

An Efficient Data Collecting Network Structure in Wireless Sensor Networks

Chi-Tsun Cheng, Chi K. Tse and Francis C. M. Lau

Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong

Email: {enccheng, cktse, encmlau}@eie.polyu.edu.hk

Abstract—In a sensor network, a single event may be captured by hundreds of sensor nodes simultaneously. Data in these sensor nodes are thus highly correlated. Data transmission can be facilitated by employing data aggregation techniques. The traditional way to do data aggregation in a sensor network is to divide the network into clusters and elect one node as the cluster head in each cluster. These cluster heads are responsible for collecting data from other nodes within its belonging cluster. Collected data will first be fused before being forwarded to the base station. Due to the design limitations, most nodes in a sensor network are only capable of handling a single connection at a time. Therefore, to collect data from n cluster members, a cluster head will need at least n time slots. In this paper, a special network structure and its formation algorithm are proposed. Simulation results show that the proposed network structure can greatly reduce the delay in data collection.

I. INTRODUCTION

Thanks to the advanced electronic technology, wireless sensor nodes can nowadays be produced in compact size and at low cost, bringing the development of sensor networks from a theoretical consideration to practical implementation. In a sensor network, thousands of wireless sensor nodes are deployed randomly over a sensing field. Sensor nodes collect data from the sensing field and report them to the base station. Since sensor nodes are usually deployed in places which are inaccessible, battery replacement or recharge are impractical. Thus, prolonging network lifetime becomes one of the most important issues in the field of sensor networks.

Much prior work has focused on extending network lifetimes by clustering. A network with clustering is divided into several clusters. Within each cluster, one of the sensor nodes is elected as the *cluster head*, the rest being the *cluster members*. The cluster head will collect data from its cluster members. Collected data will first be aggregated before being forwarded to the base station. Network lifetime is extended by reducing the number of sensor nodes involved in long distance transmission. Clustering provides a significant improvement in prolonging network lifetime. However, it also introduces longer delay in data collection. Due to the design limitations, most nodes in a sensor network are only capable of handling a single connection at a time. To avoid data collision, cluster heads may employ TDMA and collect data from their cluster members in a one by one manner. Let T be the average transmission delay among nodes, a cluster head with n neighbors

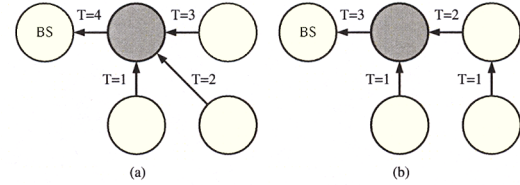


Fig. 1. (a) Data collection in 2-hop network and (b) data collection in multi-hop network. Cluster heads are represented by shaded circles while cluster members are represented by clear circles. BS represents the base station.

TABLE I

CLUSTER MEMBERS DEGREE DISTRIBUTION OF THE PROPOSED NETWORK STRUCTURE WITH NETWORK SIZE $N = 2^k$, WHERE $k = 1, 2, \dots$

Degree	1	2	\dots	$\log_2 N - 1$	$\log_2 N$
Frequency	$\frac{N}{2^1}$	$\frac{N}{2^2}$	\dots	$\frac{N}{2^{(\log_2 N - 1)}}$	$\frac{N}{2^{(\log_2 N)}}$

will take at least $n \times T$ to collect a complete set of data from its members (Fig. 1). Apart from requiring a longer delay in data collection, cluster members will need a larger buffer to handle the incoming data while waiting for the belonging cluster head to become available.

The bottleneck problem mentioned above can be alleviated by modifying the network structure. Assume that data aggregation is allowed and data in each sensor nodes are highly correlated, such that data fusion is applicable. By transforming the original network into a multi-hop network (Fig. 1), the time needed to collect a full set of data from the sensing field can be reduced by as much as 25%.

The aim of this paper is to investigate the characteristics of an efficient data collecting network structure, the algorithm for forming such a network structure and the improvement gained by this network structure. The rest of the paper is organized as follows. Section II will define the proposed network structure. Section III will explain the algorithm for forming such network. Simulations results and their analysis will be given in Sections IV and V, respectively.

II. NETWORK STRUCTURE

The proposed network structure is a tree structure, where the degree distribution of the cluster members follows an inverse exponential base-2 function, as shown in Table I. A node with degree k will connect with $k - 1$ nodes of degrees ranging from 1, 2, \dots up to $k - 1$ uniquely. At the same time, it will have one connection with a node of degree higher than itself.

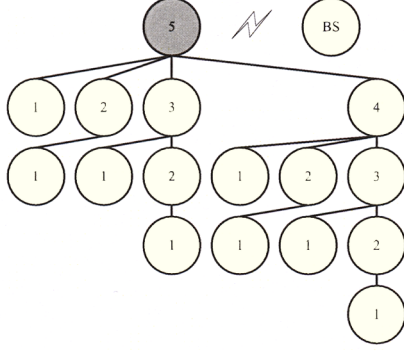


Fig. 2. Proposed network structure with network size $N = 16$. Cluster head is represented by shaded circle while cluster members are represented by white circles. BS represents the base station. Degree of each node is represented by the number inside the circle.

The cluster head will be considered as a special case. The cluster head is the one with the highest degree in the network. Instead of connecting with a node of higher degree, the cluster head will connect directly to the base station. An example of the proposed network with $N = 16$ is shown in Fig. 2.

The network structure is designed mainly for network with $N = 2^k$ nodes, where $k = 1, 2, \dots$. It will be shown in a later part that such a constraint can be relaxed by giving up some performance of the system. The time slot $t(N)$ required for the base station to collect data from the whole network of size N is given by:

$$t(N) = \lceil \log_2 N \rceil + 1, \quad N = 1, 2, \dots \quad (1)$$

III. NETWORK FORMATION ALGORITHM

A. Top-Down Approach

For networks with $N \leq 48$, the proposed network structure can be formed according to the following algorithm.

- 1) The algorithm starts with a fully connected network. For a network of $N = 2^k$ nodes, where $k = 1, 2, \dots$, each node will begin with degree equal to $N - 1$. The nodes will form the set $\hat{H}_{s=1}$. Set $b = \frac{N}{2}$.
- 2) Select b nodes from set \hat{H}_s such that the total path length within the b -subgraph is maximized. These b nodes will form set H_{s+1} . The rest of the nodes from \hat{H}_s will form set \hat{H}_{s+1} . The algorithm will then remove all connections among nodes within H_s . Set iterators $s \leftarrow s + 1$ and $b \leftarrow \frac{b}{2}$.
- 3) Repeat step 2) until $b < 1$. Set $r = 2$.
- 4) Nodes with degree $N - r$ form set L . Nodes with degree greater than $N - r$ form set U such that set L and set U are of the same size. Connections among nodes in the two sets are reduced until each node in set L is only connected to a single node in set U . The total path length of the outcome can be further optimized. Details of the optimization method is mentioned in the later part of this section. After reducing the number of connections, set $r \leftarrow r \times 2$.
- 5) Repeat step 4) until $r = N$.

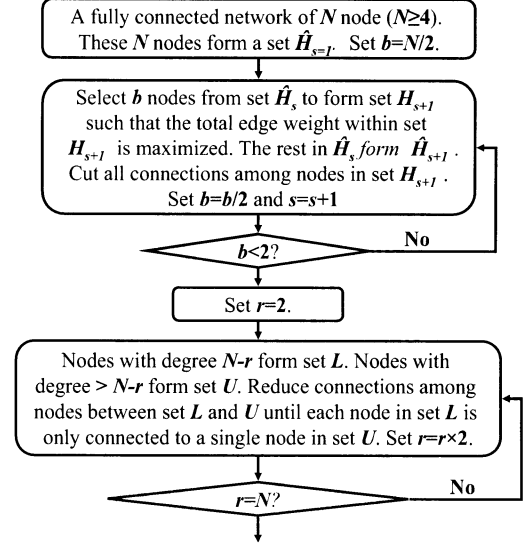


Fig. 3. Network formation of the proposed network structure using centralized approach ($N \geq 4$).

The network structures for $N = 2^0$ and $N = 2^1$ are trivial. The algorithm above is mainly for networks with $N = 2^k$ nodes, where $k = 2, 3, \dots$. A flow chart of the network formation algorithm is summarized in Fig. 3. The algorithm will produce 2 nodes with degree $\log_2 N$. Within these 2 nodes, the one with higher residual energy e_{res} will be selected as the cluster head and be connected directly to the base station. Therefore the cluster head will have a degree of $\log_2 N + 1$ which is the highest within the cluster. The above algorithm can be carried out by the base station which is a power unlimited device. Once the result is obtained, the base station will instruct each sensor node to form connections.

The procedure in step 2) is in fact the “heaviest k -subgraph” problem defined in [1]. For $N \leq 48$, the heaviest k -subgraph problem can be solved by mixed-integer linear programming. For networks with larger N , the mixed-integer linear programming becomes very time consuming and only heuristic methods such as Tabu search [2] can be used.

The nodes involved in the path length optimization in step 4) will form a distance matrix with each entry storing the distance between two nodes, with x-axis representing nodes from set L and y-axis representing nodes from set U . The two sets of nodes therefore form a bipartite graph and the path length optimization problem becomes a weighted matching problem. For $N \leq 48$, the total path length can be optimized by applying matching techniques such as Hungarian Method [3], [4] or Munkres’ Assignment Algorithm [5]. For $N > 48$, the techniques mentioned above become impractical due to their memory intensive nature. A sub-optimum method can be used instead to search for the minimum entry in the distance matrix and to record it in a connection matrix. Once recorded, all entries on the same row and column of the minimum entry in the distance matrix will be set to infinity. The method repeats itself until all entries in the distance matrix become

infinity. The sub-optimum method can provide decent results for large scale networks.

For networks with number of nodes other than $N = 2^k$, where $k = 1, 2, \dots$, dummy nodes are virtually added to expand the network such that the network can fulfill the network size requirement of the algorithm. These dummy nodes will have infinite separations with the real nodes and have infinite separations among themselves. The number of dummy nodes will always be smaller than $N/2$. Thus, at the end of the algorithm, these dummy nodes will all have degree of 1 which can be removed without partitioning the network. The time slots required for complete data collection will still be governed by equation (1).

B. Bottom-Up Approach

Due to its memory intensive nature, the top-down approach mentioned above generally works well for networks with relatively small N . For large networks such as $N \gg 48$, a bottom-up approach is more appropriate. The bottom-up approach may not deliver results as good as the top-down approach. However, with its flexible nature, the bottom-up approach is more scalable and can be implemented in both centralized or de-centralized manner. Specifically, the bottom-up approach in de-centralized manner can be described as follows.

- 1) Each node is first labeled with a unique identity and marked as level w . Variable w is a function of the number of nodes in a cluster. For a cluster of i nodes, w is equal to $\log_2 i$. Since nodes are disconnected initially, N nodes will form N single-node clusters. Each node will therefore be assigned as level 0. Within each cluster, one node will be elected as the sub-cluster head. We denote $SCH(w)$ as a sub-cluster head of level w cluster. Any $SCH(w)$ can only communicate with another $SCH(w)$. Since there is only 1 node in each cluster, all nodes begin as $SCH(w)$ with $w = 0$.
- 2) Each $SCH(w)$ broadcasts a "ranging packet" to its neighboring $SCH(w)$ s within r_{com} m. The packet contains the identity of the issuing $SCH(w)$. Upon receiving a ranging packet, a $SCH(w)$ calculates its distance to its neighbor who issues the packet.
- 3) Each $SCH(w)$ will then broadcast an invitation packet to its neighbors within r_{com} m. The invitation packet contains the level w of the issuing node, the identity of the issuing $SCH(w)$ and the identity of the nearest neighbor. If two parties are the nearest neighbor of each other, a connection will then be formed between these 2 parties. Once a connection is formed, the two parties form a composite cluster. One of the two involved $SCH(w)$ with a higher residual energy will become the sub-cluster head of the composite cluster and increment its level w by 1. It will handle all communications between the composite cluster and other parties. The involved party with lower residual energy will become silent. The sub-cluster head of the composite cluster will listen to the communication channel and reply any

lower level invitation with a rejecting packet. When a rejecting packet is received, the inviting party will include the identity of its next nearest neighbor into the invitation packet. The sub-cluster heads of higher level clusters remain listening to the channel. When no more invitation packet from lower level sub-cluster heads can be heard, sub-cluster heads of higher level clusters start communicating with their neighbors.

- 4) Repeat steps 2) and 3) until no more nodes can be connected.

When implementing in a decentralized control manner, the above algorithm may end up with multiple composite clusters when the number of nodes is not equal to 2^k , where $k = 1, 2, \dots$. SCHs of these composite clusters will communicate with the base station directly. In contrast, the above algorithm can be simulated at the base station as a centralized control algorithm. When the number of nodes is not equal to 2^k , where $k = 1, 2, \dots$, dummy nodes are added as in the top-down approach. These dummy nodes will have infinite separations with the real nodes and with themselves. The residual energy of a dummy node will be zero such that it will never be a SCH of a real node. Once the algorithm is completed, all dummy nodes will be removed.

IV. SIMULATIONS

In this section, the proposed tree based network structure will be compared with a single-hop (SH) network and a minimum spanning tree (MST) network. We will find the average data collecting time (DCT) and the average total path length (TPL) of each network structure. Simulations will be conducted in Matlab. Networks with nodes N varying from 4 to 64 will be distributed randomly on a sensing field of 50×50 m². For practical purposes, the residual energy of each sensor node will be a random function bounded by 0 J and 0.5 J. Results are taken from the average of 50 simulations.

In the SH network, minimizing DCT is impossible. Therefore, only TPL is minimized by selecting the node having the minimum separation with other nodes as the cluster head. In the MST network, the tree structure is formed by using Prim's algorithm. Since the TPL is already the shortest, optimization can only be done on the data collecting time. The DCT of the MST network is minimized by selecting the node with the minimum number of hops to other nodes as the cluster head. For better comparison, the bottom up approach is tuned to return with a single cluster. Communication radius r_{com} of the bottom up approach is therefore set to infinity.

In real scenarios, the cluster head is the only node that will communicate with the base station. The position of the cluster head may vary with different network structures. The location of the base station may introduce a biased situation to certain network formation algorithms. For a fair evaluation, the path length between the cluster head and the base station will not be included in the TPL. Hence, the time slot used for transmitting data from the cluster head to the base station will not be included in the DCT. Simulation results for networks with size varying from 4 to 64 are shown in Figs. 4 and 5.

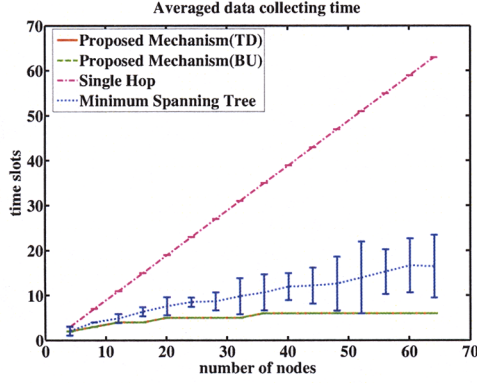


Fig. 4. Averaged data collecting time (DCT) of tree structures formed by the proposed algorithm using a top-down approach (TD), bottom-up approach (BU), single-hop (SH) and minimum spanning tree (MST).

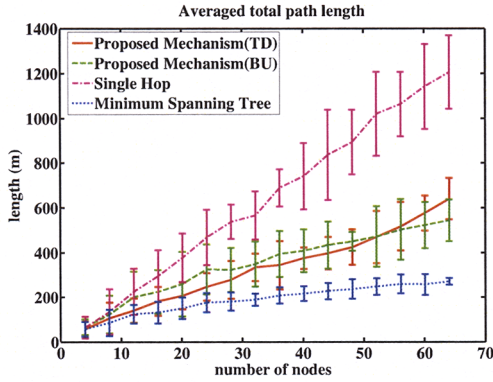


Fig. 5. Averaged total path length (TPL) of the tree structures formed by the proposed algorithm using a top-down approach (TD), bottom-up approach (BU), single-hop (SH) and minimum spanning tree (MST).

V. ANALYSIS

Simulation results show that the network formed by the proposed algorithm obtains the shortest DCT. With a single-layer structure, the SH network has expectedly the longest DCT. The MST network has a medium DCT. Furthermore, because of its minimum spanning nature, the MST network can always deliver the shortest TPL. When $N \leq 52$, the performance of the top-down approach in terms of TPL is a better than that of the bottom-up approach. This is because the optimization methods used in the top-down approach are more effective for small-scale networks. However, these optimization methods may easily be trapped in a local optimal point as the network size increases. Therefore, the top-down approach is recommended for small-scale networks, and the bottom-up approach for large-scale networks. The TPL for the SH network is the highest among the four structures.

As the network size becomes larger, it is obvious that the top-down approach of the proposed network formation algorithm and the SH network is not comparable to the other two algorithms in terms of DCT and TPL. Therefore, only the bottom-up approach of the proposed algorithm and MST are further investigated in networks with larger N .

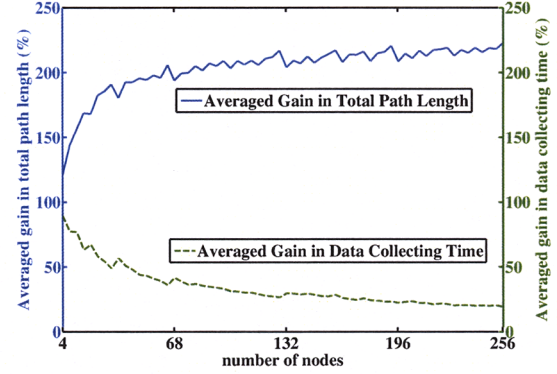


Fig. 6. Averaged gain in total path length (TPL) and data collecting time (DCT) of networks formed by the proposed bottom-up (TU) algorithm to that formed by MST

By dividing the results of the bottom-up approach with that of the MST, the averaged gain in DCT and the averaged gain in TPL of the proposed algorithm to the MST are expressed in percentage, as shown in Fig. 6. A low gain in data collecting time implies that the network formed by the proposed algorithm requires a shorter DCT while a high gain in TPL implies the network structure formed by the proposed algorithm has a longer TPL. It can be observed that the network formed by the bottom-up approach of the proposed algorithm can greatly shorten the DCT by about 4 times at the expense of doubling the TPL. The reduction in DCT becomes more significant as the number of nodes increases.

VI. CONCLUSIONS

In this paper, an efficient data collecting network structure and its formation algorithms are proposed. To cater for different applications, network formation can be implemented in either centralized or decentralized manner. Two network formation approaches are derived to provide optimized results for networks with different sizes. Comparing with the minimum spanning tree network and the single-hop network, our proposed network structure is shown to be the most efficient in terms of data collecting time. Although an increase in the total path length is observed, the gain in data collecting time outweighs the loss in total path length. Performance can be further optimized by applying different optimization methods in the network formation process.

ACKNOWLEDGMENT

This work is supported by HK PolyU under grant G-YF51.

REFERENCES

- [1] A. Billionnet, "Different formulations for solving the heaviest k-subgraph problem," *Information Systems and Operational Research*, vol. 43, no. 3, pp. 171-186, August, 2005.
- [2] E. M. Macambira, "An application of tabu search heuristic for the maximum edge-weighted subgraph problem," *Annals of Operations Research*, vol. 117, no. 1-4, pp. 175-190, November, 2002.
- [3] H.W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 52, no. 1, pp. 7-21, February, 2005.
- [4] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, New York: Dover Publications INC., 1998.
- [5] J. Munkres, "Algorithms for assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, March, 1957.