Available online at www.sciencedirect.com

ScienceDirect

Ad hoc Networks X (2016) 1–32

Ad hoc Networks

www.elsevier.com/locate/procedia

CBA: a Cluster-Based client/server data Aggregation routing protocol

Saeid Pourroostaei Ardakani*, Julian Padget, Marina De Vos

Computer Science Department, University of Bath, UK

Abstract

Client/server routing forwards data samples from the source nodes to the sink through single or multi-hop paths which are formed over a flat or hierarchical infrastructure. Depending on the routing infrastructure, the intermediate nodes may perform in-network data aggregation to collect and combine the data samples which are measures of the environmental events. Minimising energy consumption is a vital requirement due to resource constraints in wireless sensor networks. Data collection delay should be minimised as it is the key to data freshness. At the same time, the number of collected data samples should be maximised, as it should lead to increased accuracy and robustness in data collection. Owing to these, we define the system objective to be maximising the number of delivered data samples, while minimising energy consumption and data collection delay. We propose a cluster-based client/server data aggregation routing protocol called Cluster-Based client/server data Aggregation routing protocol (CBA). It dynamically partitions the network into a set of data-centric clusters using a lightweight clustering approach based on the Hamming distance. The cluster-heads then form a Minimum Spanning Tree (MST) as the network backbone to forward aggregated results to the sink. A parallel collision-guided technique is used to minimise the establishment cost of the tree infrastructure. Compared with the conventional routing protocols like MR-LEACH and Directed DiFFusion, CBA reduces energy consumption and data collection delay and increases accuracy (the number of captured data samples). In addition, our protocol reduces the impacts of the network architecture and the event source distribution model (distributed and centralised) on the performance of data aggregation routing.

Keywords: Wireless Sensor Networks, Client/Server, Data Aggregation and Path Planning

1. Introduction

Wireless Sensor Networks (WSNs) comprise a number of sensor nodes which typically measure and report environmental data. The nodes are typically networked in a self-organising manner without any specific infrastructure or centralised control [54]. The key objective of WSN protocols is to minimise the cost of ambient data collection. Ambient data samples should be collected and forwarded through minimum cost links (in terms of hop count and consumed energy) to data consumer access point (sink) for further analysis and manipulation. The WSN architecture is generally classified in of two ways: distributed (flat)

*Corresponding author.

Email addresses: spa23@bath.ac.uk (Saeid Pourroostaei Ardakani), jap@cs.bath.ac.uk (Julian Padget), mdv@cs.bath.ac.uk (Marina De Vos)

and hierarchical [38]. In the former, the nodes are randomly scattered in the field, whereas in the latter, sensor nodes are organised with a specific distribution topology such as grid, cluster or tree.

The sensor nodes themselves are typically highly resource constrained (in terms of energy, computation, communication and storage) and able to perform three key tasks:[2] (i) measuring a physical quantity (such as temperature or light) from the surrounding environment, (ii) processing (and storing) the sensed data and (iii) transmitting the information to collection points (called *Sinks*) for either future processing or consumer access.

WSN routing is the field of research that focuses on the interconnection of sensor nodes via either single or multi-hop paths to forward data packets from event regions to the sink. However, the routing overhead increases if raw data packets are forwarded from each source region to the sink. Data aggregation is a technique that collects and combines data packets to express the collected information in a summary form. It reduces the number and size of transmissions and eliminates redundant data packets. WSN Routing can be performed in two ways with data aggregation [47]: mobile agent and client/server. The former routes mobile agent(s) to collect and aggregate data samples from the sensor nodes, whereas the latter establishes an hierarchical network in which data packets are aggregated and forwarded from the ambient event regions to the sink in a convergent manner.

Client/server data aggregation routing establishes the paths according to the network architecture that can be flat or hierarchical [10]. In flat networks, the routes are established from the source nodes converging towards the sink as all nodes play same roles. Apart from sink, intermediate nodes can perform in-network data aggregation if they receive multiple data packets. However, no node is particularly selected to perform data aggregation in flat networks. In hierarchical networks, the nodes play different roles such as network bridge, intermediate aggregator or data consumer access point. Hierarchical routes are usually established from the source nodes to the sink via intermediate nodes which carry out the process of data aggregation.

There are five key issues that need to be considered by designers/developers of WSN client/server data aggregation routing whether flat or hierarchical [39], [52]:

1. **Energy consumption:** power resources need to be used efficiently in WSNs as they are highly constrained. Forwarding data packets over long paths, overhearing and message conflicts/collisions are the behaviours that increase energy consumption in flat WSNs. On the other hand, the cost of establishing and maintaining an hierarchical infrastructure must be minimised if the costs are not to outweigh the benefits.

2. **Network congestion:** simultaneous access to the limited wireless channels increases network congestion and consequently increase the probability of message failures in WSNs. It can increase network resource consumption as the source nodes need to re-transmit failed data packets. Network congestion is decreased in hierarchical networks, as compared to flat, due to the smaller number of nodes which need simultaneously to access the wireless channels. Hierarchical WSNs partition the network into a set of groups in which a small number of nodes (group leaders/representatives) are in charge of managing the group communications. However, network congestion could be problematic in hierarchical WSNs as the number of group and/or leaders increases.

3. **Overhearing:** receiving network packets which do not belong to the receiver nodes increases network resource consumption in WSN. Hierarchical infrastructure has the potential to reduce overhearing (compared to flat networks) as the communications can be locally limited to the node groups. Depending on the size of groups, however, overhearing is increased if groups are large and/or dense.

4. **Delay:** end-to-end delay (ETE) should be minimised in data collection as it is key to data freshness. ETE depends on network traffic and path length (hop count) from the source regions to the sink.

5. **Data collection/aggregation from Event-Radius (ER) and Random-Source (RS) event sources:** The event occurs in a single point of the sensing field in ER (i.e 100% detection), whereas the event sources are randomly distributed in RS (i.e random detection) [22]. RS data collection increases network congestion, delay and resource consumption especially in a flat network, as each source node needs to establish a path to forward data to the sink. It can be resolved in hierarchical networks by

grouping the source nodes in which the group representatives forward the aggregated data of group node to the sink. This results in a reduction of routing overhead, network traffic and resource consumption. However, the group leaders miss collecting data samples from source nodes which are not joined to the hierarchical infrastructure. For this reason, the hierarchical infrastructure needs to minimise the establishment and maintenance cost and maximise coverage of event regions either in RS or ER.

In the remainder of this article, Section 2 outlines well-known client/server data aggregation routing protocols to highlight their advantages, features and techniques. Section 3 describes the CBA protocol and the key techniques which are used to enhance the performance and resolve the existing drawbacks of client/server data aggregation routing. Section 4 focuses on the experimental plans to test the performance of CBA. Section 5 evaluates the performance of CBA according to five metrics: total consumed energy, total number of delivered data samples (accuracy), average end-to-end delay, average hop count and total transmitted traffic which are usually used to test the performance of client/server data aggregation routing protocols. The results are measured and discussed to evaluate the performance of CBA in comparison to two client/server data aggregation routing protocols namely MR-LEACH [14](hierarchical) and Directed DiFFusion (DDiFF) [20] (flat). These protocols are selected as they are well-known in the literature, widely simulated and implemented both in the real world and for our chosen experimental platform of OMNET++. This last contributes to the correctness and credibility of our evaluation, because we are able to compare CBA against two client/server routing protocols that have been independently written and verified for OMNET++. Comparison against more recent protocols is also desirable, but this is not feasible without appropriately-verified implementations in OMNET++ yet lacks credibility if authored ourselves. Section 6 concludes the key advantages and disadvantages of CBA protocol and then highlights the research issues which need to be addressed as future works.

2. Related Work

This section introduces and compares a set of client/server routing protocols (both flat and hierarchical architectures) have been proposed for data aggregation in WSNs. This section does not provide a statistical analysis of the routing protocols, but it explains and highlights the distinctive techniques, features and schemes that are used in the introduced protocols.

2.1. Flat Architectures

Flat data aggregation routing establishes low-cost routes from the source nodes to the sink to forward data samples. Data packets are reactively aggregated at intermediate nodes while they are being transmitted. Flat routes may be established either in address-centric (AC) or data-centric (DC). The former focuses on establishing and recognising the paths based on the address of nodes, whereas the latter considers data attributes (e.g datatype) at the intermediate nodes to establish the paths. DC routing is commonly used to provide in-network data aggregation as it forwards through the nodes that are aware of data content and are able to perform in-network data aggregation.

Sensor Protocols for Information via Negotiation (SPIN) [23] is a negotiation-based data-centric (DC) routing protocol that uses meta data rather than the original data to establish the routes over a flat WSN. First, each source node advertises its data (using a high-level data indicator message) in its single-hop neighbourhood. Each neighbour node that is interested in collecting, aggregating and reporting data replies back by sending a request packet. Finally, the source node sends the original data packet to any of its neighbours who already asked for. The intermediate nodes perform a similar scheme to forward data packets until the sink receives them. However, SPIN is not able to guarantee data delivery because of the utilising negotiation-based data transmission scheme to forward data packets. This means that source nodes do not forward data packets if they do not receive a request for data if there is a hole (disconnection) or request packets are lost. Moreover, SPIN can not be considered energy-efficient as the intermediate sensor nodes waste energy resources to stay available over long periods to receive data advertisements.

Table 1: Flat WSN routing protocols

Protocol	Key advantage	Key drawback	communication	Routing metric
SPIN	reducing the traffic of query broadcasts	data delivery not guaranteed	multi/unicast	data centric
Directed DiFFusion	reducing energy/delay of query broadcasts	bottlenecks	unicast	1- minimum hop count 2- data centric
MCFA	easy setup	increasing message conflicts	uni/multicast	minimum hop count

Directed DiFFusion (DDiFF) [20] is a query-driven routing protocol that utilises data naming technique to forward data packets. First, the sink propagates data queries containing a set of attribute values (i.e data type, freshness ratio and/or geographical area) which describes the interesting data to collect. The queries are periodically propagated by the sink to check/refresh the availability of possible routes. Upon receiving the query packets, intermediate nodes record the routing information of packets in their local tables to establish/reserve return links from the source nodes to the sink. The recorded routing information is used for in-network data aggregation when the data packets are transmitted from the source nodes through the return paths. The query propagation is repeated by intermediate nodes until they are received by the source nodes that have interesting data. As source nodes receive a number of similar queries that are forwarded through variant routes, they need to select the optimal path to report data packets. In turn, they therefore consider a set of factors (like end-to-end delay or hop-count) to establish a low cost or delay links. The selected path (called gradient) is used to forward the data packet to the sink. Other possible routes (which are recorded at the intermediate nodes) are used as alternatives when the gradient fails. Data packets are aggregated and forwarded by intermediate nodes that reside on the gradient until they are received by the sink. However, DDiFF is not particularly useful for continuous data collection as nodes need to consume a great deal of energy to transmit multiple queries and data packets. In other words, nodes (bottleneck) residing on the gradient consume a greater deal of energy to transmit queries and data packets which are frequently transmitted.

Minimum Cost Forwarding Algorithm (MCFA) [49] establishes multiple routes from data regions to the sink in which the path with the minimum cost (hop count) is selected to forward data packets. Prior to network deployment, each node is assigned a cost that is initially infinity. The cost is updated when a cost packet is received by a node. The cost packet is used to let the nodes know how many hops they are away from the sink. The sink initialises the hop count value to zero and then broadcasts the cost packet. The hop count value is incremented by one at each node which receives it. The cost value of each node is updated to the new hop count value and then the cost packet is forwarded for the neighbouring nodes. It is repeated until all nodes have set their cost according to the cost packet hop count. Then, data packets are aggregated and forwarded via intermediate nodes that have a smaller cost value (are closer to the sink) until they are received at the sink. However, the key drawback of MCFA is that a large number of redundant cost packets is blindly re-transmitted as the intermediate nodes forward any received cost packet to their neighbouring nodes. In consequence, the message conflict ratio will increase especially when the network deployed is dense. MCFA has been extended in [18] by utilising a technique that allows the intermediate sensor nodes to wait for a period before re-transmitting the cost packets. Using this technique, the receiver would be able to consider/aggregate a number of cost packets before re-transmitting them. It would result in a reduction of network traffic and consequently decrease the energy consumption.

The drawbacks of flat data aggregation routing are: (i) establishing shortest path between each source nodes and the sink consumes network resources especially when the network deployed is large and dense, (ii) message failure and network congestion is increased as the source nodes simultaneously try to access the wireless channels to forward data packets, (iii) re-transmission of data packets (which are lost/collided) increases energy consumption, (iv) data packet failures reduce data aggregation accuracy and robustness as the number of collected data samples at the sink reduces, (v) transmitting data packets through paths that have variant end-to-end delays can increase data collection end-to-end delay and influence data freshness [3]. This means that data collection delay can vary as data packets are delivered to the sink through different

paths with different hop counts. The flat routing protocols are highlighted and compared in Table 1.

2.2. Hierarchical Architectures

Hierarchical data aggregation establishes a hierarchical infrastructure to collect, aggregate and transmit data packets from the source nodes to the sink. It has the potential to resolve the drawbacks in flat data aggregation. In hierarchical data aggregation, source nodes do not try to transmit data packets to the sink, but they forward data packets to a set of intermediate nodes which are specifically selected for the duty of performing in-network aggregation. Hence, data packets get aggregated earlier in hierarchical networks as compared to flat. In fact, instead of any random node that resides on a joint path, intermediate aggregators hierarchically aggregate data samples and forward then the results to the sink. Owing to this, network traffic and congestion is reduced in hierarchical data aggregation routing as the number of forwarding data packets and/or relay nodes is normally lower [19]. It results in a reduction of collided/lost messages, data collection delay and increased data collection accuracy.

There are a number of different techniques that are used to form hierarchical infrastructures in WSNs. Aggregation trees, clusters and chains are the most commonly used ones as we explain below:

1. **Aggregation tree:** a tree infrastructure is formed in which data packets are hierarchically forwarded by the source nodes to the parent nodes to perform in-network data aggregation. The aggregated results are forwarded by the parent nodes at each level until they are received by the sink. The objective of establishing the aggregation tree is to minimise energy network resource consumption and maximise the number of collected data samples [4]. In other words, the aggregation tree should be established with the maximum number of interesting source nodes in an energy efficient manner. Tiny AGgregation service for ad-hoc sensor networks (TAG) [29] proposes a dynamic hierarchical data aggregation protocol that establishes a tree infrastructure to collect and aggregate environmental data. First, a level discovery message is broadcast by the sink to allocate a level number to each node. Each node that receives the level discovery message increases the level value by one and then re-broadcasts the message for the next hop nodes. This procedure is repeated until all nodes receive level values. Data packets are forwarded from the source nodes to their upper level nodes (parents) to collect and aggregate. They are forwarded through the tree infrastructure until the aggregated results are received by the sink. Temporal coherency-aware In-Network Aggregation (TINA) [37] similarly establishes a tree infrastructure to collect and aggregate data samples. The difference of TAG and TiNA is that the latter utilises temporal coherency tolerances to reduce energy consumption over the aggregation tree. This means that the source nodes in TiNA do not transmit all the measured data, but they just forward data packets whose values differ with data which is already reported. For this reason, a new parameter is added to sink queries called ”tct”. It shows the consumer preference tolerant degree to report a data sample. A data sample is reported if it differs with the last reported value greater than ”tct”. According to the simulation results [37], the number of transmissions as well as energy consumption is reduced in TiNA as compared to TAG.

2. **Clustering:** this technique partitions the network into a set of groups named clusters. The clusters are formed based on the similarity of nodes according to a set of distinctive features like location, measured data and/or communication and computation behaviours. Each cluster consists of a set of Cluster Members (CMs) in which a single or multiple ones are selected as the cluster representative(s). The cluster representatives are called Cluster Heads (CHs) and responsible for collecting and aggregating intra-cluster data samples. Aggregated results are hierarchically forwarded from CHs to sink via single or multi-hop paths. Low-Energy Adaptive Clustering Hierarchy (LEACH) [17] is a cluster-based routing algorithm that supports data aggregation. LEACH has two phases: setup and steady-state. The former is the process of network clustering, whereas the latter routes data packets from source nodes to sink. The cluster-heads (CHs) are periodically selected based on a distributed random algorithm in which each cluster member may become a CH for a particular round according to a probability value (P). The probability value allows a cluster member to become a CH for $\frac{1}{P}$ round. In other words, there is no chance for a node to be a CH again for the next P rounds. The

source nodes collect and transmit data samples to cluster-heads using TDMA (Time Division Multiple Access) [31] to avoid intra-cluster collisions. Cluster-heads collect and aggregate data samples and then transmit the results to the sink. CDMA (Code Division Multiple Access) [7] is used by CHs to avoid inter-cluster interference. However, there are three drawbacks in LEACH: (1) increasing network energy consumption due to periodical CH selection (to replace low battery CHs with new ones), (2) establishing single-hop inter and/or intra cluster links between the sink, cluster-heads and cluster members to forward data is not feasible for WSNs which are deployed in large areas, (3) non-balance CH distribution and uncertainty in cluster count and size . For this reason, a set of modified version of LEACH are proposed aiming to resolve the drawbacks. A Two Level LEACH (TL-LEACH) [28] resolves inter/intra cluster single hop communications by establishing a two-level clustered infrastructure. They are called primary and secondary clusters. The secondary cluster-heads collect data from the source nodes and then transmit the aggregated results to the primary ones to report to the sink. Multi-hop Routing with LEACH (MR-LEACH) [14] extends TL-LEACH by providing multi-hop paths between the CHs to transmit data packets to the sink. Each cluster head would relay data packets that are forwarded from the CHs residing in lower hierarchy level via multi-hop paths to sink. Energy aware LEACH (E-LEACH) [44] initially selects CHs similar to original LEACH (randomly) and then utilises residual energy at each nodes to select the CHs for next rounds. LEACH-centralised (LEACH-C) [16] uses the sink as a centralised point to create the clusters in an optimal way. The sensor nodes forward the required clustering information such as location, residual energy and/or connectivity degree to the sink during the set-up phase. The sink proactively forms a set of balance clusters in terms of energy, coverage and connectivity and then allocates the roles (i.e CH and/or cluster member) to the nodes. The overhead of collecting clustering information at the sink to form the clusters is a drawback of LEACH-C. Vice-CH LEACH (V-LEACH) [48] selects a vice-cluster head at each cluster to handle cluster communications is the case of CH failure. The CHs in LEACH may fail quicker (due to running out of energy) than cluster members as they usually need to perform a greater number of communication/computation tasks. For this reason, the vice-cluster head would stay in the cluster to cover CH duty if it fails. It would result in enhancing the network lifetime. LEACH Fuzzy Logic (LEACH-FL) [34] utilises fuzzy logics based on three metrics: residual energy level, density and distance from sink to select CHs. The author claims LEACH-FL has the potential to reduce energy consumption of CH selection and consequently enhance network lifetime.

3. **Aggregation chain:** the hierarchical infrastructure for data collection/aggregation is formed by a chain of the source nodes which have interesting data to report. The chain is usually rooted in a node called the leader and is responsible for reporting the aggregated result of chain members to the sink. Data samples are hierarchically forwarded and aggregated from the source region to the leader nodes. The leader(s) collect and forward the aggregated result to the sink directly or indirectly. PEGASIS (Power-Efficient GAthering in Sensor Information Systems) [25] hierarchically forms a chain-based infrastructure to route data packets. It selects a set of nodes as leader nodes according to residual energy level and/or location information to collect, aggregate and transmit data samples. Data packets are forwarded from source nodes to the next hop nodes, if they are closer to the leaders, using a greedy algorithm. Each node aggregates the received data with its own and then transmits the result until the leader node receives. The leader nodes are responsible for reporting the results to the sink. If the leader node fails, sensor nodes leave the chain to construct a new one with a new leader. The difference of PEGASIS and MR-LEACH is that the sensor nodes do not need to periodically pay the clustering cost to re-cluster the network. However, the overhead of leader selection is increased in PEGASIS when the network works over long period. The sensor nodes will require a dynamic topology adjustment to collect information (i.e residual energy which changes over time) that are required to select or re-select the leaders. Moreover, the leader becomes bottleneck if data samples are frequently transmitted to the sink. Data collection delay also is increased in PEGASIS due to the multi-hop transmissions (with variant hop count) from source nodes to the sink. Hierarchical-PEGASIS [25] is proposed by the same author aiming to resolve the delay drawback. It reduces delay using parallel transmissions from the source nodes to the sink. It uses two techniques to provide parallel communications: signal

Table 2: Hierarchical WSN routing protocols

Protocol	Architecture	Key advantage	Key drawback	Mechanism	Routing metric
TAG	Tree	simple implementation	bottlenecks at parent nodes	level discovery	shortest paths
LEACH (MR-LEACH)	AC cluster	1- reducing message collision 2- increasing lifetime	1- random and non-balance clusters 2- not defined number of iterations	rotated clustering with infinite iterations	fresh routes
PEGASIS	Chain	reducing cost of infrastructure	1- bottlenecks 2- increasing delay	greedy distance (hop count) to sink	shortest paths

Table 3: Flat vs. Hierarchical data aggregation routing

Features	Flat Networks	Hierarchical Network
Aggregator nodes	any node on the path	intermediate aggregators i.d leader/CH
Node failures	disconnect network	locally disconnect clustered or grouped area
Traffic congestion	High (packets are forwarded by any)	low (aggregators forward data packets)
Message collisions	High	low
deployment cost	low	high
Node heterogeneity	doesn't matter	stronger nodes are selected as aggregators
Delay	High (multi-hop paths from event regions to sink)	Low (early aggregation)

coding (e.g CDMA) and transmitting spatial separated data. In the former, the nodes construct a tree of chains which is rooted in the sink. Using this tree, the nodes at each level transmit data packets to their leader in parallel. Each level of transmissions is coded by CDMA that allows collision-free parallel communications. The latter allows the nodes that physically reside close to each other to transmit data packets to the leaders at each round. In other words, transmitting spatial separated data allows the source nodes to be grouped spatially and then group members independently transmit data samples to the leader.

Table 2 highlights and compares the key features of the hierarchical routing protocols discussed above.

There are two key drawbacks in hierarchical client/server data aggregation [26]: (1) infrastructure establishment and maintenance cost: the overhead of (re-)establishing and maintaining the hierarchical infrastructure (due to network topology or density changes) increases network resource consumption and reduce network lifetime, (2) Leader/CH bottlenecks: computation and communication task loads usually focus on the intermediate aggregators (i.e leaders or CHs) rather than other nodes in hierarchical data aggregation.

The key features of flat and hierarchal data aggregation routing are compared in Table 3.

3. The CBA Protocol

This section proposes a cluster-based routing protocol (CBA) which supports data aggregation in a client/server model. CBA allocates a cost value to each node according to the distance and path hop count from the sink. Then, it partitions the network into a set of data-centric (based on data type) clusters using the Hamming distance technique [45]. The data packets are aggregated at the cluster-heads and hierarchically forwarded then through a spanning tree to the sink. The tree infrastructure is rooted at the sink and formed as a result of parallel route request collisions which are forwarded from the CHs to the sink.

CBA offers three key contributions to the current literature. First, CBA is able to balance the existing trade-off between energy consumption and data collection latency during the data aggregation routing procedure. CBA reports the collected data through a tree infrastructure in which the links are established based on Euclidean distance and hop count. Euclidean distance has a high impact on the energy consumption, whereas hop count influences the communication delays. Hence, CBA is able to optimise the trade-off between the energy consumption and data collection latency if the Euclidean distance (energy consumption)

and hop count (delay) are minimised. Second, CBA utilises a parallel collision-guided technique which has the potential to reduce the establishment cost of the routing infrastructure (tree). Although establishing a spanning tree from the event regions to the sink offers increased accuracy and reduced delay in client/server data aggregation routing, it is expensive in terms of energy consumption. Parallel collision guidance is a suitable technique to form the spanning tree that minimises the establishment cost. This technique avoids forwarding useless and/or redundant control packets during the tree establishment phase. Hence, fewer control packets are forwarded and the establishment cost of the tree is reduced especially when the network is dense and large. Third, CBA supports early data aggregation which has the potential to reduce energy consumption and end-to-end delay in client/server data aggregation routing. Aggregating data packets as soon as possible (in terms of traversed hop count) results in a reduction in the amount of transmitted traffic. Hence, a fewer number of relay nodes are required to relay the aggregated results from the event regions to the sink. On the other hand, data packets are forwarded from source nodes to the aggregator nodes (i.e sink) when early data aggregation is not supported, which results in increased energy cost and end-to-end delay in client/server routing.

3.1. Cost Value Allocation

Each node in CBA is allocated a cost value according to its distance (hop count) from the sink. The value is assigned to inform the nodes how far they reside from the sink. Using the cost value, each node would be able to guide network transmissions (i.e data or control packets) to regions either closer to or farther from the sink. In fact, the nodes should avoid pure broadcasting, since that is usually utilised to forward network packets where a global addressing scheme or location information is not available. This means that the sink queriers are usually broadcast by each node throughout the network when this technique is not used, whereas they are forwarded only to the nodes which reside in farther regions from the sink if the nodes are allocated by the cost values.

CBA utilises a similar approach to the Minimum Cost Forwarding Algorithm (MCFA) [49] to allocate the costs to the nodes. The procedure is started when the sink forwards the cost packets. As Figure 1 shows, there are three fields in the header of cost packets: sender Id, Hop Count(HC) and Total Cost (TC). HC shows the hop count and TC is the total distance value of the interconnected links on the Forward Path (FP). The receiver nodes need to record both the values to select energy efficient and minimum delay routes. The routes with lower HC have lower communication delay as they are established by a fewer number of intermediate nodes. The paths with lower TC are more energy efficient as they are shorter in terms of Euclidean distance. Transmission distance has a high impact on energy consumption on the sender side that means more energy is typically required to transmit messages over greater distances. To estimate distance to a sender node, each receiver node measures Received Signal Strength Indication (RSSI) [46] value on the arrival of a cost packet. A Line-Of-Sight (LOS) model [41] is used by CBA for wireless signal propagation when the nodes communicate with each other. This model assumes that there is no obstacle between the sender and receiver nodes and they communicate over a flat surface [15]. Using this model, the receiver is able to estimate its distance to the sender using the RSSI technique when a wireless signal (cost packet) is received. However, obstacles affect the quality of the received signal and the receiver cannot estimate the distance if a Non-LOS (NLOS) model is used. In addition, it is assumed that there is no ambient (or thermal) noise affecting the wireless signals. The receiver is not able to measure RSSI and estimate the distance in case of environmental noise as it reduces the quality and reliability of wireless signals. According to Equation 1, the power of the received (P_R) and transmitted (P_T) signals depends on the distance (d) between the sender and receiver and an environmental value (n). This means that a stronger signal is received from a closer sender. It stems from the reduction of the power of wireless signals due to the fading effects over communication distances. The RSSI is measured using the power of sent and received signals according to Equation 2. A receiver node would be able to measure/estimate its shortest Euclidean path to a sender node if the RSSI value is maximised. This means that RSSI value increases when the sender node is closer as the receiving signals have greater power. The RSSI technique is suitable for WSN as its implementation cost and complexity is low [21].

$$P_R = (P_T) \times (1/d)^n \tag{1}$$

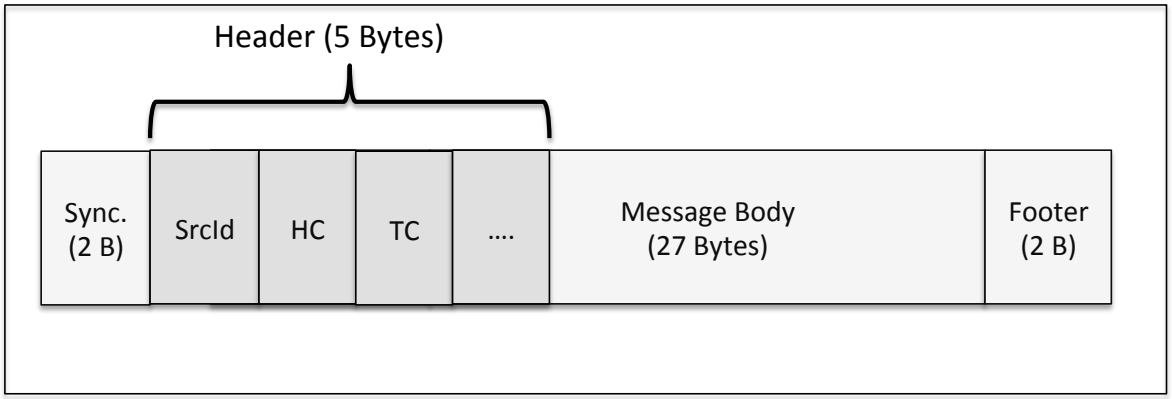


Fig. 1: The structure of cost messages in CBA

$$RSSI(dB) = 10 \log \frac{P_R}{P_T} \tag{2}$$

CBA has the ability to support low cost and/or delay communication links according to the data consumer and/or QoS requirements. When a cost message is received, the RRSI value of the link is measured and added then to TC value. Indeed, TC value gives the total RSSI weight of the route from the sink to the node. The receiver node also increases the value of HC by one. The value of HC shows the node hop count from the sink. However, multiple cost packets are received through different routes during the cost allocation phase. The packets have different TC and HC values according to the link length and intermediate hop count. A cost packet is immediately discarded if it has a greater HC (forwarded via a longer path in terms of hop count) than the node hop count. The ID of the sender node which has the minimum value of HC is recorded as TS (To the Sink) to establish a minimum delay Backward Path (BP). A BP is used to send the collected data from the event regions to the sink. If multiple messages with same HC values (with the node hop count) are received, the receiver node record, the ID of sender nodes as BackUp TS (BUTS) to establish alternative BPs if the primary one fails.

Each sensor node needs to wait for while in order to allow for the receipt of several cost messages arriving via different routes, since some may have a lower TC. Figure 2 depicts an example in which node D receives three cost packets through paths R1, R2 and R3. The network messages are normally received quicker through the paths with fewer hop count as the communication delay decreases [6]. For this reason, it is assumed that the cost packets are received in the order of R1, R3 and R2. An inaccurate TC value is transmitted to the next hop nodes if D immediately transmits the cost message which is received through R1. In this case, TC(R3) should be propagated for the next hop nodes instead of TC(R1) as the latter is greater. This drawback is resolved in a similar way to MCFA [49]. It lets the nodes wait for a short period (back-off) until other cost packets which have smaller cost are received. However, the waiting time increases network deployment delay. Moreover, it increases energy consumption as the nodes consume more energy by receiving multiple cost packets. For this reason, the back-off time should be minimised as much as possible. MCFA [49] recommends a 10 ms back-off time before re-transmitting the cost packets.

3.2. Forming Data-Centric Clusters

CBA utilises clustering to establish an hierarchical routing infrastructure for data aggregation routing. With clustering, network congestion and energy consumption is reduced due to a decrease in the number of transmitted data packets [27]. As data packets are aggregated at the CHs, the source nodes do not need to forward their data to the sink individually. In consequence, clustering results in a reduction of transmitted network traffic, which decreases energy consumption and network congestion. CBA partitions the network into a set of clusters which are dynamically formed in a data-centric manner. Data-centric (DC) clustering

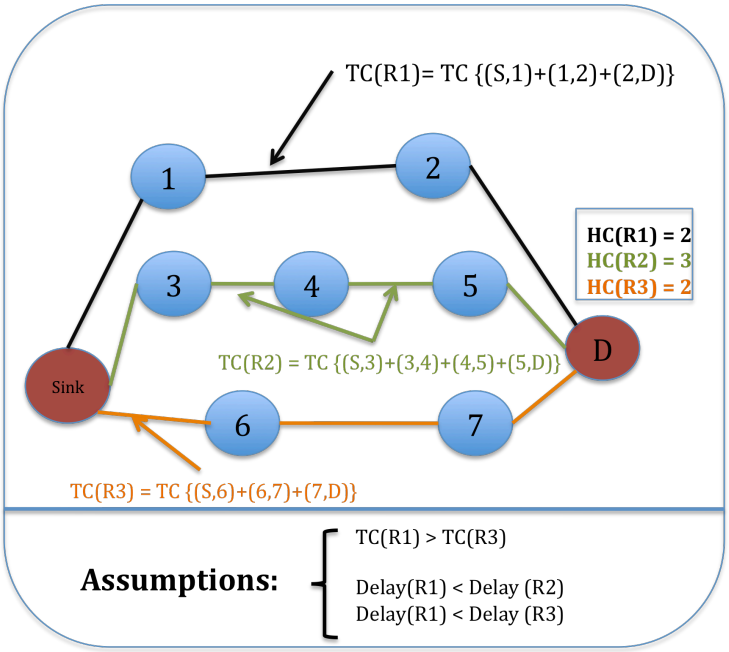


Fig. 2: Cost value allocation in CBA

in CBA refers to the clustering process in which sensor nodes are grouped based on their datatype. Each cluster can be considered as a super node which has a particular data type to report/collect.

The Hamming distance technique [45] is used to form cluster in CBA. This technique is generally used to find the difference of binary values, by counting the number of flipped bits in fixed size binary data streams and returning the value of the difference as the distance. However, this technique can also be used to cluster WSNs in a data-centric manner. The sensors nodes code the meaningful features of measured data such as data type, measuring time and/or location into binary vectors. Then, sensor nodes figure out the similarities of data features (e.g data type) by calculating the Hamming distance. For example, sensor nodes that have a zero Hamming distance (same data type) are grouped into data-centric clusters in a distributed manner without requiring a centralised control [5].

Using Hamming distance, the source nodes are grouped into a set of DC clusters according to the measured data type. Each cluster is managed by a single CH. The clustering procedure is started when the source nodes broadcast data advertisement messages (ADV_{msg}). The header of an ADV_{msg} consists of four fields which are encoded with the Hamming: node data type, residual energy level (reliability), hop count (location) and TC (link cost). These last three allow the nodes to select the most powerful nodes (in terms of having sufficient energy resources), with minimum link cost to the sink as the CHs, while the data type is used to form data-centric (same data type) clusters. The clustering procedure is performed in a distributed manner by the source nodes which are connected (single-hop) and have the same Hamming code for their data types. When a ADV_{msg} message is received, the node updates its routing table with the sender ID, data type, energy level and HC. The nodes, which have the same Hamming code for data and HC, are grouped into the clusters. Then, each node considers its routing table to find the strongest node which has the greatest level of energy. If there are multiple nodes which have same (greatest) energy level, the node with smallest TC is selected as CH because it would be able to communicate with the sink via a minimum cost link if there is still a tie. However, the node with smallest ID is selected as CH if multiple nodes have the same (greatest) energy level and (smallest) TC. The source nodes send (in unicast) a MY_{CH} message to the candidate CH to inform it about the selection. The node which receives the messages becomes the CH. Finally, the CMs are

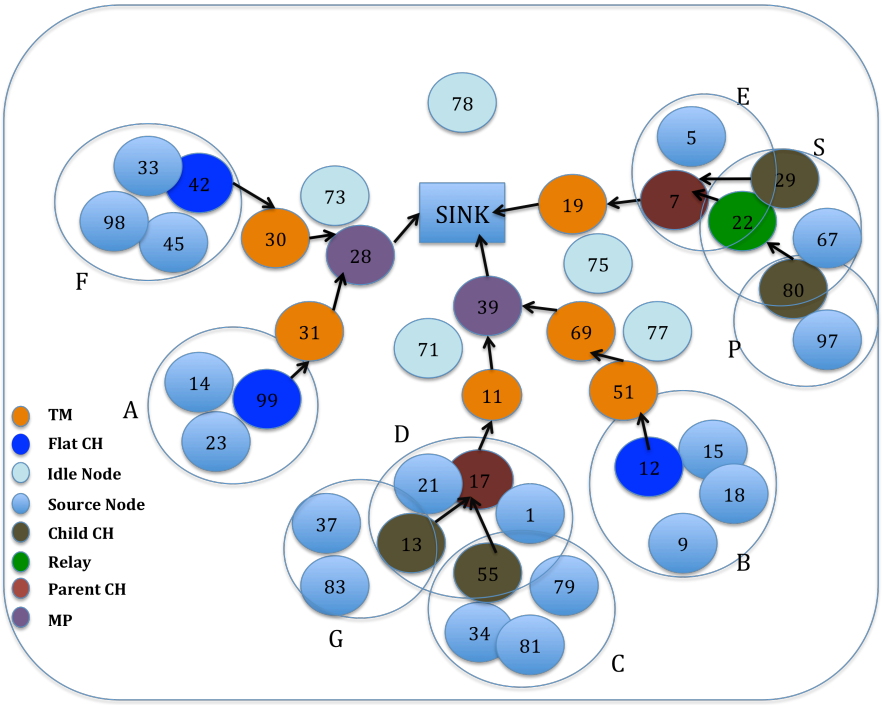


Fig. 3: Clustering in CBA

allocated by a cluster number according to their CH ID and data type.

CBA clusters are formed under two schemes: flat and hierarchical. A flat cluster is formed with no interconnection to any other clusters, whereas an hierarchal cluster is connected to at least one node which belongs to the next cluster. In the former, a CH does not know any other node in its single-hop neighbourhood which has a greater level of energy or a different cluster number. On the other hand, the CH has a connection to at least one node that belongs to another cluster among the hierarchical clusters. The hierarchal clusters are interconnected through inter-cluster links which are used during the data aggregation routing to forward data packets. The cluster in the topmost level of the hierarchy (with smallest HC) is called the parent cluster. According to Figure 3, for example, cluster A is flat as its CH (99) is not connected to any other cluster, whereas clusters P, S and E (parent) are hierarchical as they are interconnected.

Table 4: A Glossary for Tree Establishment Algorithm

Acronym	Definition	Acronym	Definition
RREQ	Route REQuest packet	HC	Hop Count
RREP	Route REPly packet	TC	Total Cost
srcid	the last sender ID	BP	Backward Path
FP	Forward Path	BHC	BP Hop Count
BTC	BP Total Cost	TS	To the Sink
BUTS	BackUp TS	TM	Tree Member
FRR	Failed Route Request	MP	Meet Point
APC	Aggregation Path hop Count	CH	Cluster Head

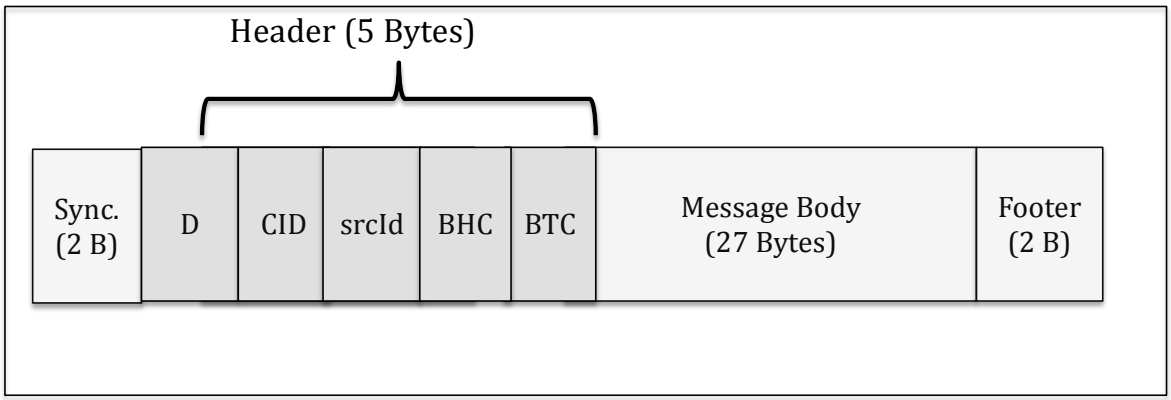


Fig. 4: The structure of RREQ in CBA

3.3. Data Aggregation Routing

To reduce the cost of data reporting, CBA forms a tree-based backbone from the CHs to the sink to collect and aggregate data via minimum hop-count links. The tree infrastructure is established to aggregate and forward data packets from the event regions to the sink in an hierarchical manner. It establishes minimum hop count paths from the event regions to the sink, in which data packets get aggregated as soon as possible. Hence, forwarding data packets through the tree infrastructure leads to a reduction in the total number of transmitted data packets and consequently the data reporting cost. This section describes the tree construction algorithm. Table 4 lists the acronyms which are used in data aggregation routing algorithm.

The tree infrastructure is established by interconnecting the minimum hop count routes from the CHs to the sink. It is rooted in the sink and formed in a data-centric manner using route request packets (RREQs) and route reply packets (RREPs) which are forwarded between the CHs and sink. The RREQ messages are forwarded from the CHs to establish inter-cluster routes and/or interconnect the data regions (DC CHs) to the sink, whereas RREP messages are forwarded from the sink to the CHs to form the tree branches. In addition, it allows CBA to work over heterogeneous WSNs in which the nodes have variant data types to report. This means that CBA is able to collect interesting data samples from source nodes with different data types as the tree infrastructure is dynamically formed from DC CHs to the sink based on the data consumer interests.

To form the tree backbone, CBA utilises a parallel collision-guided technique. It is conceptually similar to [8], but differs in implementation and execution. The parallel collision-guided approach has the potential to reduce the overhead of the backbone establishment. It avoids forwarding redundant and useless route requests (RREQs) through the links which are already established. The RREQ messages are forwarded in parallel from each data region whether flat or hierarchical to the sink. In hierarchical clusters, RREQs are forwarded only by the parent CHs. This is because the child clusters are hierarchically interconnected to the parent cluster(s) during the clustering procedure. Hence, establishing a path from a parent cluster to the sink results in interconnecting all its child clusters to the sink. This reduces the number of transmitted RREQs and routing overhead in hierarchical clusters. For example, cluster-head 17 is the only node that forwards a RREQ to the sink from clusters C, D and G which are hierarchically interconnected according to Figure 3.

As Figure 4 shows, the header of a route request packet (RREQ) has five fields: cluster data type (D), CH id (CID), last sender ID (srcId), BP Hop Count (BHC) and BP Total Cost (RSSI) value (BTC). The values of BHC and BTC are measured as the path hop count and RRSI on Backward Paths from the node to the sink. Although hop count and path RSSI value are measured on FPs using HC and TC values, they are calculated similarly on BPs using BHC and BTC to inform the sink about the location of DC CHs and/or select the tree paths to forward the route reply packet (RREP) through. Besides, BTC and BHC can be used to support uni-directional communications when RREP messages (to establish a BP) are forwarded

Algorithm 1: RREQ Forwarding Algorithm

Data: Routing Table (RT)
RREQ initialisation at CH:
if *Node Role = (Flat or Parent CH)* **then**
  $RREQ \leftarrow (D, CID, srcId, BHC = 0, BTC = 0)$;
  if *TS is not available* **then** $TS \leftarrow Select(BUTS)$;
  $RREQ \rightarrow TS$;
end
RREQ forwarding at intermediate nodes (Node X):
`/* Update RT */`
if $RREQ.CID \in RT$ **then**
  if $RREQ.BHC < RT.BHC$ **then**
    `/* RT is updated */`
    $RT(CID) \leftarrow (CID, RREQ.srcId, RREQ.BHC, RREQ.BTC)$;
  else
    Discard (RREQ);
  end
else
  Insert RT(CID, srcId, BHC, BTC);
end
`/* Forward RREQ */`
if *Role = (Sink)* **then**
  Delete RREQ;
else if *(Role = CH or CM) and (X.CID ≠ RREQ.CID)* **then**
  $X \leftarrow Parent(CID)$;
  Delete RREQ;
else
  if *Count (RT) > 1* **then**
    `/* RREQ collision */`
    Delete RREQ;
  else
    $srcId \leftarrow X.Id$;
    $BHC \leftarrow BHC + 1$;
    $BTC \leftarrow BTC + RSSI(link)$;
    $RREQ \leftarrow (D, CID, srcId, BHC, BTC)$;
    $RREQ \rightarrow Select(TS \parallel BUTS)$;
  end
end

via the same FP from the CH to the sink. Wireless communication routes are formed unidirectionally due to variable communication power and signal propagation (especially in heterogeneous networks), ambient noise and/or wireless communication fading [33]. This is stated formally in Algorithm 1.

The RREQ messages are forwarded from flat and/or parent CHs to the sink. According to Algorithm 1, the CHs firstly check the availability of their TS nodes to forward the RREQ messages. The messages are forwarded if the TS nodes are available. Otherwise, a BUTS node (or neighbour node which has a smaller HC) is selected by each CH to forward the RREQ message. It reduces the overhead of forwarding RREQ as compared to message broadcasting. Each intermediate node, which receives a RREQ, would record srcId value of the message as a potential link to a desirable CH. The receiver node updates the RREQ by changing the srcId to its id, increments BHC and adds the last link RSSI value to the BTC. RREQ messages are forwarded in the same manner until the following situations arise:

1. The RREQ is received by the sink. In this case, the srcId value of the received RREQ would show a single-hop neighbour which knows a potential path with BHC hop count and BTC cost to a desirable CH.

2. The RREQ collides with another at an intermediate node. In this case, the intermediate node stops forwarding the RREQ messages(s) to the sink to conserve network resources.

3. The RREQ is received by CM/CH which has lower HC value. The cluster which receives the RREQ becomes a parent of the sending cluster.

Route reply packets (RREP) are forwarded from the sink through the shortest paths which have the minimum BHC/BTC to the desirable CHs (in terms of data). The header of RREP maintains the message sequence number, CID (the target cluster id), srcId (the next hop neighbour node) and APC (Aggregation Path hop Count). APC will be used during data aggregation routing to find the path distance (hop count) from the CH to the sink or the (closest) intermediate node which has the potential to aggregate data. The value of APC is initially zero and incremented at each intermediate node until the following situation arises (see Algorithm 2):

1. The RREP is received by a cluster (CM or CH). In this case, the cluster only forwards the RREP if it is connected to any other cluster with higher HC value (child cluster). A copy of the received RREP is forwarded to each child cluster if there is still a tie. The value of APC is set to zero in copy RREPs. The copy RREPs are forwarded until they are received by the target cluster.

2. The RREP is received by a node which already has received multiple RREQs. The node becomes an MP and the original RREP is forwarded to the target cluster. In addition, a copy of RREPs (APC=0) is forwarded to each cluster which is connected via a single or multi-hop link.

3. The RREP is received by a node which already has received a single RREQ. The node becomes an TM and the RREQ is forwarded to the destined cluster.

Data aggregation is started in a bottom-up manner from the child and/or flat clusters. The data samples are locally aggregated at the CHs. Then, the aggregated results are forward from the CHs to the parent clusters or the sink through intermediate TMs. Each CH selects the path with the minimum APC (aggregation path) to forward its data packets if multiple RREP messages are received. This means that CBA supports early data aggregation in which data packet get aggregated at the closest aggregator on the tree as soon as possible (with minimum hop count). The data results from each cluster are aggregated at MPs or parent CHs and then hierarchically forwarded until finally received by the sink. It is the minimum hop count tree in CBA that has the potential to minimise end-to-end delay. The tree branches provide the minimum hop count links between the CHs, TMs and MPs in which data packets are aggregated with minimum delay. According to Figure 3, for example, aggregated results from cluster G, C are combined at D. Then, CH (17) transmits the aggregated result to MP 39 to be combined by the result of cluster B. MP 39 is then responsible for forwarding the aggregated result of cluster G, C, D and B to the sink.

Algorithm 2: RREP Forwarding Algorithm

Data: RREP Table (PT))
RREP initialisation at Sink:
if *Role = (Sink)* **then**
  $RREP \leftarrow (CID, srcId, APC = 0, Query)$;
  $RREP(s) \rightarrow srcId(s)$;
end
RREP forwarding at intermediate nodes (Node X):
if $CID \in PT$ **then**
  Discard (RREP);/* redundant RREP */
end
Insert PT(CID, srcId, APC);
if *Role = (CH or CM)* **then**
  if *X has a child (CID)* **then**
    $RREP \leftarrow (CID, srcId, APC = 0, Query)$;
    $RREP(s) \rightarrow CID(s)$;
  else
    Delete RREP;/* a tree branch is established */
  end
else
  if *Count (RREQ)* $\not> 1$ **then**
    /* checks the number of received RREQ messages */
    $TM \leftarrow X$;
    $RREP \leftarrow (CID, srcId, APC++, Query)$;
    $RREP \rightarrow srcId$;
  else
    $MP \leftarrow X$;
    /* for the CH whose RREQ is received by the sink */
    $RREP \leftarrow (CID, srcId, APC++, Query)$;
    $RREP \rightarrow srcId$;
    /* for the CHs whose RREQ collided at MPs */
    $RREP \leftarrow (CID, srcId, APC = 0, Query)$;
    $RREP(s) \rightarrow srcId(s)$;
  end
end

4. Experimental Plan

To test and evaluate CBA, we use simulation. OMNET++ [32], is an open-source simulator for which there are implementations of MR-LEACH [14] and Directed DiFFusion [20]. It has a modelling framework called MiXiM [42] that offers detailed models of radio wave propagation, interference estimation, radio transceiver power consumption and wireless MAC protocols such as B-Mac in WSN. We used MiXiM to model, implement and test the CBA.

The experiments measure five metrics which are those typically used in the literature to evaluate the performance of data aggregation routing protocols[11]: total consumed energy, total number of delivered data samples (accuracy), average end-to-end delay, total hop count and total transmitted traffic[1]. Our aim is to show that CBA improves upon MR-LEACH and Directed DiFFusion in each of these aspects. MR-LEACH and Directed DiFFusion are used as benchmarks to evaluate our proposed protocols because they are well-known and used in a number of published papers to evaluate the proposed routing protocols [25], [40], [24], [50], [30]. Importantly, they are also already implemented in OMNET++. Moreover, these protocols have been the subject of attention by researchers in this field for a long period. Indeed, since 2002 (when the basic LEACH was proposed) until 2010, a number of researchers worked on LEACH to enhance its performance according to the advances in WSN and in requirements. The results of these research were published as modified/extended versions of LEACH [28], [14], [44], [16], [48], [34]. Among all these, MR-LEACH [14] is used in our experiments as the last version of LEACH.

1. **Total consumed energy:** represents the total amount of energy that is consumed to establish, deploy and maintain the routing infrastructure and route the data packets to the sink. Minimising energy consumption is a significant factor in maximising network lifetime. Reducing the overhead of routing infrastructure deployment/maintenance and communication are two key factors that have potential to reduce total energy consumption.

2. **Total delivered data samples (accuracy):** calculates the number of data samples collected by the sink through direct or indirect links. This metric has a (positive) correlation with data aggregation robustness. This means that maximising the number of data samples in the data aggregation procedure results in greater accuracy for the data consumer. However, data transmissions fail due to message collisions, network traffic and/or lack of routing capability. For this reason, accuracy is measured to show the ability of the routing protocols to reduce the message collision/traffic and establish efficient and reliable paths.

3. **Average hop count:** measures the capability of routing protocols in establishing optimal/shortest paths to forward data samples. It is calculated as the average intermediate hop count from the event regions to the sink for each delivered data sample. The average hop count influences the network energy consumption and data aggregation delay. The network resource consumption and data collection delay drops if data packets are delivered to the sink through minimum hop count links. On the other hand, establishing random/blind paths increases the number of intermediate hops on routes resulting in increased communication cost and delays. For this reason, efficient routing protocols aim to reduce hop count, which reduces data collection delay and saves on network resources.

4. **Average end-to-end delay:** this measures the average end-to-end delay (ETE) of data aggregation routing. ETE is measured from when the data samples leaves the source nodes until they are collected and aggregated at the sink. It is influenced by communication and computation delays such data packet reception and transmission, routing and aggregation latency. ETE has the potential to influence data accuracy and freshness. This means that data packets are expired or lose their usefulness if they are delivered to the sink late. For this reason, another routing protocol objective is to minimise the end-to-end delay (ETE).

[1] This term refers to the total sent and received network traffic

5. **Total transmitted traffic:** represents the amount of network traffic transmitted (sent/received), including control and data packets, in the entire network. Control packets are transmitted to deploy the network, establish/maintain the routing infrastructure and route the data packets, whereas data packets are forwarded to deliver the measured data samples to the sink. Routing control packets include: Hello, route request/reply, route errors and maintenance, routing updates and acknowledgements. Increasing the network traffic results in higher network resource consumption. Furthermore, end-to-end delay (ETE) rises due to increased wireless channel access and communication delays. Hence, reducing transmitted network traffic results in reduced energy consumption and data aggregation end-to-end delay.

4.1. Simulation Setup

The simulation experiments are characterised by three parameters: area size, node count and data density. These parameters let us to observe the routing protocols' behaviour, scalability and performance according to varying area size (small, medium and large), node count (sparse and dense), and data density (25, 50, 75 and 100 percents). The experiment parameters are explained as below:

1. Area size: area size influences the wireless communication type (single or multi-hop) and consequently the performance of routing. The sensor nodes can communicate with single-hop in small networks, whereas they need multi-hop when the network size increases.

2. Node count (node density): the number of nodes in the network increases to test the protocol scalability.

3. Data density: it is defined as the number of source nodes whose data match the consumer interests. This parameter allows us to observe the ability of routing protocol to aggregate and forward interesting data samples when the proportion of the source nodes is varied in the network.

The network can be deployed with three different area sizes in a two-dimensional field: small (200×200 m^2), medium (400×400 m^2) and large (800×800 m^2). This allows observation of the protocol's behaviour and performance in large and small networks. In small ones, the communication between nodes (with 75 meter radio range) is mostly single-hop, whereas they may become multi-hop as the area increases. The sensor nodes are randomly scattered in the field.

To test protocol scalability, a varying node count is considered for each area size. Deploying networks with a variable node count lets us observe the protocol's behaviour, scalability and performance in sparse and dense networks. A minimum required number of nodes ($Count_N$) to deploy a wireless network is calculated based on Equation 3[2] [53] where N is the number of nodes, R is the maximum radio range, O is the overlapping area in node radio range, and M and K are the dimensions of the network field. Accordingly, each network is set up with the minimum number of nodes required to provide a connected network in the area. The small network (200×200) is deployed with node count of 16, 32 and 64, whereas medium (400×400) and large (800×800) networks are setup with 64, 128, 256 and 256, 512, 1024 nodes respectively in order to produce similar levels of node density. The node density is calculated using Equation 4 [51].

$$Count_N = \left\lceil \frac{0.5 \times (M \times K)}{(R - (0.5 \times O))^2} \right\rceil \tag{3}$$

Each experiment features one of four proportions of source nodes which have interesting data samples (data density) to report. Each node count in each area size is allocated with four different data densities (25%, 50%, 75% and 100%). For example, the performance of data aggregation routing protocol is tested

[2]Factor 0.3125 should be updated to 0.5 in this equation to find out the number of sensor nodes which is required to fully cover a 2D grid area. This is because of that the original equation (with factor 0.3125) does not consider the area not covered which is formed between four sensor nodes placed in a 2×2 grid. Owing to this, factor 0.5 should be used as one node is required to fill the uncovered area for each four nodes.

Table 5: The setup simulation parameters

Parameters	**Simulation Time**	**Repetition**	**Network initialisation time**
Range	3600s	300	50s
Parameters	**Environmental noise**	**Node distribution model**	**Sink location**
Range	enabled	random	a single sink in the centre
Parameters	**Node battery capacity**	**MAC protocool**	**Node radio range**
Range	99999 mAh (3.3 V)	B-MAC	75 meters
Parameters	**Node count**	**Area size**	**data density**
Range	small (16, 32, 64) medium (64, 128, 256) large (256, 512, 1024)	small (200×200 m^2) medium (400×400 m^2) large (800×800 m^2)	25%, 50%, 75% and 100%

in a small network (200×200 m^2) when 25 (8 nodes), 50 (16 nodes), 75 (24 nodes) and 100 (32 nodes) percentages of sensor nodes have interesting data to report.

$$Density = \frac{N}{M \times K} \tag{4}$$

All the experiments are run 300 times. Statistical power analysis [43] was used to determine the necessary repetitions. This technique is used in experimental design to calculate the number of repetitions (sample size) using the population standard deviation and according to a given confidence degree. We run the experiments over small networks for 50 times (sample size) and measured then the standard deviation of total consumed energy for each protocol. The greatest standard deviation is used to calculate the minimum number of required repetitions as it shows the widest confidence interval. This means that the protocols (e.g CBA) which have a smaller standard deviation need a fewer number of repetitions to achieve the confidence degree as compared to the protocols (e.g DDiFF) which have greater ones. According to [36], 300 is the minimum number of required repetitions to achieve 90% confidence using the assumed population standard deviation. The setup simulation parameters are summarised in Table 5.

The performance of CBA is evaluated against two leading protocols namely MR-LEACH [14] and Directed DiFFusion [20]. Each performance parameter (total consumed energy, total number of delivered data samples (accuracy), average end-to-end delay, total hop count and total transmitted traffic) is measured in 36 different scenarios according to different area size (×3) and node count (×3), and data density (×4) in the network. According to this, three graphs will be presented in Section 5 for each performance parameter focusing on the network size (small, medium and large). Each of the graphs (e.g small network 200×200) shows the results of the protocols with three node count (e.g 16, 32 and 64 for small network) when four different percentages of nodes (25%, 50%, 75% and 100%) have interesting data to report. The next section will discuss the behaviour of the performance parameters for each protocol when node count, data density and area size change.

4.2. Isolated Nodes

Random node distribution can lead to isolated groups of nodes. A sensor node is isolated if it is not able to communicate with the sink (or any other node) either through single or multi-hop links. Isolated nodes influence the performance of routing protocols in terms of coverage and network connectivity since both are reduced by the isolated (groups of) nodes. Hence, the area under their coverage is totally disconnected to the sink and becomes out of reach for the network consumer. In addition, isolated nodes waste network resources – mainly energy – with data sample collection and computation of routes that are not used.

The minimum required number of sensor nodes to cover a $M \times K$ meter area is calculated (i.e 15 nodes for the sparsest network in the smallest area) and used then to set up the experiments using Equation 3. However, this Equation measures the minimum required number for a uniformly-placed (grid) network rather than a

random one. Therefore, we use Equation 5 [9] to calculate the probability of having isolated sensor nodes in our experiments as the nodes (except the sink which always resides in the middle) are randomly placed. R is radio range, P is the probability of no isolated nodes in a network, N is the number of (random-placed) nodes and D is network density (calculated using Equation 4). This equation shows satisfactory probabilities (i.e 98.6% for 16 nodes in $200 \times 200m^2$ area) to deploy a network with no isolated nodes according to our experiment setup.

$$R \geq \sqrt{\frac{-\ln(1 - P^{\frac{1}{N}})}{D \times \pi}} \tag{5}$$

According to [9], the probability of a network deployed with no isolated node (there is at least one link between any pair of nodes) is calculated using Equation 5. However, it does not consider the case of a group of nodes which are able to communicate with each other but are out of reach of the sink. None of the grouped nodes is isolated, but the group cannot reach the sink to report data. For this reason, we measure the number of isolated nodes which cannot hear the sink in our experiments. Figure 5 shows the total number of isolated nodes for 300 repetitions per each node count and network size. As it is observed from the figure, the number of isolated node increases when area size increases. However, isolated node count decreases when node count (network density) increases. For example, there is no isolated node when the deployed network is dense. As it can be observed, the proportion of isolated nodes for each experiment (dividing the total number of isolated nodes by 300 experiment repetitions) is too small to influence the connectivity of network in our experiments.

5. Results

This section evaluates the performance of CBA, MR-LEACH [14] and Directed DiFFusion (DDiFF) [20] based on the routing performance metrics that are described in the previous section.

5.1. Total Energy Consumption

Energy saving is a vital requirement in WSN because of the sensor node power resource constraints. One objective of designing WSN protocols is to enhance the network lifetime by reducing power consumption. This section evaluates the energy efficiency of CBA in comparison to MR-LEACH and DDiFF.

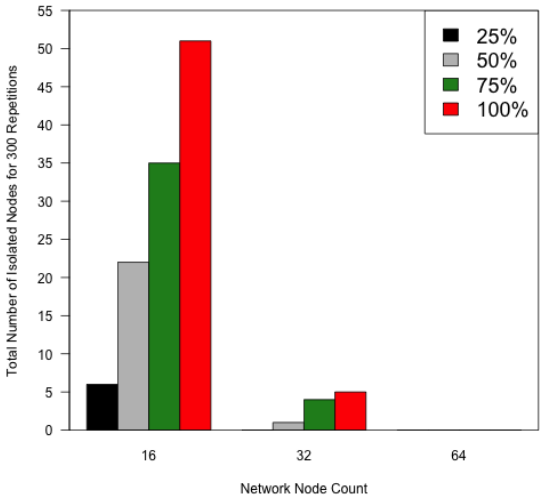
As Figure 6 shows, CBA consumes less energy to collect and aggregate data samples over a clustered network in comparison to MR-LEACH. This is for the following reasons:

1. Utilises the Hamming distance technique to form the clusters: the amount of transmitted information to form clusters is reduced as compared to MR-LEACH, which needs periodically to collect routing information to form the clusters and select the cluster-heads. The clustering cost and routing information collection is higher in MR-LEACH, especially when the node count increases, due to more overhearing and the number of node which participate in the data clustering procedure.

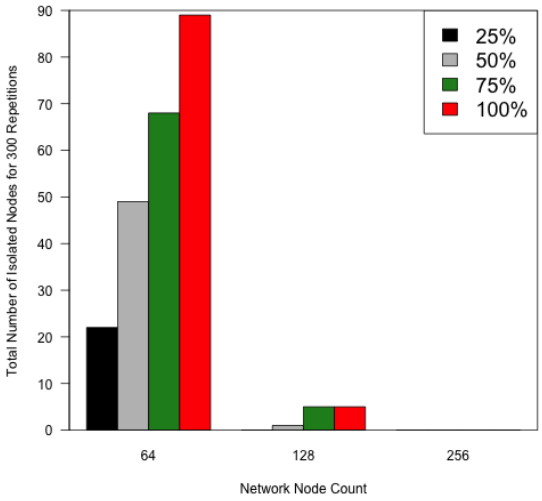
2. Forming DC clusters: this reduces the total network energy consumption (especially when the proportion of interesting source nodes is low) because the nodes only participate in the clustering procedure if they have interesting data to report. However, all nodes need to participate in clustering with an address-centric clustering approach such as MR-LEACH.

3. Avoids heuristic packet routing: CBA utilises a tree backbone to forward aggregated results from the clusters to the sink. This results in a reduction of the cost of transmitting data packets from the cluster-heads to the sink. The data packets are forwarded through shortest paths (spanning tree), which are established from the data regions to the sink. They are hierarchically aggregated through the tree infrastructure until the final result is received by the sink. Hence, the number of messages and communication hop count are reduced as compared to MR-LEACH, in which the paths are heuristically established from the data regions to the sink.

Fig. 5: The number of isolated nodes.



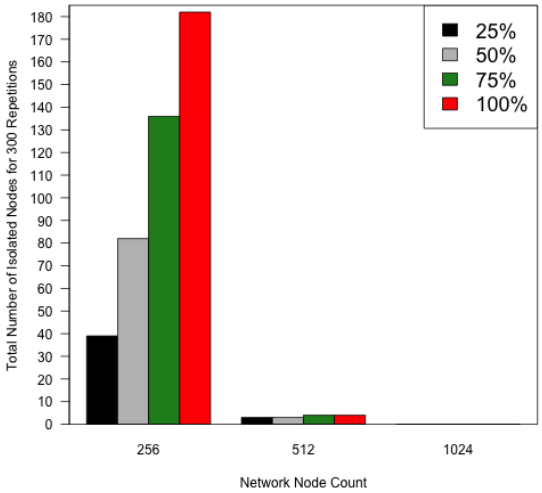
(a) small area (200×200)m^2



(b) medium area (400×400)m^2

Key points of the figures:

1. The number of isolated node is increased when area size increases.
2. Isolated node count is reduced when node count increases.



(c) large area (800×800)m^2

4. Hierarchical aggregation over tree infrastructure: data packets in CBA are hierarchically aggregated and forwarded through a tree infrastructure from the event regions to the sink. This results in a reduction of the number of data packet broadcasts, as the TMs (tree members) utilise unicast to forward the data to the sink. However, lack of such infrastructure in MR-LEACH increases the number of data transmissions as it needs more intermediate/CH nodes to broadcast/multicast the data packets.

5. Utilises parallel guided-collision: CBA utilises parallel guided-collision to reduce the overhead of establishing the tree infrastructure. This technique avoids forwarding/broadcasting useless/redundant control packets during the data aggregation infrastructure establishment phase. This also results in fewer control packets and consequently decreases the total transmitted network traffic.

Energy consumption in CBA is higher, compared to DDiFF as data density rises. DDiFF does not establish a hierarchical routing/aggregation infrastructure for data aggregation. Hence, it consumes less energy in comparison to CBA. Energy consumption in CBA rises as the number of desirable source nodes which participate in the DC clustering and tree establishment phases increases.

5.2. Total Number of Captured Data Samples (Accuracy)

Accuracy is measured as the number of interesting data samples that are collected at the sink. Data samples are reported and aggregated through flat or hierarchical infrastructures until they are received by the sink. The objective is to maximise accuracy, as it has the potential to enhance data aggregation robustness. This section evaluates the accuracy of CBA against MR-LEACH and DDiFF.

From Figure 7, it is observed that CBA outperforms DDiFF and MR-LEACH as node count and/or data density (the proportion of desirable nodes) increases in the network. It stems from the following:

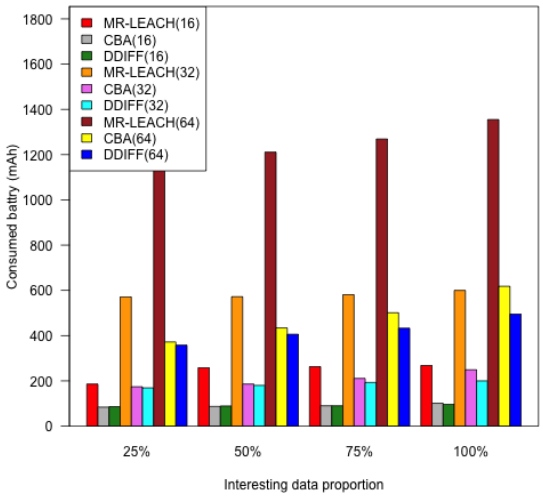
1. Clustering: it has the potential to reduce the message collision as data transmissions would be limited into inter/intra-cluster communications. Using the cluster-based infrastructure in CBA, data packets are forwarded from the source nodes to CHs through intra-cluster DC links. The CHs collect and aggregate intra-cluster data samples and then report the aggregated results to the sink. For this reason, fewer nodes (CHs) need to access wireless channels to transmit their data as compared to flat networks (i.e DDiFF) in which any source node individually forwards its data. This reduces the probability of message collision caused by communication invention especially when data density increases in the network. Owing to this, the accuracy of CBA is enhanced (in comparison to DDiFF) when node count and/or the proportion of desirable source nodes increases.

2. Formation of DC clusters: clustering the network in a DC manner has the potential to reduce message collisions in comparison to AC clustered network. DC clustering reduces cluster count and consequently the number of CHs which try to access wireless channels to forward data packets to the sink, especially when data density is low. On the other hand, the probability of message collision increases in MR-LEACH as the number of CHs increases when whole the network is divided into AC clusters without considering data content at the source nodes.

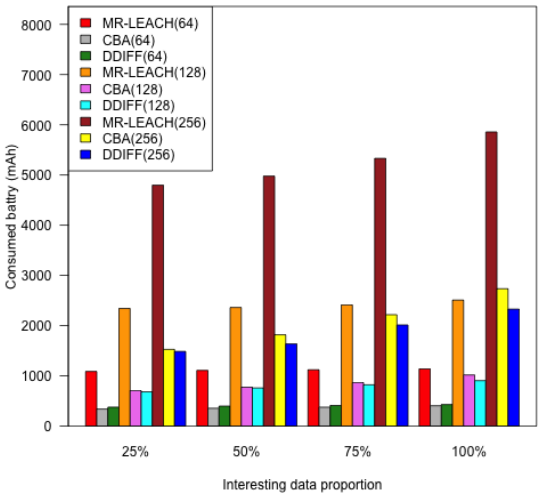
3. Use of aggregation tree to forward data packets from event regions to sink: the CHs utilise a reliable infrastructure in CBA that is established to collect, aggregate and forward data samples from the source nodes to the sink. This reduces the probability of message conflict/loss in comparison to MR-LEACH, which heuristically route the data packets from source nodes to sink. Lack of a reliable infrastructure from the event regions to the sink increases the probability of message conflict/loss, especially when node count and/or data density is high. Data packets are lost/collided if many CHs dynamically route the packets from the event regions to the sink.

The accuracy of CBA increases with network size. Utilising the tree backbone to forward the data packets from the source nodes to the sink, is the key reason for the increase in accuracy in CBA. In large networks, the probability of message conflict/loss is increased in routing protocols which heuristically route data packets (e.g MR-LEACH) as the hop count between source nodes and sink increases. Increasing the

Fig. 6: Energy consumption of client/server routing protocols.



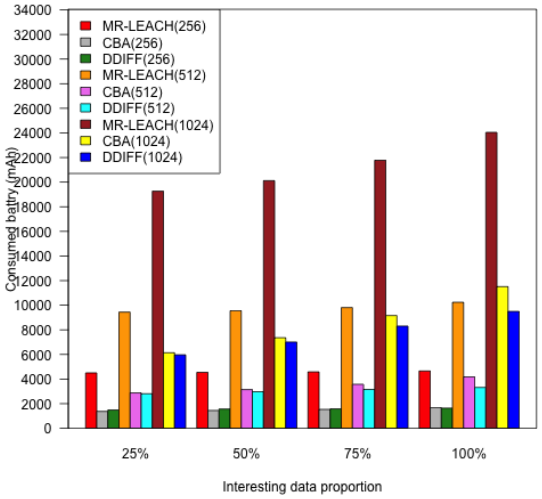
(a) small area (200×200)m^2



(b) medium area (400×400)m^2

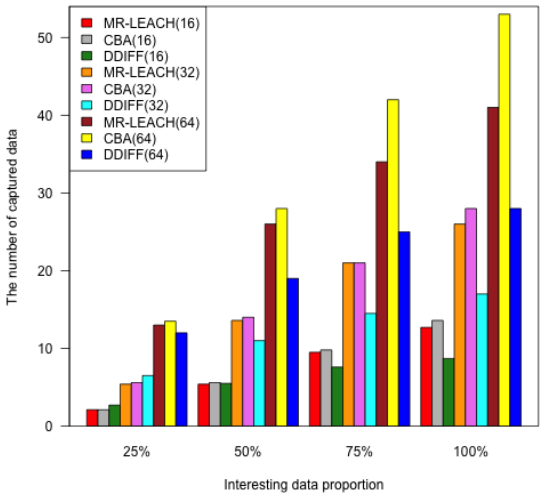
Key points of the figures:

1. CBA outperforms MR-LEACH in terms of energy consumption.
2. DDiFF outperforms CBA especially when the data density increases.

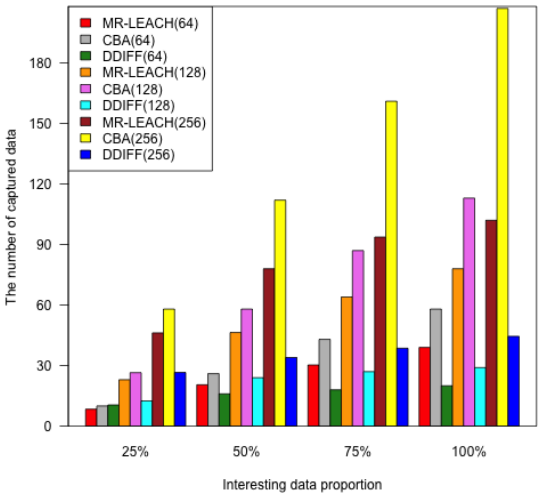


(c) large area (800×800)m^2

Fig. 7: Accuracy of client/server routing protocols.



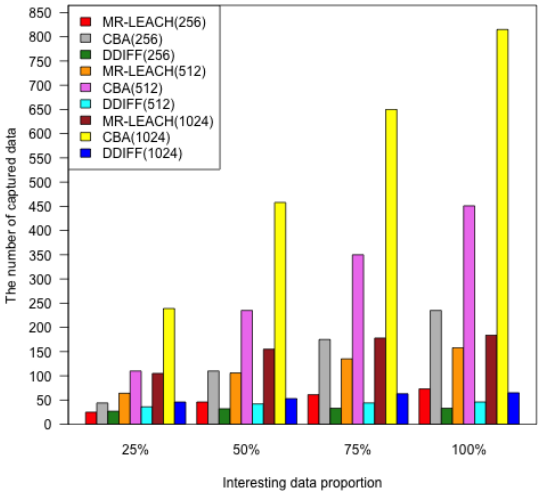
(a) small area (200×200)m^2



(b) medium area (400×400)m^2

Key points of the figures:

1. CBA outperforms the benchmark protocols in terms of accuracy.
2. CBA increases accuracy when network deployed is dense and large.



(c) large area (800×800)m^2

hop count between the event regions and sink increases the number of intermediate nodes which participate in routing data packets. Hence, data messages have a higher chance of failure due to collision, as a greater number of nodes try to access the wireless channels to forward them. Moreover, the probability of message loss goes up with the number of relay nodes – these heuristically route the data packets from the event regions to the sink. On the other hand, CBA utilises a tree backbone in which data packets are forwarded through reliable and/or shortest path to the sink. The probability of message collisions/loss is reduced as the number of relay nodes is limited to the tree members (TMs). Each node which resides in the tree uses unicast communication instead of broadcast/multicast to collect the data packets from child nodes and then forwards the aggregated result to the parent nodes. Hence, routing the data packets via the tree infrastructure results in a reduction of communication and consequently message collisions/loss.

5.3. Average Hop Count

Average hop count is measured as the average path length from the event regions to the sink to report each data sample. The objective of a routing protocol is to reduce the average hop count as much as possible. The average hop count influences ETE and energy consumption. This means that increasing the average hop count results in increasing the number of intermediate nodes which participate in sending/receiving the data packets.

According to Figure 8, CBA reduces the average hop count in comparison to MR-LEACH and DDiFF. Data-centric clustering and utilising the spanning tree for data aggregation routing are the two techniques that affect the average hop count in CBA. DC clustering and spanning tree reduce path hop count as below:

1. DC clustering: CBA forms DC clusters to collect and aggregate local data samples according to the sink interests. Data samples are forwarded via intra-cluster links to the CHs for aggregation. Thus, data packets do not need to traverse long paths until they are heuristically aggregated at intermediate nodes or the sink, as they are aggregated at CHs. On the other hand, data aggregation is not guaranteed in address-centric CHs as AC clusters (intra-cluster links) are formed based on the address of nodes and not the content of data. In AC clusters (e.g MR-LEACH), a node performs data aggregation if it resides on a route through which multiple data packets are forwarded. Thus, data packets need to traverse longer paths (especially when data density is low) until they are aggregated at AC aggregator nodes, in comparison to DC clusters in which data aggregation is performed at each CH. Hence, the average hop count of MR-LEACH is higher than CBA.

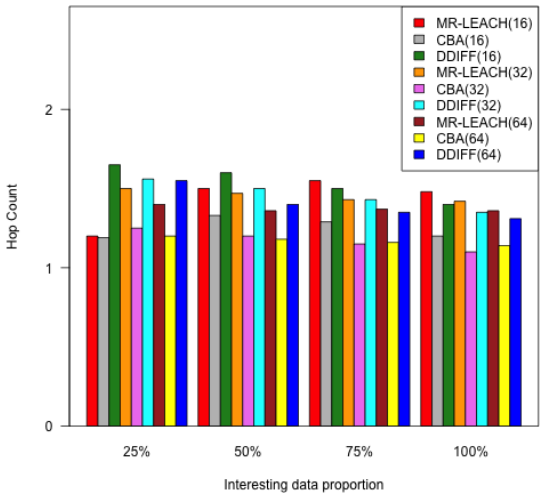
2. Use of a tree backbone: the aggregated result of each cluster is forwarded to the sink through a spanning tree which is formed by the shortest paths (using HC value) between the event regions and the sink. Data packets are hierarchically forwarded and aggregated until the final result is received by the sink. Hence, the data packets traverse minimum hop count paths from the event regions to the sink. On the other hand, the lack of a such the infrastructure in DDiFF and MR-LEACH to aggregate data packets is the reason for its higher average hop count. This means that data packets need to traverse longer paths until they are aggregated. As a result, the average hop count in DDiFF is more than CBA.

5.4. Average End-to-End Delay

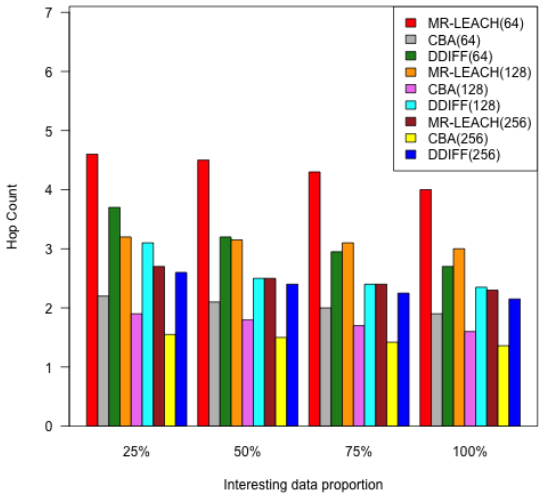
Average end-to-end delay (ETE) is measured as the average time from when a packet leaves the source node until it is received by the sink. The objective of data aggregation routing protocols is to reduce the average delay as it enhances data freshness. Reporting the data samples to the sink as quickly as possible provides the data consumer with fresh data for further analysis. Hence, the data consumer can make better decisions using the collected data.

As Figure 9 shows, CBA reduces ETE in comparison to DDiFF and MR-LEACH especially when the area size and/or node count increases. It is because of three reasons as below:

Fig. 8: Average hop count of client/server routing protocols.



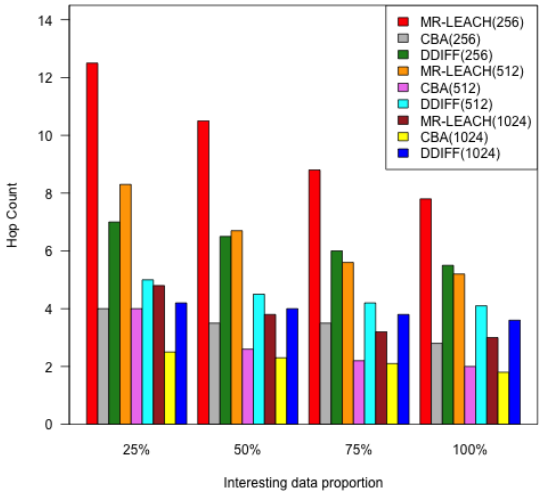
(a) small area (200×200)m^2



(b) medium area (400×400)m^2

Key points of the figures:

1. The average hop count is reduced when the node count increases.
2. CBA forwards data packets through shorter paths as compared to the benchmark protocols.



(c) large area (800×800)m^2

1. Establishing shortest paths to report the data packets: CBA forwards data packets through a spanning tree, so data packets traverse minimum hop count routes until arrived at the sink. The communication delay is reduced as a minimum number of intermediate nodes participate in routing data packets. On the other hand, MR-LEACH heuristically routes the data packets from the CHs to the sink. It increases the path hop count (according to Figure 8) which results in higher ETE in MR-LEACH, especially when the network deployed is large and dense.

2. Hybrid routing: CBA utilises a hybrid routing scheme in which data packets are reactively forwarded through intra-cluster links, while they are routed proactively on the tree backbone. Each CH collects data samples from its CMs and then forwards the aggregated result to the sink through the spanning tree. The TMs (tree member nodes) do not need to collect routing information to route the data packets as they already know the required information. Hence, end-to-end delay is reduced as compared to MR-LEACH in which CHs reactively route the data packets from the event regions to the sink.

3. Early data aggregation[3]: data packets are aggregated in CBA as early as possible at the CHs and/or TMs. This results in reduction of ETE as compared to MR-LEACH and DDiFF which do not use early data aggregation. Early data aggregation has the potential to reduce the number of data transmissions. This means that a fewer number of data packets needs to be forwarded when they are aggregated as early as possible (in terms of hop count). This results in a reduction of network node requests to access the wireless channels and consequently decreases idle-listening and access time. However, lack of early data aggregation increases the access time and consequently higher ETE in protocols such as DDiFF. Forwarding data packets from each source nodes to the sink increases the number of sensor nodes which try to access the wireless channels in DDiFF. This results in increased access time delay and consequently increased ETE as node count and/or area size increases.

5.5. Total Transmitted Traffic

Total transmitted traffic is measured as the total amount of data/control packets (sent/received) transmitted by the network nodes. Data packets are used to report the ambient data, whereas the control packets are transmitted to deploy the network, form the routing infrastructure and/or control wireless channels. The routing performance is determined by transmitted traffic due to the following reasons:

1. Increasing the network traffic results in reduction of network lifetime: both sender and receiver sides of communication consume energy to transmit network packets. Hence, increasing network traffic results in higher energy consumption and consequently reduced network lifetime.

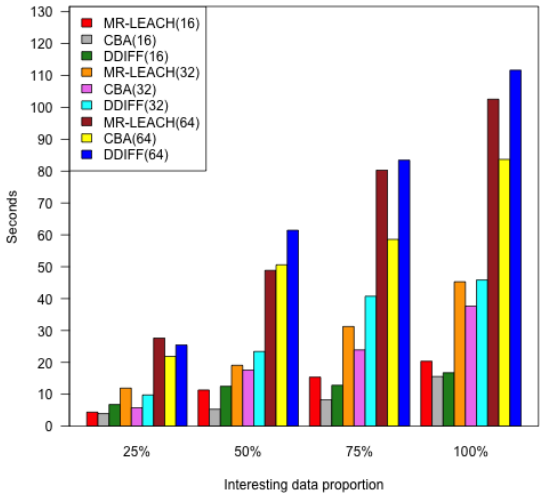
2. Increasing network traffic has the potential to reduce routing throughput: it increases network congestion and message failure.

3. Increasing network traffic increases ETE: data packets need to be queued until the wireless channel become available (idle listening) if network traffic increases. This means that increasing network traffic increases waiting time for the sensor nodes to access the wireless channels for communication. The packets queue for a longer period until the wireless channel become available. Moreover, increasing network traffic increases data packet failures and consequently delivery time. The probability of message failure is increased due to message collisions when network traffic rises. Hence, nodes need to re-transmit data packets until they are correctly delivered to the destination. Data collection ETE rises when the network is dense.

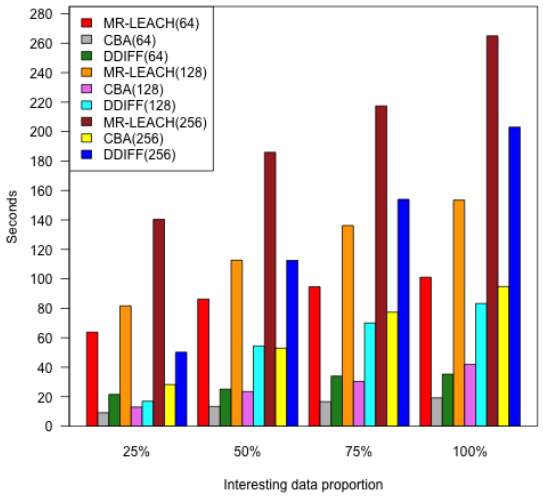
As Figure 10 shows, CBA reduces the total transmitted traffic compared to MR-LEACH. Reducing clustering overhead and utilising the parallel guided-collision to establish the tree backbone are two key techniques that effect the reduction of network traffic in CBA.

[3]this term refers to aggregating data packets as soon as possible in terms of hop count

Fig. 9: End-to-end delay (ETE) of client/server routing protocols.



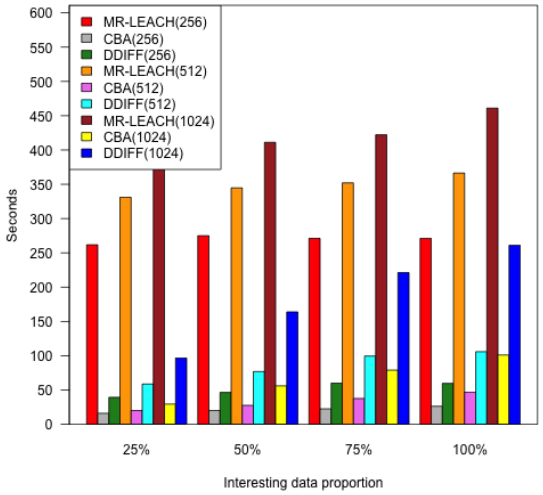
(a) small area (200×200)m^2



(b) medium area (400×400)m^2

Key points of the figures:

1. CBA outperforms the benchmark protocols in terms of ETE.



(c) large area (800×800)m^2

1. Reduces clustering overhead: CBA dynamically groups the desirable source nodes into DC clusters according to the sink queries. This results in a reduction of network traffic for two reasons: (i) reducing the cluster count in CBA: the number of clusters formed is reduced in CBA compared to MR-LEACH, when data density is low in the network. In CBA, the clusters are formed if they are required to collect and aggregate interesting data samples for the sink. On the other hand, MR-LEACH partitions the entire network into AC clusters without considering the content of available data at the source nodes. All nodes need to collect and forward the required routing information to form the clusters. For this reason, a fewer number of clusters are formed in CBA compared to MR-LEACH, especially when the number of desirable source nodes (data density) is low. (ii) Periodical clustering and CH selection in MR-LEACH: the CMs need to transmit control packets periodically to select the new CHs in MR-LEACH. This results in a significant increase in the number of transmitted control packets in MR-LEACH especially when the network is large and dense.

2. Parallel guided-collision technique: utilising parallel guided-collision during the aggregation tree construction has the potential to reduce the number of control packets. The nodes avoid forwarding useless and/or redundant messages from the sensor nodes to establish the Backward Paths (BPs) to the sink. On the other hand, the route requests are heuristically forwarded by the nodes in MR-LEACH to find/establish the required paths to route the data packets from the event regions to the sink. This results in more network traffic, especially as area increases.

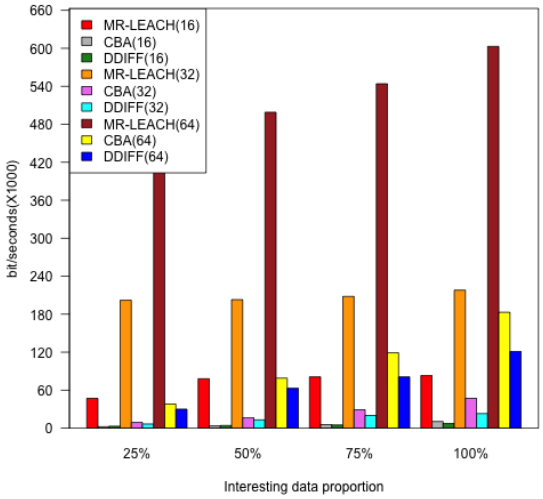
Less traffic is transmitted in DDiFF compared to CBA and MR-LEACH, especially when the number of interesting source nodes and/or node count increases. DDiFF does not use a specific hierarchical infrastructure to aggregate and/or route the data packets. Hence, the sensor nodes do not need to forward control packets to establish/maintain the infrastructure to aggregate and forward packets. On the other hand, CBA and MR-LEACH need to forward a greater number of control packets to construct data aggregation infrastructures. This results in more network traffic (and overhearing) especially when the network is dense.

6. Conclusion and Future Works

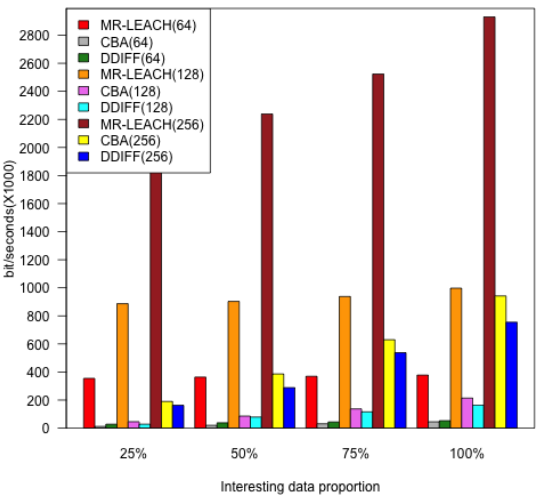
CBA partitions the network into a set of DC clusters and then establishes a tree backbone to forward and aggregate the results of each cluster to the sink. The proposed protocol aims to maximise energy efficiency and data aggregation accuracy, and minimise end-to-end delay. According to the results, a satisfactory performance of CBA is observed that satisfies its objectives compared to the MR-LEACH and DDiFF routing protocols. Dynamic data-centric clustering gives CBA the ability to collect and aggregate desirable data samples regardless of their distribution model and/or type heterogeneity. This means that CBA has the potential to work over WSN in which sensor nodes are equipped with multiple sensing modules (e.g. TelosB mote) to measure a range of ambient events that may be distributed in the sensing field in either RS or ER model. CBA reduces energy consumption as it forms clusters using the lightweight Hamming distance technique. This avoids involving sensor nodes whose data is not interesting for the data consumer. Moreover, utilising the collision guided technique reduces the number of control packets and consequently decreases energy consumption during the routing infrastructure establishment phase. The end-to-end delay of data aggregation is reduced in CBA by avoiding random routing and establishing shortest paths (minimum hop count) from the data regions (clusters) to the sink. In addition, reducing routing delay by utilising proactive routing over the tree backbone for data packets helps in reducing ETE. The accuracy of data aggregation is enhanced in CBA as the number of data packet delivery is increased. Reducing the message collision/loss due to reducing network traffic, forming data centric clusters and establishing a network backbone to route data packets, enhances data delivery and consequently accuracy in CBA.

In future, the performance of CBA can be improved further if synchronisation is considered during data aggregation routing. Data packets are aggregated at the aggregator nodes and forwarded until received by the sink. However, reporting data samples without time synchronisation reduces the effectiveness of data aggregation. This means that the data aggregator nodes miss collecting and aggregating data packets which are received late. In this case, data packets are forwarded without aggregation to the sink. This results in

Fig. 10: Total transmitted traffic of client/server routing protocols.



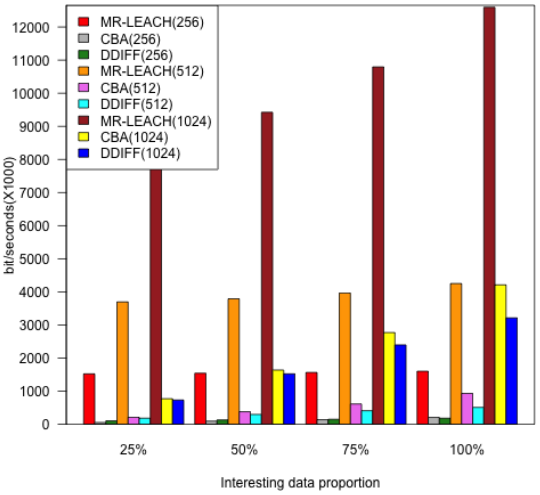
(a) small area (200×200)m^2



(b) medium area (400×400)m^2

Key points of the figures:

1. CBA transmits less network traffic as compared to MR-LEACH.
2. DDiFF transmit less network traffic as compared to CBA when data density in the network increases.



(c) large area (800×800)m^2

higher cost of data collection, as the number of data packets is not as low as it might be. CBA partially resolves this issue using BHC (Backward path Hop Count) values which let the aggregator nodes (MPs of CHs) know the distance (in terms of hop count) to the CHs which have interesting data to report (or their RREQ messages are already received). In consequence, the aggregator nodes wait for a minimum required period, according to the link hop count to source CHs, to receive and aggregate the data packets. However, this technique needs to be optimised as the CHs are hierarchically interconnected to other child CHs whose data packets may take longer to be received for aggregation.

Extending CBA to support mobile sensor nodes (MWSN) is an issue that also needs to be addressed as future work. In MWSN, the establishment/maintenance cost of routing infrastructure (the spanning tree in CBA) is increased due to the frequent network topology changes caused by node mobility. Social networking [35] is a potential technique to reduce the update cost of routing infrastructure according to topology changes, especially when the nodes are mobile and/or the network is highly dynamic [13], [1]. Social networking patterns such as content-based relations (or common interest relationships) [12] can be used by the disconnected nodes (caused by mobility and/or topology change) to (re-)join the routing infrastructure (i.e spanning tree). Using this technique, the disconnected nodes would firstly try to contact sensor nodes which have a better communication history in terms of frequency and/or duration with the nodes residing on the infrastructure (TMs). This would increase the probability of relaying data messages from the disconnected nodes to the available nodes which reside on the routing infrastructure, according to the communication histories which show previous an/or potential connections. Hence, the disconnected nodes join the routing infrastructure more quickly and by transmitting fewer control packets.

7. References

References

[1] Aggarwal, C. C., Abdelzaher, T. F., 2011. Social Network Data Analytics. Springer, Ch. Integrating Sensors and Social Networks, pp. 379–412.
[2] Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E., 2002. Wireless sensor networks: a survey. Computer Networks 38, 393–422.
[3] Al-Karaki, J. N., Kamal, A. E., 2004. Routing techniques in wireless sensor networks: A survey. Ieee Wireless Communications 11(6), 6–28.
[4] Anisi, M. H., Abdullah, A. H., Razak, S. A., September 16-18, 2011. Efficient data aggregation in wireless sensor networks. International Conference on Future Information Technology (ICFIT'11), Singapore, Singapore 13, 305–310.
[5] Arasu, A., Ganti, V., Kaushik, R., 2006. Efficient exact set-similarity joins. the 32Nd International Conference on Very Large Data Bases (VLDB'06), Seoul, Korea, september 12-15, 918–929.
[6] Ardakani, S. P., Padget, J., Vos, M. D., 2014. Hrts: A hierarchical reactive time synchronization protocol for wireless sensor networks. Ad Hoc Networks 129, 47–62.
[7] Ares, B. Z., Fischione, C., Johansson, K. H., 2007. Wireless Sensor Networks. Vol. 4373 of Lecture Notes in Computer Science. Springer Berlin/Heidelberg, Delft, The Netherlands, Ch. Energy consumption of minimum energy coding in CDMA wireless sensor networks, pp. 212–227.
[8] Basurra, S. S. A., October 2012. Collision guided routing for ad-hoc mobile wireless networks. Ph.D. thesis, Department of Computer Science, University of Bath.
[9] Bettstetter, C., 2002. On the minimum node degree and connectivity of a wireless multihop network. The 3rd ACM International Symposium on Mobile Ad Hoc Networking (MobiHoc '02), Lausanne, Switzerland, June 9-11, 80–91.
[10] Biswas, P. K., Qi, H., Xu, Y., 2008. Mobile-agent-based collaborative sensor fusion. Information Fusion 9 (3), 399–411.
[11] Boulis, A., Ganeriwal, S., Srivastava, M. B., 2003. Aggregation in sensor networks: an energy/accuracy trade-off. Ad Hoc Networks 1, 317–331.
[12] Daly, E., Haahr, M., 2007. Social network analysis for routing in disconnected delay-tolerant manets. the 8th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc 07, Montreal, Quebec, Canada, September 9-14, 32–40.
[13] Dinh, T. N., Xuan, Y., Thi, M. T., 2009. Towards social-aware routing in dynamic communication networks. 28th International Performance Computing and Communications Conference, IPCCC 2009, , Phoenix, Arizona, USA, 14-16 December, 161–168.
[14] Farooq, M. O., Dogar, A. B., Shah, G. A., 2010. Mr-leach: Multi-hop routing with low energy adaptive clustering hierarchy. Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM 2010), Venice/Mestre, Italy, July 18 - 25, 262–68.
[15] Haslett, C., 2008. Essentials of radio wave propagation. Cambridge University Press.
[16] Heinzelman, W. B., Chandrakasan, A. P., Balakrishnan, H., 2002. An application specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications 1(4), 660–70.
[17] Heinzelman, W. R., Chandrakasan, A., Balakrishnan, H., 2000. Energy-efficient communication protocol for wireless microsensor networks. The 33rd Hawaii International Conference on System Sciences (HICSS'00), the Island of Maui, 4-7 January, 3005–3014.

[18] Henderson, W. D., Tron, S., 2006. Verification of the minimum cost forwarding protocol for wireless sensor networks. 11th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Prague, Czech Republic, September 20-22, 194–201.
[19] Hu, F., Cao, X., May, C., 2005. Optimized scheduling for data aggregation in wireless sensor networks. International Symposium on Information Technology: Coding and Computing (ITCC 2005), Las Vegas, Nevada, USA, April 4-6 2, 557–561.
[20] Intanagonwiwat, C., Govindan, R., Estrin, D., 2000. Directed diffusion: A scalable and robust communication paradigm for sensor networks. The 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00), Boston, Massachusetts, August 6-11, 56–67.
[21] Khan, A., Tamim, I., Ahmed, E., Awal, M. A., 2012. Multiple parameter based clustering (mpc): Prospective analysis for effective clustering in wireless sensor network (wsn) using k-means algorithm. Wireless Sensor Network 4, 18–24.
[22] Krishnamachari, B., Estrin, D., Wicker, S., 2002. Modelling data-centric routing in wireless sensor networks. The 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), New York, USA, June 23-27 2(4), 1–11.
[23] Kulik, J., Rabiner, W., Balakrishnan, H., 1999. Adaptive protocols for information dissemination in wireless sensor networks. The 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom' 99), Seattle, Washington, August 15-20, 174–185.
[24] Lee, M., Wong, V. W., 2005. An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks. IEEE Pacific Rim Conference on Communications, Computers and signal Processing (PACRIM'05), Victoria, B.C., Canada, August 24-26, 300–303.
[25] Lindsey, S., Raghavendra, C. S., Sivalingam, K. M., 2002. Data gathering algorithms in sensor networks using energy metrics. IEEE Transactions on Parallel Distributed System 13 (9), 924–935.
[26] Liu, K.-W. F. S., Sinha, P., August 2007. Structure-free data aggregation in sensor networks. IEEE transactions on mobile computing 6 (8), 929–942.
[27] Liu, X., 2012. A survey on clustering routing protocols in wireless sensor networks. Sensors 12, 113–153.
[28] Loscr, V., Morabito, G., Marano, S., 2005. A two-levels hierarchy for low-energy adaptive clustering hierarchy (tl-leach). 62nd Vehicular Technology Conference (VTC' 2005-fall), September 25-28, 1809–13.
[29] Madden, S., Franklin, M. J., Hellerstein, J. M., Hong, W., 2002. Tag: A tiny aggregation service for ad-hoc sensor networks. Fifth Symposium on Operating Systems Design and implementation (OSDI 02), Boston, MA, USA, December 9 - 11, 131–146.
[30] Manjeshwar, A., Agrawal, D. P., 2001. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. 15th International Parallel and Distributed Processing Symposium (IPDPS), San Francisco, USA, April 23-27 3, 2009 – 2015.
[31] Mao, J., Wua, Z., Wuc, X., February 2007. A TDMA scheduling scheme for many-to-one communications in wireless sensor networks. Computer Communications 30 (4), 863–872.
[32] OMNET++, 2012. Omnet++ simulator. `http://www.omnetpp.org/`, Retrieved (March 2012).
[33] Ramasubramanian, V., Chandra, R., Mosse, D., 2002. Providing a bidirectional abstraction for unidirectional ad hoc networks. The 21st Annual Joint Conference of the IEEE Computer and Communications Societies, New York, USA, June 23-27, 345–354.
[34] Ran, G., Zhang, H., Gong, S., 2010. Improving on leach protocol of wireless sensor networks using fuzzy logic. Journal of Information & Computational Science 7(3), 767775.
[35] Scott, J., 2000. Social Network Analysis: a handbook. SAGE Publications Ltd.
[36] Serqeant, E., 2014. Sample size to estimate a single mean with specified precision. `http://epitools.ausvet.com.au/content.php?page=1Mean\&Stdev=45\&Conf=0.95\&Error=20`, Retrieved (December, 2014).
[37] Sharaf, M. A., Beaver, J., Labrinidis, A., Chrysanthis, P. K., 2003. Tina: a scheme for temporal coherency-aware in-network aggregation. Third ACM International Workshop on Data Engineering for Wireless and Mobile Access, MobiDE 2003, San Diego, California, USA, September 19, 69–76.
[38] Sohraby, K., Minoli, D., Znati, T., 2007. Wireless Sensor Network Technology, Protocols and Applications. John Wiley & Sons, Inc.
[39] Solis, I., Obraczka, K., 2006. In-network aggregation trade-offs for data collection in wireless sensor networks. International Journal of Sensor Network (IJSNet) 1 (3/4), 200–212.
[40] Tan, H. O., Körpeoğlu, I., 2003. Power efficient data gathering and aggregation in wireless sensor networks. SIGMOD Record 32, 66–71.
[41] Uthansakul, P., Bialkowski, M. E., Durrani, S., Bialkowski, K., Postula, A., 2005. Effect of line of sight propagation on capacity of an indoor mimo system. IEEE Antennas and Propagation Society International Symposium 2005, 3-8 July, Washington, DC, 707–710.
[42] Viklund, A., 2013. Mixim code. `http://mixim.sourceforge.net/index.html`, Retrieved (December, 2013).
[43] Walker, I., 2013. Switching to r: A guide for the behavioural sciences. Tech. rep., University of Bath.
[44] Xiangning, F., Yulin, S., 2007. Improvement on leach protocol of wireless sensor network. International Conference on Sensor Technologies and Applications (SensorComm 2007), Valencia, Spain, October 14-20, 260–4.
[45] Xiong, Z., Liveris, A. D., Cheng, S., September 2004. Distributed source coding for sensor networks. IEEE signal processing magazine 21 (5), 80–94.
[46] Xu, J., Liu, W., Lang, F., Zhang, Y., Wang, C., 2010. Distance measurement model based on rssi in wsn. Wireless Sensor Network 2(8), 606–611.
[47] Xu, Y., Qi, H., 2008. Mobile agent migration modeling and design for target tracking in wireless sensor networks. Ad Hoc Networks 6 (1), 1–16.
[48] Yassein, M. B., Al-zou'bi, A., Khamayseh, Y., Mardini, W., 2009. Improvement on leach protocol of wireless sensor network (vleach). Journal of Digital Content Technology and its Applications 3(2), 132–136.
[49] Ye, F., Chen, A., Lu, S., Zhang, L., 2001. A scalable solution to minimum cost forwarding in large sensor networks. The 10th International Conference on Computer Communications and Networks, Scottsdale, Arizona, USA, October 15-17, 304–309.

[50] Younis, O., Fahmy, S., 2004. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE TRANSACTIONS ON MOBILE COMPUTING 3(4), 366–79.

[51] Youssef, M. A., Youssef, A., Younis, M. F., December 2009. Overlapping multihop clustering for wireless sensor networks. IEEE transactions on parallel and distributed systems 20 (12), 1844–1856.

[52] Yuan, W., Krishnamurthy, S. V., Tripathi, S. K., 2003. Synchronization of multiple levels of data fusion in wireless sensor networks. the Global Telecommunications Conference, (GLOBECOM '03. IEEE), San Francisco, USA, December 1-5, 221–225.

[53] Zaidi, S. A. R., Hafeez, M., Khayam, S. A., Mclernon, D., Ghogho, M., Kim, K., 2009. On minimum cost coverage in wireless sensor networks. The 43rd Annual Conference on Information Sciences and Systems (CISS 09), Johns Hopkins University, Baltimore, MD, March 18-20, 213–218.

[54] Zhu, X., Zhang, W., 2010. A mobile agent-based clustering data fusion algorithm in wsn. International Journal of Electrical and Computer Engineering 5(5), 227–280.