# Dynamic Route Sharing Protocol for Wireless Sensor Networks

Chih-Yung Chang, Sheng-Wen Chang, Ming-Hsien Li, Yu-Chieh Chen
Tamkang University
cychang@mail.tku.edu.tw, swchang@wireless.cs.tku.edu.tw, 695410505@s95.tku.edu.tw, ycchen@wireless.cs.tku.edu.tw

***Abstract*—A wireless sensor networks (WSN) comprises of a large number of sensor nodes and a few sink nodes. When multiple sink nodes are interested in collecting the readings of the same monitoring region, it's conducive to exploit the sharing route in order to save bandwidth and power consumption and prolong WSN's lifetime. This paper proposes a dynamic route sharing protocol (DRSP) which constructs sharing routes based upon different attributes (for example, frequency, packet length or delay time) of the commands requested from different sink nodes. The proposed DRSP dynamically adjusts data transmission route to achieve the goals of routes sharing and route length reduction. Simulation study shows that DRSP saves more energy and bandwidth consumptions than the existing work and thus prolongs the WSN's lifetime.***

## I. INTRODUCTION

Wireless sensor networks (WSNs) are potentially applied in many fields like military monitoring, location tracking, environmental sampling, health care monitoring, and so forth. A WSN is mainly composed of few sink nodes and a large number of sensor nodes. The sink nodes play the role of interface between users and WSNs, enabling users to send the data gathering commands to a region of sensor nodes in a wireless manner. Each sensor node has sensing, computation and wireless communication capabilities. Deployed in the monitoring region, the sensor nodes are responsible for collecting environmental information and reporting their readings according to the query commands sent by the sink nodes. Since sensor nodes are powered by batteries, energy conservation is of most important issue in developing routing protocol.

Braginsky and Estrin [1] pointed out that if there are only a few short messages sent between source and destination in WSN, it is not necessary to build a communication route for data exchange. On the contrary, when there is a need for long time data transmission between sink node and sensor node, building new route can be considered to reduce the overhead of data transmission. In [2], Kim et. cl. emphasized the importance of building a shared route for multiple sink nodes to retrieve the data of WSN and proposed a route sharing protocol to reduce the data traffics for transmitting the same data, saving the energy and bandwidth consumptions. However, the pre-constructed route cannot be adaptively changed according to the new request of sink nodes which build routes later.

Kim et al. [3] adopted a greedy algorithm to build a tree structure to construct a shared route for collecting the same data from a specific region to different sink nodes. Then the branch nodes in tree filter the data to each route branch according to the frequencies defined by the sink node.

In study [4], a network structure, which uses multicast tree to gather data from administrator within the same area, has been proposed. When a sink node intends to join the multicast tree, the sink will send the joining message to the root of the multicast, and the root of the multicast tree subsequently relays the joining message to the sensor node on the tree nearest to the sink node. After that, the sensor node will calculate the gravity point to build a sharing route for the new sink node. However, the location of the gravity point might have none of sensor nodes, and hence the gravity point might not be the optimal branch point. Moreover, how to build multicast tree after finding gravity point is not explicitly stated in the paper. Usually, building such a tree requires flooding operations to attain the goal. A large amount of control packets are thereby produced. On the branch point of each multicast tree, the same data must be periodically updated to support the request of each sink node. Thus, the branch point will meet the same power-consumption problem which exists in research [3], and pre-built multicast tree will be destroyed.

This paper proposes a dynamic route sharing protocol which constructs routes based upon data requests from multiple sink nodes. The established sharing routes can be dynamically modified according to newly–built route by subsequent sink node under cost-efficient condition, making data collection method is equipped with the property of dynamic immediacy. Thus, the data collections from the coordinator to multiple sink nodes can be proceeded in a way of route sharing which saves more energy and bandwidth consumptions than existing work, prolonging the network lifetime.

The remaining sections in the paper are organized as follows: Section 2 introduces the network model. Section 3 details the proposed dynamic route sharing protocol (DRSP). Section 4 discusses experiments on DRSP and their results, and finally, a brief conclusion is presented in Section 5.

## II. NETWORK ENVIRONMENT AND PROBLEM STATEMENT

This section presents the network model and assumptions. Some notations which will be used in the proposed DRSP are given.

### *2.1 Environment and Problem Statement*

The WSN we discuss is composed of few sink nodes, a large number of sensor nodes and a few coordinators. All coordinators and sensor nodes are aware of their own location information. The sensor nodes are responsible for performing monitoring tasks including sensing the environmental information and periodically transmitting their readings to the coordinators according to the sinks' requests. Based upon user's request, sink node will send command to all the sensor nodes within a specific region so as to meet the requirement of gathering data periodically. The coordinator takes the responsible of data collection from sensor nodes within the specific region and then proceeds with the data calculation and reports results to the sink nodes. It is permitted that the times of sending requests by different sink nodes can be different. The content of request sent by sink node includes the returning frequency that data collection requires, the attributes of the content of the collected data, the expected share degree of data collecting route, the permitted delay time

**978 -1-4244-1866-4/08/$25.00 © 2008IEEE**

of returning data and the time interval for data collection. Due to the need of returning data periodically, when a sink node sends a request to a coordinator, the coordinator will transmit collected data through the constructed route from itself to the sink node in a multi-hop manner. In literature, many papers have focused their attention on how coordinator gathers data from sensor nodes within a specific regions in an energy saving and efficient way. The paper won't discuss the issue of gathering data from sensor nodes to the coordinator. Alternatively, it will be discussed how to build efficient routes with high sharing degree from the coordinator to the multiple sink nodes according to the requests issued by multiple sink nodes at different times.

### *2.2 Notations*

To clearly present the details of the proposed protocol, the following notations are defined. Assume that there are $k$ sink nodes $K_i$ in a WSN, where $1 \le i \le k$, and there are $m$ sensor nodes $s_i$, where $1 \le i \le m$. Notation $R$ denotes coordinator. The command requested from each sink node $K_i$ contains the following possible attributes. Notation $f_i$ denotes the data returning frequency required in collecting data, notation $a_{ij}$ denotes the attributes of the content of collected data. The expected share degree of data collecting route is denoted by $l_i$. The permitted delay time in data returning is denoted by $d_i$. Notation $TI_i$ represents the time interval for data collection. When the sink node $K_i$ sends RREQ packets to the coordinator $R$ using directional flooding approach, if node $s_x$ sends RREQ packet to node $s_y$, we refer node $s_x$ as the upstream candidate of node $s_y$. Suppose a route can be thus decided, the route constructed by the forwarding nodes is denoted as $Route(K, R) = \{K=s_0, s_1, \ldots, s_n=R\}$. We further assume that node $s_i$ is the upstream node of node $s_{i+1}$, and node $s_{i+1}$ is the downstream node of node $s_i$.

The so-called existing route denotes a built route from some sink nodes to the coordinator, and current route denotes as under-constructed route from the new sink node to the coordinator. The current node denotes the sensor node that is on the current route and executes the route construction task. Moreover, to construct a low-cost sharing route, $C_{\overline{xy}}$ is defined as the cost of $link_{\overline{xy}}$, which is measured by the cost of power consumption and time delay. Let $C_{i\to j}$ denote the cost of a route from sensor node $s_i$ to node $s_j$, we have

$$C_{i\to j} = \sum_{k=i}^{j-1} C_{\overline{S_k S_{k+1}}}$$

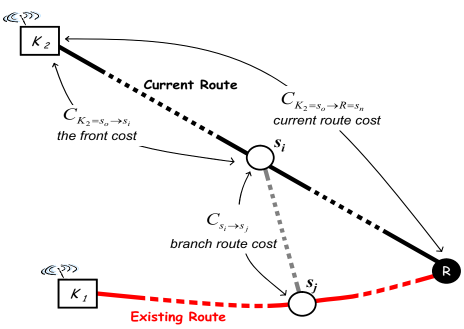


Figure 1: The illustration of the cost of different routes.

As shown in Fig. 1, we define the terms of the cost for each route. The *current route cost* $C_{K_2=s_0\to R=s_n}$ denotes the cost for building route between sink node $K_2$ and coordinator $R$. On the current route, the cost of the route from node $s_i$ to sink node $K_2$ is referred as *the front cost of current route* or *the front cost* in short, and denoted by the symbol $C_{K_2=s_o\to s_i}$. Furthermore, the cost of route between node $s_i$ and any node $s_j$ on existing route is referred as *branch route cost*, and is denoted by $C_{s_i\to s_j}$.

## III. Dynamic Route Sharing Protocol

This paper focuses on the route construction between multiple sinks and one coordinator and proposes a Dynamic Route Sharing Protocol (DRSP). The proposed DRSP protocol is composed of three major Phases: Route Request, Route Reply, and Route Sharing Phases. In the Route Request Phase, a sink node sends user's commands to coordinator via applying directional flooding operations in multi-hop manner. Then, an initial route will be built between the sink node and the coordinator in Route Reply Phase. After that, the Route Sharing Phase will dynamically adjust the constructed route to establish an optimal sharing route according to several key information including the attached node and branch node it found, the requested frequency of each sink node, and the cost calculation. The following describes the details of DRSP.

### *3.1 Route Request Phase*

When user issues commands to a certain sink node $K_i$, the sink node $K_i$ will initiate the Route Request Phase. In this phase, the sink node $K_i$ sends route request packet (*RREQ*) to the coordinator $R$ using directional flooding approach in a multi-hop manner. On receiving the *RREQ* packet, sensor nodes should also stay in the *Route Request Phase*. To relay the route request packets to the coordinator $R$ as soon as possible but avoid the packet flooding over the entire sensor network, as shown in Fig. 2, DRSP will confine packet flooding within Request Zone, which is defined according to the locations of sink node $K_1$ and the coordinator.

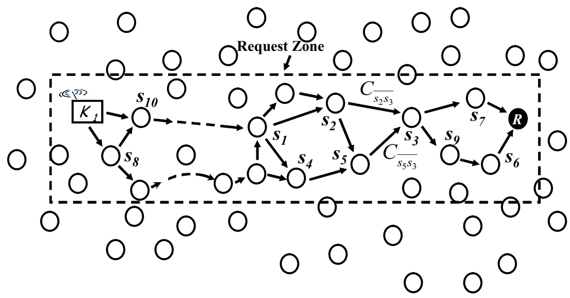


Figure 2: In Request Zone, sink node $K_1$ uses directional flooding approach to send *RREQ* packet to the coordinator $R$.

Figure 2 is taken as an example to illustrate the operation of Route Request Phase. The sink node $K_1$ appoints a certain Request Zone to forward *RREQ* packets to coordinator $R$. In the process of transmitting *RREQ* packet, we consider the situation of transmission of sensor nodes which is in the vicinity of sensor node $s_3$. Assume that node $s_3$ first receives the RREQ packet from its neighbor node $s_2$. In the meantime sensor node $s_3$ will treat node $s_2$ as an upstream candidate, evaluate and record both $C_{K_1\to s_2}$ and $C_{\overline{s_2 s_3}}$ in its own *cost table*. Based upon the content of RREQ, a new value of *the front cost* $C_{K_1\to s_3} = C_{K_1\to s_2} + C_{\overline{s_2 s_3}}$ is calculated and placed in the fields of *RREQ* packet to send. Whenever sensor node $s_3$

receives another *RREQ* packet from node $s_5$, similarly, sensor node $s_3$ will treat node $s_5$ as its upstream candidate, and calculate the value of link cost $C_{\overline{s_5 s_3}}$ and *the front cost* between itself and node $s_5$. If node $s_3$ finds that the cost value $C_{K_1 \to s_3} = C_{K_1 \to s_5} + C_{\overline{s_5 s_3}}$ of route passing through node $s_5$ is better than the cost $C_{K_1 \to s_3}$ of route passing through only node $s_2$, node $s_3$ will put the value $C_{K_1 \to s_3}$ into *RREQ* packet and sends the RREQ packet again. If it is not this case, the information $C_{K_1 \to s_5}$ and $C_{\overline{s_5 s_3}}$ will be saved in cost table for the use of later route breakage.

### *3.2 Route Reply Phase*

In this Phase, coordinator *R* will determine a route between sink node $K_i$ and coordinator *R*, and send back a Route Reply packet (*RREP*) to sink node $K_i$ in a multi-hop manner. Moreover, the coordinator will notify sink node $K_i$ whether or not it is the first sink node to retrieve the data from the coordinator by using the *RREP* packet. In the meanwhile, the neighboring nodes in the vicinity of current route will treat the *RREP* packet as *S-RREQ* packet which is going to find outward sharing route.

Figure 3 is taken as an example to illustrate the detailed operations of Route Reply Phase. Herein, we assume that sink node $K_1$ is the first sink node that issues command to the coordinator *R*. The coordinator *R* has two neighbors of upstream sensor nodes $s_6$ and $s_7$ at the moment. We further assume that coordinator *R* can derive the minimal *current route cost* $C_{K_1 \to R}$ by constructing a route passing through node $s_7$. Thus, coordinator R will fill the values of $C_{K_1 \to R}$ in the two fields of *RREP* packet: *current route cost* and *the front cost*, and then send *RREP* packet to its chosen upstream node $s_7$. By the same reason, sensor node $s_7$, which is appointed to forward *RREP* packet, will calculate new the front cost $C_{K1 \to s_7} = C_{K_1 \to R} - C_{\overline{s_7 R}}$, record the information into cost table, and choose the upstream node $s_3$ ,that fits the cost, as the most suitable neighboring nodes to forward *RREP* packet. When node $s_7$ transmits *RREP* packet to node $s_3$, the front cost field in the *RREP* packet will be updated to be $C_{K_1 \to R} - C_{\overline{s_7 R}}$. Then, operation is done by the same way as aforementioned until *RREP* packet is reached to sink node $K_1$.

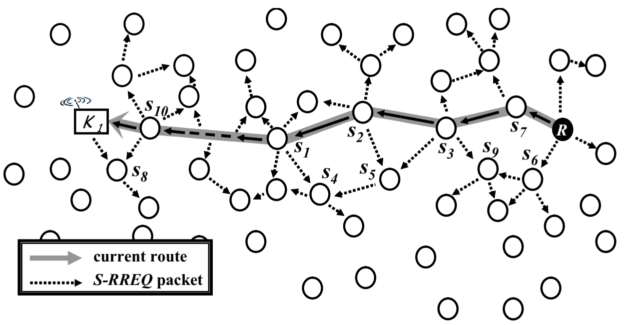


Figure 3: Coordinator *R* responds *RREP* packet to the sink node $K_1$ according to the cost table. Then optimistic node sends *S-RREQ* packet to construct the sharing route.

Figure 3 illustrates the operations of the other situation that sink node $K_1$ is not the first sink node to issue the command to coordinator. Because that there are existing routes to coordinator *R*, the problems of sharing routes to coordinator exists between sink nodes $K_i$ and coordinator *R*. The first route field in the *RREP* packet is filled in 0 by coordinator *R*, which informs sink node $K_i$ to exploit the opportunity of building sharing routes and notifies the current node, which is responsible for forwarding *RREP* packet, to construct sharing routes in its surroundings. Here, we define the $l_i$-hop neighbors in the vicinity of all the current nodes on current route as the *optimistic node*, where the value $l_i$ denotes the sharing degrees decided by the sink node. The 1-hop neighbors of coordinator *R* and current nodes also receive *RREP* packet. However, different from the mechanism of current nodes on current route, the optimistic nodes will treats *RREP* packet as Shared Route Search Request Packet(*S-RREQ packet*) since the first route field in *RREP* packet is set to 0. After entering the Route Sharing Phase, the optimistic nodes will send *S-RREQ* packet to find outward sharing routes, and the hop distance of sending *S-RREQ* packet outward is followed the value of the share degree $l_i$ field in *RREP* packet. Therefore, all the sensor nodes within the requested hop count of *S-RREQ* packet will play the role of optimistic node.

### *3.3 Route Sharing Phase*

In Route Sharing Phase, optimistic node will try to find an adequate couple of attached node and branch node to change the route for achieving the goal of route sharing. Because the requested returning frequency of sink node is not the same as that of coordinator, the following route-changing strategy is given as a foundation of building sharing route.

***Criteria 1***：When building sharing route, the route with lower requested returning frequency must attached to those routes with higher requested returning frequencies.

When the two routes with different returning frequencies intend to retrieve data from the same coordinator at the same time, the route with higher returning frequency can share similar data with the route with lower returning frequency. Therefore, *criteria 1* help to reduce the cost of data transmission.

***Criteria 2***：If optimistic node $s_j$ satisfies the following Expression (1), it can continue to send *S-RREQ* packet. Sensor node $s_j$ on existing route whose cost calculation meets Expression (1) returns the request of changing route from the current nodes.

$$C_{K_2 = s_o \to s_i} + C_{s_i \to s_j} \le C_{K_2 = s_o \to R = s_n} \quad (1)$$

In Figure 4, it can be found that after node $s_{13}$ delivers the *RREP* packet to its upstream node $s_{19}$, optimistic node $s_{14}$ will treat the *RREP* packet as *S-RREQ* packet, and check if the cost of the route satisfies the requirements as stated in *Criteria 2*. If it is not the case or $C_{K_2 \to s_{13}} + C_{s_{13} \to s_{14}} \ge C_{K_2 \to R}$, $s_{14}$ will keep some information its cost table for the use of later route breakage. These information include *link cost*, *current route cost*, *the front cost* and *branch route cost*. On the contrary, if *Criteria 2* is satisfied, node $s_{14}$ will update the value of *branch route cost* in *S-RREQ*, and continue to send *S-RREQ* packet to its neighboring nodes $s_7$ and $s_{15}$, aiming to construct the share route. Similarly, nodes $s_7$ and $s_{15}$ will decide whether or not to continue to send *S-RREQ* packet. As shown in Fig. 4, the *S-RREQ* packet is sent to two nodes $s_2$ and $s_3$ on existing route. Then nodes $s_2$ and $s_3$ evaluate whether or not the cost of route meet the requirement of route adjustment based on *Criteria 2*, play the role of attached node according to *Criteria 1*, and then reply to the current node $s_{13}$ with an *S-ACK* message. At this moment, the current node $s_{13}$ is referred as branch node, and attached nodes $s_2$ and $s_3$ will add cost information in communication into the fields of *S-ACK* as a reference for optimistic nodes and branch nodes to further construct the sharing routes.

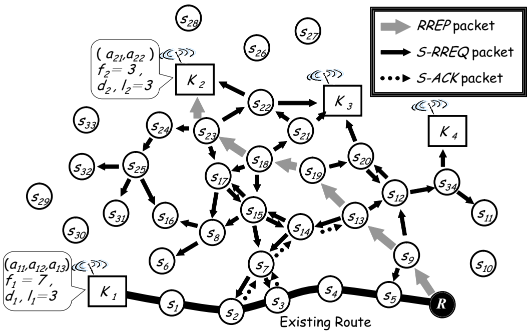


Figure 4: Sink node $K_2$ uses *RREP* and *S-RREQ* packet to create possible sharing routes.

Each current node sends *S-RREQ* packet independently to find possible attached nodes. For an optimistic node, it might receive different *S-RREQ* packets from different current nodes. As shown in Fig. 4, current node $s_{18}$ sends *S-RREQ* packet to the optimistic node $s_7$ via optimistic node $s_{15}$. Before deciding to send *S-RREQ* packet, node $s_7$ will check the *S-RREQ* packet and find that the remaining hop is 1. Then node $s_7$ checks the value $C_{K_2 \to s_{13}} + C_{s_{13} \to s_7}$ in its cost table and evaluates that the cost of route passing through the optimistic node $s_{14}$ is 22.2 whereas the cost of route passing through the optimistic node $s_{15}$ is 17.6. Thus, the requirement of *Criteria 2* is satisfied. Therefore, the optimistic node $s_7$ will further continue to forward the *S-RREQ* packet. When current node $s_{23}$ sends *S-RREQ* packet to the optimistic node $s_7$ via optimistic node $s_{15}$, it will find that the remaining hop count of *S-RREQ* packet is 0. Thus, optimistic node $s_7$ will only keep the information in the cost table instead of continuing forwarding *S-RREQ* packet.

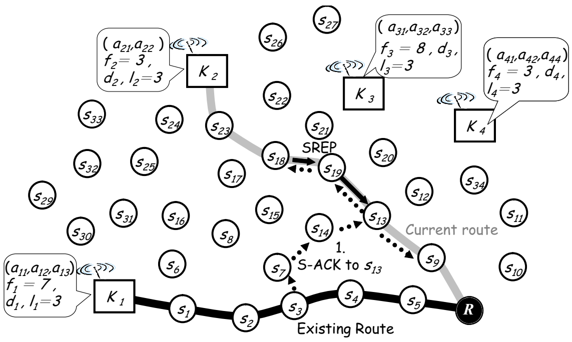


Figure 5: An adequate branch node chosen among current nodes using *RSP* and *SREP* packets.

The behavior of sending *S-RREQ* packet and *S-ACK* by the current node and optimistic node might identify many attached nodes and branch nodes on the existing route. Thus, in order to decide an optimal couple of attached node and branch node and construct a low-cost sharing route, the following mechanism is stipulated. As shown in Fig. 5, upon receiving *S-ACK*, which is sent from attached node $s_3$, current node $s_{13}$ will continue to forward *S-ACK* to current nodes $s_9$ and $s_{19}$. When node $s_{19}$ receives *S-ACK*, it will continue to forward *S-ACK* to the current node $s_{18}$ after calculating of the route cost and determining to change the route to a better route passing through node $s_{13}$. In the meantime, $s_{18}$ determines that the sharing function can be better than $s_{18}$ by changing route itself. Then node $s_{18}$ will return *Share Route Existed Packet* (*SREP*) to $s_{13}$ in order to inform $s_{13}$ of abandoning playing the role of branch node. Otherwise, if node $s_{13}$ only receive *Route Success Packet* (*RSP*) from the sink node $K_2$, it changes the current route to connect to the attached node $s_3$.

### *3.4 Adaptive Route Sharing*

Previously, we have mentioned the strategy of building sharing route when current route has sensor nodes with low returning frequency. In what follows, we will state the case when the returning frequency of current route is higher than that of the existing route.

Different from the aforementioned strategy, the member nodes on existing route which receives *S-RREQ* packet will evaluate the necessity of building sharing route between current route and existing route. If it is the case, the member node on existing route will fill the join field in *S-ACK* with 1, representing that the existing route will join current route for the sake of forming sharing route. As depict in Fig. 6, suppose node $s_7$ receives a *S-RREQ* packet from the current node $s_{12}$, it's not better to change route according to evaluation. Thus, node $s_7$ will only inform attached node $s_2$ about the existence of current route. Upon receiving the *S-RREQ* packet sent from current node $s_{20}$, node $s_{23}$ decides to build a sharing route with node $s_{20}$, and then play the role of branch node as step 1 depicts. For the sake of informing the attached node $s_2$ of the request of changing route to the current route, node $s_{23}$ will continue to send *S-RREQ* packet to the attached node $s_2$. In the meantime, if node $s_{23}$ receives *SREP* packet from other member nodes on existing route, it will abandon playing the role of branch node. Otherwise, as step 2 depicts, node $s_{23}$ received *RSP* packet sent from the attached node $s_2$, it continues to play the role of branch node, and return an *S-ACK* message to node $s_{20}$ to join current route in step 3. Upon receiving the *S-RREQ* packet sent from node $s_7$, node $s_2$ helps combine original existing route and current route as a new sharing route and the original existing route can reduce the cost by changing to a new route via node $s_7$. As step 4 depicts, the attached node $s_2$ returns *S-ACK* to node $s_{12}$, and perform the route changing scheme to achieve the goal of route sharing.

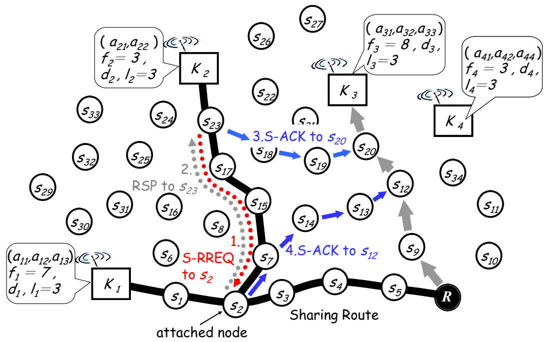


Figure 6: The existing route formed by sink node $K_1$ and$K_2$, and the illustration of steps in building the sharing route among sink node $K_3$.

The proposed protocol tries to build a sharing route for those sink nodes that request to collect the data from the same coordinator with different requested returning frequency. For the data with different attributes, another route to coordinator

$R$ will be built by sink node itself. As shown in Fig. 7, assume that the coordinator $R$ choose nodes $s_{10}$, $s_{11}$, and $s_{34}$ to forward data from the coordinator $R$ to the sink node $K_4$. However, the same part in the attributes of the requested data by $K_4$, $K_1$, $K_2$, $K_3$, and other sink nodes includes only the contents such as ($a_{41}$, $a_{42}$), but different part in data attribute includes ($a_{44}$). At the moment, after receiving responded *S-ACK* from $s_{12}$, node $s_{34}$ can know the attribute of shared data which is offered by node $s_{12}$ is ($a_{41}$, $a_{42}$). Therefore, node $s_{34}$ will continue to transmit *S-ACK* to current node, and collect the data with different frequency on the 2 routes after determining to play the role of attached node. Finally, the data will be integrated and sent back to sink node $K_4$。

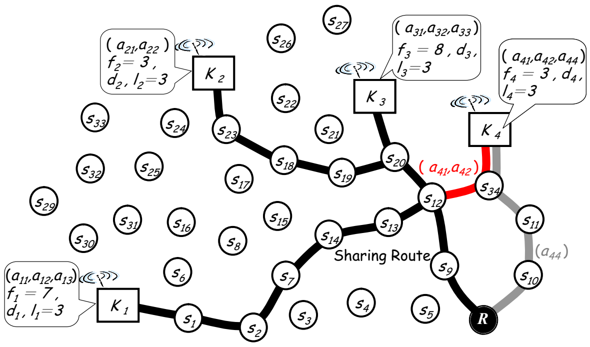


Figure 7: The sharing route that sink nodes $K_1$, $K_2$, $K_3$ and $K_4$ build to connect coordinator $R$.

## IV. Simulation

This section evaluates the efficiency of proposed Dynamic Route Sharing Protocol. The main goal of the proposed Dynamic Route Sharing Protocol is construct a shared route for multiple sink nodes.

To simulate the environment of WSN that satisfies the characteristic of hardware, the simulation result will be produced according to MICA2 which is used in [5]. In the simulation, the sink nodes and sensor nodes are randomly deployed in the monitoring region with size 2000m × 2000m. The number of sensor nodes is varied ranging from 300 to 500 and the number of sink nodes varies ranging from 1 to 10. A sensor node is chosen to play the role of coordinator which is responsible to collect and transmit data to multiple sink nodes according to each sink node's query. Each sensor node and sink node knows its own location, and all sink nodes are also aware of the location of the coordinator. The common communication range of sink node and sensor node is 50m. The dissipated energy of sensor node in the data transmission and data reception is 0.080W and 0.025W, respectively. Three mechanisms, the proposed DRSP, previous work SAFE [2], and flooding are compared.

Figure 8 discusses the relationship between number of sink nodes and the average number of sharing route. The number of deployed sensors is set at 500. In general, the average number of sharing routes constructed by DRSP and SAFE increases with the number of sink nodes. The DRSP and SAFE have similar performance and significantly outperform the flooding mechanism in terms of the average number of shared routes. For instance, when the number of sink node in the network is 5 and 10, DRSP constructs 4.21 and 8.678 sharing routes, respectively, while SAFE constructs 4.35 and 8.783 sharing routes, respectively. The main reason that SAFE constructs more shared routes than DRSP is that SAFE uses flooding approach to build sharing route. Thus, SAFE exploits more opportunities for finding the sharing route than DRSP. The Flooding scheme builds route from each sink node to the coordinator by applying flooding operations, but the construction of sharing route is never considered. Therefore, the possibility of generating sharing route increases with the number of sink nodes.

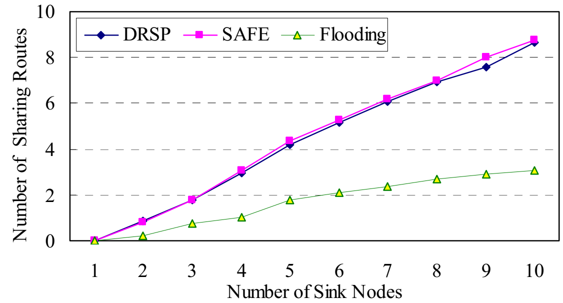


Figure 8: The relationship between the number of sink nodes and the average number of sharing routes.

Figure 9 shows the overhead of building route between sink node and coordinator. It can be easily found that DRSP use local Flooding approach to find routes and its control overhead is roughly 989 when there are 10 sink nodes in the network. But the control overhead of SAFE is 11257, Flooding is 31762. This is because the more control packets are sent by SAFE and Flooding in building routes compared with DRSP.

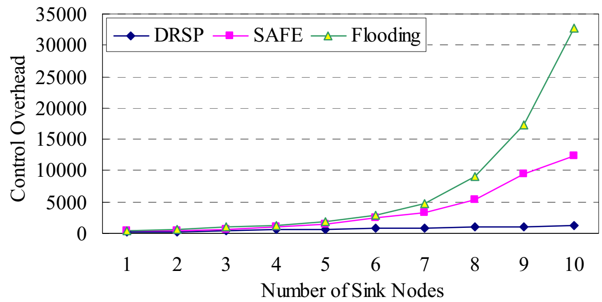


Figure 9: The relationship between the number of sink nodes and control overhead.

In Figure 10 depicts, DRSP consumes less power than SAFE and Flooding. The main reason is that when DRSP builds sharing route, it will adjust sharing route to the optimal sharing route depend on the requested data attribute and returning frequency of sink node. Therefore, DRSP consumes less power than SAFE which applies greedy algorithm to build route in data transmission.

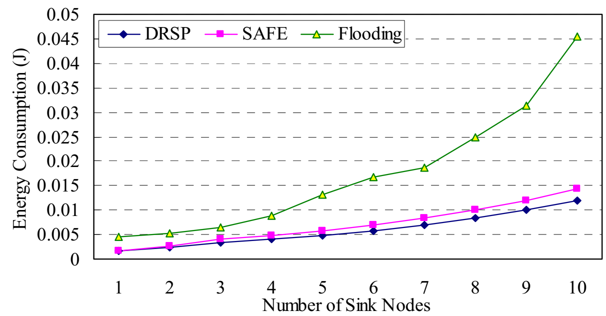


Figure 10: The power dissipation in data dissemination.

## V. Conclusion

In the paper, a protocol, that supports building sharing route among multiple sink nodes and coordinator in the same area, is proposed. Through the concept of sharing route, a data collection protocol with multiple initial nodes. The number of forwarding nodes, the bandwidth waste, and transmission of replicate data are thus reduced, reaching the goal of power saving. For sink node which requests for dissimilar data, we try to make its route a sharing route, and dynamically adjust the route according to requested attribute and returning frequency which is sent from each sink node at different time interval, making the routes to be sharing ones. Finally, the simulation shows that DRSP did have good Performance.

## REFERENCES


[1] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," in *Proceedings Workshop on Wireless Sensor Networks and Applications (WSNA02)*, 2002.

[2] Sooyeon Kim, Sang H. Son, John A. Stankovic, Shuoqi Li, and Yanghee Choi, "SAFE: A Data Dissemination Protocol for Periodic Updates in Sensor Networks," in *Workshop on Data Distribution for Real-Time Systems (DDRTS)*, May 2003.

[3] Hyung Seok Kim, Tarek Abdelzaher, Wook Hyun Kwon, "Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks," *ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, USA, Nov 2003.

[4] S. Bhattacharya, H. Kim, S. Prabh, and T. Abdelzaher. Energy-conserving data placement and asynchronous multicast in wireless sensor networks. in *Proceedings on Mobile Systems, Applications, and Services (MobiSys)*, pp. 173–186, M03.ay 2003

[5] I. Crossbow Technology. MICA2 Wireless Measurement System Datasheet, URL *http://www.Crossbow.com/Products/Wireless_Sensor_Networks.htm*. Crossbow Technology, inc. 2003.