DATA CENTRIC STORAGE IN WIRELESS SENSOR NETWORKS

Prof Narendra Kumar,
Dean DIT School of Engineering,
Ph.D. Research Scholar, Manav Bharti University, Solan, Himachal Pradesh (India)

ABSTRACT:

Wireless Sensor Networks (WSNs) have matured as the technology most suited for monitoring of the environment and data collection. But harnessing the powers of a WSN presents innumerable challenges. Issues related to networking (routing), storage, and transport of the huge amount of data that keep flooding the Network, needs to be planned in advance, before the data reaches the sink and one has to minimize creation of hotspots on the sensors around the sink.

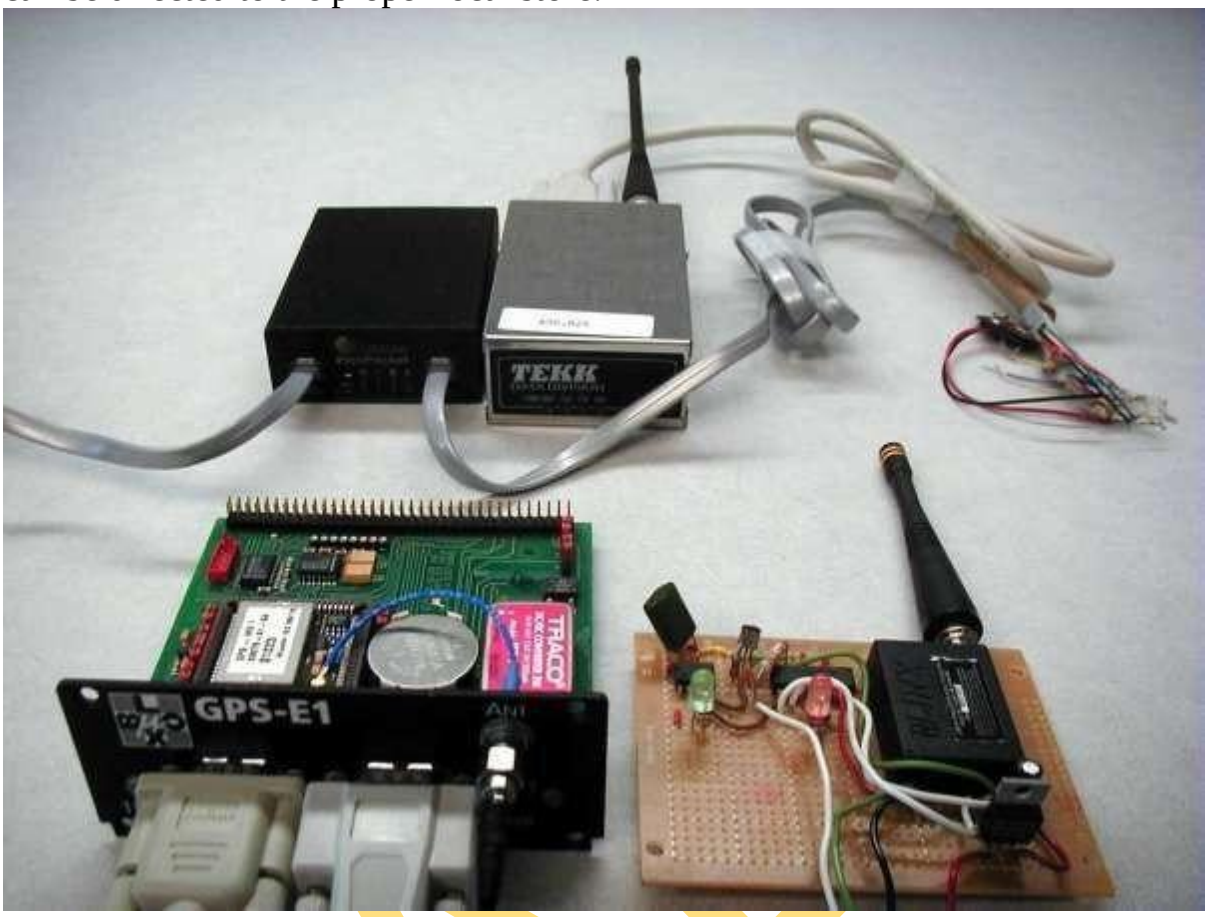
Introduction

A *sensornet* is a distributed sensing network consisting of a large number of small devices, each having some computational, storage and communication capability. Such networks best operated *in an unattended mode* to record detailed information about their surroundings. They are thus well suited to applications such as *location tracking* and *habitat monitoring*, etc. Communication between nodes requires the expenditure of energy, a scarce commodity for most sensornets. Thus, making effective use of sensornet data will require scalable, self-organizing, and energy-efficient data dissemination algorithms Making effective use of the vast amounts of data gathered by large-scale sensor networks (sensornets) will require scalable, self-organizing, and energy-efficient data disseminationalgorithms.

The requests for data are normally broadcast to all the sensors, since the sink can hardly know in advance the identity of the sensors that eventually manages to sends the relevant data that sink needs on regular basis. Communication abstractions that are *data-centric* become the best model, as the data are "named" and the communication abstractions refer to these names rather than to the node network addresses. *The* directed diffusion *data-centric routing scheme* is most promising and energy-efficient data dissemination method for sensornet environments. The three canonical methods described in the previous section certainly do not exhaust the design space; combinations of them yield hybrid methods specialized for particular needs. Forexample:

Data-Centric Storage for location guidance. For certain applications, one might combine LS and DCS by storing detailed event information locally and using DCS to inform a querier of an event's location so that subsequent queries

can be directed to the proper local store.



(Typical Hardware of the Wireless Sensor Networks connected with the GPS)

Data-Centric Storage for context. In the course of processing local data, nodes may find it useful to have some context about global parameters. For instance, datacentric storage could give nodes access to the number of other animals sighted when a node is trying to determine if a migration isunderway.

Directed queries (geographically targeted). The canonical methods are designed for cases where one does not *a priori* know the event location. If one already knows the location of the event through out-of-band techniques, then one can direct queries to that location using geographic routing methods. This LS variant stores data locally, and queries are sent (at cost $O(\sqrt{n})$) to the relevant locations to retrieve the desired data. It avoids the cost of flooding in the canonical LS approach, and the cost of storing each event in the canonical DCSapproach. In-network storage and in particular Data Centric Storage (DCS) stores data on a set of sensors that depend on a meta-datum describing the data. DCS is a paradigm that is promising for Data Management in WSNs, since it addresses the problem of scalability (DCS employs unicast communications to manage WSNs), allows in-network data pre-processing and can mitigate hot-spotsinsurgence.

Storage actions

Like most of the distributed hash table systems the WSN - DCS also provides a *key/ value* -based ***associative*** memory. Even here the *events* are named with ***keys***. Both the *storage of an event* and ***its retrieval*** are performed using these *keys*. DCS is naming-agnostic in that any naming scheme that distinguishes events that users of the sensornet wish to identify distinctly suffices. The two operations DCS supports are:-

Put ***(k, v)*** *that* stores ***v*** (the observed data) according to the key ***k***, the name of the data.

Get ***(k)*** retrieves whatever value is stored associated with the key ***k***.

Design criteria

The challenges faced by the designers of a DCS system involves meeting the scalability and robustness criteria despite the system's fundamentally distributed nature. Sensornets represent a particularly challenging environment for a distributed storage system ***due to*** the following :

Node failures - are routine due to low battery power and permanent/ transient failure in a harsh environment in most of the r e a l i s t i c sensornet deployment.

Topology changes - will be more frequent than on traditional wired networks, compounded by Node failures, node mobility and weakening received signal strength/irregular variations disturbing neighbor relationships among the nodes

System scale in nodes may be very great. Sensor nodes may be deployed extremely densely (consider the limit case of smart dust), and may be deployed over a very wide physical region, such that the total number of devices participating in the DCS system may be on the order of 106 or more nodes.

Energy constraints will often be severe as survival of nodes depends on battery power.

The challenges listed above can be met by stringent albeit ingenious design criteria that ensure scalability and robustness of the system in order to make the distributed storage system survive even in the extremes of operations. The following are envisaged forthis:

Persistence: A ***(k, v)*** pair stored in the system *must remain available* to queries, despite sensor *node failures* and ***changes*** in the sensor network topology.

Consistency: A query for ***k*** must be routed correctly to a node where ***(k, v)*** pairs are currently stored. Also, if this node changes place (due to any reason like a node failure), the queries and the associated/ related stored data, must choose a new node in consistence with the previous ones for persistencesake.

Scaling in database size: As the number of ***(k, v)*** pairs stored in the system increases, whether for the same or different ***k'*s**, the storage should not get concentrated at any one particular node.

Scaling in node count: As the number of nodes in the system increases, the

system's total storage capacity would increase but it must ensure that the communication cost of the system does not grow unduly. Nor should any node become a concentration or choke point of communication.

Topological generality: The system should work well on a broad range of network topologies.

Use of Geographic Hash Table (GHT)

The core step in GHT is the hashing of a key ***k*** into geographic coordinates. Both a **Put***()* operation and a **Get***()* operation on the same key ***k,*** *actually puts the* ***hash k*** to that very location. A *key–value pair* is thus stored at a node in the vicinity of the location to which its key hashes. Choosing this node consistently is central to the mechanism of building aGHT.

If we assume a perfectly static network topology and a network routing system that can deliver packets to the desired location/ positions, such a GHT will cause storage requests and queries for the same ***k*** to be *routed to the* same node, and will distribute the storage request and query load for *distinct k values* ***evenly*** across the area covered by thisnetwork.

The service provided by GHT is similar in character to those offered by other distributed hash table systems. However, in the case of WSN systems, much of the nuances of the GHT system-design emerge specifically as a consequence to ensuring robustness and scalability in the face of the many failures that plague such distributed systems.

GHT makes use of a novel *perimeter refresh protocol* that ensures both persistence and consistency when nodes fail or shift their location while on the move. This protocol replicates stored data for ***key k*** at nodes around the location to which **k hashes**, and ensures that one node is chosen consistently as the *home node* for that *k*, so that all storage requests and queries for *k* can be routed to that node (*home node)*.

Eventhough the protocol is efficient; it typically relies on extensive utilisation of the local communication systems, especially when nodes are deployed densely. By hashing keys, GHT attempts to spread ***the storage and communication load*** between different keys ***evenly*** throughout the sensornet. When many events are stored ***with the same key*** as its reference, the GHT avoids creating a hotspot of communication and storage at their shared ***home node*** by employing *structured replication*, whereby events that hash to the same home node can be divided among multiplemirrors.

The DCS finds use in Data Management in the middleware for WSNs. Since WSNs can feature different paradigms for data routing (geographical routing as also the more traditional tree routing), this paper introduces ***two separate*** DCS protocols for these ***two different*** kinds of WNSs. Of these, the Q-NiGHT is based on geographical routing as it can manage the quantity of resources that areassignedtothestorageofdifferent*meta-data*,andimplementaloadbalance

for the data storage over the sensors in the WSN.

Designing the WSN Applications

The design of WSN applications is ***challenging*** since they normally have to deal with their ***own business logic***, and with the issues that naturally arise when WSNs are taken into account, such as network formation, data transport and data management, security and energy saving. This last requirement arises since nodes are typically battery-powered, and the energy budget of the nodes is limited, hence energy saving techniques are to be implemented to avoid energy starvation and the subsequent death of nodes. Dealing with these issues can be done either explicitly, thus adding complexity to the applications, or implicitly by means of a middleware, that is a software layer that abstracts from common issues of theWSNs.

Relevant WSN Features

The particular routing protocol that is used for WSN becomes a ***middleware parameter.*** The application developer then switches between different routing mechanisms at the time of network-creation, rather than at the time of designing the application. Extracting/ ***exacting*** a given high-level behaviour from a set of sensors is a challenging problem that gets compounded when explicitly dealing with WSN issues and the complexity can be overwhelming. In a complex scenario arising in WSN applications, a number of theoretical and practical problems emerges when the level of the abstraction of the middleware was deeply pinpointed. This thesis aims at analyzing some of them. For example, if an application wants to deliver data to more than one sink, or if the sink is inaccessible for some time, it can be useful to cache data into the network before sending them to the user application. From this scenario, Data Centric Storage (DCS) was born. The paradigm states that data are stored inside the network, in a subset of the sensors that is dependent on a meta-datum that describes the data, and the sinks query the WSN to retrieve the data they are interested in. DCS in general is considered as a useful mechanism to decouple data production from datacollection.

Wireless Sensor Networks applications

A large number of low-capacity sensors must be orchestrated into a common behaviour to give support to complex WSN applications. Despite the complexity that must be faced during the development, several kinds of applications are supported by WSNs, since they are flexible enough to be applied in diverse areas, such as: Critical Infrastructure Protection. According to the European Commission, Critical Infrastructures consist of "those physical and information technology facilities, networks, services and assets which, if disrupted or destroyed, would have a serious impact on the health, safety, securityoreconomicwell-beingofcitizensortheeffectivefunctioningof

governments in the Member States. Home systems and health monitoring. A "smart house" is usually made up of several intelligent devices that can control home appliances or "feel" their surroundings, and it can feature devices that intermix control on "intelligent" furniture and that monitor user position and/or state. This kind of infrastructure can lead to applications supporting rehabilitation after illness, such as providing assistive technology for cognitive rehabilitation and independent aging. Their technique incorporates a node's residual energy into the cost metric that is computed when determining what route to send packets on. The applications for such networks include inventory management, product quality monitoring, factory process monitoring, disaster area monitoring, biometrics monitoring, andsurveillance.

Perfect Data Centric Storage

A middleware based on Data Centric Storage (DCS) would represent a novelty into the WSN scene and could expand current WSN capabilities. On the other hand, the most studied implementations of DCS have limitations:

The set of sensors that store the data depends on the topology of the WSN, and it is not selectable by the WSN application developer. The storage primitive of DCS assumes uniform distribution of the sensors, and is thus inefficient when the sensors have a different distribution. Dependability is ensured by means of pure replication, in the sense that all the sensors that store a given meta-datum, have to store a copy of all related data. In this thesis such issues have been addressed by introducing new mechanisms for the Data Management layer of WSN middleware. Reviewed current Dependability mechanisms used for WSNs, and introduced a new way of exploiting erasure coding that can enhance the Data Management layer without increasing the complexity of DCS; Proposed a novel Data Management layer mechanism that combines gossip routing and erasure coding to disseminate data in a WSN in an efficient way. DCS systems can implement efficient in-network data storage and retrieval, since they require only unicast communications. However existing approaches disregard issues related to load balancing of the sensors, and QoS. The work on Q-NiGHT addressed these issues in WSNs that use a geographical routing protocol. In particular, Q-NiGHT uses a rejection hashing technique that produces pairs of coordinates by taking into account the actual sensor density, and then relies on a novel dispersal technique that enforces QoS and provides load balance. Simulation results show that Q-NiGHT significantly balances the storage load on the sensors and it adapts to different sensors' distributions. The Data Centric Storage systems can be combined with memory-efficient erasure codes. The use of erasure codes is however not immediate, since it requires the sensors to perform the encoding of the data before the storage (encoding that is not necessary in traditional DCS since they exploit pure replication). In fact it is necessary that each sensor be assigned with a coding parameter, and this assignmentiscriticalfromthepointofviewofcorrectdatacodingand

decoding. For this reason a probabilistic model was proposed, and it allowed the estimation of correct coding and decoding probability; this model was adapted to estimate this probability in three cases of study: local storage, DCS-GHT, and Q-NiGHT. From the analytical and simulative results, it was shown that correct coding/decoding can be achieved with high probability with the three systems for different network configurations. Finally, the research considered the application of erasure coding to spatial gossiping, and it showed that by using spatial gossip, it is possible to quickly establish an in-network erasure coding of the generated data. The gossip algorithm has a near linear message cost. Using erasure codes, generated data can be recovered from a subset of the nodes with high probability. The application of this storage scheme is useful to combat node failures and to help with efficient data collections with data mules. The thesis also presented a variation of the basic gossip algorithm in which the data mule can effectively reconstruct data as soon as it collects the code words. A future direction on DCS systems that take into account sensor distribution, as it is the case of Q-NiGHT, is the on-the-fly estimation of the distribution of the sensors, that can be exploited to better tailor the generalized hash function that associates meta-data to locations in the sensing area and to deal with topology changes due to sensors' failures and/or sensors'mobility.

References:

[1] P. .A. Dinda: Archetype-based design: Sensor network programming for application experts, not just programming experts. In: IPSN '09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks, p. 85-95(2009)

[2] An Evaluation Framework for middleware approaches on WSN. In: Seminar on Internetworking, Helsinki University of Technology, Finland, April 27th2009

[3] S. Chessa: Sensor Network Standards. (book). In: J. Zheng and A. Jamalipour, Wireless Sensor Networks: A Networking Perspective, Wiley-IEEE Press, ISBN: 978-0-470-16763-2, p. 407-431, September2009