A Fault-tolerant Aggregation Schema in WSN

Wang Na and Li Liping
School of Computer and information, Shanghai Second Polytechnic University, Jinhai RD, Shanghai, China
wnoffice@126.com

Keywords: Wireless Sensor Network, Two-Valued, Aggregation, Fault-tolerant, Byzantine Attack.

Abstract: One of the critical tasks in designing a wireless sensor network is to monitor, detect, and report various useful occurrences of events. The result of data aggregation is very important for running of networks. Fault-tolerant is critical to the efficiency of data aggregation scheme. In this paper, we considered a special condition of two-valued data and present a data aggregation scheme on account of byzantine attack. It is proved that the algorithm in this scheme has polynomial complexity and can tolerant about one fourth fault in the whole network. Simulations showed the effect of our model with different number of nodes. Comparison with EFSA showed our model has higher fault detection rate.

1 INTRODUCTION

Wireless sensor network consisting of large number of micro sensor nodes can complement the collaborative awareness in many areas. One of the critical tasks in designing a wireless sensor network is to monitor, detect, and report various useful occurrences of events in the network domain. An event can be defined as an exceptional change in the environmental parameters such as temperature, pressure, humidity. However, an event may occur in many ways such as a sudden change of sensed parameters or may have a gradual and continuous change over time. Data aggregation is a popular research recently to decide an event.

Nodes have several hardware and software components that can produce malfunctions. For example, the enclosure can suffer impacts and expose the hardware of the sensor node to the extreme conditions of the environment.

When the battery of a node reaches a certain stage, sensor readings may become incorrect. Hardware failures will generally lead to software failure. A Data aggregation application will not perform properly if the underlying sensors are providing incorrect readings. So fault tolerance is critical to the efficiency of data aggregation scheme. In the paper discussed, although the node cannot be used to provide correct sensor readings it still can be used to route packages in the sensor network.

The rest of the paper is organized as follows. The detailed aggregation schema is depicted in Section 3. The analysis and evaluation of trust model are given in Sections 4. The related work and our conclusions are presented in Sections 2 and 5.

2 RELATED WORKS

2.1 Node Faults

Nodes have several hardware and software components that can produce malfunctions. For example, the enclosure can suffer impacts and expose the hardware of the sensor node to the extreme conditions of the environment. When the battery of a node reaches a certain stage, sensor readings may become incorrect. Hardware failures will generally lead to software failure. A Data Acquisition application will not perform properly if the underlying sensors are providing incorrect readings. Nevertheless, some hardware failures do not affect all the services in a sensor node. In the example discussed, although the node cannot be used to provide correct sensor readings it still can be used to route packages in the sensor network.

Organizing a network in clusters is an approach used in many applications, for example to extend the lifetime of the network. A small number of nodes are selected to become cluster heads. They are responsible for coordinating the nodes in their clusters, for instance by collecting data from them and forwarding it to the base station. In case that a cluster head fails, no messages of its cluster will be

Liping L. and Na W.
A Fault-tolerant Aggregation Schema in WSN.
DOI: 10.5220/0006027504220417
In *Proceedings of the Information Science and Management Engineering III (ISME 2015)*, pages 422-417
ISBN: 978-989-758-163-2

A Fault-tolerant Aggregation Schema in WSN

forwarded to the base station any longer. The cluster head can also intentionally or due to software bugs forward incorrect information. Depending on the application case, the impact of such a failure can vary from quality degradation of measurements to alarm messages not being delivered to a back-end system.

While forwarding messages, nodes can aggregate data from multiple other nodes in order to reduce the amount of data sent to the base station. One common simple approach is to calculate the average of correlated measured values such as temperature, humidity and pressure, sending only one message to the back-end. If a node generates incorrect data, the data aggregation results can suffer deviations from the real value. Also, if a node responsible for generating the aggregated data is subject to a value failure, the base station will receive incorrect information of an entire region of the network.

2.2 Related Models

Fault tolerance has been recognized by some authors. Yunxia Feng presents a fault Tolerant Data Aggregation Scheduling with Local Information. Authors divided a fault tolerant data aggregation protocol into two parts: basic aggregation scheduling and amendment strategies (Yunxia Feng, 2002). On default, data is aggregated according to the basic aggregation scheduling strategy. The amendment strategy starts after a middle sensor node is out of service. Nodes here are identified as Directly Affected Node, Indirectly Affected Node and Non-Affected Node. It emphasizes on how to amend the tree according to various types of fault nodes.

XIAO Wei presents a Fault tolerant scheme for data aggregation in event cluster. He used K-means to aggregate data that can make approximated data in a set and proposed a concept of creditability whose value is calculated according to an average value coming from the set (XIAO Wei, 2006). The former described recovery of the aggregation tree after the recognition of fault nodes, while the latter devote to resolving faulty determination but does not consider the isolated point. YANG Yan presented an Ant Colony Algorithm MACCA which makes different speed ant colony cluster with SACA parallel to divide approximated set. But it also does not present fault recognition and isolated point (Yang Yan, 1984).

Ganeriwal S, Kumar R, Srivastava M B proposes a fully distributed K-Means algorithm (Epidemic K-Means) which does not require global communication and is intrinsically fault tolerant. The proposed distributed K-Means algorithm provides a clustering solution which can approximate the solution of an ideal centralized algorithm over the aggregated data as closely as desired(Ganeriwal S, Kumar R, Srivastava M B, 2003). But the algorithm assumes nodes communicating peer to peer which doesn't consider the communication complex and energy consumption.

3 A POLYNOMIAL ALGORITHM FOR FAULT-TOLERANT AGGREGATION

When detect abnormality, sensors must be clustered into numbers of local sensor networks according to the region they are located. Besides, each region of sensors has their own autonomy. In other words, all sensors, which are in the same region, can execute the proposed protocol without the sink and other unconcerned sensors. This can reduce the time for collecting data and designing the final result. We use LEACH to create clusters.

In classic WSN, if the detected value meets some condition, then the initial value must be set to 1, otherwise, 0. Take the fire control system for instance, if the sensor detects the temperature is higher than 50°C, then its initial value is 1, otherwise, 0 as default (Dai Hui, Han R, 2004). After clustering, the proposed protocol can let each sensor reach an agreement and do the corresponding action with the following assumptions.

- Let N be the set of all sensors in the local autonomous WSN and| N |=n.
- The total number of faulty sensors and transmission media in the local WSN is $\lfloor(n-1)/4\rfloor$.
- Each sensor needs to collect messages through $\lfloor(n-1)/4\rfloor+1$ phases of message exchange and each phase has two round.
- Each sensor has its own initial value Vi.

In the message exchange phase, each sensor collects and exchanges messages from other sensors with $\lfloor(n-1)/4\rfloor+1$ phases of message exchange. As shown in Fig. 1. In Fig 1, the algorithm contains f+1 phases, each taking two rounds. Each node has a preferred decision for each phase. At the first round of each phase, all nodes in one cluster send their preferences to each other. Let v_i^k be the majority value in the set of values received by node p_i at the end of the first round of phase k. If there is no majority, then a default value is used. In the second round of the phase, node p_k which called the king of phase sends its majority value v_k^k to all nodes. If p_i receives more than n/2+f copied of v_i^k then it sets its preference for the next phase to be v_i^k. Otherwise, it sets its preference to be the phase king's phase, the node decides on its preference.

```
Algorithm(for single node)

    Initially pref[i]=x and pref[j]=v⊥, for any j≠i
    Round 2k-1, 1<=k<=f+1;
       Send <pref[i]> to all processors
    Receive <vj> from pj and assign to pref[j], for all
0<=j<=n-1, j≠i
       Let maj be the majority value of pref[0], pref[1]……,
pref[n-1] (v⊥ if none)
       Let mult be the multiplicity of maj
    Round 2k, 1<=k<=f+1;
       If i=k then send <maj> to all precessors
       Receive <king-maj> fro, pk (v⊥if none)
       If mult>n/2+f
       Then pref[i]:=maj
       Else pref[i]:= king-maj
    If k=f+1 then y:= pref[i]
```

Figure 1: Deployment of nodes.

For example in Figure 2（the commander is not faulty）, We consider the top two level which is a cluster. In the tree, node 1 is proposed as a common node. At the second level, there are four nodes and we assume that N is 5 and f is 1. Then it needs 2 phases and 4 round to complete this two-valued aggregation. We set initial values as:

(1) Initial value: p1=1, p2=1, p3=1, p4=1, p5=1. And p3 is fault.

(2) Initial value: p1=1, p2=1, p3=0, p4=1, p5=0. p1 is the faulty.

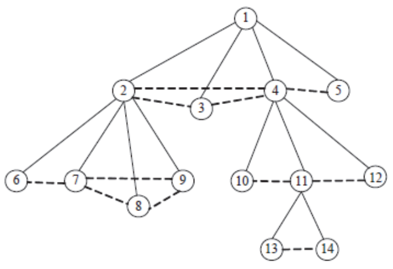

Figure 2: Hierarchical cluster tree.

With execution of algorithm in Figure 1, the result of (1) is shown in Table 1 and result of (2) is shown in Table 2.

Table 1: THE faulty nodes AND TESTING of (1).

	p1	P2	P3	P4	p5
round1	Send 1	Send 1	send 0	Send 1	Send 1
Receive	{1,0,1,1}	{1,0,1,1}	{1,1,1,1}	{1,1,0,1}	{1,1,0,1}
	Maj= 1	1	1	1	1
	Mul =4	4	5	4	4
round2	Pref=1	1	1	1	1
round3	Maj=1	1	1	1	1
	Mul>=4	>=4	>=4	>=4	>=4
Round4	Pref=1	1	1	1	1(agreement)

Table 2: THE faulty nodes AND TESTING of (2).

	p1	P2	P3	P4	p5
round1	send 0 to p2, send 1 to p3, send 0 to p4, send 1 to p5	Send 1	send 0	Send 1	Send 0
Receive	{1,0,1,0}	{0,0,1,0}	{1,1,1,0}	{0,1,0,0}	{1,1,0,1}
	Maj= 1	0	1	0	1
	Mul =3	3	3	3	3
round2	Pref=1	0	1	0	1
round3	Maj=1	0	1	0	1
	Mul=3	3	3	3	3
Round4	Pref=0	0	0	0	0(agreement)

Table 1 and 2 show when node 5 or 1 (command node) is fault, the prefer value still keep 1 since the numbers of fault nodes is no more than $\lfloor(n-1)/4\rfloor$.

4 EVALUATIONS

Our simulation experiment is based on ns3. Fifty sensor nodes are distributed in a space of 500×700, and the communication radius is set as 60. Each node has two to five neighbors in the experiment and the node's location is already known.

We analyze the efficacy of Trust model against abnormal data and fault data though generating a few abnormal nodes and faulty nodes. Firstly, we input 59 nodes and give a fixed probably faulty rate as 0.2.

The relation between networks scale and detection rate can be described as in Figure 3.

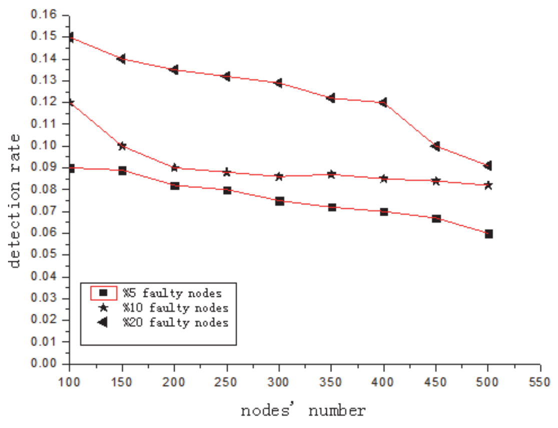

Figure 3: Detection rate with different networks scale.

This figure shows that detection rate will rise with the increasing of sensor nodes.

When fault rate is below 0.4, we may get different result of faulty detection rate. The simulation result compared with EFSA is shown in Figure 4.

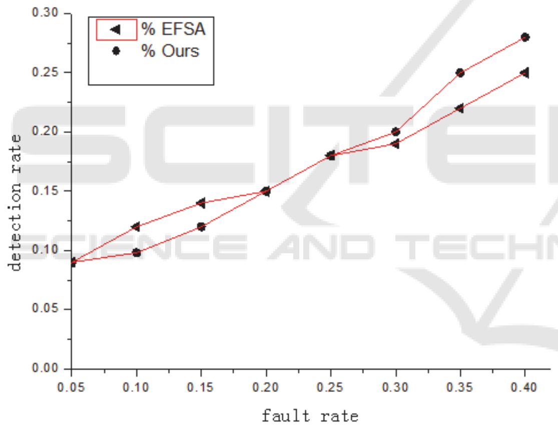

Figure 4: Detection rate with different fault rate.

This figure shows that detection rate will rise with the increasing of faulty nodes. And our model has higher detection rate than EFSA.

5 CONCLUSIONS

This algorithm is based on byzantine problem and be used in aggregation of WSN. But it only suits for two-valued data which can make a decision of normal or abnormal. It can tolerant ⌊(n-1)/4⌋ failures with executing the proposed algorithm. Compare to other algorithm, the communication complexity is reduced leading to less energy consumption. Meanwhile, the number of fault nodes that can be tolerant also decreases. Therefore, the scheme can be used in the early period of network, when the faulty increasing, other algorithm should applied in the case.

ACKNOWLEDGEMENTS

This work is supported by the key discipline of Shanghai second polytechnic University named software engineering (No.XXKZD1301). The authors would like to thank the referees for their invaluable comments and suggestions.

REFERENCES

Yunxia Feng Wireless sensor networks: a survey, Computer Networks 38 (2002) 393–422

Xiao Wei, Byzantine Algorithms in Wireless Sensors Network, ICIA 2006

Yang Yan. Using Time Instead of Timeout for Fault-tolerant Distributed Systems. ACM Trans. onProg. Lang. and Sys. 6, 2 (April 1984), 254-280.

Ganeriwal S, Kumar R, Srivastava M B. Timing-sync Protocol for Sensor Networks[C]//Proc. of the 1st ACM Conf. on Embedded Network Sensor Systems. Los Angeles, CA, USA: [s. n.], 2003.

Dai Hui, Han R. TSync: A Lightweight Bi-directional Time Synchronization Service for Wireless Sensor Networks [J]. ACM Mobile Computing and Communications Review, 2004, 18(1): 125-139.