed sensor nodes (Sensor), mobile communicators equipped on vehicles (Carrier) and a few sink nodes (Aggregator). Sensors are deployed by road-side. The "Store-Carry-Forward" strategy is used to convey delay-tolerant data. Data collected by sensors are sent to carriers when they pass through. When two carriers encounter, they exchange data over a short range wireless link. So the network connectivity is acquired by the mobility of vehicles. The design goal of the algorithm is to detect the faulty sensors and shut them down as quickly as possible with small amount of communication and computational overheads.

The rationale of the proposed algorithm is to exploit the local spatial correlation of the physical field that is being sampled by sensors, and take advantage of the carrier mobility to reduce the communication overhead while increase the responding speed to isolate the malfunctioning sensors. The proposed algorithm is simulated on real data to check its performance in reality. For detection algorithm, we analyze the simulation results to investigate how the parameters, such as the node density and the detection window size, affect the performance. For faulty node isolation, optimal parameters are found to minimize the total data traffic in the whole network.

The remainder of the paper is organized as follows. In Section II, we introduce and discuss some related work. In Section III, the faulty node detection algorithm is introduced, and the simulation results are analyzed and discussed. In Section IV, the faulty node isolation scheme is presented and the simulation results are discussed. In Section V, we draw a conclusion to this paper.

II. Related work

Centralized faulty node detection algorithms are originally introduced in [1]–[3], and can be adopted in mobile sensor networks. Accurate detection results can be achieved by centralized algorithms, which are based on data from the whole

978-1-4673-2569-1/12/$31.00 ©2012 IEEE

network. However, it will lead to large overhead and long delay in DTNs.

Distributed faulty node detection is first investigated by B. Krishnamachari and S. Iyengar in 2004 [4]. The algorithms introduced in [4]–[7] utilize the spatial correlation of the data collected nearby and scale well to large-scale fully connected WSNs. However, it is difficult to apply these algorithms to sparse mobile sensor networks because of the intermittent connectivity. As for each node in MVSNETs, there are no neighbors existing within the range of communication generally. So lots of sinks, which can obtain all the sensing data in their pre-divided detection areas respectively, should be deployed all over the network to carry out the algorithm periodically. Whereas, it is money and energy consuming to deploy so many sinks in metropolis and demands a lot on sensors' communication ability. The work in [4] takes only 0/1 decision predicates. In [5], the proposed algorithm can accept any kind of scalar values as inputs, and achieve high fault detection accuracy with adequate neighborhood size. In [6] and [7], the probability distribution function of the observation is needed, which is not feasible in real scenarios.

The work in [8] and [9] do fault detection by modeling WSNs as Neural Networks. It is useful for fully connected static networks and the knowledge of topology is required. Besides, the computational complexity of the algorithm is pretty high and scales poorly, which is not affordable for cost-effective large-scale sensor networks.

To the best of our knowledge, work on Fault Detection and Isolation (FDI) always concerns itself with monitoring a system to identify when a faulty node emerges and then pinpoint the type of the fault and its location. Further in WSNs, few works consider how to isolate the sensor from generating faulty data as soon as possible when it is predicated as a faulty node. The work in [10] investigates a centralized faulty node isolation method by blacklist dissemination. It is proved to be able to reduce faulty data transmission in the network, whereas the cost of blacklist dissemination is very high. Due to the centralized scheme, the method only works well in dense fault scenarios, otherwise the cost brought by blacklist dissemination itself is much higher than the fault diffusion.

III. Distributed Faulty Node Detection

In this section, we introduce the distributed faulty node detection algorithm in mobile sensor networks and analyze the simulation results based on real data. The detection is targeting in the fixed sensor nodes by road-side in our MVSNET model. The sensor whose readings bias from the true value abnormally due to either mechanical failure or stochastic events is defined as a faulty node in this paper.

A. Methodology

The distributed faulty node detection algorithm is based on the data's spatial correlation, that is, the collected data are correlated in space, while the faulty data are always not. Considering that the mobility of carriers is utilized to achieve network connectivity, our algorithm is proposed upon the notion that data collected by a carrier continuously in a limited time interval are statistically correlated. The on-board detection algorithm is purely distributed. It is carried out by each carrier based on its local data without any communication with others. The algorithm can be formalized as follows.

Consider a time interval T, n sensors S_i $(i = 1,\ 2,\ ...,\ n)$ passed by a carrier in the last T time slots constitute a detection set $N = \{S_1,\ S_2,\ ...,\ S_n\}$. Note that the geographical distribution of N is determined by the carrier's mobility pattern, so that S_1 may be quite far from S_n. Let $x_i(i = 1,\ 2,\ ...,\ n)$ denote the reading of node S_i. Calculate med, the median of the set $\{x_1,\ x_2,\ ...,\ x_n\}$, as an estimatin of the "center" of the sample set. The offsets between the measurement x_i and the center is

$$d_i = x_i - med. \tag{1}$$

Then standardize the offset set $D = \{d_1,\ d_2,\ ...,\ d_n\}$. Let μ and σ denote, respectively, the mean and standard deviation of the set D, i.e.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} d_i, \quad \sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d_i - \mu)^2}. \tag{2}$$

Thus the standard offset set $E = \{e_1,\ e_2,\ ...,\ e_n\}$ is derived, where

$$e_i = \frac{d_i - \mu}{\sigma}, (i = 1,\ 2,\ ...,\ n). \tag{3}$$

DECISION: If $|e_i| \geq \theta$, identify S_i as a faulty node. Here $\theta(>1)$ is a preselected threshold according to the specific application scenario and the expected detection resolution. Larger θ means more tolerance of error. In essence, it is a low-pass filter. In respect that the near-surface temperature field is a 2-dimensional band-limited signal varying slowly, the low-pass filter can effectively remove high frequency noises. The decision algorithm we present above is closely related to the classic voting algorithms studied in distributed statistic applications, which has been proved to be optimal [4]. However, in delay-tolerant vehicular networks, the spatial correlation of the data samples diminishes so much that the effectiveness of the algorithm is reevaluated by simulations in the next subsection.

B. Simulation and Results

As shown in Fig. 1(a), the thermal image of San Francisco Bay Area supplied by the USGS Landsat Thematic Mapper [11] is used as the simulation data source and the simulation system is for urban temperature collection. A square region of $30km \times 30km$ in reality is selected from the image as the database (Fig. 1(b)). To simulate the vehicular mobile sensor network and get data's spatial correlation as realistically as possible, we utilize the map of the corresponding region from Google Map [12] (Fig. 1(c)). In the simulation, nodes are only deployed on the trace of roads (Interstate 680, 580 and CA 4, 13, 24, 242) instead of being distributed in the whole region uniformly, because in real system, sensors are deployed

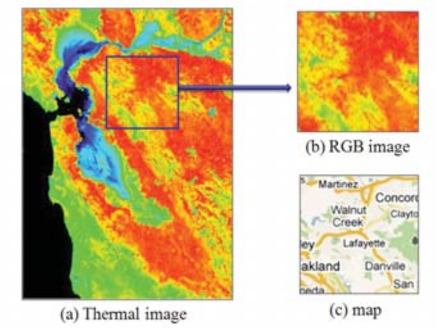

Fig. 1. Data source process

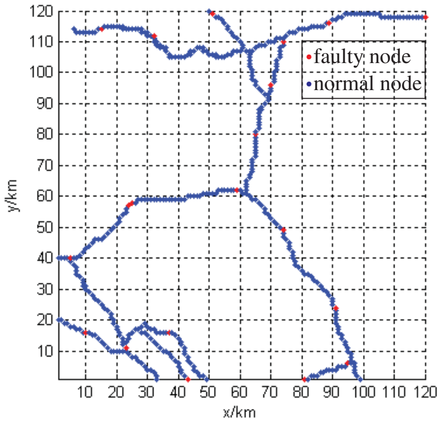

Fig. 2. The distribution of nodes

by road-side(Fig. 2). Then the RGB thermal image (Fig. 1(b)) is mapped into a matrix, whose elements are normalized to [0,1] indicating the temperature.

As mentioned, nodes are distributed as Fig. 2 according to Fig. 1(c) overall uniformly, among which some are set along the boundary of the map, while some are set next to each other on purpose, so that we can check how the algorithm performs in such special cases. All the carriers are limited to move randomly along the paths in Fig. 2. To imitate vehicle's behavior, some more restrictions, like U-turn, are imposed on them in case that some carriers keep circling locally, which are unusual in reality but make the performance better. Parameters used to evaluate the algorithm are listed in Table I.

ρ	Node density
N	Total number of nodes
E	The number of faulty nodes
S	Detection window size
B	Error bias of faulty nodes

TABLE I
PARAMETERS IN FAULTY NODE DETECTION SIMULATION

The False Alarm Rate(FAR), Miss Rate(MR) and normalized error reduction(α) are metrics to evaluate the performance of the faulty node detection algorithm [4]. Let N_{fa} denote the number of nodes, which are mistaken as faulty nodes but normal in essence, and N_m denote the number of missed faulty nodes by the algorithm. The relations between the metrics are

$$FAR = \frac{N_{fa}}{N-E}, \quad MR = \frac{N_m}{E}, \tag{4}$$
$$\alpha = \frac{E-N_m}{E} - \frac{N_{fa}}{E} = 1 - \frac{N_m + N_{fa}}{E}$$

MR increases with θ, while FAR decreases. In the following simulations $\theta = 1.4$. To investigate the performances under different error bias B, detection window size S and node density ρ, a group of simulations have been done, and each simulation has been run for 6000 times to get the average results. Parameter settings are listed in Table II.

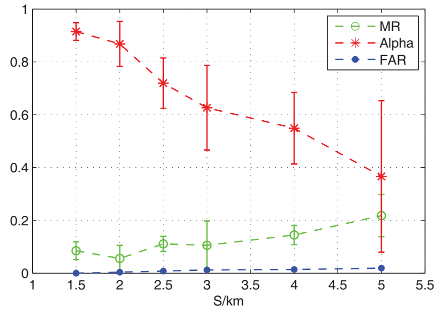

Fig. 3. FAR, MR and α in simulation 1

$No.$	ρ(node/km)	N	E	B
1	4	476	20	20%
2	2	218	20	40%
3	1	111	20	40%
4	1	111	20	60%
5	0.5	105	10	40%
6	0.5	105	10	60%

TABLE II
PARAMETER SETTINGS

In simulation 1, nodes are deployed along the roads every 0.25 km and the readings of faulty nodes bias 20% from the true value. As shown in Fig. 3, FAR and MR increase with S, while α decreases. In terms of the minor 20% measurement bias, when the detection is carried out per 2.5 km, more than 70% faulty nodes can be identified. If we do the detection per 4 km, more than half of the faulty nodes can be picked out. When the detection window becomes larger, the performance deteriorates further as the spatial correlation of the detection data set weakens.

The results of simulation 2, 3 and 5 are shown in Fig. 4. Apart from the influence of detection window size, the error reduction decreases when the node distribution becomes sparser as the number of samples decreases. Hence, in order to ensure good performance of the detection algorithm, ρ and S should be maintained at a certain high level, which is related to the expected detection resolution. That is, the more precise detection resolution is expected, the more nodes

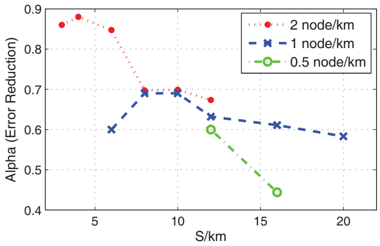

Fig. 4. Error reduction in 40% bias with various node density

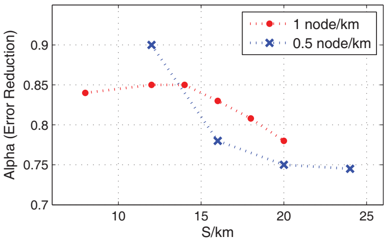

Fig. 5. Error reduction in 60% bias with various node density

should be deployed and the more frequently the algorithm should be run on board. For instance, to achieve an over 70% error reduction ratio in 40% measurement bias, we should deploy a node per kilometer and run the detection program once every ten kilometers. And according to the curves in Fig. 5 based on simulation 4 and 6, if node density is one per 2 kilometers and the detection program is run once every 20 kilometers, the measurement bias within 60% can be identified by above 75%. Table III is a list of parameter settings and the estimation of performances that can guide implementation in such metropolitan-area delay-tolerant temperature sensor networks.

Detection resolution (Bias) B	Node density ρ(node/km)	Detection window size $S(km)$	Error reduction α
20%	4	2.5	70%
40%	1	10	70%
60%	0.5	20	75%

TABLE III
PARAMETERS AND PERFORMANCES OF DETECTION ALGORITHM

Detection resolution (Bias) B	Node density ρ(node/km)	Neighborhood size (km^2)	Error reduction α
20%	4	7.5*7.5	70%
40%	1	10*10	75%
60%	0.5	20*20	75%

TABLE IV
PARAMETERS AND PERFORMANCES OF ALGORITHM IN [5]

We compare the performance of our algorithm with that in [5], where Min Ding et al. investigate a localized faulty node detection algorithm for fully connected static sensor networks. In that algorithm, the reading of each sensor is compared with its neighbors'. The algorithm is simulated upon the database used above and performs well as shown in Table IV. Compared with Table III, Table IV shows that the error reductions of two algorithms are comparable and under certain condition the algorithm in [5] performs a little better. However, note that the algorithm in [5] has a very high demand on sensor communication distance, and the performance deteriorates when the size of neighborhood deceases. To keep the error reduction at a high level, the neighborhood size should be as large as hundreds of square kilometers. It is energy-consuming and not available in the cost-effective and large-scale sensor networks.

To sum up, in this section we introduce the distributed faulty node detection algorithm for the sparse vehicular sensor network, evolving from the traditional distributed Bayesian faulty node detection algorithm in [4] for fully connected static sensor networks. A series of simulations based on real data of urban temperature and road distribution have been conducted to prove that the method works. By analyzing the simulation results, we describe how the parameters, such as the node density, detection window size and expected detection resolution, affect the performance of the algorithm. By comparison, our algorithm performs almost as well as that in [5] with a much lower demand on sensors' communication ability and works more energy-efficiently.

IV. Faulty Node Isolation

After detecting and identifying the faulty nodes successfully, isolation actions should be taken to keep them from generating and propagating faulty data through the whole network, which will harm the data reconstruction as well as introducing large wasteful communication cost. In this section, a faulty node isolation scheme based on the distributed faulty node detection is proposed and evaluated. Simulations show that the isolation scheme can effectively control the diffusion of faulty data and decrease total communication overhead. Furthermore, the optimal parameters are found to minimize the total data traffic in the whole network.

A. Scheme Description

Each carrier in the network keeps a blacklist in its buffer. When a sensor is passed through and identified as a faulty node, its ID will be added to the blacklist and then propagated in the network as an alarm. A sensor node will not shut down itself until it receives a faulty alarm. To isolate faulty nodes as soon as possible, carriers flood blacklists to each other as long as they meet. It is notable that although the isolation could control the faulty data's propagation, the transmission of blacklist itself will result in considerate additional cost. Hence a parameter BLT (Blacklist Life Time) is introduced. Each blacklist can live for BLT time slots and then expires. Larger BLT means better restraint of faulty data but more additional communication cost, and vice versa. Clearly, a tradeoff exists between the isolation performance and the overhead. The optimal BLT is discussed as follows and achieved by our simulation in the next subsection.

1) Communication Cost: The total communication overhead of the system with the isolation scheme $Cost_i$ comprises two parts, the cost brought by blacklist dissemination $Cost_b$ and the propagation of faulty data $Cost_f$. That is, $Cost_i = Cost_b + Cost_f$. While the total system overhead without faulty node isolation $Cost$ is equivalent to the communication cost of faulty data flooding. To simplify the problem, it is assumed that one unit communication cost will be introduced when forwarding either a blacklist or a data packet once in the following calculation and simulations. The encounter probability, i.e. the probability for one node(either a carrier or a sensor) to meet another in a time slot can be expressed as

$$p = \frac{2 \times (Total\ encounter\ times\ in\ T)}{T \times (Number\ of\ carriers\ and\ sensors)}, \tag{5}$$

Take a faulty sensor S_k into account that in time interval T, S_k and each carrier encounters others $\lfloor pT \rfloor$ times. So without any isolation, the communication cost resulting from S_k is

$$Cost_k = \sum_{j=2}^{\lfloor pT \rfloor} (2^{j-2} + 1) + 1 \tag{6}$$

It is assumed that T_I time slots will be cost to identify and isolate a faulty node. With the isolation scheme, the communication cost $Cost_{i_k}$ caused by the faulty node S_k is

$$Cost_{f_k} = \sum_{j=\lfloor p(T-T_I) \rfloor}^{\lfloor pT \rfloor} (2^{j-2} + 1) + 1 \tag{7}$$

$$Cost_{b_k} = \sum_{i=1}^{\lfloor pT_I \rfloor} \sum_{m=1}^{\lfloor pBLT \rfloor} 2^{m-1} = \lfloor pT_I \rfloor (\sum_{m=1}^{\lfloor pBLT \rfloor} 2^{m-1}) \tag{8}$$

$$Cost_{i_k} = Cost_{b_k} + Cost_{f_k} \tag{9}$$

i) $Cost_b$ is a monotonically increasing function of BLT and stays at a finite value when $T \geq BLT$; ii) Blacklists should live longer than the isolation time T_I, i.e. $BLT \geq T_I$. Thus, $BLT = T_I$ is the optimal solution.

2) Isolation Time: Obviously, the isolation time consists of two time periods: i) η, the time interval to detect faulty nodes and update the blacklist, which is determined by the complexity of the algorithm; ii) τ, the time cost to transmit the faulty alarm from the carrier to the sensor; i.e. $T_I = \eta + \tau$.

B. Simulation and Results

As mentioned, Fig. 2 is used as the map, which is divided into grids in size of 120×120 and 476 points among them are valid to move on representing roads. Parameters of the isolation simulation are listed in Table V. C carriers are initially set in uniform distribution on the map. During the system running time, faulty nodes emerge one by one every λ time slots. It will take η time slots for the detection algorithm to update the blacklist. A faulty node is reset to be normal, once it is isolated successfully. Parameters are set as Table VI to evaluate the isolation scheme. According to Equation 5, the encounter probability p is computed to be 14.07%.

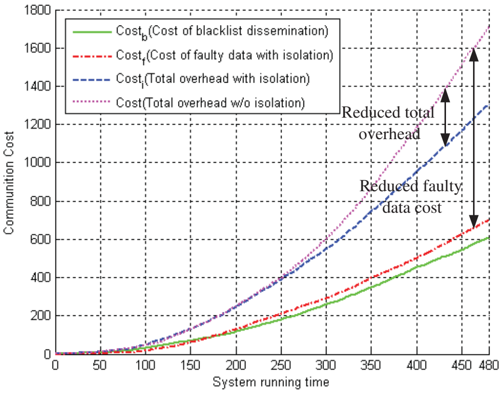

Fig. 6. Overhead comparison

C	Number of carriers
T	System running time
λ	Time interval of the occurrence of faulty nodes
η	Time interval to prepare new blacklist
BLT	Blacklist Life Time

TABLE V
PARAMETERS OF THE ISOLATION SIMULATION

$C = 40$	T=480	λ=24	η=10

TABLE VI
PARAMETER SETTINGS

1) Communication Cost: Firstly, without faulty node isolation scheme, overhead due to faulty data transmission grows exponentially as the highest curve in Fig. 6. Hence errors will contaminate the collected data quickly and harm reconstruction badly. Secondly, the faulty node isolation scheme can put a strong inhibition to the spread of faulty data as the big gap shown. Thirdly, while the diffusion of faulty data is controlled, the total system overhead also decreases. So, the faulty node isolation scheme based on distributed faulty node detection is proved to be effective on both error control and energy conservation. In the simulation corresponding to Fig. 6, $BLT = \infty$.

2) BLT Optimization: Several simulation are carried out to find the optimal BLT, which can minimize the total overhead. The probability density function of FHT(First Hitting Time) from any position to a fixed sensor (the Euclidean distance between them is not more than η steps) is shown in Fig. 7. It is well approximated by an exponential curve with the average value $\overline{FHT} = 13\ time\ slot$ [13]. Limited by the propagation of blacklists, only the carriers that share the corresponding blacklist can isolate the faulty sensor successfully, so the PDF of τ in Fig. 8 dose not follow the exponential distribution exactly but the average value $\overline{\tau} = 15\ time\ slot$ is approximately equal to $\overline{FHT}$. Thus the average isolation time $\overline{T_I} = \overline{\eta} + \overline{\tau} = 24$. Fig. 9 shows the different costs with different BLT when the system running time T=480. Obviously as the

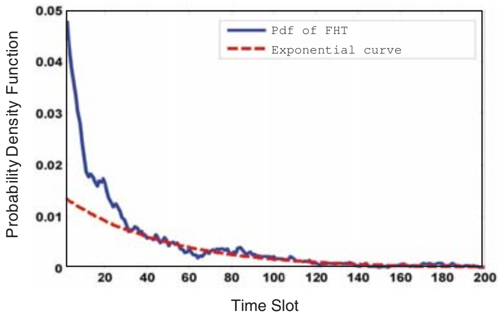

Fig. 7. PDF of FHT

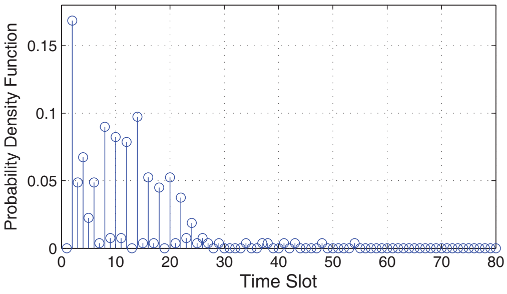

Fig. 8. PDF of τ

BLT increases, the cost of blacklist dissemination increases, while the cost of faulty data decreases in the first 30 time slots and then stays at a stable level. It is inferred that most of the faulty nodes are isolated within 30 time slots, which makes any larger BLT meaningless. Furthermore in this scenario, 26 is the optimal BLT value to make the faulty node isolation scheme most efficient according to $Cost_i$ in Fig. 9. Therefore, $\overline{FHT}$ can work as a good reference to optimize BLT in implementation. Benefitting from the isolation scheme, more than half of the faulty data can be reduced in the whole network while the total overhead decreases by 35%.

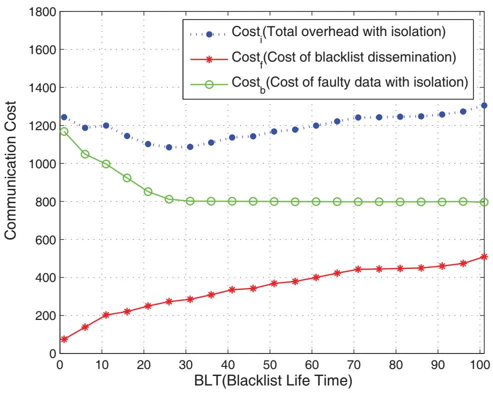

Fig. 9. Overhead in terms of BLT

V. Conclusion

This paper has designed a distributed faulty node detection and isolation algorithm for delay-tolerant vehicular sensor networks. The on-board delay-tolerant distributed faulty node detection algorithm utilizing sample data's spatial correlation demands so low on nodes' communication and computation ability that is suitable for mobile sparse sensor networks. Based on the distributed faulty node detection algorithm, a faulty node isolation algorithm is proposed aiming at both controlling the diffusion of faulty data and reducing communication cost. A temperature collection system is simulated on real data of urban temperature and roads distribution, which suggests the distributed faulty node detection and isolation algorithm effective. By analyzing the simulation results, we describe how the parameters affect the performance of the algorithms and achieve the optimal settings. A list of parameter settings in such metropolitan-area vehicular temperature sensor network is provided to help implementation. With feasible parameter settings, over 70 percent of faulty nodes can be identified. More than half of the faulty data can be reduced in the whole network while the total overhead decreases by 35%. It is noticed that a lot of sensor data, such as environmental measurements, are correlated in both space and time. So in future researches, the temporal correlation of data should be taken into account to improve the faulty node detection accuracy.

References

[1] L. Ruiz, I. Siqueira, L. e Oliveira, H. Wong, J. Nogueira, and A. Loureiro, "Fault management in event-driven wireless sensor networks," in *Proceedings of the 7th ACM international symposium on modeling*, pp. 149–156, 2004.

[2] J. Hill, R. Szewczyk, A. Woo, S. Hollar, and J. Heidemann, "A software architecture to support network sensors," Master's thesis, UC Berkeley, 2000.

[3] S. Madden, M. J. Franklin, J. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," in *OSDI*, December 2002.

[4] B. Krishnamachari and S. Iyengar, "Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, no. 3, pp. 241–250, 2004.

[5] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *in Proceedings IEEE INFOCOM*, 2005.

[6] X. Luo, M. Dong, and Y. Huang, "On distributed fault-tolerant detection in wireless sensor networks," *IEEE Transactions on Computers*, no. 1, pp. 58–70, 2006.

[7] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, pp. 65–72,, 2006.

[8] A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection," *IEEE Transactions on instrumentation and measurement*, no. 5, 2008.

[9] C. Vasar, I. Filip, and I. Szeidert, "Fault detection methods for wireless sensor networks using neural networks," in *IEEE International Joint Conferences on Computational Cybernetics and Technical Informatics (ICCC-CONTI)*, pp. 27–29, 2010.

[10] P. Zhang, "Security mechanism design for metropolitan sensor networks," Master's thesis, Tsinghua University, 2010.

[11] L. T. M. (TM), "U.S. Geological Survey (USGS)." http://eros.usgs.gov/products/satellite/tm.php.

[12] Google, "Google Map." http://www.google.com/maps.

[13] R. B. Ellis, "Torus hitting times from greens functions," 2001.