

# A Deterministic Protocol for Permutation Routing in Dense Multi-Hop Sensor Networks

Alain Bertrand Bomgni<sup>1</sup>, Jean Frédéric Myoupo<sup>2</sup>

<sup>1</sup>Department of Computer Science, University of Yaounde 1, Yaounde, Cameroon

<sup>2</sup>Department of Computer Science, University of Picardie Jules Verne, Amiens, France

E-mail: [alain.bomgni@yahoo.fr](mailto:alain.bomgni@yahoo.fr), [jean-frederic.myoupo@u-picardie.fr](mailto:jean-frederic.myoupo@u-picardie.fr)

Received January 16, 2010; revised February 20, 2010; accepted March 4, 2010

## Abstract

A large variety of permutation routing protocols in a single-hop Network are known in the literature. Since they are single hop, there is always a wireless link connecting two nodes. One way to solve this problem in a multiple hop environment is to partition nodes into clusters, where a node in each cluster called clusterhead is responsible for the routing service. In this paper, we propose a hybrid clustering mechanism to perform permutation routing in multi-hop ad hoc Networks. We first propose to partition the network in single-hop clusters also named *cliques*. Secondly, we run a local permutation routing to broadcast items to their local destinations in each clique. Next we partition the clusterheads of cliques with the hierarchical clustering technique. We show how the outgoing items can be routed to their destination cliques. We give an estimation of the number of broadcast rounds in the worse case. More precisely, we show that solving the permutation routing problem on a multi-hop sensor network need  $(n/p)|HUB_{max}|(k+1) + O(|HUB_{max}|) + k^2 + 2k$  in the worse case. Where  $n$  is the number of the data items stored in the network,  $p$  is the number of sensors,  $|HUB_{max}|$  is the number of sensors in the clique of maximum size and  $k$  is the number of cliques after the first clustering. Finally, simulation results show that our algorithm performs better than the naïve multiple gossiping. To the best of our knowledge, it is the first algorithm for permutation routing in multi-hop radio networks.

**Keywords:** Permutation Routing Problem, Wireless Sensor Networks, Hierarchical Clustering, Clique

## 1. Introduction

Wireless sensor networks (WSN for short) can be deployed to provide continuous surveillance over an area of interest, referred to as a sensor field [1,2]. Wireless sensor nodes perform collaborative work [3] via wireless communication channels to retrieve information about targets that appear in the sensor field or to exchange some information. Higher-level decision making can then be carried out based on the information received from the sensor nodes. These networks can be deployed in inhospitable terrain or in hostile environments to provide continuous monitoring and information. In some applications, a sensor may need to get all its own data items that should help him to accomplish its mission. In such applications and for confidential reasons, during the deployment, a sensor may hold items which are not necessary its own. A permutation routing between sensors

permits that every sensor can recover its own items in order to perform the task that has been assigned to him.

There are two types of wireless networks: Single hop wireless networks in which each station can transmit or communicate directly with any other station. All the stations use the same channel to communicate, and the message broadcast by one of the stations on the common channel is simultaneously heard by all other stations. In the multi-hop wireless networks intermediate nodes are used to route message from the source to the destination. In this paper only multi-hop wireless networks are considered.

*Permutation Problem:* Consider a wireless sensor network of  $p$  stations with  $n$  items pretitled on it (WSN( $n, p$ ) for short). Each item has a unique destination which is one of the  $p$  stations. Each station has a local memory of size  $O(n/p)$  in which  $n/p$  items are stored. It is important to note that in general, some of the  $n/p$

items stored in the station, say  $i$ , have not  $i$  as destination station. And even, it can happen that none of these  $n/p$  items belongs to it. On the other hand, the situation in which initially all items in  $i$  belong to  $i$  can also occur. The permutation routing problem is to route the items in such a way that for all  $i$ ,  $1 \leq i \leq p$ , station  $i$  contains all its own items at final.

### 1.1. Related Work

The number of studies specifically targeted to permutation routing in single hop wireless networks has grown significantly: It is shown in [4] that the permutation routing of  $n$  items pretitled on wireless sensor network of  $p$  stations and  $k$  channels with  $k < p$ , can be carried out efficiently if  $k \leq \sqrt{p/2}$ . In [5], we solve the problem showing how the above restriction can be left. Datta in [6] derived a fault tolerant permutation routing protocol of  $n$  items pretitled on mobile Ad-hoc network of  $p$  stations and  $k$  channels MANET( $n, p, k$ ) for short. He also assumed that  $k \leq \sqrt{p/2}$  and in the presence of faulty stations some data items are lost. We came out with our work in [7] presenting a fault tolerant protocol which avoids the lost of items. The first energy-efficient permutation routing appeared in [8]. A more efficient energy-efficient permutation routing protocol was presented in [9]. In [10] Walls et al. propose an optimal permutation routing on mesh networks. Another approach as an application of an initialization algorithm appeared in [11]. All these approaches assume that the WSN is a single hop networks. Our former work in [12] presents a randomized algorithm for the same problem in multi-hop network with high probability.

### 1.2. Our Contribution

We consider a WSN ( $n, p$ ) with  $n$  items,  $p$  stations. We first propose to partition the network into single-hop clusters also named *cliques*. Secondly, we run a local permutation routing to broadcast items to their local destinations in each clique. Next we partition the cluster-heads of cliques with the hierarchical clustering technique. We show how the outgoing items can be routed to their destination cliques. We give an estimation of the upper bound of the number of broadcast rounds in the worse case.

The rest of this work is organized as follows. Some definitions and the environment considered in this work are presented in Section 2. In Section 3, we present some useful preliminaries. The permutation routing is described in Section 4 followed by the simulation results in Section 5. A conclusion ends the paper.

## 2. Basic Definitions

A Wireless Sensor Network is a set  $S$  of  $n$  radio trans-

ceivers or sensors which can transmit and/or receive messages from each other. The time is assumed to be slotted and all sensors have a local clock that keeps synchronous time. In any time slot, a sensor can tune into one channel and/or broadcast on at most one channel. A broadcast operation involves a data packet whose length is such that the broadcast operation can be completed within one time slot. So, in the WSN with Collision Detection (CD for short), the status of an  $n$ -station WSN channel is:

-NULL: If no station broadcasts on the channel in the current slot,

-SINGLE: If exactly one station broadcasts on the channel and

-COLLISION: If two or more stations broadcast on the channel in the current time slot.

Also, all the communications are performed at time slots boundaries *i.e.* the duration operation is assumed to be sufficiently short.

1) All communications are over wireless links. A wireless link can be established between a pair of nodes only if they were within wireless range of each other. Two sensors that have a wireless link, will be said to be 1-wireless hop away from each other. They are also called neighbors.

2) Each sensor belonging to a cluster is a resident of that cluster. Hence, this sensor may in a given time unit, broadcast a message to its neighbours.

#### Definition 1

Let us consider  $p$  stations  $1, 2, \dots, p$  which communicate in a multi-hop wireless sensor network WSN ( $n, p$ ). We suppose that we have  $n$  items in the system. Then each station of a WSN ( $n, p$ ) is assumed to have a local memory of size at least  $O(n/p)$ .

#### Definition 2

We suppose that the  $n$  items denoted  $a_1, a_2, \dots, a_n$  are pretitled on a WSN( $n, p$ ) such that for every  $i$ ,  $1 \leq i \leq p$ , station  $i$  stores the  $n/p$  items. Each item has a unique destination station. It is important to note that hereafter a station knows the destination of items it holds. In fact the data item it holds is a couple  $(a(v), v)$ . Where  $a(v)$  is the real data item belonging to sensor  $v$ . For every  $v$ ,  $1 \leq v \leq p$ , let  $h_v$  be the set of items whose destination is sensor  $v$ .

The permutation routing problem is to route the items in such a way that for all  $v$ ,  $1 \leq v \leq p$ , sensor  $v$  contains all the items in  $h_v$ . Consequently, each  $h_v$  must contain exactly  $n/p$  items (see **Figure 1** for an example).

## 3. Preliminaries

We now describe some tools that we will use to derive our permutation routing algorithm. A practical way of tackling the permutation routing problem would be to build a hierarchical structure above the network in order to simulate a sort of backbone made up of nodes which are

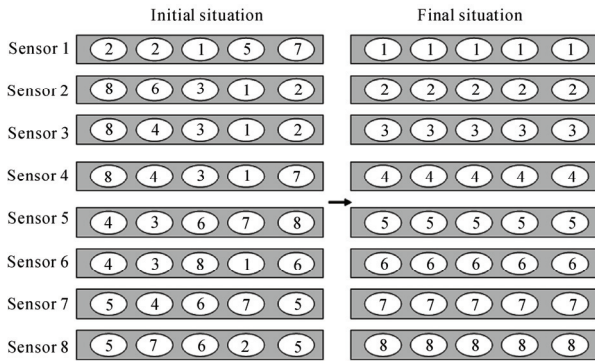


Figure 1. Permutation routing with  $n=40$  and  $p=8$ .

more “adapted” than others. This is precisely the goal of clustering. This methodology has already proven its efficiency in the past. In sensor networks the sensor nodes can be partitioned into *clusters* by their physical proximity to achieve better efficiency, and each cluster may elect a *clusterhead* to coordinate the nodes in the cluster. Certain references say that clustering with at most two hops is said to be *node-centric*[13], whereas clustering with over two hops is called *cluster-centric* [14-16]. In *node-centric* approach, clusterheads are first elected and a procedure indicates how to assign other nodes to different clusters. In *Cluster-centric* approaches, clusters are first formed, and each cluster then elects its clusterhead. Such approaches require that all the nodes in one cluster agree on the same membership before electing their clusterhead. We now summarize two clustering schemes that will be helpful to describe our permutation routing protocol.

### 3.1. A Clustering Scheme in Cliques

Our approach uses one of the protocols from [17,18] to partition network into clusters (cliques). The figure below illustrates a network in which each clique is a single hop sub network. Each clique is a single hop network.

### 3.2. Hierarchical Clustering

Banerjee and Khuller [14] proposed a clustering algorithm for multi-hop sensor networks.

Their clustering scheme is motivated by the need to generate an applicable hierarchy for multi-hop wireless environment. Their method yields a multi-stage clustering. To reach their goal they construct a depth first search tree such that each level is composed of clusterheads of the immediate low level. These clusters are by definition disjoint and the number of the nodes in a cluster remains between  $k$  and  $2k$  for some integer  $k$ . **Figure 3** shows a hierarchical clustering of a network of 25 sensors with  $k = 3$ .

## 4. Permutation Routing Protocol

We recall that we have  $p$  sensors and  $n$  data items pre-titled in these  $p$  sensors. Hence each sensor has a locale memory of size  $O(n/p)$ . The time is slotted. Our Approach to provide permutation routing in multi-hop sensor network consists of the following five phases.

### 4.1. Phase 1: Clustering Procedure in Cliques

The sensors run one of the protocols in [17,18] to create

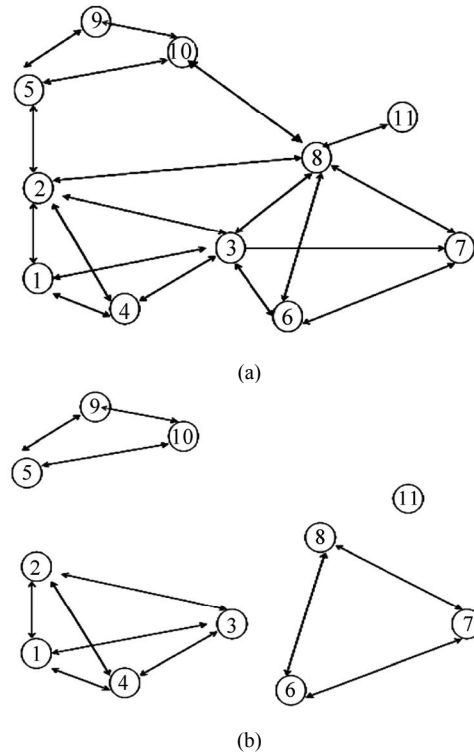


Figure 2. (a) Network with 11 sensors; (b) Resulting cluster formation in cliques.

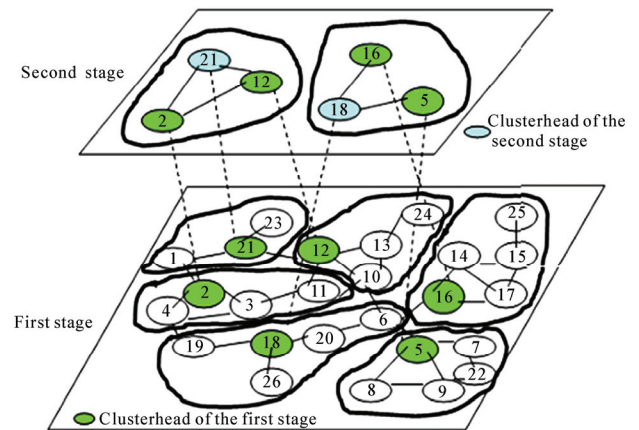


Figure 3. Hierarchical clustering with  $k = 3$ .

cliques like clusters. We assume that this phase yields  $k$  cliques (clusters), hence  $k$  cluster heads named  $CH_{\text{clique-}i}$  for the clusterhead of clique  $i$ . In fact clique  $i$  is a hub for its local sensors and will be named  $HUB(i)$ . The role of clusterhead  $CH_{\text{clique-}i}$ , is to collect all data items whose destination sensors are in  $i$ . Note  $HUB_{\text{max}}$  the clique that contains the maximum of sensors. Initially  $HUB(i)$  may contain outgoing items *i.e.* items whose destinations are not in  $HUB(i)$ . Thus the goal of the phases that follow is to describe a mechanism that permit to each sensor to collect all its own data items.

#### 4.2. Phase 2: Local Broadcasts in Cliques

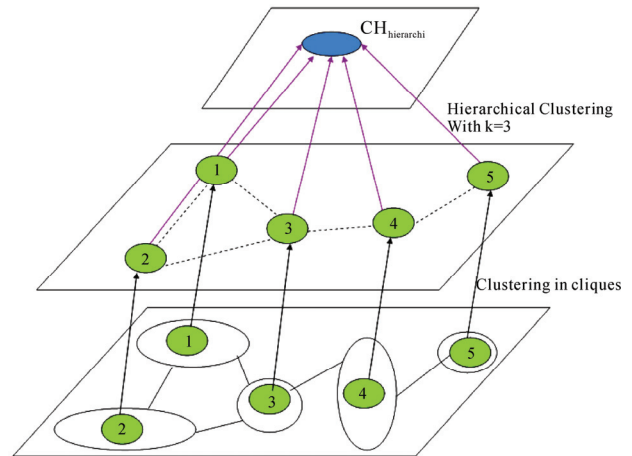
The idea of this phase is similar to the protocol single-channel-routing in [4]. The broadcast item here is a couple  $(a(v), v)$  where  $a(v)$  is the data item belonging to sensor  $v$ . It can be summarized as follows.  $CH_{\text{clique-}i}$  invites each node of  $HUB(i)$  to broadcast one by one the data items it holds. In each slot, the sensor whose identity matches the destination of the item being broadcast copies the item in its local memory. If no sensor of the clique is the destination of the broadcast data item then the clusterhead ( $CH_{\text{clique-}i}$ ) copies it in its *outgoing\_local\_memory*. Note that the cluster head has the IDs of all the residents of its cluster. The broadcasts are carried out on cliques. So the clique with the great number of sensors should help to estimate the total broadcast rounds of this phase. At the end of this phase all data items that do not belong to the sensors of a clique are saved by the clusterhead. The goal now is to route them to their final destinations.

#### 4.3. Phase 3: Hierarchical Clustering

The clustering in cliques of the first phase gives  $k$  cliques, thus  $k$  clusterheads. Now we focus only on these clusterheads and consider a network, say  $G'$ , whose sensors reduce to these  $k$  clusterheads. Clearly  $|G'| = k$ . Partition this network as in Subsection 3.2 using the hierarchical clustering such that each resulting cluster contains at least  $k/2$  sensors and at most  $k$  sensors. Hence the partition will give only one cluster, and thus one clusterhead. This clusterhead knows the IDs of all residents of its cluster, *i.e.* the IDs of the other  $k-1$  sensors (see **Figure 4**).

#### 4.4. Phase 4: Ordering Sensors for Broadcasting Outgoing Data Items

The clusterhead of the above hierarchical clustering, say  $CH_{\text{hierarchi}}$  knows all the residents of its cluster, thus has their IDs named  $CH_{\text{HUB}(i)}$  in phase 1.  $CH_{\text{hierarchi}}$  or



**Figure 4.** An example hybrid clustering: The first stage shows an example of clustering in cliques. The second stage shows the hierarchical clustering of  $G'$ .

$CH_{\text{HUB}(1)}$  broadcasts one by one the IDs of the  $CH_{\text{HUB}(i)}$ ,  $1 \leq i \leq k$  in the network  $G'$ . Each  $CH_{\text{HUB}(i)}$  counts the number of IDs that are less than its own ID. At the end of the process each  $CH_{\text{HUB}(i)}$  knows its rank in the set of the IDs in  $G'$ .

#### 4.5. Phase 5: Broadcasting Outgoing Data Items

##### 4.5.1. Step1: Collecting the Rest of the Data Items of a HUB

In this phase  $CH_{\text{HUB}(i)}$  or  $CH_{\text{hierarchi}}$  invites the sensors of  $G'$  with the lowest ID to initiate the broadcasts. It then broadcasts its outgoing data items one by one every two slot to the sensors in  $G$ . More precisely, when an item is broadcast in slot  $\alpha$ , the next slot  $\alpha + 1$  is not used and the next broadcast is carried out in slot  $\alpha + 2$ . It accompanies its last item with a special mark indicating that it is its last outgoing item. On receiving this last item, the sensor of second rank takes over the broadcasts. The sensor third rank takes over the broadcasts when it receives the last item from the one of second rank and so on till the sensor of the higher rank is reached. In this phase, an item whose destination matches with the ID of a sensor in *clique- $i$*  is saved by  $CH_{\text{HUB}(i)}$  in its local memory. During this process all sensors which are not in  $G'$  are inactive except gateway nodes. At the end of this phase the rest of the data items of  $HUB(i)$  is saved in the local memory of  $CH_{\text{HUB}(i)}$ .

##### 4.5.2. Step2: Broadcasts of the Rest of the Data Items of a HUB

In  $HUB(i)$ ,  $CH_{\text{HUB}(i)}$  broadcasts one by one the data items collected in step one above and whose destinations are sensors of  $HUB(i)$ .

The protocol is summarized by the pseudo code below:

---

**Protocol-Permut-Routing- muti-hop-Wsn**


---

**INPUT:** wsn of  $p$  sensors in which a sensor may not have its own data items

**OUPUT:** wsn in which each sensor has its own data items

**Begin**

1. run the maximum clique formations protocol in [17 or 18] to obtain HUBS
2. in parallel on HUBS, run the protocol single-channel-routing of [4] for single hop wsn
3. run the hierarchical clustering algorithm in [14] with  $k = \lfloor G'/2 \rfloor$  on the dominating set,  $G'$  derived from 1.
4. Broad casts of outgoing data Items
  - 4.1 Order sensors for broadcasting outgoing data items
  - 4.2 collect the rest of data items of a HUB
  - 4.3 broadcasts of the rest of the data items of a HUB

**END**

---

**Lemma**

Let  $p$  sensors in a multi-hop sensor network  $(n, p)$  with  $n$  items pretitled on it. Without clustering broadcast rounds, the permutation routing problem can be solved in  $(n/p) \lfloor HUB_{\max} \rfloor (k+1) + O(\lfloor HUB_{\max} \rfloor) + k^2 + 2k$  in the worse case.

**Proof:**

1) *Evaluation the broadcast rounds of phase 2*

Since  $HUB_{\max}$  detains the maximum number of sensors, it should need the maximum number of broadcast rounds in this phase. Therefore local broadcasts in cliques need  $O(\lfloor HUB_{\max} \rfloor)$  broadcast rounds (see [4]).

2) *Evaluation the broadcast rounds of phase 4*

There are  $k$  sensors in  $G'$  ( $k = \lfloor G' \rfloor$ ). The clusterhead of  $G'$  knows all its residents and must broadcast their Ids. Therefore  $k$  broadcast rounds are necessary for this scenario. However the last broadcasted ID has to travel through the network diameter, *i.e.*  $k$  in the worse case. Hence  $2k$  broadcast rounds are necessary to achieve this phase in the worse case.

3) *Evaluation the broadcast rounds of phase 5*

It is important to note that the upper bound of the number of items stored in each *outgoing\_local\_memory* is  $(n/p) \lfloor HUB_{\max} \rfloor$ . Since we have to broadcast all data items of the *outgoing\_local\_memories*,  $(n/p) \lfloor HUB_{\max} \rfloor$  broadcast rounds are necessary to broadcast all items from these *outgoing\_local\_memories*. The special mark indicating the end of broadcasts for a sensor is broadcast  $k \times (\text{diameter of } G')$  times. It reduces to  $k^2$  in the worse case. Therefore Step1 of this phase is completed in  $(n/p) k \lfloor HUB_{\max} \rfloor + k^2$  broadcast rounds in the worse case. Step2 takes  $(n/p) \lfloor HUB_{\max} \rfloor$  broadcast rounds in

the worse case. Therefore this phase is completed  $(n/p) \lfloor HUB_{\max} \rfloor (k+1) + k^2$  broadcast rounds in the worse case.

Summing up these numbers of broadcast rounds, we get the result.

## 5. Simulation Results

In this section, we present simulations results of the clustering algorithm to show the influence of the heuristic used to choose the clusterhead. These benchmarks have been run on a laptop (Pentium-m 1.7 GHz, 1 GO RAM, Windows XP SP 2, Cygwin 1.5.19) and programmed in C++. Our main problem has been to establish suitable experiments conditions. Our main issue was to set up a quite realistic mobility scenario. As WSNs are supposed to be used in rescue services, we can assume that nodes are static. Random numbers have been obtained using the well-known Mersenne Twister random numbers generator. All nodes are assumed to have the same transmission range. The experiments take place in a geographic square area of side  $L$ . In the simulation depicted by the **Figure 2**, we measure the instantaneous average number of clusters generated by the algorithm with up to 300 nodes. The presented curves are the average of 100 experiments. We have made the common assumption that two nodes are neighbors if and only if their Euclidean distance is lower than 1 Km. The nodes are in a square, which the length is  $L = 2$  km. In our implementation, the MAC layer is managed in such a way that a node can only receive one message at a time, yielding delays in the clustering process and so maintaining always a high number of clusters.

### 5.1. Average Number of Cliques

Hereafter we will need the number of cliques for the simulations. **Figure 5** shows the evolution curve of the average number of cliques with respect to the number of sensors. In a square of side 2 km, the average number of cliques evolves slowly. This situation is due to the fact that, according to the broadcast range, with a little bit more than 200 sensors, the total number of cliques in this square is almost reached. Therefore the new incoming sensors should join one of the existing cliques.

### 5.2. Average Number of Broadcast Rounds after Clustering

The number of data items is fixed to 1000. As expected, the simulation results of the **Figure 6** show that there is a significant gap between the naïve gossiping heuristic and our protocol. The procedure of gossiping used here is for



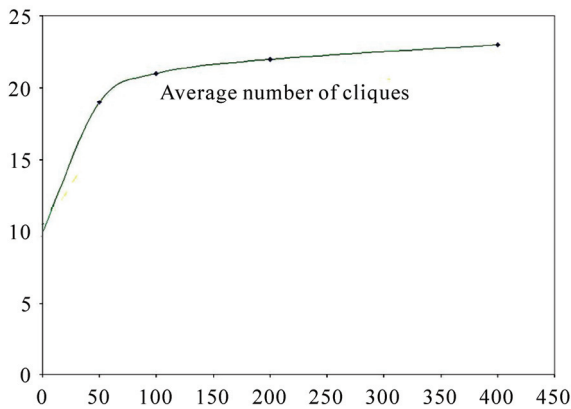


Figure 5. Average number of cliques.

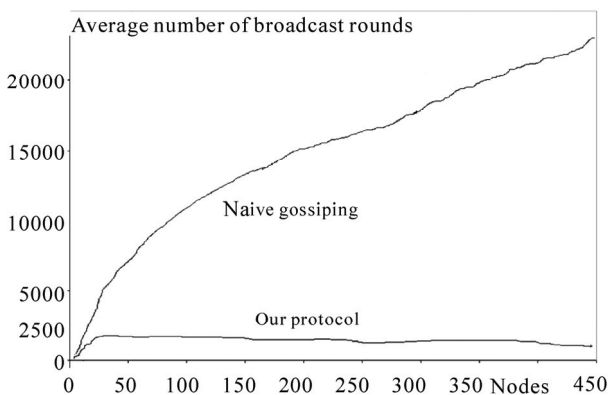


Figure 6. Comparative curves of the average numbers of broadcast rounds.

each sensor to broadcast its items one by one. Thus we have  $n/p$  simultaneous broadcasts. It is important here to note that each sensor has a collision detection capability. Our protocol performs better than the above naïve heuristic. One important remark is that when the number of nodes increases, the number of broadcast rounds decreases slowly. As we see above the number of cliques is stable when  $p$  increases. Therefore when  $p$  increases  $n/p$  decreases (since  $n$  is fixed) and the number of items to be broadcast by each sensor decreases.

## 6. Conclusion and Discussions

In this paper we show that solving the permutation routing problem on a multi-hop sensor network need  $(n/p)|HUB_{\max}|(k+1) + O(|HUB_{\max}|) + k^2 + 2k$  broadcast rounds in the worse case. Where  $n$  is the number of the data items stored in the network,  $p$ , is the number of sensors,  $|HUB_{\max}|$  is the number of sensors in the clique of maximum size and  $k$  is the number of cliques after the first clustering. Finally, simulation results show

that our algorithm performs better than the naïve multiple gossiping. To the best of our knowledge, it is the first algorithm for permutation routing in multi-hop radio networks.

The reader may wonder why is the assumption about dense networks necessary. To address this point, notice that if  $k \times |HUB_{\max}| \ll p$  then the number of broadcast rounds reduces to  $O(n)$ . And we have the same performance as the one of the protocol single-channel-routing in [4], in single hop networks.

However some open problems remain. The derivation from the idea of this paper of a fault tolerant algorithm, which guarantees the delivery of data items to non faulty nodes, is to be investigated. Also, the construction of an energy-efficient permutation routing protocol for multi-hop ad hoc network is a challenge. Another challenging problem is to secure the protocol proposed in this paper.

## 7. Acknowledgment

Thanks to the anonymous referees for their constructive comments and suggestions.

## 8. References

- [1] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, Seattle, 1999, pp. 263-270.
- [2] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, Vol. 43, No.5 2000, pp. 51-58.
- [3] F. Zhao, J. Liu, J. Liu, L. Guibas and J. Reich, "Collaborative Signal and Information Processing: An Information Directed Approach," *Proceedings of the IEEE*, New York, Vol. 91, No. 8, 2003, pp. 1199-1209.
- [4] K. Nakano, S. Olariu and J. L. Schwing, "Broadcast-Efficient Protocols for Mobile Radio Networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 10, No. 12, 1999, pp. 1276-1289.
- [5] F. Myoupo, "Concurrent Broadcasts-Based Permutation Routing Algorithms in Radio Networks," *IEEE Symposium on Computers and Communications*, 2003, pp. 1272-1278.
- [6] A. Datta, "Fault-Tolerant and Energy-efficient Permutation Routing Protocol for Wireless Networks," *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, Nice, 2003, pp. 22-26.
- [7] D. Karimou and J. F. Myoupo, "A Fault Tolerant Permutation Routing Algorithm in Mobile Ad Hoc Networks," *International Conference on Networks-Part II*, 2005, pp. 107-115.
- [8] K. Nakano, S. Olariu and A. Y. Zomaya, "Energy-

- Efficient Permutation Routing in Radio Networks,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 6, 2001, pp. 544-557.
- [9] A. Datta and A. Y. Zomaya, “An Energy-Efficient Permutation Routing Protocol for Single-Hop Radio Networks”. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 4, 2004, pp. 331-338.
- [10] I. S. Walls and J. Žerovnik, “Optimal Permutation Routing on Mesh Networks,” *International Network Optimization Conference*, Belgium, 22-25 April 2008.
- [11] D. Karimou and J. F. Myoupo, “An Application of an Initialization Protocol to Permutation Routing in a Single-hop Mobile Ad-Hoc Networks,” *Journal of Supercomputing*, Vol. 31, No. 3, 2005, pp. 215-226.
- [12] D. Karimou and J. F. Myoupo, “Randomized Permutation Routing in Multi-hop Ad Hoc Networks with Unknown destinations,” *IFIP International Federation of Information Processing*, Vol. 212, 2006, pp. 47-59.
- [13] S. Basagni, “Distributed Clustering for Multi Hop Wireless Networks,” *Proceedings of the IEEE International Symposium on Wireless Communications*, Victoria, 3-4 June 1999, pp. 41-42.
- [14] R. Wattenhofer, S. Khuller, *et al.*, “A Clustering Scheme for Hierarchical Control in Multi-Hop Wireless Networks,” *Proceedings of the 20th IEEE International Conference on Computer Communications*, Vol. 3, 2001, pp. 1028- 1037.
- [15] C. L. Fullmer and J. J. Garcia-Luna-Aceves, “Solutions to Hidden Terminal Problems in Wireless Networks,” *Sigcomm*, September 1997.
- [16] A. B. McDonald and A. Zanati, “A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks,” *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, 1999, pp. 1466-1487.
- [17] K. Sun, P. Peng and P. Ning, “Secure Distributed Cluster Formation in Wireless Sensor Networks,” *22nd Annual Computer Security Applications Conference*, Las Vegas, 2006, pp. 131-140.
- [18] P. Tosic and G. Agha, “Maximal Clique Based Distributed Coalition Formation for Task Allocation in Large-Scale Multi-Agent Systems,” *Lecture Notes in Computer Science*, Vol. 3446, 2005, pp. 104-120.