International Journal of Security and Its Applications
Vol. 10, No. 4 (2016), pp.33-40
http://dx.doi.org/10.14257/ijsia.2016.10.4.04

An Efficient and Robust Data Integrity Verification Algorithm Based on Context Sensitive

Feng Xie*, Hu Chen

Naval University of engineering, Wuhan, Hubei, 430033, China
whxiefeng@163.com

Abstract

There exist two key problems about data aggregation that should be thoroughly explored - algorithm design in networking layer, and algorithm design in application layer. Those two problems should be subtlety tackled in termers of high efficiency and robustness. Therefore, the former one requires the survivability and highly reliable design at networking layer, the latter one usually asks for high efficiency and robustness at application layer. Moreover, the optimization of algorithms is also considered for further enhancement. The integrity check is a key requirement for optimization. The context-aware and cross-layer design is applied in the optimization. A dynamic fragment odd-even parity checking code is proposed, and a context-aware aggregative integrity check code is proposed.

Keywords*: Data Aggregation; context-aware aggregative integrity; IoT*

1. Introduction

With the development of Internet of Things (IoT), it is possible for relying IoT technologies for the monitoring of technical status on mechanical and electrical equipments. We call it as technical status IoT for environment (TSIE) [1-6]. This technology provides high efficient, real time, dynamical in adjustments, and highly accurate system monitoring. A large volume of data will be generated in the monitoring. Thus, the data aggregation in TSIE is a critical issue for the efficiency of data collection in such kind of IoT, and it should be addressed firstly [7-12]. [1]

Network layer aggregation protocol design problem and application layer aggregation algorithm design problem are two key research issues in the design of the network layer aggregation protocol [13-16]. On the basis of the existing research on these two issues, this paper mainly focuses on the optimization design of the algorithm. The optimization design includes two aspects: one is the efficiency of the algorithm; the one is the robustness of the algorithm. The performance of the algorithm is mainly for the calculation of the algorithm and the communication energy consumption is low. Robustness and reliability are mainly expressed as data [17-21]. The existing literature on data aggregation algorithms mainly includes the efficiency and the efficiency of the network layer algorithm design, and the efficiency and reliability of the application layer algorithm design, In this paper, we will study the optimization design problem from the perspective of context sensitive (Context-aware) and cross layer design (Design Cross-layer), and take into account the overall data aggregation algorithm design in order to further improve the algorithm in the practical application of adaptive. The main difficulty of data aggregation algorithm is that the scene is complex, and it may have different applications in different scenarios. Therefore, context sensitive design and cross layer design will benefit from the overall situation to get a feasible

Feng Xie is the corresponding author.

ISSN: 1738-9976 IJSIA

way. Therefore, the main problem of this paper is: From the context sensitive design and cross layer design, the idea of network layer and application layer design based on actual application scenarios and the idea of application layer design are proposed. Because data aggregation algorithm involves many problems, this paper mainly focuses on data integrity problem.

2. Context Sensitive Efficient Robust Data Integrity Verification Sensitive

2.1 Modeling of Data Aggregation Integrity Verification Optimization Problem

The robustness of data aggregation process and results is a basic requirement of data aggregation, and in data aggregation algorithm, the correctness of the results of data aggregation algorithm is presented. For example, when the perceived data in the transmission process of interference, the results of data aggregation may be wrong, so to protect the integrity of the data, the receiver can understand whether there is a mistake in the perception data. This section starts with the data integrity verification to study the optimization of the algorithm.

A context sensitive data aggregation integrity verification algorithm is proposed here. We sum up the common context information that can be considered include:

(Condition1) Network topology. This situation includes the changes of the network topology, and the mobility of data aware nodes, and the mobility of data aggregation nodes.

(Condition 2) Data characteristics. This situation includes the characteristics of perception data and the characteristics of clustered data.

(Condition 3) Data aggregation model. This situation includes the time period of data aggregation, and the timeliness requirements.

Without loss of generality, the modeling of the problem is given, which is convenient for the following discussion, at the same time, it makes the algorithm design is general, and the application of the algorithm is universal.

Definition 1: Data aggregation of single hop single data integrity verification problem (P1). We assume the perceived data is S, and the data sent to the data gathering node is S'. Let S'=f(S), then data aggregation nodes need to recover S from S', and it is necessary to confirm the integrity of S is maintained.

Definition 2: Data aggregation single hop multiple data integrity verification problem (P2). We assume perceived data is <S1, S2,…, Sn>, and data to the data gathering node is <S1', S2',…,Sn'>. Let Si'=f (Si), then data aggregation nodes need to recover Si from Si', and it is necessary to confirm the integrity of Si is maintained.

Definition 3: Data aggregation multi hop data integrity verification problem (P3). We assume perceived data is <S1, S2,…, Sn>, and the distance between the sensing nodes and the data nodes is multi hop. This situation can be divided into two cases: the end node aggregation and the intermediate node aggregation.

(P3-1) The end-node aggregation. We assume that the data is transmitted to the end node for data aggregation, and the intermediate node only has the function of data forwarding. We assume that the data sent to the end node is <S1', S2',…, Sn'>. Let Si'=f(Si), then data aggregation nodes need to recover Si from Si', and it is necessary to confirm the integrity of Si is maintained. It is easy to see, this situation is similar to the P2,but this case need deal with weather need to verify the integrity of the intermediate nodes.

(P3-1-1) If the integrity of the intermediate nodes is verified, the calculation of energy consumption will be generated. However, if the integrity verification of the data is found to be invalid, then the energy consumption of the communication can be avoided.

(P3-1-2) The intermediate nodes do not carry out the integrity verification, and the integrity verification is done by the end node. Then the intermediate node has no energy consumption, but may have communication energy consumption.

(P3-1) The intermediate-node aggregation. Intermediate nodes carry out the data aggregation, and then the amount of data transmitted among the nodes is reduced. This can reduce the data communication overhead. However, the intermediate nodes assume the computational overhead of the integrity verification and the computation overhead of data aggregation.

2.2 The Integrity Verification Based on DFOEP

In this section, a new method of dynamic Fragment Odd-Even Parity Check is proposed. The method can use context information, such as data features, network topology, combines with the network layer and application layer of cross layer design concept to present an efficient method for verifying the integrity of the method.

Firstly, problem P1 is discussed. This situation is a basic scene. A basic general protocol is introduced in this section. Let A the sensing node, and B express aggregation node, then A→B：{S'}. The following mainly discusses the composition of S'. First, a common method is presented to be called the basic comparison method.

Before giving common method, we first analyze the adversary model. The adversary model here mainly refers to random errors. In this case, due to the limitation of data transmission distance, the adversary cannot get the data. Therefore, in general, there will be no artificial tampering with the data, but the data may encounter electromagnetic interference, or random data perturbation error, so here must be defensive random error.

Based on the assumption about the adversary model, the usual basic plan is to assume:

$$S' = f(S) = S||T \quad (1)$$

Here, $T=\delta(S)$. $\delta()$ is a cryptographic hash function, which meets the second preimage resistance.

We call this method the basic contrast method.

$$|S'|=|S|+|\delta()| \quad (2)$$

If $|S|>>|\delta()|$, the method is appropriate. However, if S meets a Data Smooth Form, then it could be $|S|<|\delta()|$. In this case, the efficiency of this method is not high. The Data Smooth Form to be short for DSF has a feature that represents smoothness of data. If the data to be transmitted is<D1, D2, …, Dn>, DSF represents the data changes, data structure and data value range for<D1, D2, …, Dn>.

Here is an example of DSF.

DSF1: Change of data smoothing paradigm.

$$|Di-Di-1|<\Delta 1, \quad (i=2,..,n) \quad (3)$$

Here $\Delta 1$ represents a non negative value, which represents the difference degree between the data.

Back to the discussion of the $\delta()$ function. If $\delta()$ is a full check code, such as parity check code, cyclic redundancy check code, *etc*, in the basic plan, the Delta () will be reduced. However, in the error check, compared with the hash function, the error checking ability is greatly reduced.

Therefore, a new approach is proposed, which is based on the context (the context is mainly referring to the length of the perception data), to dynamically determine the selection of the δ ().That is, we propose a variable length parity check code based on the length of the S.

Problem P1 solution:

Here, let $T=\delta(S)$, and let δ () be used as a variable - length parity check code, then we can get:

$$T=|\delta(S)| = p*|S| \quad (4)$$

According to |T|, the S is divided into |T| parts, S is expressed by S[i] (i=1,…,|T|). Making Parity (S[i]||Ti) =0, and Ti is the fifth bits for T, then the function Parity (x) return the result of each binary bits XOR about x.

P is a variable length system parameter, and its dynamic adjustment strategy (p-Policy) is as follows:

First, assume that p=α+Δ. Here, the initial value of Δ is β. If the last time parity error occurred, then

$$\triangle \leftarrow \text{Min}\ (2*\triangle+\beta, 1-\alpha) \quad (5)$$

If it is successful, then

$$\triangle \leftarrow \text{Max}\ (\triangle-\beta, 0) \quad (6)$$

Usually, if |S|<10, this case meets the balance, Recommended system parameters (α , β) = (0.5, 0.01).

Here we give Delta and P changes with change of network error status change figure. Network error sample is 100 times. The errors are generated randomly, and do not appear in a row. So the adjustment is not continuous.

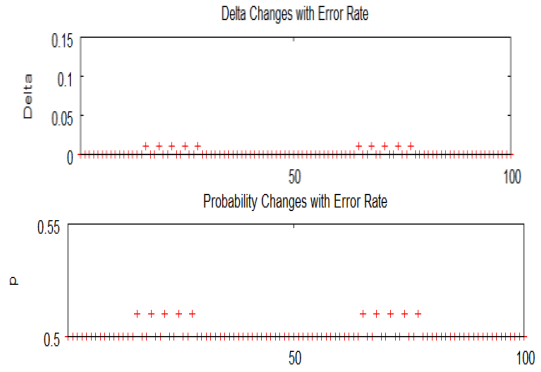

Figure 1. Delta and P Changed with the Change of Network Error State (α，β) = (0.5，0.01)

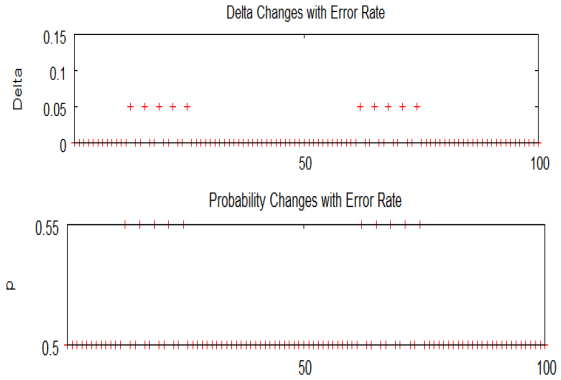

Figure 2. Delta and P Changed with the Change of Network Error State (α，β) = (0.5，0.05)

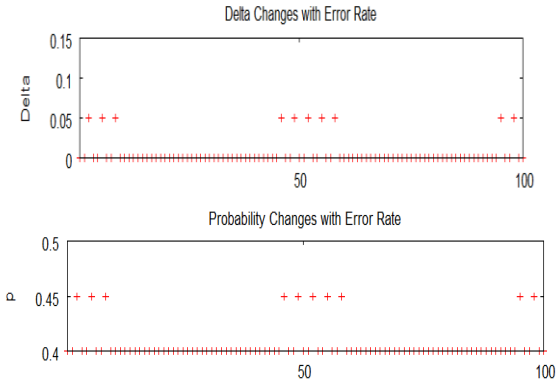

Figure 3. Delta and P Changed with the Change of Network Error State (α，β) = (0.4，0.05).

The above discussion is mainly aimed at the problem P1, and now we discuss the problem P2. Here we use the same model as the front.

Firstly, a basic general protocol is introduced. Make the perception node to be A1, …, An, and B represents data aggregation node, then we can get Ai→B：{Si'}(i=1, …, n).

Below we mainly discuss the composition of Si'(i=1,…,n). For this problem, the method to solve the problem of P1can be used. That is to say, the verification of multiple data is considered to be the order of a single data.

The methods used are as follows.

We used B to verify the Si '=' = f (Si) = Si||Ti. Here let T=δ(Si), and let δ() to be as variable length parity check codes. According to the previous section, we can get

$$T=|\delta(Si)| = p*|Si| \quad (7)$$

And we can get Parity (Si||Ti) =0. Here the function Parity (x) returns the result of each binary bit XOR about x.

The above solution is easy to think of, However, we propose here a more ingenious method, that is, verification only needs to be done once, thereby reducing the number of validation calculations and comparisons. That is reduced n-1 times.

At this time, if Parity (∑(Si||Ti)) = Parity(S1||T1||…||Sn||Tn) = 0, the data integrity verification is successful. The defect of this method is that if there is a failure of integrity verification, then we need carry out N calculation and comparison to determine which data integrity is changed. This step is used to determine which parameters cause the integrity change.

Below we discuss the problem P3. Here we also use the same model as the front section.

First discuss the case (P3-1). In this case, the intermediate node only plays the role of forwarding, not data aggregation computation. For the case (P3-1-1), each intermediate node has a complete check. If it is found that the integrity verification is false, it is considered that the data is invalid and no longer be forwarded, and the p-Policy's integrity check parameters are adjusted. For (P3-1-2), the intermediate nodes are not fully verified, and the forwarding operation is performed, and the verification is performed by the end node. In this case, although the intermediate nodes do not have the integrity of the calculation of energy consumption, but there may be unnecessary data forwarding, resulting in poor communication energy efficiency. In the case of P3-1, the communication energy consumption in the case is higher than that of P3-2, because there is no data aggregation in the intermediate nodes, and the amount of data transmission is higher than that of P3-2.

The following is a discussion of the P3-2 problem. In this case, the intermediate nodes carry out data aggregation and data forwarding. The data aggregation in the intermediate nodes can reduce the amount of data forwarding and save energy consumption. Moreover, in the intermediate nodes for data validation, you can understand the current situation of communication interference, and it is conducive to make appropriate adjustments.

We assume that the sensing data generated by the sensing node Ai is Si=<Si1, Si2,…, Sin>, and the data will eventually converge in the aggregation node B. Between A and B, there are multiple intermediate nodes C1, C2, …, Cm. We need to find a method of encoding data integrity, and convert the Si to Si '=<Si1', Si2', …, Sin'>. And the Si ' take intermediate node aggregation computation in the Cj. The final result will be completed in B.

For this problem, data aggregation is faced with the requirements of a distributed aggregation, that is, it requires the data to be independent in multiple intermediate nodes, and can guarantee the integrity of the maximum. In this case, we can extend to the idea of P1-Solution, but also add the use of the topological structure of the context to make the integrity check code is more flexible and efficient. The main programmer (P3-1-Solution) is as follows:

Without loss of generality, we assume that Ai (i=1,…,m) 's sensing data Si '(i=1,…,m) is converging in Cx, and Aj (j=m+1,…,n) 's sensing data Sj '(j=m+1,…,n) is converging in Cy. Cx and Cy bring together the results to B to converge. If Ck (k=x, y) finds a problem, the p-Policy policy will be adopted to adjust the p value. If Ck (k=x, y) finds no verification problem, then we calculate the final result ∑Si'（i=1,…,m） and ∑Sj'（i=m+1,…,n）after the convergence, and adopt p-Policy strategy to adjust the p value, and add the corresponding T. The advantage of this scheme is that the intermediate nodes can adjust the strength of the integrity verification according to the current disturbance, moreover, the integrity verification is done in the middle of one by one, and it is sensitive to the interference of environmental integrity.

2.3 Performance Analysis

Firstly, the definition of performance evaluation index is given.

Define 4: Length Overhead (LO). It is the ratio of the length of the data integrity to the length of the original data.

Define 5: False Negative (FN). It is data integrity verification is not wrong but the actual data integrity is changed.

Define 6: False Positive (FP). It is, the data integrity check is wrong, but the actual data integrity has not changed.

It is easy to see that LO characterizes the efficiency of the integrity of the encoding method, and FN and FP characterize the accuracy of the integrity of the encoding method.

Proposition 1: In the basic comparison method, we can get

$$LO=| Delta () |/|S| \quad (8)$$

Proof. |S'|=|S|+|δ()|.The δ() is determined by the hash function, which is the same length, and usually meet the |S|<| DSF Delta (). Therefore, LO>1.

Proposition 2: In policy P1-Solution, LO=p<1.

Proof. Because |S'|=|S|+p*|S|, we can get LO=p*|S|/|S|=p<1.

Proposition 3: In the basic contrast method, FP=0, FN=0. Li.

Proof. By the property of the hash function, we can get the conclusion.

Proposition 4: In P1-Solution, FN=p, FP=0.

Proof. When |T|=p*|S|, if the number of errors is less than |T|, it can be detected. When the number of errors is greater than |T|, it cannot be detected. So we can get FN=|T|/|S|=p*|S|/|S|=p.

3. Conclusion

In this paper, we first study the problem of data aggregation integrity verification, and then analyzes the composition of context information. The completeness of data aggregation is given, and a new method for verifying the integrity of data is presented. For verifying the efficiency of the algorithm, a data integrity verification model is proposed, and we prove the lower bound of the data integrity check model. Lastly, we propose a context sensitive clustering scheme for the integrity verification.

References

[1] P. Jesus, C. Baquero, P.S., Almeida, "A Survey of Distributed Data Aggregation Algorithms", IEEE Communications Surveys & Tutorials, no.99, pp.1,1.
[2] Z. Ye, A.A Abouzeid, Ai, Jing, "Optimal Stochastic Policies for Distributed Data Aggregation in Wireless Sensor Networks", IEEE/ACM Transactions on Networking, vol.17, no.5, Oct. **(2009)**pp.1494-1507.
[3] S. Srinivasan, A. Azadmanesh, "Survivable Data Aggregation in Multiagent Network Systems with Hybrid Faults", IEEE Transactions Computers, vol.62, no.10, Oct. **(2013)**, pp.2054-2068.
[4] H. Salarian, K-W Chin, F. Naghdy, "An Energy-Efficient Mobile-Sink Path Selection Strategy for Wireless Sensor Networks", IEEE Transactions on Vehicular Technology, vol.63, no.5, Jun **(2014)**, pp.2407-2419.
[5] D. Takaishi, H. Nishiyama, N. Kato, R. Miura, "Toward Energy Efficient Big Data Gathering in Densely Distributed Sensor Networks", IEEE Transactions on Emerging Topics in Computing, vol.2, no.3, Sept. **(2014)**, pp.388-397.
[6] W Zhao, X Tang, "Scheduling Sensor Data Collection with Dynamic Traffic Patterns", IEEE Transactions on Parallel and Distributed Systems, vol.24, no.4, April **(2013)**, pp.789-802,
[7] A. M. Alsheikh, S Lin, D Niyato, H-P Tan, "Machine Learning in Wireless Sensor Networks": Algorithms, Strategies, and Applications, IEEE Communications Surveys & Tutorials, vol.16, no.4, Fourthquarter **(2014)**, pp.1996-2018.
[8] H Zhang, H Shen, "Balancing Energy Consumption to Maximize Network Lifetime in Data-Gathering Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, vol.20, no.10, Oct. **(2009)**, pp.1526-1539.
[9] P.T Shaw, S. Peaslee, M.O Ferguson, "Integrated and distributed Position Navigation and Timing (PNT) data in shipboard environments[C]", Proc. of OCEANS04, vol.2, no., pp.796,801, vol.2 ,pp.9-12 Nov. **(2004)**.
[10] R. Swanson, S.C. Cash, W.C Pettway, C.A Peterson, K. Sharp, "A current overview of NAVOCEANO's Ocean Projects Department's roll-on/roll-off data collection vehicles and support systems[C]", Proc. of OCEANS02, vol.4, no., pp.2054-2059 , vol.4, pp. 29-31 Oct. **(2002)**.
[11] G.J. Dobeck, "Algorithm fusion for automated sea mine detection and classification [C]", Proc. of OCEANS01, vol.1, no., pp.130-134, vol.1, **(2001)**.
[12] M. Bagaa, Y. Challal, A. Ksentini, A. Derhab, N., Badache, "Data Aggregation Scheduling Algorithms in Wireless Sensor Networks: Solutions and Challenges", IEEE Communications Surveys & Tutorials, vol.16, no.3, pp.1339-1368, **(2014)**.
[13] J. Lin, N. Xiong, A.V.; Vasilakos, G. Chen, W. Guo, "Evolutionary game-based data aggregation model for wireless sensor networks", IET Communications, vol.5, no.12, August 12 **(2011)**, pp.1691-1697,
[14] XH Xu, M Li, XF Mao, S Tang, SG Wang, "A Delay-Efficient Algorithm for Data Aggregation in Multihop Wireless Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, vol.22, no.1, Jan. **(2011)**, pp.163-175.
[15] H Jiang, S Jin, C Wang, "Parameter-Based Data Aggregation for Statistical Information Extraction in Wireless Sensor Networks", IEEE Transactions on Vehicular Technology, vol.59, no.8, Oct. **(2010)**, pp.3992-4001.
[16] D.C. Hoang, R. Kumar, S.K. Panda , "Optimal data aggregation tree in wireless sensor networks based on intelligent water drops algorithm", IET Wireless Sensor Systems, vol.2, no.3, September **(2012)**, pp.282-292.
[17] H Jiang S Jin, C Wang, "Prediction or Not? An Energy-Efficient Framework for Clustering-Based Data Collection in Wireless Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, vol.22, no.6, June **(2011),** pp.1064-1071.
[18] H.R. Dhasian, P. Balasubramanian, "Survey of data aggregation techniques using soft computing in wireless sensor networks", IET Information Security, vol.7, no.4, December **(2013)**, pp.336-342.
[19] J He, S Ji, Y Pan, Y Li, "Constructing Load-Balanced Data Aggregation Trees in Probabilistic Wireless Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, vol.25, no.7, July **(2014)**, pp.1681-1690.

[20] T H.O, I Korpeoglu, I. Stojmenovic, "Computing Localized Power-Efficient Data Aggregation Trees for Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, vol.22, no.3, March **(2011)**, pp.489-500.
[21] Y Ma, Y Guo, X Tian, Ghanem, M, "Distributed Clustering-Based Aggregation Algorithm for Spatial Correlated Sensor Networks", IEEE Sensors Journal, vol.11, no.3, March **(2011)**, pp.641-648.

Author

Feng Xie. He graduated from Shandong University in Information Science and engineering, and his master's degree in Southeast University. Now he is engaged in the research of data aggregation and development of mobile dynamic positioning system.

Hu Chen. He graduated from the University of Naval Engineering, computer science. His master also enrolled in the school of Electronic Science and Engineering. Now he is mainly engaged in the research of data aggregation and efficient network design algorithm.

