A Randomly Delayed Clustering Method for Wireless Sensor Networks

Xiaobo CHEN, Zhisheng NIU
Tsinghua-Hitachi Joint Lab on Ubiquitous IT
Room 3-329, FIT Building, Tsinghua University, Beijing 100084, China

Abstract—In recent years, wireless sensor networks have become a very hot research area, and are expected to find lots of applications in ubiquitous monitoring and information collecting. In this paper, a random delay based hierarchical routing protocol, RDCM (Randomly Delayed Clustering Method), is proposed for mobile targets detection and positioning in wireless sensor networks. RDCM is a new fully distributed algorithm, with the ability of dynamically forming clusters on-demand for reactive sensor networks. It requires broadcasting only once for cluster head selection and intra-cluster communication schedule. Then RDCM is compared with LEACH (Low-Energy Adaptive Clustering Hierarchy) and the optimal routing scheme for our system model—ideal CNS (Center at Nearest Source), in terms of communication energy costs. Simulation results show that RDCM can save more energy than LEACH while introducing a small implementation overhead when compared with ideal CNS.

1. INTRODUCTION

Recent advances in MEMS, distributed computing, sensors and wireless communications have enabled the development of wireless sensor networks, which consist of hundreds to thousands of tiny, inexpensive, autonomous and battery-powered sensor nodes[1]. Wireless sensor networks are a kind of large scale, multi-hop wireless ad hoc networks[2]. They are expected to find a wide range of applications in the near future, including battlefield surveillance, environment and habitat monitoring, civil infrastructure and factory machine monitoring, disasters recovery, health and other commercial applications[1]. Wireless sensor networks of the future are envisioned to revolutionize the paradigm of information collecting and processing in diverse environments.

Compared with traditional wireless networks, wireless sensor networks have four basic features. First, sensor nodes have very limited resources, especially the very scarce energy resources[1-4]. Sensor nodes are often battery-powered and autonomous. Power efficiency directly influences the network lifetime and hence is of prime importance in the design of sensor networks[1]. Second, wireless sensor network is a kind of data-centric network. The typical mode of communication is from multiple data sources to one sink[3]. Third, adjacent sensor nodes often sense the common phenomena and have similar data[1-4]. Thus, it should be carefully considered to support in-network data aggregation in routing protocol design. Finally, position awareness of sensor nodes is important since data collection is normally based on the location[2]. All these factors make the design of wireless sensor networks very different from that of other wireless networks, and raise some new challenges, including energy-efficient routing protocol design, MAC layer design, in-network data aggregation algorithms developing, targets recognition and positioning algorithms developing and the design of hardware and middleware for sensor nodes. Among them, the design of an energy-efficient, robust and scalable routing protocol is surely of the most importance.

Several routing protocols have already been proposed for wireless sensor networks, such as LEACH (Low-Energy Adaptive Clustering Hierarchy)[5,6], APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol)[8], PEGASIS (Power-Efficient GAthering in Sensor Information Systems)[9,10], Directed Diffusion[11,12] and GEAR (Geographical Energy Aware Routing)[13]. LEACH is a fully distributed cluster-based routing protocol for wireless sensor networks. It randomly selects a few sensor nodes as cluster heads and rotates this role to evenly distribute the energy load among the sensors in the network. It is very representative for wireless sensor networks, and is the fundamental of most work on cluster-based routing, such as APTEEN and PEGASIS. APTEEN is an improved version of LEACH. It uses the same cluster head selection algorithm with LEACH, but introduces three parameters—hard threshold, soft threshold and count time—to reduce node's data report times. PEGASIS is another improved version of LEACH. Its key idea is to organize all sensor nodes into a chain, in which those nodes that are closest to each other are neighbors. Then, the nodes in the chain take turns being a cluster head. Directed diffusion is a flat routing protocol. It can set up multiple paths between a node and the sink through the interest diffusion process initiated by the sink, and choose the best path for data report. GEAR uses energy-aware and geographically informed neighbor selection heuristics to route a packet toward the destination region. The key idea is to restrict the number of interests in directed diffusion by only considering a certain region rather than sending the interests to the whole network. A detailed survey on wireless sensor network routing protocols is given in [7]. However, for different kinds of applications, the requirements to sensor networks may totally different. So the design of routing protocols in wireless sensor networks should be application specific.
In this paper, a random delay based hierarchical routing protocol, RDCM (Randomly Delayed Clustering Method), is proposed for mobile targets detection and positioning in wireless sensor networks. RDCM is a new fully distributed algorithm, with the ability of dynamically forming clusters on-demand. It utilizes the query-driven routing model, and introduces a random delay based cluster head selection algorithm, which requires broadcasting only once for cluster head selection and intra-cluster communication schedule. Then RDCM is compared with LEACH and the theoretical optimal routing scheme for our system model—ideal CNS. Simulation results show that RDCM consumes less energy than LEACH under the same situation. Besides, it introduces a quite small implementation overhead to ideal CNS. Thus, it is energy-efficient.

The rest of the paper is organized as follows. We first give our system model in section II. The motivation of our work is given in section III. RDCM, which is designed for our system model, is given in detail in section IV. In section V, we compare RDCM with LEACH and ideal CNS by simulations. And we conclude our paper in section VI.

II. SYSTEM MODEL

We consider a sensor network with \(N\) nodes randomly scattering in a sensing field. A data sink is deployed to collect sensing data from it. There are some mobile objects moving inside the sensing field. The task of the sensor network is to detect and position these mobile objects. For simplicity, we assume that all nodes know their own positions. All nodes are assumed to have a communication radius \(R\), a corresponding maximum transmission power \(P_{\text{max}}\) and a sensing range \(S\). A node has a sensing range of \(S\) means that all nodes within a distance of \(S\) of an object are considered to be data sources. (This is called Event Radius (ER) model in [4]) We also assume that \(R>2S\), under which all nodes that detect the same object can communicate with each other directly. As for the routing model, the sink first sends out a query/interest. Then the sensor nodes which have appropriate data will report to the sink. This means that the reactive routing model is adopted. Figure 1 shows an example of the system model. In this example, 100 nodes are randomly scattering in a 100*100m\(^2\) area. The sink is placed at (0, 0). The sensing range \(S\) is set to 15m and the communication radius \(R\) is set to 30m.

III. MOTIVATION

From [7], we learn that Directed Diffusion and GEAR are two typical routing protocols that can be directly applied to our system model. However, when nodes can only detect mobile objects in a small fraction of the whole monitoring time, it will waste a lot of energy to setup and maintain gradient fields for such nodes. Besides, they both are flat routing protocols. Each node independently chooses a path from its gradient field. Although data aggregation can be done in an intermediate node, it is obviously not as energy efficient as doing data aggregation in a local cluster head. These deficiencies make them not as energy efficient as cluster-based routing protocols. However, in the cluster-based routing protocols, such as LEACH, APTEEN and PEGASIS, they all have assumed that all nodes could directly communicate with the data sink (or base station) if needed. They are mainly focus on how to select cluster heads and how to schedule the intra-cluster communication. Routing from cluster heads to the sink is not considered under this assumption. It is not supported in our system model because each node has a limited communication radius \(R\). So they can not be applied directly. That is why we want to develop a new cluster-based routing protocol for this kind of applications to get higher energy efficiency.

In [3,4], the authors have proved that for the ER model, when \(R>2S\), CNS(Center at Nearest Source) is the optimal routing protocol in the sense that it requires the least data transmissions, if not considering any implementation overhead costs involved in setting up and maintaining the routing paths from sources to the sink. In CNS, the source which is nearest to the sink acts as a cluster head. All other sources send their data directly to it. It then aggregates all the receiving data with its own data, and forwards the aggregation information to the sink along the shortest path. CNS only needs \((k-1)+\min(d_i)\) transmissions, where \(k\) is the number of sources, and \(d_i\) is the distance of the shortest path from the \(i\)th source to the sink. We call the CNS without considering any implementation overhead costs ideal CNS in the following. However, how to set up the shortest path between a source node and the sink and how to find out the source node nearest to the sink are not mentioned in [3,4]. In this paper, a random delay based hierarchical routing protocol, RDCM, is presented for the above system model. RDCM can be seen as a simple...
implementation of CNS. It utilizes the query-driven model, and can set up a path, which is used for cluster head data forwarding, between each node and the sink though the query (Also called "interest" in [11,12]. And in this paper, there is no difference between them) diffusion process. It introduces a random delay based cluster head selection algorithm, and requires broadcasting only once for cluster head selection and intra-cluster communication schedule. It is a fully distributed cluster-based routing protocol, with the ability of dynamically forming clusters on demand.

IV. RDCM: A RANDOMLY DELAYED CLUSTERING METHOD FOR WIRELESS SENSOR NETWORKS

The operation of RDCM is broken up into three phases: interest diffusion, CH (Cluster Head) selection and cluster forming, and CH data forwarding. During the interest diffusion phase, the sink sends its query/interest to task sensor nodes in the sensing field. Each node chooses one from all its neighbors that is nearest to the sink (in hops) to be its immediately previous hop in this phase too. Through the immediately previous hop, routing paths from nodes to the sink are established, but will only used for CH data forwarding. After receiving the interest, all nodes that have appropriate data will perform a random delay based CH selection and cluster forming algorithm periodically, and report their data to the sink.

4.1 Interest diffusion

The sink first sends out a query/interest to tell sensor nodes the type of data wanted. In this paper, the attribute-value pairs based naming scheme in [11,12] is used for query describing. The query’s attributes of the initial time, the duration time, the type of sensing data required and the data reporting rate are included in this scheme. Correspondingly, this naming scheme is used for sensor nodes to describe their sensing data, too.

Each sensor node keeps an interest cache. Each item in the cache corresponds to a different interest. Interest entry in the cache contains information about the interest attributes, the immediately previous hop, the receiving signal strength and the hops to the sink.

For each active task, the sink periodically broadcasts an interest message to each of its neighbors with the maximum transmission power $P_{\text{max}}$. The initial hop count to the sink is set to 0. When a sensor node receives the interest, it will check its interest cache to see if the interest already exists. If no matching entry, the sensor node will create an interest entry for it, and forward it to its neighbors with transmission power $P_{\text{max}}$ after adding the message’s hop count by 1. Otherwise, the node will not forward it, but compare the receiving message’s hop count with the record in the interest cache. If the former is smaller than the latter by 1 and if the receiving signal strength is larger, the immediately previous hop, the receiving signal strength in the interest entry should be updated. If the former is smaller than the latter and the difference between them is larger than 1, the immediately previous hop, the receiving signal strength and the hops to the sink in the interest entry should be updated.

At the end of the interest diffusion phase, each node chooses the one with the minimum hops to the sink (if not unique, the one with the maximum receiving signal strength among them will be selected) from all its neighbors to act as its immediately previous hop. Through the immediately previous hop in the interest cache, each node sets up a path to the sink. In RDCM, this path is only used for CH data forwarding.

4.2 CH selection and cluster forming

Upon receiving the interest, a node starts to collect samples with an interval of $T_s$, where $T_s$ is the sampling period required by the interest. When a node detects the mobile object, it first chooses a random time value $t$ between $(0,T)$, where $T$ is a parameter related to the node density $\rho$ of the network, the sensing range $S$ of a node and the time $T_0$ for receiving and forwarding a packet. Then it turns on its transceiver and waits for a time $t$ to transmit. If it does not hear any CH broadcast during the time $t$, it will serve as a CH and send out a CH broadcast with power $P_{\text{max}}$ at $t$. Otherwise, it will choose the node from which the CH broadcast with strongest receiving signal is heard to be its CH, adjust its transmission power according to the receiving signal strength, and report its data to CH at $t$.

In this scheme, a CH node will receive all data packets from non CH nodes in time $T$. So the parameter $T$ should be set to guarantee that all non CH nodes could send their data to CH. We will give a detailed description of how to set $T$ later.

4.3 CH data forwarding

All non CH nodes send their data to CH in $T$. CH aggregates all receiving data with its own sensing data, and then transmits the aggregated data to its immediately previous hop node according to the record in its interest cache. A node that receives this aggregated data packet will forward it according to the interest cache until it finally reaches the data sink. During the data forwarding phase, data aggregation can also happen in an intermediate node that receives multiple data packets from different neighbors.

4.4 How to set the parameter $T$

During the CH selection and cluster forming phase, each source node has to randomly choose a continuous time value $t$ between $(0,T)$. As mentioned before, the parameter $T$ is determined by the node density $\rho$ of the network, the sensing range $S$ of a node and the time $T_0$ for receiving and forwarding a packet, to guarantee all non CH nodes can send their data to CH during time $T$. However, as we need time $T_0$ to transmit
and receive a data packet, the choosing of a continuous time value $t$ is not appropriate. A more practical approach is to divide $T$ into time slots, with a length of $T_i$. Obviously $T_i$ should be larger than $T_m$ so that we can receive and forward a data packet in a slot.

In our system model, the node number in a cluster is equal to the number of nodes which detect the same mobile object simultaneously. It is a random variable with an expectation $N_0$ of $\rho \pi S^2$. Then we could set $T$ equal to $k \cdot N_0 \cdot T_i$, where $k$ is a constant and is chosen to guarantee that there are enough time slots. When a source node detects a mobile object, it first randomly chooses a time slot to transmit its data packet or CH broadcast. The larger $k$ is, the smaller the probability that two different nodes choose the same slot. But a larger $k$ means longer time for CH to collect data. So we can choose $k$ based on the nodes distribution. In our system model, nodes are randomly scattering in the sensing field, and can set $k$ to 3.

In wireless sensor networks, nodes always have a report time $T_r$ to reduce data transmissions. When a node detects a mobile object, it reports its data to sink every $T_r$ seconds, during which a node can have multiple samples. The value of $T_r$ is determined by application requirements. In these applications, in order to facilitate the end user to recognize data reported at different time, the total time needed for a node to report its data to the sink should less than $T_r$, i.e. parameter $T$ plus the delay from CH to the sink should be less than $T_r$ in our scheme. If this is satisfied, then different $T$ makes no differences to the end user.

V. PERFORMANCE EVALUATION

We compare RDCM with LEACH and ideal CNS in terms of communication energy costs in this section. Normally, the communication energy costs are measured by the data transmission times as in [3, 4]. This has two deficiencies. In sensor networks, receiving data packets is also an energy consuming process. Distributing a broadcast packet to neighbors will consume much more energy. Furthermore, the communication energy between two nodes depends on their distance too. In this paper, we use the first order radio model in [5] to compute the total amount of energy dissipated. In the first order radio model, the radio dissipates $E_{\text{elec}}=50\text{J}/\text{bit}$ to run the transmitter or receiver circuitry and $E_{\text{amp}}=100\text{pJ}/\text{bit}/\text{m}^2$ for the transmit amplifier to achieve an acceptable signal to noise ratio. For more details, see [5]. Using the first order radio model, to transmit a $k$-bit data packet to a distance $d$ will expend an energy of

$$E_t(k,d)=E_{\text{elec}} \cdot k + E_{\text{amp}} \cdot k \cdot d^2,$$  \hspace{1cm} (1)

and to receive this packet will expend an energy of

$$E_r(k)=k \cdot E_{\text{elec}}.$$  \hspace{1cm} (2)

We have simulated two scenarios. One is a mobile object moving with a constant velocity in the sensing field. The other is a mobile object moving with a random velocity in the sensing field.

In our simulation, the sensing field is set to be a 100*100m$^2$ square area. The sink is placed in one of its vertices, with the coordinate of (0, 0). The number of sensor nodes is 100. We assume that all data packets have a length $L$ of 512 bits. For simplicity, we also assume that the aggregation function is such that each intermediate node in the routing transmits a single aggregated packet even if it receives multiple input packets.

In the simulation of RDCM and ideal CNS, each sensor node has a communication radius $R$ of 30m, a sensing range $S$ of 15m, a data rate $r$ of 40kb/s (Mica[14] has a maximum data rate of 40kb/s) and a report time $T_r$ of 1s. The parameter $T$ in RDCM is set to 0.5s for the following reason: the time $T_0$ to transmit and receiving a packet is

$$T_0=L/r=12.5\text{ms}.$$  \hspace{1cm} (3)

And the time $T$ is long enough to transmit and receive 40 data packets. The sensor nodes density $\rho$ is $\rho=100/(100\times100\text{m}^2)=0.01\text{m}^{-2}$. (4)

Then the average number $N_0$ of source nodes in a cluster is

$$N_0=\rho\pi S^2=7.07.$$  \hspace{1cm} (5)

So there will be enough time slots for non CH nodes to send their data to CH.

In the simulation of LEACH, each node has a communication radius $R$ of 100m, a sensing range $r$ of 15m. The expected number of clusters producing by LEACH is set to 5. The duration of each round is 60s. In the first second of each round, the distributed clustering algorithm is performed for forming clusters. Then sensor nodes report their data to the sink once a second. For the fairness of our comparison, only the nodes that have detected the mobile target are allowed to report their data.

We first simulate the scenario in which the mobile object is moving from point (0, 0) with the velocity of $v_x=v_y=1\text{m/s}$. The total simulation time is 150s. In the simulation, we assume that when the mobile object is out of the sensing field, no sensor nodes can detect it. Figure 2 shows the real track of the mobile target and its estimation, where we use the average coordinate of all sensor nodes that have detected the target as the target’s position estimation. Figure 3 shows the communication energy cost comparison of RDCM, LEACH and ideal CNS. In the second scenario, the target is moving in the given area with a random velocity. The total simulation time is 150s. Figure 4 shows the moving track of the target. The corresponding communication energy cost comparison of RDCM, LEACH and ideal CNS is showed in Figure 5.
In Figure 3 and 5, the red line shows the communication energy cost of the ideal CNS, the blue line shows the communication energy cost of RDCM, the green line shows the communication energy cost of RDCM without including the energy dissipated by CH broadcast, the magenta line shows the average communication energy cost of LEACH for simulating 100 times, and the black dash-dot line shows one instance of the communication energy cost of LEACH in these simulations. From Figure 2 and 4, we can see that RDCM consumes much less energy than that of LEACH. We can also see that the CH broadcast is the main reason for the difference of RDCM and ideal CNS.

As we have mentioned before, RDCM is motivated by the idea of ideal CNS. It can be seen as a simple implementation of ideal CNS. When compared with ideal CNS, RDCM consumes more energy for two reasons. One is that RDCM introduces a CH broadcast every data report time for CH selection and intra-cluster communication schedule. The other is that RDCM does not use the optimal route. Which one affects more? The first one is computable while the second one is hard to do so. Using the first order radio model, we could easily get the average energy consumption $E_{\text{broadcast}}$ caused by distributing a CH broadcast packet to neighbors in our simulations:

$$E_{\text{broadcast}} = N_0 * E_{\text{elec}} * L + e_{\text{amp}} * L * R^2 = 227.072 \mu J. \quad (6)$$

If we multiple the $E_{\text{broadcast}}$ value in Eq. (6) by the total data report times in our simulations, we can see that the CH broadcast in RDCM is the main cause for implementation overhead. The green curves in Figure 3 and 5 show the communication energy cost of RDCM when the energy dissipated by CH broadcast is not included. We can see that if we do not consider the CH broadcast, RDCM and ideal CNS have a very close energy costs. So it is very important to reduce the energy dissipated by CH broadcast to get higher energy efficiency. From Eq. (6) we can see that the energy dissipated by a CH broadcast is proportional to its packet
length. In order to save more energy, shorter CH broadcast packets are preferred.

In RDCM, it requires broadcasting only once for CH select and intra-cluster communication schedule. We have pointed out that the CH broadcast in RDCM is the main reason for implementation overhead. When the targets are mobile, source nodes in sensor network may change rapidly. And it is reasonable to form clusters on-demand every data report time. Thus, RDCM is energy-efficient.

VI. CONCLUSION

In this paper, we propose a random delay based hierarchical routing protocol, RDCM, for mobile targets detection and positioning in wireless sensor networks, with the object of achieving higher energy efficiency. In RDCM, a simple randomly delayed clustering method is introduced. It is a new fully distributed algorithm, with the ability of forming clusters on-demand for reactive sensor networks. We compare RDCM with LEACH and ideal CNS in terms of communication energy costs. Simulation results show that RDCM can save much more energy than LEACH while only introducing a small implementation overhead when compared with ideal CNS. Thus, RDCM is energy efficient. We also find that the CH broadcast in RDCM is the main reason for implementation overhead. In order to save more energy, shorter CH broadcast packets are preferred.

REFERENCES