ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

Data Dissemination Protocols in Wireless Sensor Networks - a Survey

Jisha Mary Jose[1], **Jomina John**[2],

Student, Department of Computer Science & Engineering, Rajagiri School of Engineering & Technology, Kochi, India[1]

Assistant Professor, Department of Computer Science & Engineering, Rajagiri School of Engineering & Technology, Kochi, India[2]

Abstract: A wireless sensor network (WSN) is a network made up of a large number of self-organizing nodes distributed in adhoc fashion. They are popularly used for monitoring and control of environment parameters. They are set up in remote locations to form a wireless communication system and it collects data samples for critical domains such as military, industry, environment etc. It is necessary to spread data and code through wireless links after the nodes are deployed in order to adjust parameters of sensors, update the sensor programs or distribute management commands to sensors. This is known as data dissemination or reprogramming in WSNs. Dissemination protocols are vital because almost all WSNs are deployed in hostile environments and thus manual reprogramming of such nodes is not possible. Many data dissemination protocols have been introduced with time and each one of them help in dissemination of program code, configuration parameters, queries, commands, bulk data etc.

Keywords: Dissemination, Reprogramming, Wireless Sensor Networks, Trickle, protocols.

I. INTRODUCTION

A wireless sensor network (WSN) consists of a number of nodes used for monitoring purposes which pass the information collected via the network to a main location primarily a base station. The development of wireless sensor networks was motivated mainly by military applications. But today WSN are used popularly in many applications like remote control and monitoring, environmental monitoring, healthcare management, construction safety, emergency response information, logistics and inventory management etc.

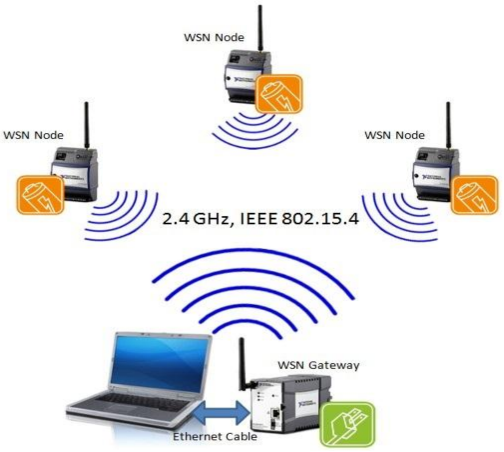


Fig. 1. Example of a wireless sensor network [9]

The WSN consists of nodes ranging from a few to several hundred, where each node is connected to one or several sensors. Each node has several components: microcontroller, radio transceiver, a circuit for interfacing with the sensors and a battery. Sensor networks are usually setup in remote and hostile environments. So size and cost are strict constraints which result in corresponding constraints on resources such as computational speed, energy, memory and communications bandwidth [1]. The topology of WSNs can be either a star network or a multi-hop wireless mesh network.

WSNs must often operate for long periods of time and usually don't get any human administration or intervention. Also evolving analysis, conditions and environment can change application requirements, causing the need to alter the networks behaviour by introducing new code. The remote nature of WSNs requires the propagation of new code over the network as manual updating of such networks is not possible. This process is known as dissemination.

But this poses a lot of challenges in system and network design. One such challenge is effective dissemination of information to all or a group of sensor nodes in the network. This is not an easy task since the number of nodes in a sensor network can be huge and the environment is dynamic in nature, i.e., nodes can die or move, and thus topology can change constantly. Also depending on the application, the information to be disseminated can be originating at a single node, such as the base station, or at multiple nodes, such as sensor nodes themselves. So data dissemination is an area to be studied in deeply.

II. DATA DISSEMINATION PROTOCOLS IN WSN

A. *Drip*

Tolle et al. introduced Sensor Network Management System (SNMS), which is an application-cooperative management system for WSN and Drip is the dissemination protocol that is used in it [2]. Drip is the simplest of all dissemination protocols and is based on Trickle algorithm and establishes an independent trickle for each variable in the data. Every time an application wants to transmit a message, a new version number is generated and used. This will cause the protocol to reset the Trickle timer and thus disseminate the new value.

Drip provides a standard message reception interface in WSN. Each node that wishes to use Drip will register with a specific identifier, which represents a dissemination channel. All messages received on that channel will be delivered directly to the node. Each node is also responsible for caching the data extracted from the recent message received on each channel to which it subscribes, and returning it in response to periodic rebroadcast requests. Drip achieves great efficiency by avoiding redundant transmissions if the same information has already been received by the nodes in the neighbourhood.

B. CodeDrip

It is a data dissemination protocol proposed by Nildo et al. and can be used in Wireless Sensor Networks. This protocol is mainly used for dissemination of small values. Network Coding is a mechanism that combines packets in the network thus increasing the throughput and decreasing number of messages transmitted. CodeDrip uses Network Coding to improve reliability, and speed of dissemination [3]. Rather than simply retransmitting received data packets, sensor nodes try to combine various packets of small data items into one, and re-transmit the combined packet to its neighbours. Thus, packet loss is avoided since lost packets might be recovered through the decoding of others combined packets. By avoiding of frequent retransmissions, the dissemination process finishes faster.

CodeDrip uses the Trickle algorithm for dissemination. It is similar to Drip except for the fact that here messages are sometimes combined and sent. To combine messages, coding protocols use different operators, here XOR operator is used, which is a Galois field of 2. This choice allows Drip to run efficiently on resource constraint nodes. Here the packet format for Drip is modified to accommodate the control fields required by network coding. The decoding side must know which messages were combined to create the given payload. So add to the message header a field to indicate what messages where combined. At the destination we use this field to determine whether a message received is an original message or a combined one.

C. Dip

DIP (Dissemination Protocol) is a data detection and dissemination protocol proposed by Lin et al. [4]. It is a protocol based on the Trickle algorithm. It works in two parts: detecting whether a difference in data in nodes has occurred, and identifying which data item is different. It uses the concept of version number and keys for each data item.

In the steady state all nodes are up to date and have the same versions.DIP uses Trickle to calculate and send hashes that cover all of the version numbers. Nodes that receive hashes which are the same as their own know they have consistent data wrt their neighbours. If a node hears a hash that differs from its own, it knows that a difference exists, but does not know which data item has a newer version. Identifying which data item is different and which node has the newer version requires exchanging of the actual version numbers.

In addition to the version number, DIP also maintains a soft state estimate of whether a given item differs from a neighbours item or not. Whenever a node receives a packet and the hashes are same the estimate is decremented to a minimum of 0. Otherwise if hashes differ the estimate is incremented. This goes on until the estimates converge to zero which means all have the same data.

D. DHV

It is a code consistency maintenance protocol given by Dang et al. [5]. It tries to keep codes on different nodes in a WSN consistent and up to date. Here data items are represented as tuples (key, version).This protocol tries to overcome the disadvantages of previous protocols like DRIP and DIP by reducing the complexity involved in the updating of data in the network. It is based on the observation that if two versions are different, they may only differ in a few least significant bits of their version number rather than in all their bits. Hence, it is not always necessary to transmit and compare the whole version number in the network. Here the version number is given as a bit array. DHV uses bit slicing to quickly determine the out of date code, resulting in fewer bits being transmitted in the network.

DHV includes two important phases: detection and identification. In detection, each node will broadcast a hash of all its versions in a SUMMARY message. Upon receiving this, a node compares it to its hash. If they are not similar, there are one or more code items with a different version number. In identification, the horizontal search and vertical search steps are used to identify which versions differ. During horizontal search, a node broadcasts a checksum of all its versions in a HSUM message. Upon receiving this, a node compares the checksum to its own checksum to identify which bits are different and moves to the next step. In vertical search, the nodes will broadcast a bit slice, starting at the LSB of all versions, which a VBIT message. If the bit indices are matching, and the hashes are different, the node will broadcast a bit slice of index 0 and increase the bit index to find various locations until the hashes become same. After getting a VBIT message, a node compares it to its own VBIT to identify the locations corresponding to the differing tuples. After identifying this, the node broadcasts those (key, version) tuples in a VECTOR message. Upon receiving a VECTOR message, a node compares it to its own (key, version) tuple to decide who has the newer version and whether it should broadcast its data. A node with a newer version will broadcast its data to nodes with an older version.

E. Deluge

Hui et al. gave Deluge which is a reliable data dissemination protocol for propagating large data objects (by dividing those to fixed sized pages) from one source node to other nodes over a multi-hop, wireless sensor network [6]. Dissemination of large data objects i.e. program images poses many issues like large size of programs, toleration of varying node densities and ensuring complete reliability in transfer etc.

Deluge achieves reliability in unpredictable wireless environments and robustness when node densities can vary by factors of a thousand or more. This protocol is based on Trickle algorithm. Here each node follows a set of strictly local rules to achieve data dissemination in the network. A node at regular intervals advertises the most recent version of the data item it has to whichever nodes that can hear its broadcast. Consider B receives an advertisement from an older node A, and then B will respond with the information that it has. From the information received, A determines which portion of the data items need updating and requests them from any neighbour that advertises the availability of the needed data, including B. Nodes receiving these requests then broadcast any requested data. Thus nodes advertise newly received data in order to propagate it further to other nodes.

F. Typhoon

It is a reliable data dissemination protocol used in wireless sensor networks given by Liang et al. [7]. It is mainly used for dissemination of bulky data similar to Deluge. So here also large data objects are divided into fixed sized pages and then again sub-divided into fixed sized packets. Unlike other protocols, Typhoon sends data packets in unicast fashion. This approach allows receivers to acknowledge the receipt of packets and thus quickly recover lost packets if any. While data packets are sent in unicast manner, interested nodes can receive those packets by snooping on the wireless medium. Thus through the combination of unicasting and snooping, this protocol achieves prompt retransmissions and data delivery to all the nodes in a broadcast domain through a single transmission. Typhoon uses Trickle timers for dissemination of meta-data. Here meta data includes object ID, size and version to indicate the existence of a newly created data object. Depending on comparisons of meta data nodes decide to accept or not accept new data objects.

Here all protocol decisions are aiming to minimize the idle listening time of nodes i.e. not transmitting or receiving data packets. So the nodes always try and aim to push data into the network as fast as possible. Also spatial reuse is used, through which nodes in different parts of the network can be transmitting at the same time. Other techniques that can be employed are duty cycling, turning nodes off when not in use and so on.

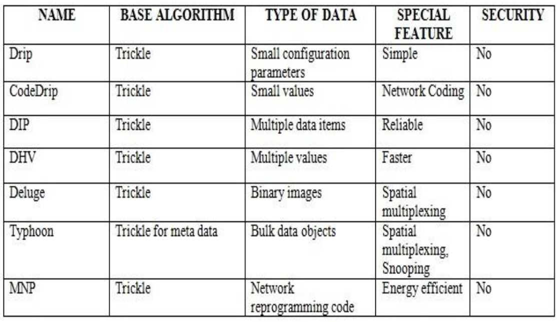
G. MNP

Sandeep et al. proposed a multihop network reprogramming protocol (MNP) [8]. It provides a reliable service to propagate new program code to all sensor nodes in the network. The main aim of this dissemination protocol is to ensure reliable, low memory usage and fast data dissemination. It is based on a sender selection protocol in which source nodes compete with each other based on the number of distinct requests they have received. In each neighbourhood, a source node sends out program codes to multiple receivers. When the receivers get the full program image at their side, they become source nodes, and send the code into their neighbourhood. But here there can be issues of collisions. This is solved by selecting a suitable sensor node based on some parameters maintained by the nodes and some advertisement and download messages exchanged by the nodes. It is like a greedy algorithm. Pipelining can be included in this protocol to enable faster data propagation in the case of larger networks. To do pipelining, programs are divided into segments, each of which contains a fixed number of packets. Once a sensor node receives all the segments of a program, it can reboot with the new program. This continues till all the nodes are hence updated.

III. A COMPARISON

TABLE I

COMPARISON OF DISSEMINATION PROTOCOLS



NAME	BASE ALGORITHM	TYPE OF DATA	SPECIAL FEATURE	SECURITY
Drip	Trickle	Small configuration parameters	Simple	No
CodeDrip	Trickle	Small values	Network Coding	No
DIP	Trickle	Multiple data items	Reliable	No
DHV	Trickle	Multiple values	Faster	No
Deluge	Trickle	Binary images	Spatial multiplexing	No
Typhoon	Trickle for meta data	Bulk data objects	Spatial multiplexing, Snooping	No
MNP	Trickle	Network reprogramming code	Energy efficient	No

IV. CONCLUSION

Wireless Sensor Networks is a wide and open area in networking research, which is increasingly being deployed for monitoring applications. This demands the need for quickly and efficiently disseminating data and code to sensor nodes to reprogram them to suite the current needs of the application. This is achieved by making use of data dissemination protocols. In this paper, a brief survey work is done on the existing various data dissemination protocols for wireless sensor networks and their performances were compared. It can be concluded that none of these methods provide any security to the data that is disseminated. So there is a need of developing secure dissemination protocols.

ACKNOWLEDGMENT

I wish to acknowledge my guide and my institution for all the support they have given me in completing my humble endeavour.

REFERENCES

[1] Rudranath Mitra, Tauseef Khan, "Secure and Reliable Data Transmission in Wireless Sensor Network: A Survey", International Journal Of Computational Engineering Research, ISSN: 2250–3005.

[2] G. Tolle and D. Culler, "Design of an application-cooperative management system for wireless sensor networks", in Proc. EWSN, pp. 121–132, 2005.

[3] Nildo Ribeiro Junior, Marcos A. M. Vieira, Luiz F. M. Vieira, and Omprakash Gnawali,"CodeDrip: Data Dissemination Protocol

with Network Coding for Wireless Sensor Networks", in Proceedings of the 11th European conference on Wireless sensor networks (EWSN 2014), Feb. 2014.

[4] Lin, K., Levis, P.: Data discovery and dissemination with dip. In: Proceedings of the 2008 International Conference on Information Processing in Sensor Networks (IPSN 2008), Washington, DC, USA, IEEE Computer Society (2008) 433–444.

[5] T. Dang, N. Bulusu, W. Feng, and S. Park, "DHV: a code consistency maintenance protocol for multi-hop wireless sensor networks," in Proc. 2009 EWSN, pp. 327–342.

[6] Hui, J.W., Culler, D.: The dynamic behavior of a data dissemination protocol for network programming at scale. In: Proceedings of the 2nd international conference on Embedded networked sensor systems (Sensys 04), New York, NY, USA, ACM (2004) 81–94.

[7] C.-J. M. Liang, R. Musaloiu-Elefteri, and A. Terzis. Typhoon: A reliable data dissemination protocol for wireless sensor networks. In Proceedings of 5th European Conference on Wireless Sensor Networks (EWSN), pages 268–285, 2008.

[8] S. Kulkarni and L. Wang. Mnp: Multihop network reprogramming service for sensor networks. In 25th International Conference on Distributed Computing Systems, June 2005.

[9] Javed Shaikh, DoS Avoidance using gateway cluster, Wireless Sensor Network and Emerging Technologies.

BIOGRAPHIES

JISHA MARY JOSE, Completed BTech in Computer Science and Engineering in 2012 and pursuing MTech in Computer Science and Engineering with specialization in Information Systems, from Mahatma Gandhi University, Kerala at Rajagiri School of Engineering and Technology, Kochi.

JOMINA JOHN, Completed B.Tech in Computer Science and Engineering in 2009 and MTech in Computer Science and Engineering with specialization in Data Security in 2011, from CUSAT, Kerala. Currently working as Asst.Prof at Rajagiri School of Engineering and Technology, Kochi.