Geographic Hash Tables with QoS in non Uniform Sensor Networks

Michele Albano[†], Stefano Chessa[†‡], Francesco Nidito[†], Susanna Pelagatti[†]

1 Introduction

A wireless sensor network is composed by a large number of low power, low cost sensors (also called nodes) [1] which self organize into a (multi-hop) ad hoc network. A sensor is a micro-system which also comprise one or more sensing units, a radio transceiver and an embedded battery.

Sensors are spread in an environment (the *sensor field*) without any predetermined infrastructure and cooperate to execute common monitoring tasks which usually consists in sensing environmental data from the surrounding environment. Sensed data are collected by an external sink node when it is available (connected to the network). The sink node, which could be either static or mobile, is in turn accessed by the external operators to retrieve the information gathered by the network.

As the sensor network scales in size, so does the amount of sensed data which should be processed and collected by the network. In the effort to provide efficient access to data and to tolerate disconnections between the network and the sink node, it has been recently proposed the use of Geographic Hash Tables (GHT) to implement a Data Centric Storage (DCS) within the network[4].

GHT uses a hash function to map a data name to a geographic position attempting to distribute data uniformly across the network. GHT hashes the name of data to be stored to a location s in the sensor field. Then, the perimeter mode of GPSR [2] is used to select the closest sensor to s, which becomes the *home node* for the data. GHT stores a copy of the data in the home node as well as in all the sensors belonging to the perimeter. Storing on all the perimeter is essential to guarantee data persistency also in presence of node faults. GPSR can later be used to locate the home node given the geographic position of data.

Generally speaking, we can define the QoS of GHT as the dependability required by the data, which in turn may be expressed using different metrics and ranges according to the particular redundancy technique used. For instance, using pure replication we may have different levels of QoS depending on the number of actual replicas ensured for a given datum. It should be observed that GHT methods have no control over the 'quality of service' (QoS) deserved for a given datum. In fact, although the average level of redundancy of a data (i.e. average number of sensors for storage associated to a data) is constant, in practice it can vary significantly due to the fact that each geographic point is surrounded by a different perimeter.

A second problem with GHT is that it uses an uniform hashing function independently of the real distribution of sensors. This can bring to heavily unbalanced networks. The load unbalance is due to the different amount of data that must be managed by an equal number of sensors. Consider the following example. Suppose that the distribution function of the sensor network is a *Gaussian* and the hash function is uniform and its range is defined by the size of the *Gaussian* at the 0.99 percentile. The hash function places data with equal probability in all the deployment area. If we take two squares, with same surface, but the first one in one corner (S_{corner}) and the second in the center (S_{center}) of the

[†]Dipartimento di Informatica, Univarsità di Pisa, Italy
[‡]Istituto di Scienza e Tecnologie dell'Informazione, Area della Ricerca CNR di Pisa, Italy
emails: `albano@cli.di.unipi.it` `{ste, nids, susanna}@di.unipi.it`

Work funded in part by the European Commission in the framework of the "SatNEx II" NoE project (contract N. 27393).

deployment area, the hash function places the same number of data, n, in both squares. The problem is that in S_{corner} the number of sensors is much smaller than the number of sensors in S_{center}, thus a sensor on the border of the deployment area must manage a quantity of data that is larger than the quantity managed by a sensor in the center of the network.

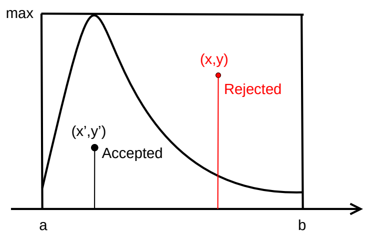
We approach these problems by introducing a new protocol (Q-NIGHT) which combines non-uniform hashing with new strategies for data distribution and retrieval.

2 Hashing functions for non uniform distributions

Our proposed solution to the problem of building a balanced GHT on networks with non-uniform distribution is to have hash functions which scatter data *approximately* with the same distribution as sensors. To do this we start with the consideration that an hash function $h(K)$ is a kind of pseudo-random number generator: starting from a seed (in our case the key K) it produces an output (in our case in a value in $\mathbb{R}^2$) such that for *near* values of key the hash values must be *distant*.

With this consideration in mind we solve the problem using as new hash function that use a strategy similar to the one used in the *rejection method*[3].

```
RejectionHash(K, f): <x,y>
begin
   i = 0;
   while(true)
     let <x,y,z> uniformly hashed form K+i in box(f)
     i = i+1
     if z < f(x,y) return <x,y>
end
```



In function `RejectionHash` the probability function `f` is boxed and we generate uniform hash values form the key in the box. If the value generated is below the distribution function the value is accepted and returned. Otherwise we randomly generate new points in the box until values are below the function. At each iteration the value is changed by a suitable operation (in our pseudo-code denoted with the + operator) to guarantee that the hash result will change at each iteration and to guarantee that the change is deterministic. We use the rejection method because h must be generic and must be distribution independent.

3 Quality of Service in GHT

We now describe Q-NIGHT protocols implementing data insertion (`put`) and data retrieval (`get`) on the sensor network. In the following we denote with $B_{(x,y)}(r)$ the *ball* centered in (x, y) of radius r, that is the set of sensors that are within a euclidean distance r from (x, y).

To our purposes the interface of the `put` includes, along with the meta-data K and the data D, also a parameter Q expressing the desired QoS. Q-NIGHT adopts pure replication and Q expresses the number of sensors on which the data should be replicated.

Data insertion is involved with `put(K,D,Q)`. We assume Q ranges in $[1, Q_{max}]$ and gives the number of sensors on which the data should be replicated.

Let s be the source node of a `put(K,D,Q)` operation. s firstly computes $H_c(K)$, the hash function conditioned with the sensor distribution in the sensing field, as discussed in Section 2. $H_c(K)$ returns a pair of geographic coordinates (x, y) as the destination of the packet $\texttt{P}_p$=`<(x,y),<K,D,Q>>`.

The packet in turn is sent to the destination using the GPSR protocol. As in GHT we call *home node* the sensor d (of coordinates (x', y')) geographically nearest to the destination coordinates. The home node naturally receives the packet as a consequence of applying GPSR.

Upon reception of packet $\texttt{P}_p$, sensor d begins a *dispersal protocol* which selects Q sensors to store a copy of `<K,D>`.

The dispersal protocol is iterative. In the first iteration d broadcasts a replica of `D` to all the sensors included in the ball $B_{(x',y')}(\overline{r})$. $\overline{r}$ is chosen in order to reach the Q sensors nearest to (x, y) with high probability.

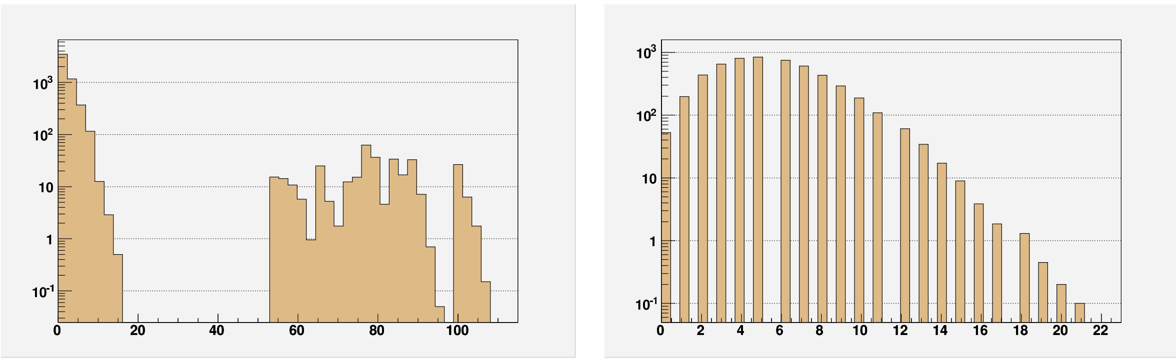


Figure 1: Average load of sensors of GHT (left) and Q-NIGHT (right) with Gaussian distribution of sensors and Gaussian hashing

Each sensor receiving a replica responds with an acknowledgment to d.

Sensor d confirms the $Q-1$ acknowledgments received from the sensors geographically nearest to (x, y) and disregards the others. The confirmation requires an extra packet sent by d. Sensors which receive the confirmation keep the data while the other sensors will disregard the data after a timeout. If d receives $Q' < Q$ acknowledgments, then it executes another iteration of the dispersal protocol with $\overline{r} = 2\overline{r}$ in which it considers only the sensors in $B_{(x',y')}(2\overline{r}) - B_{(x',y')}(\overline{r})$.

The dispersal protocol stops as soon as Q sensors have been hired or the outermost perimeter has been reached.

When a node s of coordinates (r, z) executes `get(K)` it firstly computes $(x, y) = H_c(K)$, and sends a query packet $\mathtt{P}_g$`=<(x,y),<(r,z),K>>` using the GPSR protocol. In turn, packet $\mathtt{P}_g$ will reach the perimeter surrounding (x, y) and it will start turning around the perimeter. Eventually, the packet will reach either the home node or another node containing a replica of the data D associated to K. This node will stop packet $\mathtt{P}_g$ and will send the required data back to s.

4 Discussion and Conclusions

Preliminary simulations show that the non-uniform hash function discussed in Section 2 archives good load balancing. Figure 1 shows on the x the different load (number of data) on each node and on the y axis the number of nodes storing exactly this number of data* . In the Figure we can see that in GHT the load is quite unbalanced. On the other side Q-NIGHT is able to produce a more balanced network load with a better distribution of data. Other simulations also show that `put` and `get` operations do not introduce overhead with respect to GHT.

Future work on protocol Q-NIGHT will consist in an extensive testing activity with protocol implementation an simulation with the ns-2 simulator. We will test Q-NIGHT with different configurations of non-uniform networks and with different metrics of QoS. We will also study techniques and protocols to measure the nodes distribution on the sensing field.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[2] Brad Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the 6th MobiCom*, pages 243–254, Boston, MA, USA, August 2000.

[3] J. Von Neumann. Various techniques used in connection with random digits. In A. H. Taub, editor, *John von Neumann, Collected Works*, volume 5, pages 768–770. Pergamon Press, Oxford, 1951.

[4] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-centric storage in sensornets with GHT, a geographic hash table. *Mob. Netw. Appl. (MONET)*, 8(4):427–442, 2003.

*Values on the y axis follow a logarithmic scale for better comprehension