Indian Journal of Science and Technology, Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/108022, December 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Data Dissemination Framework for IoT based Applications

S. Amrutha, T. Mohanraj, N. Chakrapani Ramapriya, M. Sujatha, R. Ezhilarasie and A. Umamakeswari*

School of Computing, SASTRA University, Thanjavur - 613401, Tamil Nadu, India; amrutha@gmail.com, tmohanraj@gmail.com, chakpriya@gmail.com, sujatha@cse.sastra.edu4, ezhil@cse.sastra.edu, aum@cse.sastra.edu

Abstract

Objective: There is a need for a distributed data storage mechanism to reduce the probability of data loss and cope with node failure as well. The increase in amount of data generated in the wireless sensor network necessitates an in-network data aggregation protocol that creates summaries of data as information is disseminated in the network. **Methods:** An effervescent aggregation technique has been formulated that makes use of the existing data redundancy in the dissemination-based network and combines it with consistency checks to filter false aggregate information. A greedy approach is formulated to keep the copies of the data in the nearby neighbour who has greater capacity. **Findings:** Architecture designed for in-network aggregation helps in dealing with a large amount of data by eliminating redundant information and at the same time distributes the aggregate information in the nearby node since dissemination deals with memory shortage and node failure. **Applications:** The benefits of the system include that the redundant copies are maintained in some other nodes which has higher memory and power this may, in turn, increase the reliability of the data in nodes of sensor networks and the availability is ensured during the communication with sinks.

Keywords: Data Dissemination, Data Replication, Wireless Sensor Networks

1. Introduction

There has been a significant increase in the development of Internet of Things (IOT) over the past few years which has its aim focussed on connecting a large population of devices together wirelessly. IOT has applications in various systems such as Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSN). One prominent technology in the IOT context is Vehicular Ad hoc Networks (VANET)[1] where vehicles are connected to a network, communicate with one another rather than centralizing the communication. In such large scale IOT-based networks, where there is no permanent connectivity, there is a need to cope with the possibility of data loss and node failure. Data loss occurs when the nodes are depleted of memory resources as they are left unattended for a long period of time. The sensed data from the nodes are cleared from their memories only when they are sent to the base station or a sink. This calls for the need for improving the system robustness of the network and prevent the loss of data that stems from node failures. Data is distributed and replicated across the network to maintain the redundant copies in the interconnected nodes to reduce the probability of data loss. Redundant copies are maintained so that if any such node failure occurs, the data can be restored from the copies of it stored in other nodes. There arises a need to constantly check for storage and energy capacity of the nodes carrying redundant copies of other nodes in the network. This can be done by distributing nodes to neighbours that are selected based on their available memory and battery levels. Introducing such a replication-based system, makes the system reliable but at the same time it is necessary for the system to be checked for data consistency. In a large scale system, node failure can cause a disruption in the overall consistency of data the network. Dissemination

*Author for correspondence

schemes provide a way to overcome this problem by establishing redundancy of data in the network. The redundancy that is introduced in the network is used as a means of providing a data consistency check during the process of aggregation. The primary aim of this project is to produce an aggregate that is devoid of any incorrect data values. We propose a mechanism that performs in-network aggregation on a dissemination-based system to ensure the accuracy of data obtained from the network. The aggregation performs a data consistency check before computing an aggregate that has to be sent to the sink. Once the data set is clear of false data, it is aggregated and sent to the sink node. We experiment our approach of performing aggregation on a dissemination-based system and its performance on the Cooja Network Simulator which is run on Contiki OS. Contiki is an open-source operating system for the Internet of Things. It provides an environment where the network can be run on an emulator before it is run on hardware components. Contiki provides an in-built network protocol stack and multi-threading. The OS requires 30 Kilobytes of RAM which also includes a graphical User Interface. The network simulator, Cooja, makes use of emulated nodes that are compiled and executed in the network environment. In this project we propose an aggregation technique that makes use of the existing data redundancy in the dissemination-based network and combines it with consistency checks to filter false information before producing an aggregate. Various dissemination techniques have been proposed over the past years[2]. There are two approaches to distributed storage in an IoT network – data centric storage and fully distributed data storage. Data centric approach is a more application specific approach which is predicted analytically where it outperforms other data dissemination approaches[3]. Fully distributed data storage has all the nodes participating at the same time in the network. They retrieve data periodically and select nodes with maximum storage capacity called the "donor nodes". These "donor nodes" are selected based on a mechanism that involves collecting the memory availability of nodes in the system and deciding which node has the maximum memory available for storage of data[4]. One such important contribution in this area is Data Farm[5]. It is important to analyse the energy efficiency of various data dissemination schemes. There are two categories to dissemination algorithms: reactive and proactive[6]. Energy consumption and dissemination time is evaluated and improved in protocols such as Deluge and Typhoon[7]. A detailed survey has been done on various data dissemination models for specific applications[8] such as Periodic broadcast and On-demand broadcast. One such application is VANET in which there is a need to provide high energy efficient data dissemination approaches to maintain active communication among the network participants[9]. Security schemes analysed and proposed by using a new structure called Event Warning Certificate (EWC). There have been methods proposed to overcome node failures by using data replication strategies. There can be nodes that can be selected and then the data to be distributed among them. Otherwise an alternative approach is to send the data to the neighbouring nodes with the largest memory availability. This is done by controlling the number replicas that exist in the network. In-network aggregation is a used as an algorithm that helps scalability of a system and hence is used in large-scale networks[10]. Aggregation is used to reduce the overhead on data consistency of the network and provide data integrity[11]. It is also done to provide security in the network against insider attacks. A protocol called RDA is used to provide high reliability of packets in the transmission of data by adjusting the degree of redundancy[12]. A statistical analysis is made on redundancy based systems[13]. Clustering is performed on the nodes that is followed by Pathlist filtering and Outlier detection. The first step comprises of filtering the disjoint paths of a network through which there are chances of data from the same node flowing in two paths. This is followed by detecting and eliminating outliers in the data that is retrieved from the disjoint paths. In this paper cryptographic signatures have been attached to the data to provide security in the network[14]. Important factors in aggregation that have been worked on are Data Storage Management and Opportunistic Data Exchange. These areas have been enhanced and analyzed performance wise to provide effective communication among the nodes in the network where data is collected at a particular node[15]. Outliers can occur due to faulty nodes or due to occurrence of an event that is observed by a sensor node[16]. The outliers are classified into erroneous or caused due to an event while at the same time resource consumption of the network is also checked and maintained to a minimum. A Univariate Statistics-based scheme is used to determine the upper and lower bounds of a particular node value and detect outliers based on the bound values[17]. Implementation and analysis is done on the Contiki OS which is an event-driven multitasking OS that is built for networked devices. This OS is applied to individual

processes and allows pre-emptive multithreading[18]. The applications are simulated on Cooja that is an in-built simulator in Contiki. Rime is a layered communication stack that is used by applications that are run on contiki. It provides lightweight data structures and functions compared to the any other protocol that is used in contiki. Rime stack is scalable such that any application that runs on top of the rime stack can make use of additional protocol that are not included in the rime stack.

2. Methodology

The nodes of the IT observation system collect data at regular intervals of time. This data is not periodically sent to a base station as there is no continuous connectivity with the sink. The base station queries for the data from the network, after responding to which the local memories of the nodes are replenished. Periodic data retrieval is necessary as the nodes have limited resources. The process of data retrieval involves aggregating the sensed data and forwarding the same to the base station. To deal with the irrecoverable data loss due to node failure or memory shortage, nodes work in the following manner. The data sensed by a node is stored in a number of other nodes. This involves sending out multiple copies of the sensed data to its neighbors that have available memory. The details of available memory and energy level on each node are broadcasted periodically (Figure 1.) so that all its neighbors can stay updated on its current status. Every node maintains a neighbor table as shown in Figure 2, with an entry for each of its detected neighbors, constantly updating its entries as and when the information broadcasts are received. The replication mechanism used in this architecture is greedy, as it selects the 'best' neighbor from its table of neighbor entries to store the redundant copies of the data created. The sensed data is dropped if there are no neighbors with available memory or energy and if there is no local memory available. At every Tm interval, a node broadcasts its memory and energy availability status to all its 1-hop neighbors. Each status update consists of the following data: 1. Node ID 2. Available memory 3. Energy consumed 4. Sequence number or identifier for the message. A table of neighbors is maintained at every node to store the rime addresses and received information of all the neighbors of a node which is depicted in Figure 2. The neighbor table stored in the local storage of every node consists of a single entry per neighbor along with the most recent information broadcast received. The table can store only a fixed size of entries. If the local memory is completely unavailable, the node stores only the entries of 'best' neighbors and discards the rest. The greedy replication mechanism creates at most N copies of each data unit generated by a node and distributes them further. At time t, node I senses a data. If the node has local memory available, it stores the sensed data in its memory as an entry in the data table, setting the remaining copies to be generated as N-1. If the local memory if full, the data is sent to a neighbor chosen from the table. The 'best' node for storing replicas is chosen as those with the largest available memory and the most recent update received from that node. The energy remaining in the node is also checked with a threshold so as to ensure that the replica is stored in a node that has sufficient energy. Otherwise, if the local memory of node i is full, or multiple copies are to be stored, node i selects, from the memory table, a neighbor node to store a copy of the data unit. In particular, node i select the neighbor node, called donor with the largest available memory space, sufficient energy remaining and the most recent information. The heuristic used for selecting the 'best' node is given. Where G denotes the best neighbor chosen at time t and $t_j < t$ denotes the time at which the memory update was received as $B_j(t_j)$ of node j and $E_j(t_j)$ denotes the energy update received. The energy consumed by node j is checked if it exceeds limit ET, beyond which the lifetime of the node becomes insignificant. If no suitable neighbors are found, the data is stored only on the node that sensed the data or it is dropped. When the neighbor node receives a data unit, it stores the data in its local memory and selects the subsequent 'best' neighbor from its neighbor table to store the remaining N-2 replicas. The heuristic used to select the subsequent neighbors is given as follows. Where $P^{(r-1)}$ is the set of selected neighbors for the previous N-1 copies. Once a subsequent neighbor is selected, the n^{th} node sends a copy and decrements the number of replicas further required by 1. The replication continues until N replicas are created or until an intermediate n^{th} node is not able to find any suitable neighbor. If the second case occurs, the number of replicas created is less than R. The value of N, which is the maximum number of replicas in the network, can be controlled. This is better than broadcasting replicas across the network, which creates innumerable copies of the sensed data. The introduced data redundancy in the network has to be eliminated before sending values to the sink. In a remote network such as this, there are potential threats in the form of faulty nodes that do not fail completely

but circulate incorrect observations through the network. These values, when collected at a node for aggregation, modify the aggregate produced. The elimination of these outliers is done just before aggregating the replicas from the network. Outlier detection mechanisms for univariate data largely depend on the mean and standard deviation of the observations. Values that do not lie inside the upper and lower threshold limits are eliminated as outliers. Increasing the number of replicas significantly shifts the threshold more towards the mean. This increases the fault tolerance of the outlier detecting mechanism. False values that largely vary from majority of the replicas are easily eliminated in this method. The observed values for aggregation are collected at nodes that are elected as cluster heads by the cluster-head election algorithm. There are various methods using which false outlier detection rates can be improved, namely the cumulative sum chart and the EWMA control chart. These methods are not well suited for our data. We assume that the deterministic-stochastic process is stationary. Assuming that, the standard normal deviate of the observed values is , where α is the level of significance. The level of significance specifies the trade-off between false outliers and missed detection rate. The value of SND signifies the percentage of fault that is tolerated by the outlier detecting mechanism. SND values are usually to be following algorithm has been used for eliminating outliers:

UT -> Upper Threshold

LT -> Lower Threshold

If LT ≤ y ≤ UT

Add y to Collected Observations

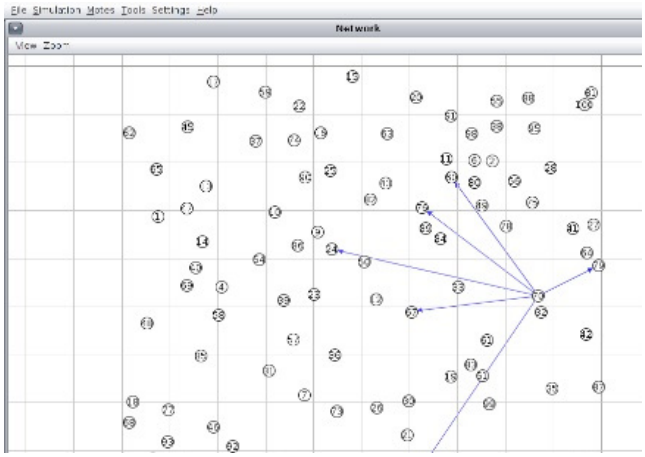

Figure 1. Available memory and energy level distribution to neighbour nodes.

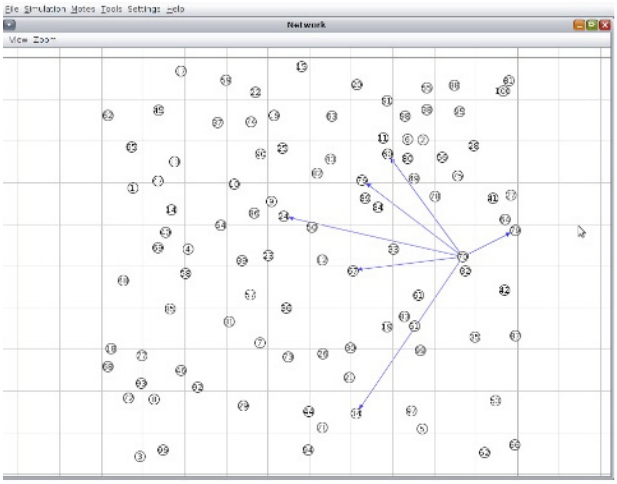

Figure 2. Finding the node that holds the nighbour table.

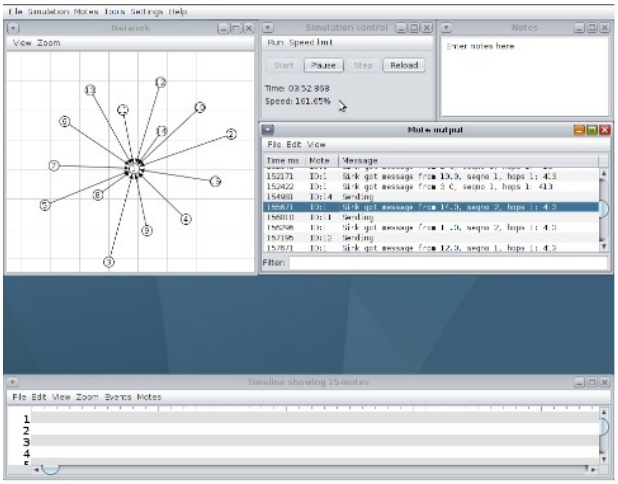

Figure 3. Information received from nodes to sink.

Any observed value that lies out of the bounds are considered outliers and are eliminated. The final aggregate is then computed as an average of the remaining observations. The aggregated value at each node is the sent to the respective cluster heads as shown in Figure 2. The cluster heads in-turn detects outliers, aggregate the values received from nodes that are cluster members, and send it to the sink. Figure 3 depicts the how the sink receives aggregated data from all cluster heads that have collected data from various members of their respective clusters.

3. Conclusion

The proposed architecture is found to increase the reliability of the network at the large scale. The controlled redundancy offered by the dissemination scheme works

hand in hand with the aggregation mechanism that detects outliers. The replication based distributed storage mechanism increases reliability against complete node failure because the outlier detection increases the trustworthiness of the data against inside attacks or false data. The proposed work can further be enhanced to accommodate outlier detection for multivariate sensor data, thereby increasing the overall accuracy of the aggregated values produced by the remote observation system.

4. Acknowledgement

The authors wish to express their sincere thanks to the Department of Science & Technology, New Delhi, India (Project ID: SR/FST/ETI-371/2014). The authors also thank SASTRA University, Thanjavur, India for extending the infrastructural support to carry out this work.

5. References

1. Stefan D, Gürtler J, Kargl F. A resilient in-network aggregation mechanism for VANETs based on dissemination redundancy. Ad Hoc Networks. 2016 Feb; 37(1):101–9.
2. Hongping F, Kangling F. Overview of data dissemination strategy in wireless sensor networks. International Conference on E-Health Networking, Digital Ecosystems and Technologies (EDT); Shenzhen, China. 2010. p. 260–3.
3. Ratnasamy S, Karp B, Shenker S, Estrin D, Govindan R, Yin L, Yu F. Datacentric storage in sensornets with GHT, a geographic hash table. Mobile Networks and Applications. 2003 Aug; 8(4):427–32.
4. Adam D, Grönvall B, Voigt T. Contiki-a lightweight and flexible operating system for tiny networked sensors. 29th Annual IEEE International Conference on Local Computer Networks; 2004. p. 455–62.
5. Chaqfeh M, Lakas A, Jawhar I. A survey on data dissemination in vehicular ad hoc networks. Vehicular Communications. 2014 Oct; 1(4):214–5.
6. Mekki K, Zouinkhi A, Derigent W, Rondeau E, Thomas A, Abdelkrim MN. USEE: A uniform data dissemination and energy efficient protocol for communicating materials. Future Generation Computer Systems. 2016 Mar; 56:651–3.
7. Liang CJM, Musăloiu ER, Terzis A. Typhoon: A reliable data dissemination protocol for wireless sensor network. EWSN'08 Proceedings of the 5th European Conference on Wireless Sensor Networks; 2008. p. 268–85.
8. Palomar E, Fuentes JMD, Tablas AIG, Alcaide A. Hindering false event dissemination in VANETs with proof-of-work mechanisms. Transportation Research Part C: Emerging Technologies. 2012 Aug; 23:85–97.
9. Pandey GK, Singh AP. Energy conservation and efficient data collection in WSN-ME: A Survey. Journal of Science and Technology. 2015 Aug; 8(17):1–11.
10. Scheuermann B, Lochert C, Rybicki J, Mauve M. A fundamental scalability criterion for data aggregation in VANETs. Proceedings of the 15th Annual International Conference on Mobile Computing and Networking; 2009. p. 285–96.
11. Bagaa I, Challal Y, Ouadjaout A, Lasla, NadjibBadache N. Efficient data aggregation with in-network integrity control for WSN. Journal of Parallel and Distributed Computing. 2012 Oct; 72(10):1157–70.
12. Dong LI, Bin TIAN, Shou-shan LUO, Yi-xian YANG. A reliable and security method for data aggregation in WSNs. The Journal of China Universities of Posts and Telecommunications. 2011 Sep; 11(S1):142–46.
13. Pietro G, Ferrari G, Gay V, Leguay J. Data dissemination scheme for distributed storage for IoT observation systems at large scale. Information Fusion. 2015 Mar; 22:16–25.
14. Dietzel S, Schoch E, Konings B, Weber M, Kargl F. Resilient secure aggregation for vehicular networks. Network. 2010; 24(1):26–31.
15. Tseng YC, Wu FJ, Lai WT. Opportunistic data collection for disconnected wireless sensor networks by mobile mules. Ad Hoc Networks. 2013 May; 11(3):1150–64.
16. Fawzy A, Mokhtar HMO, Hegazy O. Outlier's detection and classification in wireless sensor networks. Egyptian Informatics Journal. 2013 Jul; 14(2):157–64.
17. Gil P, Martins H, Januário F. Detection and accommodation of outliers in Wireless Sensor Networks within a multi-agent framework. Applied Soft Computing. 2016 May; 42:204–14.
18. Dunkels A. Poster abstract: Rime a lightweight layered communication stack for sensor networks. Proceedings of EWSN 2007, Poster/Demo session; 2007.