

An adaptive hybrid schema for data-centric storage in wireless sensor networks

International Journal of Distributed
Sensor Networks
2016, Vol. 12(10)
© The Author(s) 2016
DOI: 10.1177/1550147716666665
ijdsn.sagepub.com
 SAGE

Pooya Hejazi

Abstract

Storage, query cost, and energy efficiency are among the most important data dissemination issues in wireless sensor networks. Data-centric storage approach can be viewed as energy efficient and low cost solution to these issues, which itself suffers from hotspot storage problem. Some provided solutions tried to handle this mentioned problem and also brought load balancing and scalability to the network. In this article, we present a method called adaptive resilient data-centric storage which reduces the storage cost and also improves the load balancing and fault tolerance in the network by making data-centric storage adaptive based on the frequency of sensed events. Then, we propose a mathematical model to determine the threshold for the frequency of occurrence of sensed events. Finally, we developed a discrete-event-based simulator for two-dimensional graphical view of the network and collected the results. Simulations and analytical results show the superiority of the proposed method over the compared approaches with the miscellaneous frequency of events.

Keywords

Data-centric storage, adaptive resilient data-centric storage, wireless sensor networks, load balancing, fault tolerance, energy saving

Date received: 24 February 2016; accepted: 3 August 2016

Academic Editor: Shigeng Zhang

Introduction

Wireless sensor networks (WSNs) are interacting sensor nodes with energy and connectivity constraints. They are deployed in order to gather raw information from pervasive environments or industrial applications. In all the applications related to WSN, the most challenging issue is how to store sensed data through the network. The effective use of the vast amount of data gathered by large-scale sensor networks will require scalable, self-organizing, and energy-efficient data dissemination algorithms.¹ Two main approaches were proposed in literature: distributed data storage (DDS) and data-centric storage (DCS).² In DDS, each node sensing the data is responsible for data storage. Although the cost of data storage is low in this model, since the station has to query all the nodes within a

network using flooding method, the cost of query process increases dramatically as the network size increases. On the other hand, DCS methods use geographic hash tables (GHTs)³ to map an event-type to a specific geographic location. As a result, whenever an event is sensed by a sensor, data will be sent toward a geographic location using geometric routing algorithm like Greedy Perimeter Stateless Routing (GPSR).⁴

Department of Computer Engineering and Information Technology, Eqbal Lahoori Institute of Higher Education, Mashhad, Iran

Corresponding author:

Pooya Hejazi, Department of Computer Engineering and Information Technology, Eqbal Lahoori Institute of Higher Education, Sarafrazan 9th Street, Piroozzi Boulevard, Mashhad 9197113113, Iran.
Email: hejazi@eqbal.ac.ir



Creative Commons CC-BY: This article is distributed under the terms of the Creative Commons Attribution 3.0 License

(<http://www.creativecommons.org/licenses/by/3.0/>) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<http://www.uk.sagepub.com/aboutus/openaccess.htm>).

Then, data will be stored in the nearest node to that hash location. Queries are also sent toward the storage location instead of flooding the network. Therefore, the performance of DCS is much better than DDS in a similar situation.

Some restrictions of WSN were not considered in the basic version of DCS. Since sensor nodes are more prone to failure, data loss will occur in WSN. Therefore, data availability becomes very low. Use of data replication mechanisms will help to avoid such situations.⁵⁻⁷ Approaches such as structured-replication data-centric storage (SR-DCS)¹ and resilient data-centric storage (R-DCS)⁸ have enhanced the basic DCS in order to overcome the mentioned limitation. These methods bring information redundancy by replicating the data from storage node to the nodes called replicas. In R-DCS, to store the sensed data in its specific area, network is divided to a collection of zones. This reduces the load on replicas by distributing the data storage load across the network, consequently increases the average lifetime of the WSN. Also, the monitor nodes are involved in the storage and query process for directing the data, queries and responses. Using monitors for data query and data storage management causes traffic reduction over the replicas and their neighbors. Although R-DCS brings reliability and load balancing to WSN storage and query process, it has its own issues: First, since monitors participate in storage and query process, the network traffic will increase, especially when the frequency of occurrence of events is high. Second, monitors work as a gateway in storage and query process. Therefore, their neighbors tolerate high amount of traffics and lose their energy fast. This brings more destinations to reach monitors and finally unreachability of them.

In order to overcome the R-DCS issues, a novel hybrid method called adaptive resilient data-centric storage (AR-DCS) is proposed in this article, which is an adaptive version of R-DCS.⁸ In AR-DCS, the number of zones is determined adaptively based on the frequency of occurrence of sensed events. Whenever the frequency of events rises over the predetermined threshold in a specific zone, division happens and subzones are created. On the other hand, as the frequency of events returns to its normal state, zones are aggregated and create their widespread parent zone. This makes a WSN adaptive based on the frequency of sensed events. The benefit is threefold: load balancing, reduction in the storage cost, and avoidance of the unreachability of the monitors with a negligible effect on the query cost.

The rest of the article is organized as follows: section "Related work" covers related works previously carried out in different areas contributing to design and implementation of DCS solutions. Section "AR-DCS" proposes our method and compares the suggested method with both basic DCS and R-DCS and section

"Simulation" evaluates the results of simulation. Section "Conclusion" provides the conclusion of the article.

Related works

The DCS basic method introduced in Ratnasamy et al.¹ uses a hash function to map an event-type to a specific location. Therefore, DCS avoids flooding mechanisms used in DDS and brings energy saving and scalability to WSN. However, some issues such as node failure and mobility remain unsolved. Hence, various methods were proposed to solve the mentioned issues and bring data recovery and fault tolerance to the basic DDS method. DCS methods that work on the mentioned issues are categorized into two groups: load-balancing-based DCS and reliable-based DCS.

The load-balancing DCS methods offer load balancing over data dissemination on storage and query process which brings energy saving and consequently better network lifetime. Grid-based dynamic load-balancing (DLB)⁹ resolves the node failure problem and provides load balancing using multi-threshold levels in each grid. DLB divides the whole network into a grid of cells, and each cell consists of the nodes which are within one hop distance. Sink nodes use the hash function to map an event-type to a grid ID. After detecting an event, the node sends a put packet to the grid ID and uses GPSR to forward the mentioned packet to the closest node to that grid point.⁷ The main drawback of this method is less data availability due to node failure. Query hotspot is one of the problems that happen when the stored data in few specific nodes are targeted by numerous queries. To resolve this problem, several methods have been proposed such as zone partitioning and zone partial replication¹⁰ based on Multi-dimensional Range Query which locally detect and dissolves query hotspots. Also, in time-parameterized DCS,¹¹ the time at which the sensed data is generated determines the node which is used to store the data. Since data zones are altered periodically, both storage and query process are scattered across the network. Structured replication DCS³ is a method to support load balancing and scalability. In this method, events are assigned to a hierarchical depth based on the decomposition of the space key. This mechanism can be considered as a median method between DDS in which there is no storage cost but there is high query cost, and the basic DCS in which the storage and query cost are at an average level. In SR-DCS, if a clustered failure occurs, the network will not be able to retrieve the data of the specific event-type. Also the root storage node is a single point of failure. Adaptive structured replication DCS (ASR-DCS)¹² is another novel mechanism to bring load balancing into the WSN and reduce energy consumption

on the bottlenecks, especially in the time of high frequency of sensed events. In addition to solve the problem of root node bottleneck and high query cost, it provides SR-DCS benefits such as load balancing on storage nodes and scalability to the network. Two thresholds are defined for this method. If the event frequency exceeds the first threshold, storage node will create the first level of hierarchy and event data are sent to first-level replicas. If event frequency also exceeds the second threshold, first-level replicas will act like the root node and will create the second level of hierarchy. Upon the return of the network to its normal state and the decline in the event frequency, replica nodes of each level will be merged into their immediate upper-level root node. This makes SR-DCS adaptive based on the various frequency of the sensed events.

The reliable DCS methods work on data recovery due to node failure by replicating the stored data over multiple storage nodes. Aly et al.¹³ introduced K-D trees-based DCS schema (KDDCS) to address the node failure problem in DIM.¹⁴ In this method, the whole network is divided into equal-sized data regions with equal number of sensors, leading to a balanced K-D tree. Although KDDCS increases the quality of data and data persistence, it causes more delay in query process. A dynamic DCS mechanism is suggested in Lai et al.¹⁵ This method is able to reduce the storage cost by mapping the sensed data to the storage points. Despite the increase in the overhead on data storage process, network robustness is improved by enhancing GPSR routing protocol to store several copies of data. A new dynamic DCS mechanism¹⁶ can be a solution which changes storage node regularly over a fixed period of time called epoch. This makes it possible having temporal queries to the previous storage nodes and providing access to the historical data, which results in more data availability. But storing all events during an epoch in the selected storage nodes brings node failure and high storage cost problem to the network. Ahmed and Gregory¹⁷ proposed a method called data-centric storage with metric-based similarity searching (DCSMSS) which is particularly useful where users seek data within a WSN that is either a match or close to a match. In this method, the sensed data which is related to an event-type is sent toward a specific node called sector head (SH). SH nodes aggregate the data and then transmit the result toward the storage nodes. This enhances reliability and provides efficient similarity searching within a distributed network. Adaptive DCS¹⁸ offers a hybrid method to solve the node failure problem, which dynamically determines the network conditions such as query rate and event production for each event-type. Decision-making is done in the sink where a suitable storage method is determined, based on the rate of sensed data and query for each event-type. This method provides more energy efficiency and increment in

network lifetime, with the cost of less data availability. Gonizzi et al.² presented a memory-based replication method based on the remaining local memory capacity of nodes. In this method, each node broadcasts the information about memory availability to its neighbor. On the other hand, each node receives an update from other node indicating memory availability, stores the data in its in-memory table. Whenever lack of storage happens, the most memory-available neighbor is chosen as a replica. Thus, it brings reliability by the cost of local broadcast between neighbors. Also, it may be a situation where all the neighbors of the storage node suffer from lack of memory. Then, there is no neighbor to be chosen as a replica so that the storage node flushes its memory and data loss will happen. Ghose et al.⁸ presented the resilient-DCS (R-DCS) which proposes two classes as “control nodes” and “data nodes” in the strategic location through WSN. This method reduces the energy consumption of a single node and brings scalability to the network. In R-DCS, the sensor network space divides into “Z” zones. Using this method, data related to an event-type is stored on data nodes called replicas. On the other hand, control nodes called monitors are responsible for data storage and query management. Having increment in storage nodes related to an event-type, and preserving control and summary data in multiple nodes, R-DCS leads to reduction in average storage and query cost and makes network scalable. Another method named as efficient resilient schema DCS¹⁹ uses multiple geographical locations to store an event-type in mobile sensor networks. Locations will be selected from the different distributed points across the network. In this method, hash function works based on the event-type, data importance, and initial collection zone. In another research, Joung et al.²⁰ offered an adaptive and cost-optimal mechanism called Tug-of-War (ToW). In this approach, rather than using just a single home location, the mechanism can dynamically adjust the number of home locations based on the storage and query rate to minimize the total communication cost.

AR-DCS

Data storage is one of the most challenging issues in WSN because it is directly related to the sensor lifetime and network topological structure. As a result, various approaches try to optimize the storage cost. DDS and DCS are the main approaches in the literature dealing with storage process optimization. In DDS, sensed data are stored locally so that queries are flooded through the network to gain specific data. This brings high traffic of query. On the other hand, DCS uses GHT³ to map an event-type to a specific location. Nodes sensing the data send storage traffic toward the hash location

and store the data in the nearest node to the mentioned location using geometric routing algorithm like GPSR.⁴ Queries are also forwarded to the hash location of specific event-type instead of being flooded through the network.

As DCS introduced in Ratnasamy et al.,¹ some serious challenging issues such as node failure and mobility were not considered. Therefore, improvements in the field of DCS were continued until methods such as SR-DCS¹ and R-DCS⁸ have been proposed to handle data recovery from node failure and mobility. These methods bring information redundancy by replicating the data from storage node to the nodes called replicas. This makes WSN fault tolerant against the mentioned situations. In SR-DCS, storage is done in a hierarchy of replicas. Each event-type is mapped to a specific location and a specific depth of hierarchy. In R-DCS, network is divided into partitions called zones. In each zone, in addition to replicas, another node is used to manage storage and query traffic called monitor. Both R-DCS and SR-DCS are proposed to cover data loss due to node failure but they have their own issues. In SR-DCS, due to static structure of hierarchy of replicas, load on replica's neighbors increases dramatically, especially when the frequency of occurrence of event rises. This will increase the discharged nodes near the replica. Finally, the replicas become unreachable. Also, it suffers from single point of failure and the traffic of query process is relatively high. Therefore, in Hejazi and Amin,¹² a method called ASR-DCS is proposed to make SR-DCS adaptive, based on the frequency of occurrence of events. As the frequency rises, the depth of hierarchy increases so that more replicas are used in the storage process. Whenever it returns to its normal state, the depth of hierarchy decreases in order to reduce the query cost. On the other hand, R-DCS has two main challenging issues: first, the monitors also participate in storage and query process, which cause more traffic of storage and query of data. Second, as the monitor is a gateway node and is responsible for data query and storage management, its neighbors discharge fast and the monitor becomes a neighborless node. Therefore, storage and query traffic cannot reach the monitor and the storage and query process will be disturbed. To solve R-DCS issues while keeping its capabilities, we proposed a novel hybrid approach called AR-DCS. In AR-DCS, the number of zones within a WSN network is determined adaptively based on the frequency of occurrence of sensed events. Whenever this frequency passes a predefined threshold, each zone divides into four subzones which have their own replicas and a monitor. As a result, the traffic of storage does concentrate on monitors of subzones instead of one root monitor. This brings traffic reduction over the neighbors of the monitor. Also, as zones are divided, the distance between sensing nodes, the monitor, and

the replica reduces and this will cause traffic decrease in the storage process. However, when the frequency returns to its normal state, subzones are merged to the parent zone with the main monitor and the main replica for the reduction of the query traffic. The simulation results for basic DCS, R-DCS, and AR-DCS demonstrate that AR-DCS keeps R-DCS capabilities in reliability and data-loss recovery and also it performs better on the storage cost. AR-DCS relatively increases the query cost, but this increment is less than the rate of reduction of the storage cost. AR-DCS also has some points to consider:

1. Rapid swing in the frequency of occurrence of events: These swings will increase the rate of division and aggregation of zones. As a result, stored data are always in transfer between root zone and its subzones replicas. This will bring more traffic over the network and neighbors of replicas. To avoid this, rapid swings are handled in our simulation. Sampling of frequency is done on every storage packet arrived at monitor but evaluation is done after every five received storage packets.
2. Queries sent during the divisions and aggregations: The divisions and aggregations of zones are done based on the frequency of storage packet received at monitor and it is independent of traffic of query. Therefore, there may be a situation where queries are sent during division or aggregation of zones toward the replicas. In this situation, queries are received by a node which is not a replica anymore. If this happens at division time, queries will be received at the root replica and it can answer it directly, which it is not an issue. However, if the situation mentioned above happens at aggregation time, there will be no data to answer. Therefore, replicas in subzones keep their data valid for the specific amount of time to answer queries during aggregation. This traffic is also evaluated in the query cost of our simulation.
3. Threshold definition for zone division and aggregation: The main advantages of AR-DCS over ASR-DCS are single point of failure avoidance due to monitor node usage and dynamic determination of threshold for division and aggregation. Assume that r denotes the sensing range of each node and d is used for the minimum distance between nodes. In order to calculate the threshold adaptively based on the frequency of the occurrence of sensed events, the average number of sensed data per object must be counted. So, to determine the mentioned average, the minimum number of sensors (s_{min}) and maximum number of sensors (s_{max})

which sense the specific object should be determined. It is obvious that the minimum number is equal to zero. To calculate the maximum number, we assume a circle covers the radius of $r + (d/2)$ with the target at its center position. The number of the circles with the $d/2$ radius which can be packed within the mentioned circles will determine the maximum number of sensors sensing a specific object. The center of each packed circles is the place for deploying a sensor node. The $d/2$ radius brings the minimum distance of d between nodes. Also, the maximum distance with the object and the sensor will be r . Table 1 reveals the maximum values for different d s between nodes. The data are gathered by an algorithm used for packing circles.²¹ It should be mentioned that the sensing range is set to 20 m. Hence, the maximum radius range of sensors is assumed to be 50 m based on the averaged data proposed by TELOSB,²² and the maximum value of d is 50. Also, since nodes are distributed randomly throughout the network, the average number of node sensing an object (s_{avg}) will be determined based on the uniform distribution as follows

$$s_{avg} = \frac{s_{max} + s_{min}}{2} \quad (1)$$

In order to compute the density of nodes (ρ), one can use the following equation

$$\rho = \frac{n_{sim}}{n_{max}} \quad (2)$$

where n_{sim} is the number of nodes used for simulation and n_{max} is the maximum number of the nodes that can be deployed in the simulated network. The latter parameter is calculated by equation (3) where w is the network width and h is the network height

$$n_{max} = \frac{h \times w}{\pi(d/2)^2} \quad (3)$$

Finally, to calculate the threshold, equation (4) is used where t denotes the number of targets and z is used for the number of zones within the network

$$\text{Threshold} = \frac{s_{avg} \times \rho \times t}{z} \quad (4)$$

4. AR-DCS is complicated: In the aspect of process cost, AR-DCS is more complicated than basic DCS and R-DCS. However, there is a rule that declares the following: The cost of process of 3000 instructions is equal to send 1 bit over 100 m.³

Table 1. Maximum number of sensors sensing an object, excerpted from Graham et al.²¹

r	d	s_{max}
20	10	19
20	20	7
20	30	3
20	40	2
20	50	1

Therefore, converting communication cost to process cost will bring growth in the node lifespan and the network lifetime.

Table 2 characterizes the mentioned methods based on the dominant parameters in WSN scenarios.

AR-DCS is a hybrid method based on R-DCS⁸ and ASR-DCS¹² bringing load balancing and fault tolerance over the nodes involved in storage and query process. This will cause balanced energy consumption over these mentioned nodes and brings better performance in WSN lifetime. As mentioned above, AR-DCS is a suitable method for decreasing the traffic load over neighbors of replicas so that it has positive impact on network lifetime. Also, AR-DCS has R-DCS capabilities in storage and query process using monitors and redundant replicas. Therefore, recovery of lost data and reliability are better preformed than basic DCS. It should be mentioned that although AR-DCS has better performance than R-DCS and basic DCS in storage process, due to zone division, traffic of query process is partly raised. Finally, the total amount of traffic in AR-DCS is near the boundary of basic DCS.

Simulation

Energy consumption is the most critical issue in WSN. Wireless communications between sensor nodes need energy. Therefore, data dissemination approaches in WSN try to optimize communications in order to improve the node battery lifetime. However, mapping communication and energy consumption are complicated tasks which are related to parameters such as hardware and transmission pattern but experiences show that parameters such as “Total number of packets sent for storage: $TotalQ$,” “Total number of packets sent for query: Q ,” and “Total number of packets sent for query response: Dq ” are the dominant parameters for energy consumption measurement.¹ In order to improve our evaluation over AR-DCS, R-DCS, and DCS, a parameter called “Total number of packets: $OverallQ$ ” is added too. Communications with outside of the network is done by means of single or multiple workstation(s) querying data within WSN. We assume that nodes are fixed and stable. In real world, there

Table 2. Performance characterization of methods.

	DCS	SR-DCS	ASR-DCS	R-DCS	AR-DCS
Energy saving	✓	✓	✓	✓	✓
Load balancing	—	—	✓	—	✓
Fault tolerance	—	✓	✓	✓	✓
Resolving single point of failure	—	—	—	✓	✓
Complexity of method	—	—	✓	—	✓
Performance in storage process	—	—	✓	—	✓
Performance in query process	✓	—	—	—	—
Security	—	—	—	—	—

DCS: data-centric storage; SR-DCS: structured-replication data-centric storage; ASR-DCS: adaptive structured-replication data-centric storage; R-DCS: resilient data-centric storage; AR-DCS: adaptive resilient data-centric storage.

Table 3. Input parameters for the network.

Network parameter	Value
Number of network nodes	150 nodes
Node initial energy	1000 J
Wireless communication radius	50 m
Send cost per meter	0.2 J
Receive cost	0.001 J
Sensing cost	0.01 J
Network size	1024 × 512 m
Propagation delay per meter	0.001 s

may be a situation that a faulty node disturbs traffic of data. Also, nodes can be mobile but this is not a usual circumstance because the node movement interval is much longer than storage or query process time. Quality of radio channel does not change immediately during time so that neighbors of the nodes remain constant. We assume node capabilities are equal although in real world it may be some situations that nodes with the different capabilities such as battery and storage capacity are added to the network. Table 3 reveals the input parameters for the network and Table 4 shows the input parameters for the simulation.

As shown in Tables 3 and 4, input parameters are categorized into two major groups: The first group contains parameters that describe network characteristics. The second one contains simulation-run parameters. Each parameter has a default value, which the simulation is based on. Default values are determined according to Ratnasamy et al.¹ and Ghose et al.⁸

For evaluation and analysis of the mentioned methods, we developed a discrete-event system simulator based on event-scheduling approach explained in Jerry²³ in which events and their effects on the state of the system during the predetermined amount of time are used for system conceptualization and modeling. The superiority of the developed simulator against the one introduced in Ratnasamy et al.¹ can be seen in each

Table 4. Input parameters for the simulation.

Simulation parameter	Value
Sensing time interval for sensors	5 s
Sensing delay	1 s
Query interval	20 s
Graph update interval for GPSR	By node death
Simulation run time	300 s
Number of moving object	1–40
Data dissemination algorithms	DCS R-DCS AR-DCS
Propagation delay per meter	0.001 s

DCS: data-centric storage; R-DCS: resilient data-centric storage; AR-DCS: adaptive resilient data-centric storage.

run of simulation, where the three mentioned methods are evaluated serially based on the same circumstances and target movements. Also, each run is repeated three times and the collected data are averaged. Therefore, the collected data are closer to real-world scenarios. Another advantage of this simulator is random propagation of nodes according to uniform distribution instead of using predetermined positions.

The final advantage is the sensing which is performed from the moving targets within the network instead of injecting unreal sensed data to the nodes. This simulator was developed with three built-in engines mentioned below:

- **Simulation engine:** This engine is the implementation of event-scheduling algorithm based on Jerry.²³
- **Two-dimensional (2D) view engine:** As its name illustrates, this engine is used for graphical drawing of the network.
- **Network engine:** DCS, R-DCS, and AR-DCS are implemented within a simulated WSN by this engine.

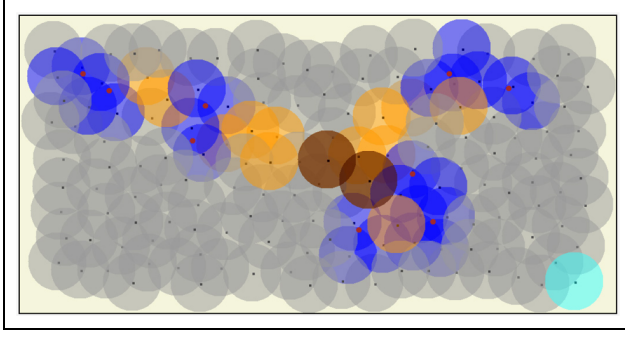


Figure 1. Storage data propagation from the sensing nodes to the replicas (DCS).

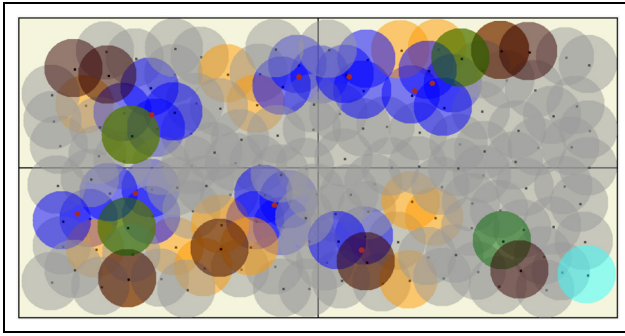


Figure 2. Storage data propagation from the sensing nodes to the monitors and replicas (R-DCS).

The 2D graphical views of the simulator using DCS, R-DCS, and AR-DCS methods are shown in Figures 1–3, respectively. Figure 1 shows the simulation sample for data dissemination of DCS; sensing nodes are illustrated in blue. The yellow nodes are receiving sensed data and the brown ones are replicas. The light-blue node specifies a Personal Digital Assistant (PDA) querying the network. Figure 2 shows the simulation sample with R-DCS. As the figure illustrates, in each zone, there is a green node which acts as a monitor for storage and query management. Figure 3 shows the behavior of AR-DCS in the burst traffic of storage process. When the frequency of storage packets sensed by the monitor passes the second-level threshold, AR-DCS acts like Figure 3. If the frequency of storage packets sensed by the monitor is between the first-level and second-level thresholds, then its behavior is like Figure 2 and whenever the mentioned frequency falls below the first-level threshold, AR-DCS performs data dissemination like Figure 1.

To determine the performance of the mentioned methods, output parameters including $TotalQ$, Q , Dq , and $OverallQ$ were evaluated during the 600 s simulation runs. Since the movements of the targets are random, there can be a situation where the traffic falls although the number of targets is increased within the

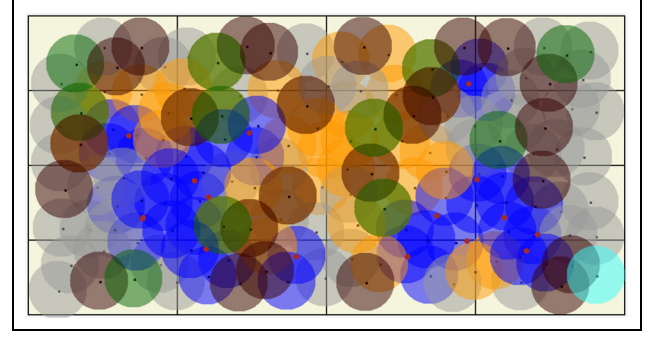


Figure 3. Storage data propagation from the sensing nodes to the monitors and replicas (AR-DCS burst fashion).

WSN network; this situation also influences the frequency of received data in monitor node, which has a direct effect on the query traffic. In order to divide zones easily, the default size of the network is 1024×512 m so that in each division, width and height of the zone are divided by two to create four subzones. By means of monitoring map,⁸ monitors were informed of the zone division and aggregation so that whenever a monitor observes the rising of frequency of received events over thresholds, it will inform other monitors to divide their zones and create four subzones within the parent zone. Also, when the frequency of received events returns to its normal state, zone aggregation will happen and subzones will be merged on their parent zone. The result of simulation is shown in Figures 4–7.

Figure 4 illustrates the $TotalQ$ for a various number of targets varies from 1 to 40. As the figure shows, AR-DCS acts like DCS when the number of targets is low. The reason for this phenomenon is the zone aggregation where the behavior of AR-DCS is like Figure 1. Also, whenever the number of the targets increases, zones are divided: more monitors and replicas are created (Figure 3) and it will cause storage process to be in the closer position of sensing node. As a result, the storage process cost will decrease and AR-DCS shows the best performance. It should be mentioned that, R-DCS becomes unstable, when the target number rises. Therefore, we set the initial energy of nodes to 2000 J in order to overcome the high amount of node failures in R-DCS. The reason for this behavior of R-DCS is related to the task of monitors. Each monitor acts like a gateway in the storage and query process so that its neighbors tolerate high amount of traffics and discharge fast. So, the routes to the monitor become longer, which bring more traffics to the nodes of network. Finally, all the nodes in the neighbor of the monitor discharge. This will cause storage and query process traffic to circle around the monitor and never has access to it. On the other hand, in AR-DCS, as the number of targets increases, zone division happens and

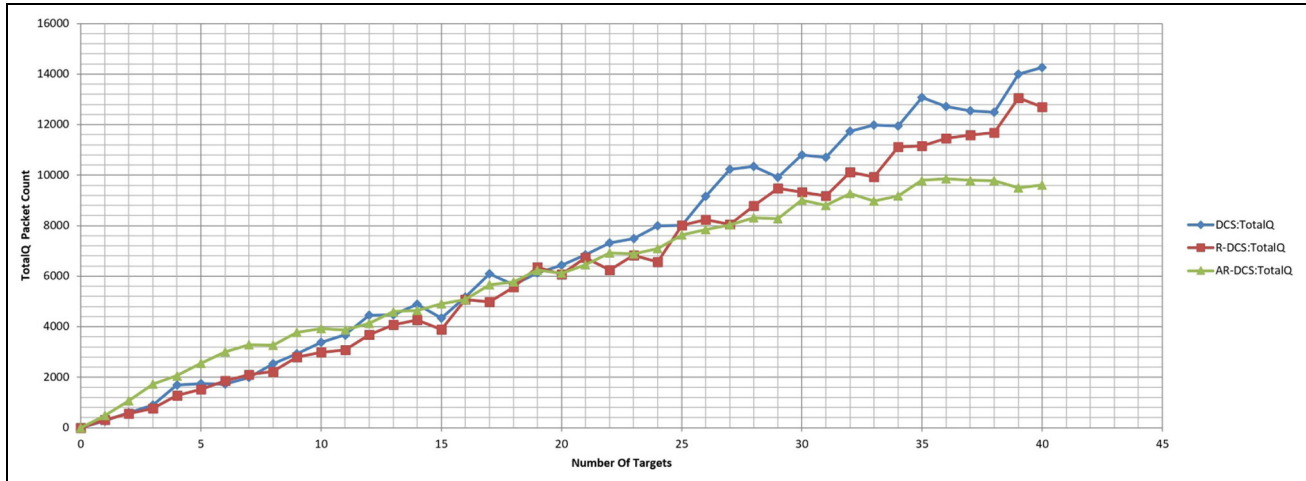


Figure 4. Number of packets sent for storage process (*TotalQ*).

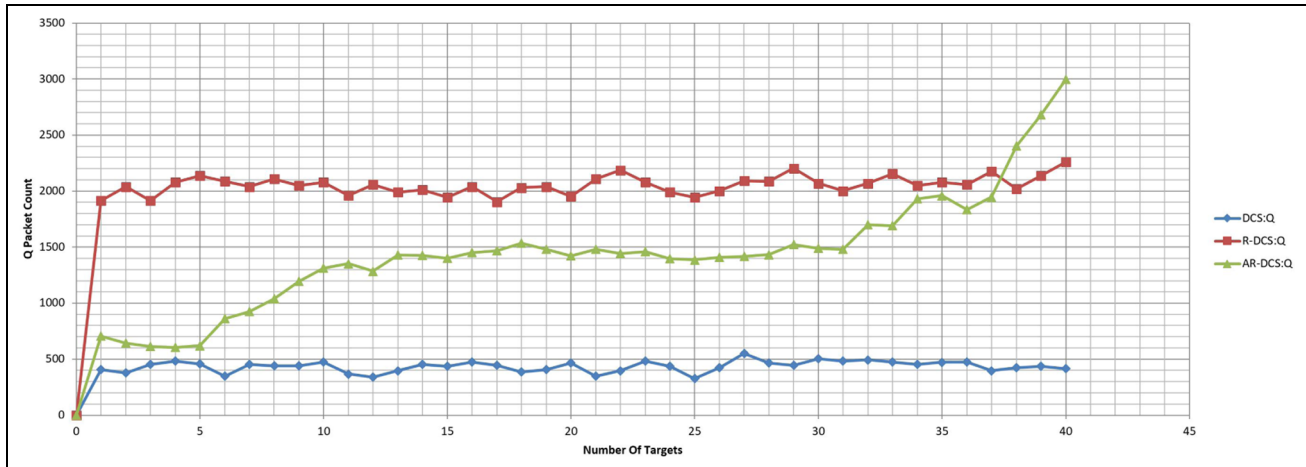


Figure 5. Number of packets sent for query process (*Q*).

more monitors will be available for the storage process. This will bring load balancing over the nodes of the network and decreases the storage traffic.

Figures 5 and 6 illustrate the total packets in query/response process. As the figures show, AR-DCS partly increases the traffic of query/response process compared with DCS and R-DCS. The reason for such a behavior is the zone division. Queries must be sent to more monitors. Monitors will query more replicas and more replicas will answer to the PDA. But AR-DCS adapts itself to the frequency of packets of storage process so that whenever the mentioned frequency is low, zones are aggregated and queries will be sent to less monitors and replicas. This will cause overall downgrade in the query process cost. That is why query/response process increases relatively. If we compare the increase in query process and decrease in storage process of AR-DCS with the DCS, we will recognize that they are similar in overall performance (Figure 7). But

it should be mentioned that AR-DCS covers node failure in the storage process by means of replicating the data over replicas.

According to the results, AR-DCS not only brings load balancing and fault tolerance to the WSN, but also shows better performance in the storage process against R-DCS and basic DCS. The only drawback of AR-DCS is a short increase in query/response process cost compared with basic DCS. It should be mentioned that the overall costs of the storage and query/response process of AR-DCS and basic DCS are similar. However, AR-DCS is resilient against the node failure which is not the feature of basic DCS.

Conclusion

DCS approach can be viewed as energy efficient and low cost solution for data dissemination in WSN. However, in the basic version of the DCS, nodes are

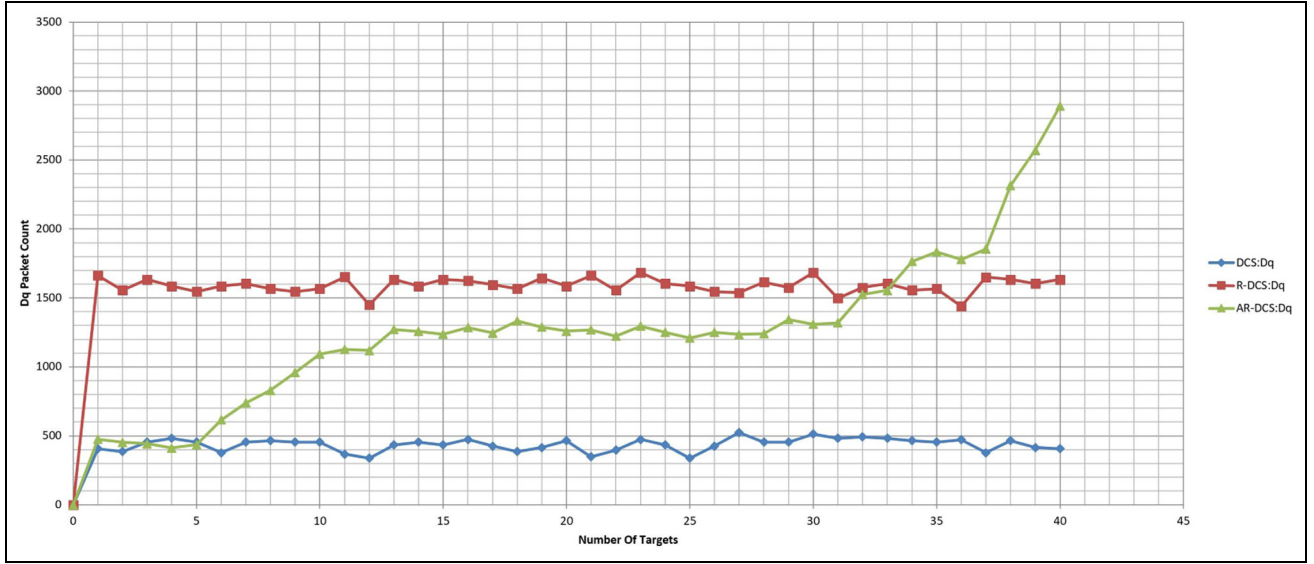


Figure 6. Number of packets sent for response process (Dq).

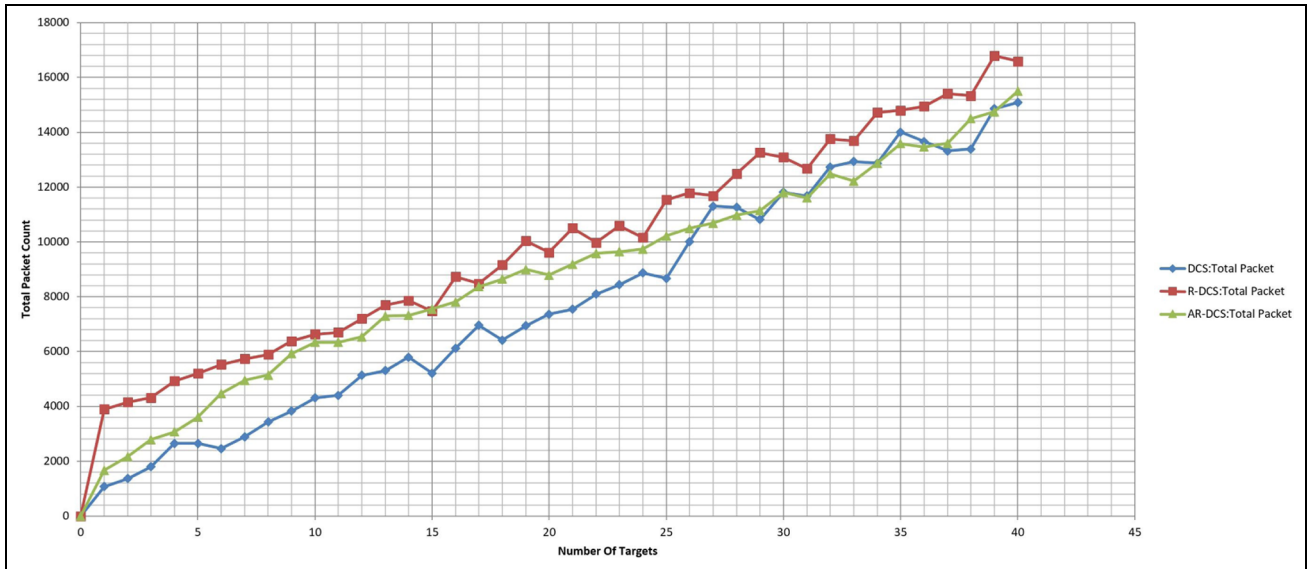


Figure 7. Total number of packets sent for storage and query process ($OverallQ$).

reliable and the network topology will not change during the long period of time so that the node failure is not considered. Therefore, methods like R-DCS have been proposed in order to overcome such an unsolved issue. This article proposed a novel hybrid approach called AR-DCS which makes R-DCS adaptive to the various circumstances of the network. As explained in section “AR-DCS,” instead of a predetermined threshold which introduced in Hejazi and Amin,¹² a mathematical one is used for zone division and aggregation.

This makes storage process more adaptive to the number of targets and their movements which bring load balancing and fault tolerance to the WSN. Also, a discrete-event-based simulator is developed in order to analyze the basic DCS, R-DCS, and AR-DCS in various circumstances of target number and their movements. The analytical results show that AR-DCS has better performance against R-DCS and DCS in storage process but also brings a short overhead in query/response process.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

1. Ratnasamy S, Karp B, Shenker S, et al. Data-centric storage in sensornets with GHT, a geographic hash table. *Mobile Netw Appl* 2003; 8(4): 427–442.
2. Gonizzi P, Ferrari G, Gay V, et al. Data dissemination scheme for distributed storage for IoT observation systems at large scale. *Inform Fusion* 2015; 22: 16–25.
3. Ratnasamy S, Karp B, Yin L, et al. GHT: a geographic hash table for data-centric storage. In: *Proceedings of the 1st ACM international workshop on wireless sensor networks and applications*, Atlanta, GA, 28 September 2002, pp.78–87. New York: ACM.
4. Karp B and Kung H-T. GPSR: Greedy Perimeter Stateless Routing for wireless networks. In: *Proceedings of the 6th annual international conference on mobile computing and networking*, Boston, MA, 6–11 August 2000, pp.243–254. New York: ACM.
5. Liu H, Wang H and Chen Y. Ensuring data storage security against frequency-based attacks in wireless networks. In: Rajaraman R, T Moscibroda, A Dunkels, et al. (eds) *Distributed computing in sensor systems*. Berlin, Heidelberg: Springer, 2010, pp.201–215.
6. Maia G, Guidoni DL, Viana AC, et al. A distributed data storage protocol for heterogeneous wireless sensor networks with mobile sinks. *Ad Hoc Netw* 2013; 11(5): 1588–1602.
7. Nair MN and Sebastian Terence J. Survey on distributed data storage schemes in wireless sensor networks. *Indian J Comput Sci Eng* 2014; 4(6).
8. Ghose A, Grossklags J and Chuang J. Resilient data-centric storage in wireless ad-hoc sensor networks. In: *Proceedings of the International Conference on Mobile Data Management*, 21 January 2003, pp.45–62. Berlin, Heidelberg: Springer.
9. Liao W-H, Shih K-P and Wu W-C. A grid-based dynamic load balancing approach for data-centric storage in wireless ad-hoc sensor networks. In: *Proceedings of the International Conference on Mobile Data Management*, 21 January 2003, pp.45–62. Berlin, Heidelberg: Springer.
10. Aly M, Chrysanthos PK and Pruhs K. Decomposing data-centric storage query hot-spots in sensor networks. In: *Proceedings of the 3rd annual international conference on mobile and ubiquitous systems-workshops*, San Jose, CA, 17–21 July 2006, pp.1–9. New York: IEEE.
11. Park Y, Seo D, Yun J, et al. An efficient data-centric storage method using time parameter for sensor networks. *Inform Sciences* 2010; 180(24): 4806–4817.
12. Hejazi P and Amin HH. An adaptive method for structured replication data-centric storage in wireless sensor networks. In: *Proceedings of the international conference on information technology and multimedia (ICIM)*, Kajang, Malaysia, 14–16 November 2011, pp.1–5. New York: IEEE.
13. Aly M, Pruhs K and Chrysanthos PK. KDDCS: a load-balanced in-network data-centric storage scheme for sensor networks. In: *Proceedings of the 15th ACM international conference on information and knowledge management*, Arlington, VA, 5–11 November 2006, pp.317–326. New York: ACM.
14. Li X, Kim YJ, Govindan R, et al. Multi-dimensional range queries in sensor networks. In: *Proceedings of the 1st international conference on embedded networked sensor systems*, Los Angeles, CA, 5–7 November 2003, pp.63–75. New York: ACM.
15. Lai Y, Wang Y and Chen H. Energy-efficient robust data-centric storage in wireless sensor networks. In: *Proceedings of the international conference on wireless communications, networking and mobile computing (WiCom 2007)*, Shanghai, China, 21–25 September 2007, pp.2735–2738. New York: IEEE.
16. Cuevas A, Urueña M, de Veciana G, et al. Dynamic data-centric storage for long-term storage in wireless sensor and actor networks. *Wirel Netw* 2014; 20(1): 141–153.
17. Ahmed K and Gregory MA. Distributed efficient similarity search mechanism in wireless sensor networks. *Sensors* 2015; 15(3): 5474–5503.
18. Babaei S and Sabaei M. Adaptive data-centric storage in wireless sensor networks. In: *Proceedings of the 3rd international conference on computer research and development (ICCRD)*, Shanghai, China, 11–13 March 2011, vol. 1, pp.163–167. New York: IEEE.
19. Dudkowski D, Jose Marron P and Rothermel K. An efficient resilience mechanism for data centric storage in mobile ad hoc networks. In: *Proceedings of the 7th international conference on mobile data management (MDM 2006)*, Nara, Japan, 10–12 May 2006, pp.7. IEEE.
20. Joung Y-J, Huang S-H and Lin S-H. Making data-centric storage adaptive and cost-optimal. *Comput Netw* 2012; 56(1): 213–230.
21. Graham RL, Lubachevsky BD, Nurmela KJ, et al. Dense packings of congruent circles in a circle. *Discrete Math* 1998; 181(1): 139–154.
22. TELOS B, http://www.willow.co.uk/TelosB_Datasheet.pdf (accessed 25 September 2015).
23. Jerry B. *Discrete-event system simulation*. Upper Saddle River, NJ: Prentice Hall, 2010. Print.