October 2011

Analyzing the Performance of Data Dissemination Algorithms to Application Requirements in Wireless Sensor Network

Sukant Kishoro Bisoyi

C. V. Raman College of Engineering, Bhubaneswar, India, sukantabisoyi@yahoo.com

Mohit Ranjan Panda

C. V. Raman College of Engineering, Bhubaneswar, India, mohit1146@gmail.com

Sangeeta Mishra

C. V. Raman College of Engineering, Bhubaneswar, India, smsinu@gmail.com

Follow this and additional works at: <https://www.interscience.in/ijssan>

Part of the [Digital Communications and Networking Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Bisoyi, Sukant Kishoro; Panda, Mohit Ranjan; and Mishra, Sangeeta (2011) "Analyzing the Performance of Data Dissemination Algorithms to Application Requirements in Wireless Sensor Network," *International Journal of Smart Sensor and Adhoc Network*: Vol. 1 : Iss. 2 , Article 22.

DOI: 10.47893/IJSSAN.2011.1038

Available at: <https://www.interscience.in/ijssan/vol1/iss2/22>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Smart Sensor and Adhoc Network by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

Analyzing the Performance of Data Dissemination Algorithms to Application Requirements in Wireless Sensor Network

Sukant Kishoro Bisoyi, Mohit Ranjan Panda & Sangeeta Mishra

C. V. Raman College of Engineering, Bhubaneswar, India

E-mail : sukantabisoyi@yahoo.com, mohit1146@gmail.com , smsinu@gmail.com

Abstract - A Wireless Sensor Network (WSN) is a collection of thousands of tiny sensor nodes having the capability of wireless communication, limited computation and sensing. Many data dissemination protocols have been proposed for multi-hop communication in sensor networks, each evaluated in some scenario. Many protocols are designed to exploit application requirements, then no one protocol can be optimized for all applications. Directed diffusion is a prominent example of data-centric routing in sensor networks, since it is based on application layer data and purely local interactions. However; its functioning relies heavily on expensive operations, like network-wide flooding. The purpose of this paper is to explore different directed diffusion protocols and their performance as the number of sinks and sources increases, the traffic rate and node placement varies, and with and without geographic proximity in node placement. We present the simulation-based performance evaluation using NS2 and tuning of data dissemination in wireless sensor networks. Hope our result would be useful to match dissemination algorithms to application performance requirements.

Keywords - data dissemination, directed diffusion, application performance, NS2

I. INTRODUCTION

Wireless sensor networks consist of specially distributed autonomous sensor, to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, Pressure etc. As sensor network deployment grows, we expect to see a growing range of application techniques. Today, a common class of applications uses a sensor net to communicate data from the net to a single sink, possibly with opportunistic data processing along the way. Data dissemination approaches in sensor networks have adopted application-specific, *data-centric* communications protocols to reduce overhead by avoiding levels of abstraction and to support application involvement in communication. Application involvement in sensor-network communications is an important complement to basic dissemination algorithms.

Fig. 1 : Example of a Sensor Network

II. CLASSIFICATION OF ROUTING PROTOCOLS IN WIRELESS SENSOR NETWORKS

Routing in wireless sensor networks differs from conventional routing in fixed networks in various ways. If there is no infrastructure, wireless links are unreliable, sensor nodes may fail, and routing protocols have to meet strict energy saving requirements [1]. Many routing algorithms were developed for wireless networks in general. All major routing protocols proposed for WSNs may be divided into seven categories as shown in Table 1.

Table 1: Routing protocols for WSN

Category	Representative Protocols
Location-based Protocols	MECN, SMECN, GAF, GEAR, Span, TBF, BVGF, GeRaF
Data-centric Protocols	SPIN, Directed Diffusion, Rumor Routing, COUGAR, ACQUIRE, EAD, Information-Directed Routing, Gradient-Based Routing, Energy-aware Routing, Information-Directed Routing, Quorum-Based Information Dissemination, Home Agent Based Information Dissemination
Hierarchical Protocols	LEACH, PEGASIS, HEED, TEEN, APTEEN
Mobility-based Protocols	SEAD, TTDD, Joint Mobility and Routing, Data MULES, Dynamic Proxy Tree-Base Data Dissemination
Multipath-based Protocols	Sensor-Disjoint Multipath, Braided Multipath, N-to-1 Multipath Discovery
Heterogeneity-based Protocols	IDSQ, CADR, CHR
QoS-based protocols	SAR, SPEED, Energy-aware routing

From different types of routing algorithm shown in above table1 we use Data Centric Protocols which include Directed diffusion routing itself. Data-centric protocols differ from traditional address-centric protocols in the manner that the data is sent from source sensors to the sink. In address-centric protocols, each source sensor that has the appropriate data responds by sending its data to the sink independently of all other sensors. However, in data-centric protocols, when the source sensors send their data to the sink, intermediate sensors can perform some form of aggregation on the data originating from multiple source sensors and send the aggregated data toward the sink. This process can result in energy savings because of less transmission required to send the data from the sources to the sink. In this section, we review some of the data-centric routing protocols for WSNs[2].

III. SUMMARY OF DIFFUSION ALGORITHM

Directed Diffusion Routing

In wireless sensor networks many small sensors can work well together as sensor networks they provide several advantage of traditional centralize sensing, Generally Communication in a sensor network is a data-centric i.e. all communication happens for named data, i.e communicating states are identified by data(not by address) they provide [3].

Directed diffusion is a data-centric communication in regarding that has proved to be an energy efficient routing infrastructure suitable for all appliance. As a data centric protocol applications in sensors label. The data using attribute value pairs. A nodes that demands the data generate a request where an interest is specified according to the attribute-value based scheme defined by the application. Directed diffusion is a data centric dissemination protocol for sensors network, provides high quality path , to found the every over entire network to explore path.

Data centric models are more desirable than address centric models for data gathering WSNs, address centric models assigns sensor nodes. Unique ID (name/labels) based on topology information which emphasize on data holders and suits quires that are issued to individual's sensors. But WSN centre around data rather than nodes holding data and in most cases queries are issued to whole network which applicable in directed diffusion which support data centric mechanism.

Mechanism and Algorithm behavior of Directed Diffusion (DD)

Table 2: Directed diffusion mechanism

Protocol	Sink	Source
Two phase-pull	Interest * (every interest internal)	Exploring data *

	Positive reinforcement Refuse to explore data)	Data node defined by application
One phase-pull	Interest * (every interest internal)	Data
Push	Positive reinforcement (response to explore data)	Data

Fig. 2: (a) interest propagation (b) initial gradient setup (c) Data Delivery

1. Pull diffusion

Pull diffusion is of two types i.e. one phase-pull and two phase-pull diffusion.

One phase-pull diffusion

One-phase pull is a subscriber-based system that avoids one of the two phases of flooding present in two-phase pull. As with two-phase pull, subscribers send interest messages that disseminate through the network, establishing gradients. Unlike twophase pull, when an interest arrives at a source it does not mark its first data message as exploratory, but instead sends data only on the preferred gradient. The preferred gradient is determined by the neighbor who was the first to send the matching interest, thus suggesting the lowest latency path. Thus one-phase pull does not require reinforcement messages, the lowest latency path is implicitly reinforced.

The sink uses a special message called reinforcement message to reinforce one such path to the source. All data message flow through the reinforced data path. This is flow the path between the source and

sinks is established [4]. The interest flow is shown in figure 3.

Fig. 3 : Interest flow in one phase-pull

Two phase pull diffusion

A subscriber, or data *sink*, identifies data by a set of attributes. This information propagates through the network in an *interest* message. Without such information, however, interests must be flooded throughout the network to find any data sources. As they are distributed, nodes establish *gradients*, state indicating the next-hop direction of other nodes interested in the data. The first data message sent from the source is marked as *exploratory* and is sent to all neighbors that have matching gradients. As with interest messages, this transfer could be limited using additional information or application involvement, but by default it is sent to all nodes. When exploratory data reaches the sink, the sink *reinforces* its preferred neighbor, establishing a reinforced gradient towards the sink. The reinforced neighbor reinforces its neighbor in turn, all the way back to the data source or sources, resulting in a chain of reinforced gradients from all sources to all sinks. Subsequent data messages are not marked exploratory, and are sent only on reinforced gradients rather than to all neighbors.

Fig. 4: Interest flow in two phase-pull

Push diffusion

Two-phase pull works well for applications where a small number of sinks collects data from the sensor net, for example, a user querying a network for detections of

some tracked object. Another class of applications involves sensor-to-sensor communication within the sensor network. A simple example of this class of application might have sensors operating at a low duty cycle most of the time, but when one sensor detects something it triggers nearby sensors to become more active and vigilant. Two-phase pull diffusion behaves poorly for this application, because all sensors actively send interests and maintain gradients to all other sensors even though nothing is detected. Push diffusion was designed for this application. Although the API is the same as two-phase diffusion (except for a flag to indicate “push”), in the implementation, the roles of the source and sink are reversed. Sinks become passive, with interest information kept local to the node subscribing to data. Sources become active; exploratory data is sent throughout the network without interest created gradients. As with two-phase pull, when exploratory data arrives at a sink a reinforcement message is generated and it recursively passes back to the source creating a reinforced gradient, and non-exploratory data follows only these reinforced gradients.

Geographic and Energy Aware Routing (GEAR)

In addition to these basic approaches, the physical nature of most sensor networks allows geographic information to be used to constrain search. GEAR (Geographic and Energy-Aware Routing) extends diffusion when node locations and geographic queries are present. Although originally designed for pull diffusion, it has also been applied to push. The physical nature of a sensor network’s deployment makes geographically scoped queries natural. If nodes know their locations, then geographic queries can influence data dissemination, limiting the need for flooding to the relevant region. GEAR (Geographic and Energy-Aware Routing) extends diffusion when node locations and geographic queries are present [5]. GEAR is an extension to an existing diffusion algorithm [6] that replaces network-wide communication with geographically constrained communication. When added to two-phase pull diffusion, GEAR’s subscribers actively send interests into the network. However, queries expressing interest in a region are sent *towards* that region using greedy geographic routing (with support for routing around holes); flooding occurs only when interests reach the region rather than sent throughout the whole network. Exploratory data is sent only on gradients set up by interests, so the limited dissemination of interests also reduces the cost of exploratory data. GEAR provides a first example of application-specific diffusion. It optimizes diffusion for applications and networks that have geographically scoped queries. GEAR-extended versions of push and one-phase pull are also available and described below

and the code uses georouting filter application shown below.

IV. SIMULATION AND RESULTS

We have conducted performance evaluation using the ns-2 simulator [7]. The simulation was set up with the view of investigating the performance of Directed Diffusion protocol. Our investigation was on the effects of the diffusion packet generated and filter used, number of nodes used. Every other parameter was fairly kept constant for all the simulated scenarios. Our simulation was deployed by creating static nodes for a typical campus sized network or with topography of 670 X 670 meters. We used a constant bit rate source because it is widely used in routing protocol comparison studies. Each of the scenarios was run for 500s. The evaluation of the mentioned protocols are done on Pull (One phase-pull and two phase-pull), and push, two phase-push with Gear and Two phasePull with Gear routing. From the figure 5, it can be observed that two-phase push protocol performs better than one phase pull and twophase-pull because it generates less number of diffusion packets as compared to other. When the number of sink node increases, push protocol performs better than onephase-pull and twophase-pull. But packet generation increases with number of sink increase for twophase-push which is shown in figure 6. Figure 7 shows that twophase-pull drops more packets as compared to twophase-push under varying sink and with one source.

When number of source increased to 1, 3, 5, 7, 9 keeping sink constant, onephasepull performs better as shown in figure 8. From figure 9, it can be concluded that twophasepull drops more packet than twophasepush under varying source. But onephase-pull protocol have less mac overhead as compared to twophasepull and push. Twophasepull protocol have more MAC overhead as compared to onephasepull and push under one-source and one-sink which is shown in figure 10. But when the source is varying with keeping sink constant, twophasepull have more MAC overhead as compared to onephase-pull and push which is shown in figure 11, but all other routing protocol generates more MAC overhead with increases of source. When sink is varying keeping source constant, twophase-pull have more MAC overhead than onephasepull and twophasepush as shown in figure 12.

From the figure 13 it can be stated that number of diffusion message is less in gear-push than push, but message diffusion decreases suddenly in each 100 second (for 1source-1sink). But in push protocol it increases and decreases randomly. Similarly figure 14 shows the number of diffusion message generated in gear-push is less than push by increasing sink as 1, 3

and keeping source constant as one. Similarly Figure 15 shows the number of diffusion message increases slowly in gear push but generates less diffusion packet as compared to push by increasing source to 1, 2, 3, 4 keeping sink constant as one. In push protocol number of diffusion increases drastically which creates more transmission problem in network. But figure 16 gear-push has more MAC overhead than push in 1source-1sink. Figure 17 shows increasing the sink keeping source constant the MAC overhead is more in gear-push as compared to push, and when increasing source keeping sink constant MAC overhead is more in gear-push as compared to push which is shown in figure 18.

Figure 19 shows number of diffusion generated from 100 to 500 seconds for 1 sender - 1 receiver slowly decrease and increase in twophase-pull, where as in twophase-pull with gear(i.e. gear-pull) it decrease and increase little randomly but is very less number of diffusion generated than twophase-pull, which specify gear-pull is better than twophase-pull. When the number of sink increases pull-gear performs better than twophase-pull since packet in pull-gear is less than twophase-pull shown in figure 20. Similarly figure 21 shows that twophasepull-gear has less diffusion packets as compared to twophase-pull under varying source and constant sink. From this it concludes that gear-pull is better than twophase-pull. Figure 22, 23, 24 show the MAC overhead is more for gear-pull than twophase-pull respectively in all case. Figure 25 show the effect of diffusion generated which is increasing of each 100 seconds with one sender and one receiver between twophase-push, pull-gear and push gear. When number of source increased to 1, 2, 3, 4 keeping sink constant push-gear protocol has less number of diffusion packet than gear-pull as shown in figure 27. But push-gear has more MAC-overhead in comparison to gear-pull when we vary number of source with sink constant as 1.

Fig. 5 : One phase-pull vs. two phase-pull Vs. Push (message Vs time)

Fig. 6 : Onephase-pull Vs. twophase-pull Vs. Push message Vs sink increasing

Fig. 10 : Onephase-pull Vs. twophase-pull Vs. Push (MAC overhead Vs source)

Fig. 7 : Twophase Pull Vs. Twophase Push (message dropped Vs sink increasing)

Fig. 11 : Onephase-pull Vs. twophase-pull Vs. Push (MAC overhead Vs source increasing)

Fig. 8 : One phase pull Vs. two phase-pull vs. Push (message Vs source increasing)

Fig. 12 : One phase-pull Vs. two phase-pull Vs. Push (MAC overhead Vs sink increasing)

Fig. 9 : Twophase Pull Vs. Twophase Push (message dropped Vs source increasing)

Fig. 13 : Push vs. Push-Gear (message Vs time)

Fig. 14 : Push Vs. Push + gear (message Vs sink increasing)

Fig. 18: Push Vs. Push + gear (MAC overhead Vs source increasing)

Fig. 15 : Push Vs. Push + gear (message Vs source increasing)

Fig. 19 : Twophase-pull Vs. Pull + gear (message Vs time)

Fig. 16 : Push Vs. Push + gear (MAC overhead Vs source)

Fig. 20: Twophase-pull Vs. Pull + gear (message Vs sink increasing)

Fig. 17: Push Vs. Push + gear (MAC overhead Vs sink increasing)

Fig. 21: Twophase-pull Vs. Pull + gear (message Vs source increasing)

Fig. 22 : Two phase-pull Vs. Pull + gear (MAC-overhead Vs source)

Fig. 26 : Push +gear Vs. Pull +gear (message Vs sink increasing)

Fig. 23 : Twophase-pull Vs. Pull-gear (MAC-overhead Vs sink increasing)

Fig. 27 : Push +gear Vs. Pull+gear (message Vs source increasing)

Fig. 24 : Twophase-pull Vs. Pull-gear (MAC-overhead Vs source increasing)

Fig. 28 : Push +gear Vs. Pull +gear (MAC-overhead Vs source)

Fig. 25: Push Vs. Pull+gear Vs. Push +gear (message Vs time)

Fig. 29: Push +gear Vs. Pull +gear (MAC-overhead Vs sink increasing)

Fig. 30 : Push +gear Vs. Pull +gear (MAC-overhead Vs source increasing)

V. CONCLUSIONS

There are different routing protocols proposed for routing in WSN, these protocols are application specific and therefore a careful selection of the underlying routing protocol for WSN is a key element to measure the performance of a WSN as a whole. In this an attempt has been made to compare different Directed Diffusion routing protocols and a detailed comparison of these routing protocols has been carried out with various parameters using NS2 simulator. From the simulation result it can be concluded that push protocol is better than one-phase and two phase-pull protocol under all circumstances. When the gear protocol is attached with two phase-pull (i.e. gear-pull), it performs better than twophase-pull. When gear is attached with push(i.e. gear-push) gear-push performs better than push protocol. But in comparison to gear-pull, gear-push is better in all scenarios. Hopefully, the result of this study can be used as reference for the future work.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Communications Magazine, Aug. 2002.
- [2] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. ACM/IEEE Transactions on Networking, 11(1):2–16, February 2003.
- [3] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, pages 56–67, Boston, MA, USA, August 2000. ACM.
- [4] Philip Sitton and Kenan Casey, "Directed Diffusion Report", COMP6360
- [5] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. INFOCOM, 2000 Mike Woo, Suresh Singh, and C. S. Raghavendra. Y.Yu, D. Estrin, and R. Govindan. "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks" Technical Report TR-01-0023, UCLA Computer Science Department Technical Report, May 2001.
- [6] C.Intanagonwiwat, R.Govindan, D.Estrin, J.Heidemann, F.Silva,"Directed Diffusion for wireless sensor networking" Networking Volume 11, Issue 1, Feb. 2003 Page(s):2-16 Digital Object Identifier 10.1109/TNET.2002.808417
- [7] [9] NS-2, The ns Manual (formally known as NS Documentation) available at <http://www.isi.edu/nsnam/ns/doc>.