A Dynamic Data Aggregation Scheme for Wireless Sensor Networks

Supriyo Chatterjea and Paul Havinga
Department of Computer Science, University of Twente
P.O. Box 217, 7500 AE Enschede, the Netherlands
Phone: +31 53 489 4619 Fax: +31 53 489 4590

URL: http://www.cs.utwente.nl/~supriyo
E-mail: {supriyo|havinga}@cs.utwente.nl

Abstract— **Wireless sensor networks are formed of tiny, energy-constrained sensor nodes that could be mobile and may be deployed in unfamiliar environments in large numbers. Considering these unique characteristics, our network architecture is modelled around a data-centric approach that allows us to make use of in-network processing and data aggregation which in turn helps to maximize network lifetime. This paper suggests methods to improve network efficiency by combining Directed Diffusion [2] with clustering and by introducing a more elaborate data aggregation scheme. Our data aggregation scheme allows nodes to process data collected from sensors and subsequently aggregate the data even in completely unfamiliar environments by including entire query definitions within interest messages. We also describe certain novel design features such as interest transformation, layered data aggregation and dynamic data aggregation points all of which would improve overall system performance.**

Keywords— **Wireless Sensor Networks; Data Aggregation, Directed Diffusion, Clustering**

I. INTRODUCTION

With the rapid progress made in the improvement of microprocessors, semiconductors and wireless communication technologies in recent years, it is envisioned that in the near future, wireless sensor networks will permeate through the very fabric of our daily lives thus bringing out the true essence of Mark Weiser's concept of ubiquitous computing [4]. These wireless sensor networks are formed of tiny, energy-constrained sensor nodes that could be mobile and may be deployed in unfamiliar environments in large numbers. Taking these unique characteristics into account, our network architecture presented in this paper is modelled around a data-centric approach that allows us to make use of in-network processing and data aggregation which in turn helps to maximize network lifetime.

Our research primarily focuses on the distributed data management aspects of wireless sensor networks by building up on prior work done on Directed Diffusion [3]. Firstly, we combine Directed Diffusion with clustering to prevent flooding during interest propagation where every single node in a sensor network takes part in the transmission of interest messages. Clustering allows transmission to be limited to clusterheads and gateway nodes.

Next, we present a new format of the interest message which allows nodes to interpret unfamiliar queries so that they can even operate in environments which they were not originally designed for at the time of network deployment. The interest message of every new query initiated by a node contains both the query and a detailed definition of the query. The definition not only mentions all the fundamental components that are required but also describes the operations that need to be performed on these components in order to generate a response that can service the query. Thus nodes need not have prior knowledge of the queries that they would encounter. This would also help to free up memory resources as nodes would not have to maintain fixed databases containing predefined queries.

This new format of the interest message also leads to the development of certain unique features such as *layered data aggregation*, *interest transformation* and *dynamic data aggregation points* all of which would eventually lead to better system performance. Layered data aggregation allows the response to a query to be formulated gradually by aggregating data in small steps thus performing gradual data reduction as the data propagates from the node servicing the query to the node which initiated the query. This feature causes data reduction to occur along the initial stages of the data propagation path thus minimising transmissions. The "interest transformation" feature takes advantage of the existing knowledge of queries possessed by a node in order to reduce unnecessary processing and memory usage. Dynamic data aggregation points ensure that as nodes generating data messages move throughout a network, the nodes chosen to perform aggregation are also changed so that aggregation is always performed as close to the data source as possible. Our design also includes a mechanism to improve the robustness of the network by allowing intermediate nodes along a data propagation path to try and *heal* the network or re-establish communication the moment a certain node along the path fails.

Since we use Directed Diffusion as the principle foundation in the work presented in this paper, for the sake of completeness, we first give a brief overview of Directed Diffusion and then go on to briefly describe the clustering scheme. This is followed by a description of the interest format and how the interest messages propagate. We also provide a detailed description of how interests are transformed, how data is aggregated in a layered manner and how data aggregation points shift around the network as the locations of the source nodes change and explain how these features would lead to improved network performance.

This is work is performed as part of the NWO funded CONSENSUS project [6] and European EYES project (IST-2001-34734) [7] on self-organising and collaborative energy-efficient sensor networks. It addresses the convergence of distributed information processing, wireless communication and mobile computing.

II. Background

A. Directed Diffusion

Directed diffusion is a data-centric communication paradigm where a sink (node requesting a service) sends out a request for data by broadcasting an *interest* to its neighbouring nodes. An interest refers to a named description of a service that a sink node requires. The neighbours subsequently broadcast the interest to their respective neighbours and this process is repeated until a "source" node, which is capable of servicing the request comes across the interest. As interests *diffuse* throughout the network, a node that receives an interest from a neighbouring node forms a gradient pointing to the sending node that indicates the direction in which data from a source node will eventually flow. The source node then generates *data* messages using its sensors which propagate back to the sink following the gradients formed along the paths through which the interests originally traversed. Every sink that receives data messages from more than one neighbour, reinforces a particular neighbour so that subsequent data messages arrive only from the chosen neighbour. This chosen neighbour also performs the same procedure on its neighbouring nodes it received a data message from. This process is repeated until data messages propagate only along the reinforced path from source to sink. If the quality of data transmission from a certain neighbour deteriorates, a node can opt to negatively reinforce its current under-performing neighbour and reinforce another better-performing neighbour instead, in order to cope with varying network dynamics.

The Directed Diffusion mechanism can be divided into two main phases, namely the *interest propagation* phase, where interest messages flow from the sink to the source and the *data propagation* phase where data messages flow from the source to the sink. In the latter phase, data messages initially flow along multiple paths towards the sink but as time progresses, the sink reinforces only a number of paths (depending on data quality) thus eventually reducing the total number of nodes that are involved in the transmission process.

However, in the initial phase, interest messages are spread throughout the network using flooding. Flooding is a highly expensive operation with respect to energy consumption as a large number of nodes in the network would have to be involved in the transmission of data. Of all the operations a node can perform, transmission is the most energy intensive operation. In the domain of wireless sensor networks, where conservation of energy resources are of paramount importance, it is imperative to reduce energy consumption at every level of the communication protocol. Thus, in order to alleviate this problem of high energy consumption during the spreading of interest messages, it is proposed that a clustering mechanism be introduced to limit the number of nodes that are actively involved in the transmission of messages.

B. Clustering

Cluster-based control structures allow more efficient use of resources when controlling large-scale, ad-hoc sensor networks that are highly dynamic. A clustered structure can also help to make the topology appear more static and this in turn alleviates certain effects of mobility.

A cluster is typically made up of one clusterhead and all nodes that are controlled by it are referred to as members. Member nodes can either be gateway nodes or regular nodes. Inter-cluster communication between adjacent clusters is maintained through the gateway nodes. There are two types of gateway nodes, a direct gateway and a distributed gateway. If a node is in direct transmission range of two (or more) clusterheads, we refer to it as a direct gateway. A node that can communicate with nodes from a different cluster, but cannot reach the corresponding clusterhead, is called a distributed gateway. For a distributed gateway, always two nodes are needed for connectivity between the two corresponding clusterheads.

By using clustering during the initial phase of interest propagation, our design ensures that only the clusterheads and gateway nodes are actively involved in the transmission of interest messages. Regular nodes generally remain silent unless they are capable of servicing a certain request.

III. Design Requirements

While we agree with the fact that unlike general-purpose networks like the LAN used in an office, sensor networks are generally task-specific, we also feel there is a possibility that in certain instances, as sensor networks, which may be part of the ambient intelligence framework, move from one environment to another, the nodes may encounter queries which were not pre-programmed into the nodes at the time of deployment. Although nodes may be faced with unfamiliar queries, the nodes may still have the physical capability to service these queries and it is therefore imperative to devise methods which are capable of taking

maximum advantage of the resources at hand instead of only using the network for the single purpose it was originally designed for. The following examples describe some scenarios which highlight the design requirements of the communication protocol.

Let us consider the case where a few persons are wearing watches that are capable of monitoring the users' pulse rates and the temperature and relative humidity of the surrounding environment. The watch is also equipped with a wireless transceiver. While on the hiking trip, the users use their watches to monitor their physical status and the surrounding environment. After the trip, the users return to their car and the climate control system is switched on. The climate control system, which has the ability of adjusting the thermal comfort level in different sections of the car by monitoring parameters such as air temperature, relative humidity and even metabolic heat production (which is function of the pulse rate), sends out interest messages asking for these parameters and making the appropriate adjustments. Later, when one of the users enters his/her apartment, the less sophisticated air-conditioning system sends out interest messages asking only for temperature readings and the watch responds appropriately.

Now consider the case where a large number of people are in a concert hall attending a concert. Due to the large area of the concert hall, and the varying concentrations of people, the environmental conditions could differ greatly in different sections of the hall, (e.g. stalls, circle, etc). Assuming people have the watches mentioned earlier, the central air conditioning system could send out interest messages asking for the thermal comfort level in various sections of the hall. Rather than having every watch return a temperature and relative humidity reading, the watches can aggregate the data, perform in-network processing, and return the comfort level parameter to the central air-conditioning system. This would allow the environmental parameters to be adjusted at a more localised level.

From the above examples, it can be clearly seen that the sensors in the watch are required to react to different interest messages as they move from one environment to another. It is simply impossible to pre-program a node with all possible queries as the possibilities are infinite. Certain queries require all parameter values, e.g. pulse rate, temperature and humidity, some require only a subset while others (e.g. the example of thermal comfort in the concert hall) simply ask for a value of a parameter that can ultimately be calculated from other parameters that can be provided by certain sensors in the network.

IV. Clustered Diffusion with Dynamic Data Aggregation

Just like Directed Diffusion, Clustered Diffusion with Dynamic Data Aggregation (CLUDDA) can be also be split up into two main phases: interest propagation and data propagation. We initially describe the interest propagation phase and later go on to elaborate on how CLUDDA is used to perform data aggregation during the data propagation phase.

A. Interest format

The format of the interest message forms an integral part of CLUDDA since it plays a significant role in *interest transformation,* the formation of dynamic *Data Aggregation Points* and most importantly, it allows a node to handle unfamiliar queries that were not pre-defined during deployment.

Due to the design requirements mentioned in Section III, the format of an interest message in CLUDDA contains not only the query but also the entire definition of the query itself. Having the entire query definition allows nodes to break down a query into its fundamental components, gather these individual components of data and subsequently process these components using the query definitions which in turn results in data reduction. This reduced data can then be transmitted by a node back to the sink. This sort of in-network processing should result in higher data reduction than simply suppressing the transmission of duplicate data messages [3]. It should also be noted that sensor nodes have a very limited memory and thus storing all possible query definitions is not feasible. In this design, sensor nodes do not need to store query definitions permanently and can instead adapt easily to varying environments.

B. Interest Propagation – Exploratory Phase

Now that the interest format has been described, this section explains how this interest propagates throughout the sensor network. Unlike Directed Diffusion where every node in the sensor network plays a part in the propagation of an interest message, in our architecture, the dissemination of interest messages only involves the clusterheads and the gateway nodes. Regular nodes carry out transmission only when they are capable of servicing a certain incoming query. Due to the dominating property of the clusterhead, every node that injects an interest message (i.e. every sink node) into the network is a member of a certain cluster. A sink node unicasts the message to its clusterhead. The clusterhead not only checks to see if it can service the query by itself but also transmits the interest message to its children made up of regular nodes, and direct and distributed gateway nodes. Upon receiving an interest message, gateway nodes retransmit the message to nodes in neighbouring clusters. Not all interest messages received by a clusterhead or gateway node are retransmitted. For instance, a node that has recently transmitted a matching interest message will refrain from retransmitting the received message. Every node receiving an interest message responds with an appropriate data message only if it is able to service the query stated in the interest message.

For every requested service, a sink node (assuming it is a child node in a cluster) sends out interest messages periodically to its clusterhead. The initial interest message can be considered to be *exploratory* while subsequent messages are sent to *reinforce* certain chosen data

propagation paths. The exploratory interest message attempts to check if the required service is available. If the sink node receives an appropriate response from a source node in the network, the sink node subsequently reinforces the data propagation path by sending new interest messages with the same query but requesting data messages at a higher frequency (i.e. shorter time interval between consecutive data messages). The rate at which data messages are required are specified within a gradient field found in every interest message.

Every clusterhead and gateway node contains an interest cache which contains an entry for every single distinct interest message it receives. Once a source node providing the required data has been found, the entries in the interest cache are used to route the data back to the sink node. Entries in the interest cache do not contain information about the sink node which originally injected the interest message into the network. As far as every receiving node is concerned, an incoming interest message originates from the neighbouring sending node.

C. Data propagation and aggregation

A node that receives an interest message first checks the main query contained in it. If the node is unable to comprehend the query, it then parses through the query definition and identifies the fundamental components that eventually make up the main query. It then checks to see if it recognises any of the fundamental components and whether it is able to provide the data for one or more of the components. A regular node that is unable to service any part of the query simply remains silent, while a clusterhead or gateway node retransmits the interest message (provided a similar message has not been transmitted recently) to neighbouring clusters. A node that can service one or more of the fundamental components, then gathers the data, assembles the appropriate data message, and checks the relevant entry in the interest cache to retrieve the ID of the neighbouring node that sent it the interest message and subsequently transmits the message. The node also checks the gradient stated in the received interest message so that it knows when the next corresponding data message needs to be sent.

Before transmitting an incoming interest message to its neighbours, a node checks its data cache to see if there are any existing recent data messages that contain data for the same components listed within the query of the received interest message. The existence of such messages would indicate that there are certain useful components of data that are passing through this node. Thus, if a match is found, the node obtains the IDs of the neighbouring nodes that have been sending it these data messages and forwards the received messages to them. Using this knowledge to route interest messages to certain specific nodes is more energy efficient than forwarding an interest message to all neighbouring gateway nodes for example.

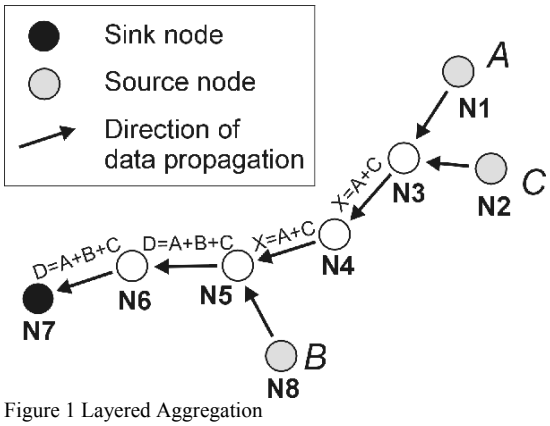

Figure 1 Layered Aggregation

1) Layered Data Aggregation

The transmission of data messages being sent back from the various source nodes across the network finally leads to the formation of *data aggregation points*. Let us consider the case where an interest message is sent out by a sink node containing a query, D which has the following definition, D = A + B + C. The parameters A, B, and C are the fundamental components that can be understood by certain nodes in the sensor network that are capable of providing the relevant data. The query D however, is not understood by any node. Let us now assume there are two nodes, N1 and N2, capable of providing data for parameters A and C which are geographically close to one another. Figure 1 shows that node N5 receives data for parameters A and C. Note that the all nodes shown in Figure 1 belong to different clusters and are either clusterheads or gateway nodes. The clusters however, have not been illustrated for the sake of clarity. Upon receiving the data messages, N3 checks all its entries in its interest cache to see if there are any queries that require parameters A and C to be returned to the sink node and in the process identifies an entry containing query D. It then goes on to check the query definition against the received parameters to see if the messages can be aggregated. In this case, the node aggregates A and C and assigns the result to a new variable, say X. N5 then returns not only the new variable X with its new value to the original sending node but also, the definition of X itself, i.e. X = A + C so that N4 understands the meaning of the aggregated data message it has received. This same data message then traverses back towards the sink until N5 which is receiving data messages for parameter B from another part of the network. N5 realises that the two incoming data messages have all the components required to aggregate the data and generate a response to service query D. Once D has been calculated, the result (together with the variable name D) is transmitted back to the sink node.

Instead of just suppressing duplicate data messages [3], the above example clearly illustrates that this design allows data to be actually *processed* within the network and subsequently aggregated thus resulting in the reduction of data as close to the source nodes as possible. To further improve the efficiency of the nodes in the network, rather than transmitting individual data messages for each parameter all the way to a node that has all the components

to perform data aggregation, when circumstances permit, our design allows *layered data aggregation*, i.e. aggregation being performed in steps. Layered aggregation is possible only because all clusterhead and gateway nodes have prior knowledge of the entire query definition.

2) Interest Propagation – Reinforcement Phase

Once a sink node receives data messages from a neighbouring node in response to an exploratory interest message it had sent out earlier, the node resends the interest. This time however, although the basic query stated in the interest message is the same, certain other elements in the message are different. Firstly, since the sink node now knows the existence of a source node in the network capable of servicing at least a certain portion of its query, it updates the gradient field in the new interest message indicating that it requires data at a higher rate, i.e. the new gradient specifies a shorter duration than the initial exploratory gradient. Secondly, since the nodes in the network already know the full definition of the query, the definition section of the interest message is now truncated. This saves considerable bandwidth during subsequent transmissions of the same interest message.

While any node that is a sink node can send out interest messages with higher gradients, only clusterheads and gateway nodes are capable of deciding which neighbour to reinforce when there are multiple neighbours sending the same data messages. This is because a regular node maintains connectivity with only the clusterhead of the node it belongs to while clusterheads and gateway nodes could be congregation points for a multiplicity of paths. The decision might depend on which neighbouring node responded first with the appropriate data message.

3) Interest Transformation

Unlike the initial exploratory interest propagation phase, where a certain cluster broadcasts a message to all its neighbouring clusters, subsequent transmissions of the updated interest message are only made to the nodes that have been reinforced by a clusterhead or gateway node. Any node that receives an interest message without a full definition checks its *query cache* to see if it had performed any aggregation previously for the received interest. Every time a clusterhead or a gateway node performs some form of data aggregation, it keeps a record in its query cache. The query cache lists the various fundamental components that

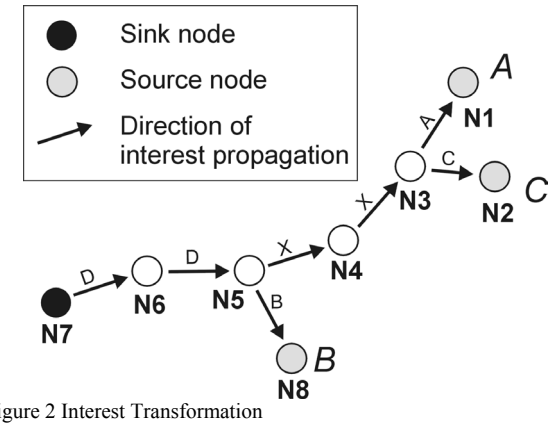

Figure 2 Interest Transformation

were combined to calculate the result that was finally sent back. It also contains the addresses of all the neighbouring nodes from which the data messages had originated. These addresses are used to propagate interest messages directly to certain specific nodes rather than broadcasting them to all the clusters. It is during the phase of subsequent transmissions of an interest message that *interest transformation* takes place.

Consider N5 shown in Figure 2. Upon receiving an interest message from N6, and checking its query cache, it realises that it had previously performed the aggregation for the received interest. The entry in the query cache states that it had received a data message for component B from N8 and component X (made up of A and C) from N4. Instead of sending out the original interest, D, from N5 to N4 and N8, N5 *transforms* the interest and sends out an interest message asking for data for parameter B from N8 and data for parameter X from N4. Note that N4 will be able to interpret parameter X as the full definition of X is stored in its interest cache. Transforming the original interest into the fundamental components allows a node to make use of its existing knowledge of query definitions (from the previous exploratory interest message) thus preventing the receiving nodes from having to parse through the entire query definition once again. This would save the node from performing redundant processing and also improve latency.

4) Dynamic Aggregation Points

Another important design aspect is that the data aggregation points are dynamic, i.e. the task of aggregating data is not assigned to any specific set of nodes in the network. The node performing data aggregation can change

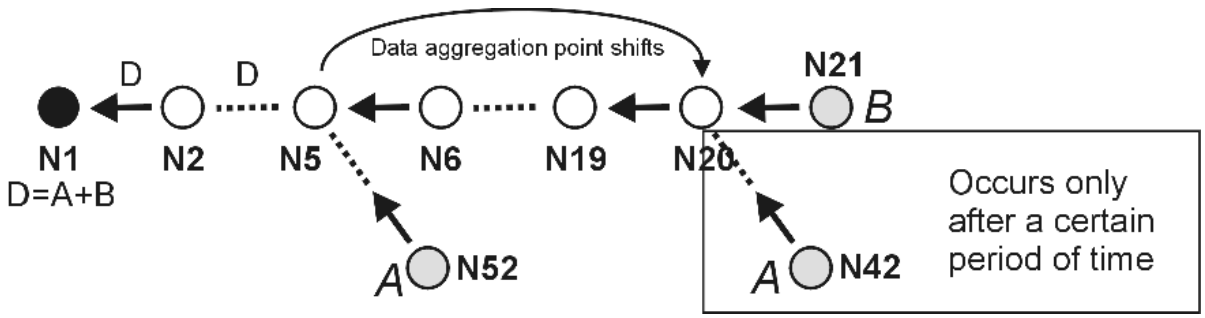

Figure 3 Dynamic Aggregation Point

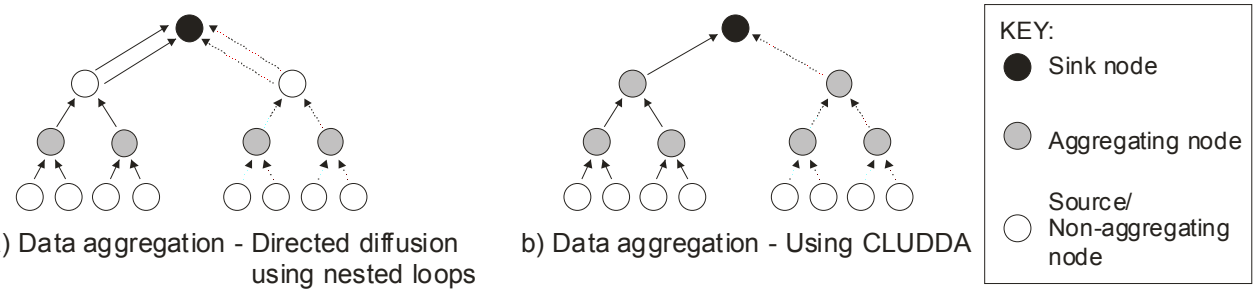

Figure 4 Dynamic Aggregation Point

as the locations of the source nodes change. Generally any clusterhead or gateway node with knowledge of the query definition is equipped to perform data aggregation. This dynamic architecture once again allows nodes to carry out aggregation as close to the source as possible.

Consider the scenario illustrated in Figure 3. Initially the sink node N1 sends out an interest message for the parameter D which is defined as: D = A + B. The interest propagates through the various clusters in the network and at first N52 provides data messages for component A while N21 provides data messages for component B. As the data messages for both A and B propagate towards the sink node, N5 encounters data messages for both the components required to calculate the requested parameter D. At this time, N5 is the node that lies closest to both N52 and N21 that is capable of performing the data aggregation. Now consider the case when after a certain period of time, N20 starts receiving data messages for parameter A from N42. Now, instead of aggregating at N5, the aggregation is performed by N20 since it has all the required components to calculate the requested parameter D. Thus the design allows the data aggregation point to shift as close to the source nodes as possible depending on the varying network conditions.

5) Network Healing

Due to the nature of wireless sensor networks where the network failures can be expected to be a common phenomenon due to nodes that run out of power or move out of transmission range, provision must be made to ensure that the network is able recover from transmission errors as soon as possible. Instead of making the sink node wait till it times out in the event of a transmission failure and *then* take the appropriate action, our architecture allows intermediate nodes that are close to the source of the error to detect it and take action immediately so as to attempt to restore communication as soon as possible. When a certain node along the data propagation path does not receive an expected data message within the stipulated time limit, it assumes that a transmission error has occurred. The node then checks its interest cache to see which queries have been affected by the data messages that did not arrive. The affected queries are then resent, together with the entire query definition, to the neighbouring clusters using exploratory gradients so that new sources for the required data can be found. Thus any clusterhead or gateway node that has the full query definition can attempt to heal the network or re-establish communication. It is not just restricted to the sink node that originally initiated the query. Apart from this mechanism, negative reinforcements are also used just like in Directed Diffusion to suppress high-delay or lossy paths.

V. Related Work

A number of concepts mentioned in this paper have certain similarities with prior work carried out in several other projects. This section highlights both the similarities and differences between the key concepts.

One of the primary features of CLUDDA is the new format of the interest message. The concept of having an interest message which has multiple definitions within it used in CLUDDA can be compared with the two-level nested queries used in directed diffusion [2]. In this approach, there are typically two types of sensors, the initial sensor and the triggered sensor. As shown in Figure 4(a), the user initially queries the triggered sensors. These sensors then sub-task the initial sensors to service the query. This architecture results in a slightly rigid design in terms of data aggregation. Firstly, a triggered sensor needs to keep track of the services provided by the initial sensors that are under its control. Thus the initial sensors need to have a publishing mechanism which could register their services with the triggered sensor. This would result in increased overhead as the publishing mechanism needs to ensure that the triggered sensor always has an updated view of its initial sensors. Secondly, only nodes that are one hop away from the initial sensors can be assigned the role of a triggered sensor and therefore perform data aggregation.

This is in stark contrast to CLUDDA which firstly, does not require the source nodes to publish their services separately. The receiving node automatically learns of the services provided by the source node by analysing the data message it has received as it travels from the source to the sink node. Next, every intermediate node that has encountered the original interest message between the source and sink nodes is capable of performing data aggregation, Figure 4(b). This is because the nodes contain the complete definition of the query and are thus capable of aggregating data whenever required, regardless of the position of the node within the tree hierarchy. This in turn leads to another feature that is inherent in the design – *layered aggregation*. This would not be possible using the

mechanism employed in directed diffusion.

Certain parallelisms can be drawn between layered aggregation and *partial aggregation* [5]. For instance, in partial aggregation, each intermediate node computes partial results that contain sufficient statistics to compute the final result. For example, if the user requests for the average readings of a set of sensor nodes, the nodes respond by sending back data messages which contain the `COUNT()` and the `SUM()`. The final aggregating node then performs the `AVG()` operation. However, what happens when a user sends out a query asking for a certain unknown operation to be worked out (e.g. `D=A+B+C`) – an operation which is not recognised by any of the nodes in the network? In the previous example, nodes are able to carry out data aggregation operations as they have the knowledge to handle queries such as `AVG()`, `COUNT()`, etc. On the other hand, it is imperative that aggregation operations can also be performed if nodes are faced with unfamiliar operations. This is the situation that CLUDDA's new interest format is designed to handle.

Directed diffusion mentions using cached data to direct interests to certain specific parts of the network to prevent flooding [3]. This helps to conserve energy by limiting transmissions. This concept of directing interests is also utilised in the *interest transformation* feature of CLUDDA. However, while directed diffusion directs the original interest sent by the sink node to a certain section of the network, CLUDDA allows intermediate nodes to make use of cached data to analyse incoming interest messages and regenerate *new* interest messages to specific parts of the network. This prevents nodes from having to parse through entire query definitions thus eliminating redundant processing, improving latency and reducing memory usage. This scheme could also be used to direct future interest messages to certain sections of the network, thus improving network efficiency over a certain period of time. We also have future plans to employ the cached data or *network metadata* of nodes to use as inputs to query plans as part of a distributed query processing engine.

VI. Future Work and Conclusion

In this paper, we have introduced a new data centric communication protocol for wireless sensor networks that uses Directed Diffusion as its foundation. We described a new format for the interest message which enables a sensor node to operate even in unfamiliar environments that it was not designed to handle before the deployment of the network. It was explained how the new format of the interest message also led to the development of key features such as layered data aggregation, interest transformation, dynamic data aggregation points and network healing all of which eventually help to improve the overall efficiency of the network.

Many of the features described in this paper depend heavily on the naming scheme used. We are currently looking into this area. The memory requirements for the protocol described in this paper will also be investigated. Once the naming scheme has been fully developed, we plan to simulate the protocol in OMNET++ and analyse the results obtained to see if the design performs up to expectations.

References

[1] S. Dulman, L. v. Hoesel, T. Nieberg, P. Havinga. Collaborative communication protocols for wireless sensor networks. Workshop on European Research on Middleware and Architectures for Complex and Embedded Systems, Apr. 2003.

[2] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. Building efficient wireless sensor networks with low-level naming. In *Proceedings of the ACM Symposium on Operating System Principles*, Banff, Canada, October 2001.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 56-67, Boston, MA, USA, Aug. 2000. ACM.

[4] M. Weiser. The Computer for the 21st Century. *Scientific American*, September 1991.

[5] Y. Yao and J. Gehrke. Query Processing for Sensor Networks. In *First Biennial Conference on Innovative Data Systems Research*, January 2003.

[6] CONSENSUS project, http://www.consensus.tudelft.nl.

[7] EYES project, http://www.eyes.eu.org.