Directional Controlled Fusion in Wireless Sensor Networks (Invited Paper)

Min Chen
Dept. of Elect & Comp Eng
Univ. of British Columbia
Canada V6T 1Z4
minchen@ece.ubc.ca

Victor Leung
Dept. of Elect & Comp Eng
Univ. of British Columbia
Canada V6T 1Z4
vleung@ece.ubc.ca

Shiwen Mao
Dept. of Elect & Comp Eng
Auburn Univ.
Auburn, AL 36849, USA
smao@ieee.org

ABSTRACT

Though data redundancy can be eliminated at aggregation point to reduce the amount of sensory data transmissions, it introduces new challenges due to multiple flows competing for the limited bandwidth in the vicinity of the aggregation point. On the other hand, waiting for multiple flows to arrive at a centralized node for aggregation not only uses precious memory to store these flows but also increases the delays of sensory data delivery. While traditional aggregation schemes can be characterized as "multipath converging", this paper proposes the use of "multipath expanding" to solve the above problems by exploiting both data fusion and load balancing. We propose a novel directional-controlled fusion (DCF) scheme, which includes two key algorithms, i.e., directional control and multipath fusion. By adjusting a key parameter named multipath fusion factor in DCF, the trade-offs between multipath-converging and multipath-expanding can be easily achieved, in order to satisfy specific QoS requirements from various applications. We present extensive simulations that verify the effectiveness of the proposed scheme.

Keywords

wireless sensor networks, data aggregation, multipath routing, load balancing

1. INTRODUCTION

Wireless sensor networks (WSNs) [1, 2] have attracted remarkable attention in the research community recently, driven by a wealth of theoretical and practical challenges and increasing number of practical civilian applications. In environments where the source nodes are close to each others and generate a lot of sensory data traffic with redundancy, forwarding all sensory data to the sink node not only wastes the scarce wireless bandwidth, but also consumes a lot of battery energy. Data aggregation is among the mechanisms to eliminate the data redundancy in order to save energy. The conventional network structure for facilitating aggregation is a tree rooted at the sink with the source nodes as leaves. By exploiting such an aggregation tree, data fusing is performed where branches merge to decrease the number of transmissions in the network, thus reducing the bandwidth usage and energy consumption caused by relaying of sensory data.

Finding the optimal aggregation points in the network is known to be NP-Complete and still an open area of research. Building an optimal aggregation trees (e.g. Weighted Steiner Tree [3, 4], or schemes approximating Steiner Tree [5]) need centralized control; i.e., before the algorithm starts, complete knowledge of all the data flows need to be collected in advance, which may cause a lot of control overhead in a large WSN.

Several heuristic schemes have been proposed, which depend on specific assumptions on the aggregation model. One of such assumptions is that the aggregation points need to wait for upstream (i.e., the flows starting from source nodes) nodes to send their data [6]. Due to the increased delay, such an assumption may be infeasible for delay-constraint applications, especially for real-time video/image transmission over WSNs, since the associated latency may cause the video/image packets to exceed their decoding deadlines.

Though recent work [5, 7] has considered both aggregation efficiency and aggregation cost in building efficient aggregation trees, contention of multiple flows at aggregation node has been largely ignored. Such contention increases control overhead, energy consumption and access delay, thus degrading the benefits from data aggregation and limiting the application scope of these previously proposed aggregation tree algorithms. This contention problem may be prevented if the data from multiple flows arrive at the aggregate point asynchronously. However, this presents a new problem of buffering delays at the aggregation nodes, which increase with the number of flows being aggregated. Such an approach also requires additional storage to temporarily store data that arrives earlier, and may not be affordable for memory-constrained sensor nodes.

In this work, we focus on data redundancy eliminating at the fusion points of multiple flows. We consider an image sensor network where static sensors are uniformly distributed over a large area. Each sensor knows its physical location, e.g., using the Global Positioning System (GPS), and geographical routing can be exploited [8–10]. In our system, several sources nodes equipped with cameras take images of a certain target object and transmit the images to the sink node. Since these cameras are located in the same target region, the captured images may be highly correlated, e.g., they may have a similar background. An example is shown in Fig. 1, in which three source nodes capture and transmit images via different disjointed paths at first. Since the path between camera-2 and the sink is the shortest one, the whole image captured by camera-2 arrives at nodes A and B earlier than those captured by camera-1 and camera-3.

QShine 2008, July 28-31, 2008, Hong Kong, Hong Kong.
 ISBN 978-963-9799-26-4
DOI 10.4108/ICST.QSHINE2008.3844

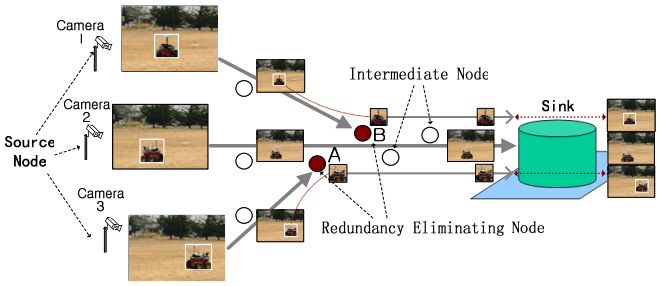


Figure 1: Example of Data Redundancy Eliminating in Directional-Controlled Fusion.

Assume that the image flow of camera-3 converges with that of camera-2 at node A. At the view point of node A, transmitting the whole picture taken by camera-3 to the sink node may be unnecessary in the case that the image captured by camera-2, which has common overlapped background, has already been transmitted to the sink. Thus, instead of transmitting the whole picture taken by camera-3, node A extracts the region-of-interest from the whole picture using an image-segmentation algorithm. Currently, the iMote2 sensor node is capable of performing in-node image-processing. When the image taken by camera-1 arrives at node B, the same operation is performed so that the large volume of imagery data is reduced into a smaller one. The segmented images are restored at the sink node, as shown in Fig. 1.

As opposed to the aforementioned properties existing in previous data fusion solutions, the directional-controlled fusion (DCF) scheme proposed in this paper has the following features:

- DCF builds the multipath fusion tree in a distributed fashion. Nodes can perform path repair and fusion without any feedback between source-sink pairs or any knowledge about the global topology. This feature makes DCF scalable for large WSNs.
- DCF explicitly considers load balancing and network lifetime while addressing the energy conservation issue.
- DCF can achieve different fusion patterns, which can be divided into two categories, i.e., multipath-converging and multipath-expanding. While the previous work can be categorized as multipath-converging, DCF first achieves multipath-expanding that exploits the efficacy of both data fusion and load balancing.

DCF is a combination of two proposed algorithms: (1) directional control algorithm for controlling the paths directions; (2) path fusion algorithm for building the multipath fusion tree with the hierarchical structure. Our main contributions are two folds: (1) introduce a new concept of "multipath-expanding" which increases the multipath aggregate bandwidth and efficiency by taking advantage of both load balancing and data redundancy eliminating; (2) achieve flexible trade-offs between multipath-converging and multipath-expanding.

The rest of the paper is organized as follows. We describe the DCF algorithm in Section 2. Simulation model and experiment results are presented in Section 3. Section 4 concludes the paper.

2. DIRECTIONAL CONTROLLED FUSION

In this section, we present the architecture and design issues of the directional-controlled fusion (DCF) scheme. We first give an overview of the network organization and our proposed redundancy eliminating model. We then describe the DCF components in detail. DCF consists of two key components, directional control and multipath fusion. We present the directional control mechanism in Section 2.2 and then the multipath fusion scheme in Section 2.3. For clarity, we also classify all possible multipath fusion patterns in Section 2.4.

Table 1: Notation

Symbol	Definition
I_i	image captured by camera i.
$\|I_i\|$	size of image I_i.
B	the common background image.
V	the set of selected source nodes.
R_i	the segmented image after removing background.
ρ	the ratio of segmented image size to the original image size.

2.1 Architecture Overview

Given Fig. 1 as an example, this paper considers a network where a number of camera sensor nodes (i.e., source nodes) are sparsely deployed among a much larger number of densely deployed low-power sensor nodes. The set of source nodes cover the target region remotely monitored by the sink. The captured images by the source nodes will be delivered to the sink node by large number of low-power sensor nodes.

In order to reduce the redundancy among the images and conserve energy, data fusion will be performed along the way when the images are forwarded to the sink. For this purpose, one source node will be selected as the *reference source* at a time based on some criteria (e.g., maximum remaining energy, closeness to the center of the center of the target region, or closeness to the sink, etc.) using the following mechanism. Initially, all of the source nodes start a so-called Reference-Source-Selection-Timer (RSS-Timer). A random value is set to each timer based on specific criterion. A smaller value indicates a source has higher eligibility as the reference source. Then, the one whose RSS-Timer expires first will be selected as the reference source, which will broadcast an election notification message within the target region. When other source nodes (called side sources) receive the notification message, they will cancel their RSS-Timers and know the reference source's location piggybacked in the message. The maximum time of RSS-Timer is set to an enough high value to avoid the situation where the RSS-Timers of two nodes expire simultaneously and both nodes begin to act as reference sources.

The reference source will then initiate the construction of the *reference path* (e.g., the shortest path to the sink), and then, the side sources will transmit control packets each specifying a different deviation angle, to form multiple side paths with different initial deviation angles. When side path intersects with the reference path or any of other side paths, data fusion will be performed at the intersection nodes, where redundancy in image data will be eliminated. The related notation used for the data fusion scheme incorporating in the application as shown in Fig. 1, is listed in Table 1.

The common background image is defined as

$$B = I_{i \cap j} \ \ (\forall i, j \in V). \tag{1}$$

We assume the captured images have a common background for sake of simplicity, although in practice, the overlapped portion may be different for various pairs of images. In a data fusion node (e.g.,

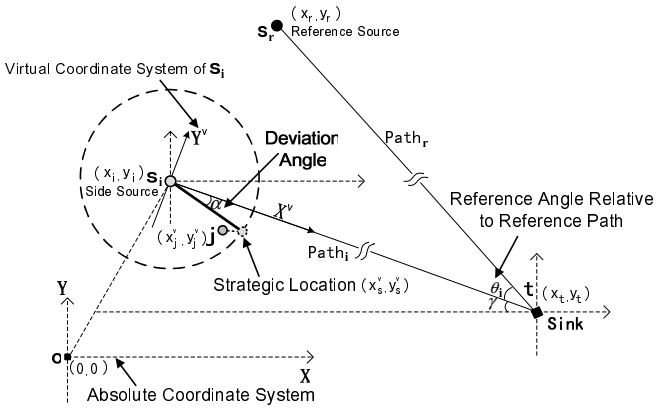


Figure 2: Illustration of the related calculations in DCF directional control.

node A or B in Fig. 1), the image is segmented into a small image (i.e., R_i, and $R_i = I_i - B$) by removing the common background. Then, the redundancy ratio ρ is equal to $\rho = 1 - \frac{|R_i|}{|I_i|}$.

2.2 Directional Control

Directional control presented in this section is one of the two key components of DCF. Based on absolute coordinates, we first present how to calculate a neighbor's virtual coordinates and the reference deviation angle of a side path in Sections 2.2.1 and 2.2.2, respectively. We then introduce the calculation of deviation angle for current node in Section 2.2.3. We propose a next hop selection mechanism based on the deviation angle in Section 2.2.5. Finally, we illustrate how to identify the polarity of a side path in Section 2.2.6.

2.2.1 Calculating the Virtual Coordinates of a Next Hop Candidate

As shown in Fig. 2, the absolute coordinates of sink t, reference source s_r, side source s_i and its neighbor j are denoted by (x_t, y_t), (x_r, y_r), (x_i, y_i) and (x_j, y_j), respectively. In this paper, we employ the virtual coordinates described in [14]. The virtual coordinates of node i's neighbor j is denoted by (x_j^v, y_j^v), which can be calculated by (2). In Eqn.(2), γ is the angle between absolute coordinate system's X-axis with the line connecting side source s_i and the sink.

$$\begin{cases} x_j^v = \cos(\gamma) \cdot (x_j - x_i) + \sin(\gamma) \cdot (y_i - y_i) \\ y_j^v = \cos(\gamma) \cdot (y_j - y_i) - \sin(\gamma) \cdot (x_j - x_i), \end{cases} \quad (2)$$

where $\gamma = \arctan(\frac{y_t - y_i}{x_t - x_i})$.

2.2.2 Calculating the Reference Deviation Angle

Let P_r denote the reference path between reference source and the sink. Let θ_i denote the reference deviation angle between P_r and the line connecting s_i and the sink. Side source s_i can calculate θ_i by the position of its reference source r (x_r, y_r), its own position (x_i, y_i), and the sink's location (x_t, y_t), as shown in Eq.(3). Note that θ_i can be either positive or negative. For instance, if s_i is below P_r,θ_i is positive; otherwise it's negative, as shown in Fig. 2. We can

Table 2: Pseudo-code for selecting the neighbor with the minimum D_j as *NextHop*

```
01 procedure NextHopSelection(V_h)
02   h is the hth hop node along the side path i;
03   V_h is the set of node h's neighbors in the forwarding area;
04 begin
05  calculate f(h) based on current hop counts;
06  calculate α_h^i based on θ_i, P_ff and f(h);
07  calculate (x_s^v, y_s^v) based on α_h^i;
08 for each neighbor j in V_h
09    calculate (x_j^v, y_j^v);
10    calculate ΔD_j according to Eqn.(9);
11 endfor
12 for each neighbor j in V_h
13   if ΔD_j = min{ΔD_k | k ∈ V_h}
14      select j as NextHop;
15      break;
16   endif
17 endfor
```

compute θ_i and the distances between any pair of the nodes as

$$\theta_i = \arccos\left(\frac{(D_t^r)^2 + (D_t^i)^2 - (D_r^i)^2}{2 \cdot D_t^r \cdot D_t^i}\right) \quad (3)$$

$$D_t^r = \sqrt{(x_t - x_r)^2 + (y_t - y_r)^2} \quad (4)$$

$$D_t^i = \sqrt{(x_t - x_i)^2 + (y_t - y_i)^2} \quad (5)$$

$$D_r^i = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2}. \quad (6)$$

2.2.3 Calculating the Deviation Angle at Each Hop

Let α_h^i denote the deviation angle at hop h along the side path P_i. It is used for the current node to make decision for next-hop selection during the construction of P_i [11]. Let P_{ff} denote the path fusion factor, which is the most important control parameter in DCF; let $f(h)$ be an adjusting function at hop h; and let H_{s_i} denote the estimated hop count between s_i and the sink. The per hop distance can be estimated as $\beta * R$, where R is the maximum transmission range. In this paper, we empirically set β to 0.7. Then, α_h^i can be calculated by P_i's reference deviation angle θ_i multiplied by $f(h)$ and P_{ff}, as given in Eqn.(7).

$$\alpha_h^i = \begin{cases} \min[90, P_{ff} * \theta_i * f(h)], & \theta_i \geq 0 \\ \max[-90, P_{ff} * \theta_i * f(h)], & \theta_i < 0. \end{cases} \quad (7)$$

where $f(h) = \frac{h}{H_{s_i}} + 1$ and $H_{s_i} = \lceil \frac{D_i^t}{\beta * R} \rceil$, for $0 < \beta < 1$.

2.2.4 Distance to Strategic Location

Strategic location means the ideal location of current node's next hop. As shown in Fig. 2, the virtual coordinates of the strategic location is denoted by (x_s^v, y_s^v). Based on the deviation angle calculation in Section 2.2.3, (x_s^v, y_s^v) can be calculated by (8).

$$\begin{cases} x_s^v = \cos(\alpha_h^i) \cdot R, \\ y_s^v = -\sin(\alpha_h^i) \cdot R. \end{cases} \quad (8)$$

Then, the distance between a next hop candidate j and the strategic location (denoted by ΔD_j) can be calculated by

$$\Delta D_j = \sqrt{(x_s^v - x_j^v)^2 + (y_s^v - y_j^v)^2} \quad (9)$$

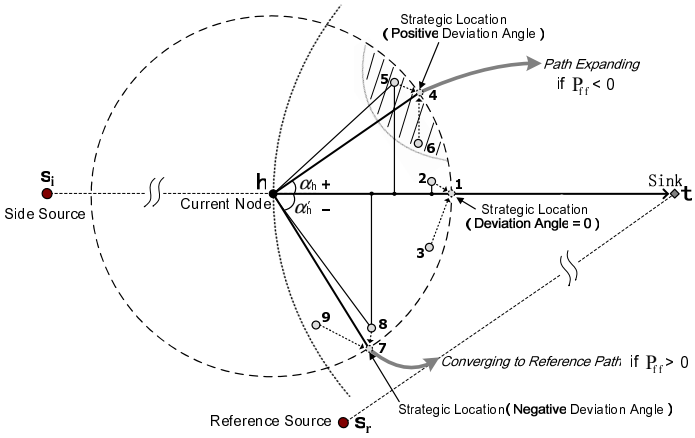


Figure 3: Illustration of Polarity Calculation.

Table 3: The impact of polarity on the effect of side path's construction

Relative to s_r	θ_i	P_{ff}	α_h^i	Effect
below	+	+	+	converging
below	+	-	-	expanding
above	-	+	-	converging
above	-	-	+	expanding

2.2.5 Direction-aware Next-hop-selection

To establish a direction-aware path, a probe message is broadcast initially by a side source for route discovery. A node receiving a probe message will calculate its (x_s^v, y_s^v) and the virtual coordinates of its neighbors. Then, DCF will select as the next hop node whose distance is closest to the strategic location, instead of the neighbor closest to the sink as in traditional geographical routing protocols. Table 2 shows the pseudo-code of the next-hop-selection algorithm in DCF. A selected next hop will continue to broadcast probe message to find its next hop, and so forth, until the sink node is reached. As a result, a path from the source node to sink is established that traverses a few predetermined strategic locations, where data fusion will be performed.

2.2.6 Polarity Map in DCF

Exploiting the adjusting function in Eqn.(7), the deviation angle becomes larger with the probe message making more progress towards the sink. Using the current adjusting function, when P_{ff} has a negative value, the adjusting function aims to evenly distribute the multiple side paths around the limited space in the proximity of the sink. Otherwise, if P_{ff} has a positive value, the side path will converge quickly to the reference path after a few hops from the side source. A suitable choice of the angle adjusting function $f(h)$ will affect the resulting formation of the multipath fusion structure.

As observed in Table 3, the effect of converging or expanding is determined by the sign of P_{ff} (i.e., "+" for converging and "-" for expanding). Thus, in the proposed DCF scheme, P_{ff} efficiently provides a flexible control knob for trade-off between path converging and path expanding.

As the example shown in Fig. 3, s_i is above its reference source s_r, and thus resulting in a negative θ_i. If P_{ff} is negative, according to Table 3, α_h is positive and the side path will be expanded. Otherwise, the path will be converged. In Fig. 3, nodes 1, 4 and 7 are the strategic locations when P_{ff} is zero, negative and positive, respectively. And nodes 2, 5 and 8 are selected as the direction-aware next hop nodes because they are the closest ones to their corresponding strategic locations.

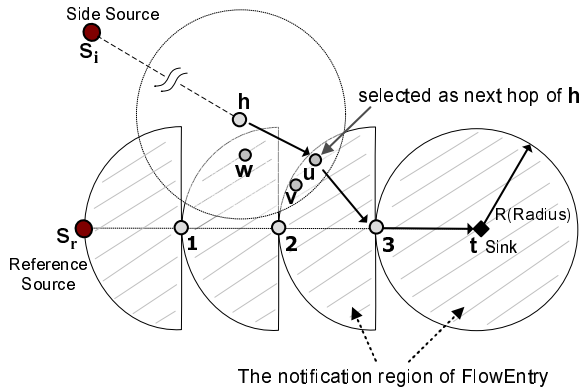


Figure 4: Illustration of the Multipath Merging Protocol.

2.3 Multipath Fusion

2.3.1 Scheme Description

Basically, when the current node receives a probe message, it will performs next-hop-selection as described in Section 2.2.5. Then, it sets up its *FlowEntry*, which is a record of the identifier of the next-hop node pointing to the sink, as the selected next hop, and notify this information to its one hop neighbors whose distance to the sink is larger than itself, as shown in Fig. 4.

In Fig. 4, during the construction of P_i initiated by s_i, the hth hop node h will check its neighbor information table. If there exists one neighbor with FlowEntry set, h will forward the probe message directly to that neighbor without direction-aware next-hop-selection. In the case that there are more than one neighbor which has already established its FlowEntry, the one whose next hop indicated in its FlowEntry is closest to the sink, will be selected as the next hop node of s_i^{k-1}. When a node with FlowEntry set receives the probe message, it will be discarded, which implicitly means the construction of P_i is terminated and the side path is merged into another path (i.e., the reference path or another side path).

As shown in Fig. 5, we divide the probe message fields into three parts, common fields, fields for next-hop-selection and for path fusion, respectively. Common fields represent the universal information for packet routing. In the fields for next-hop-selection, ReferenceSourcePOS, SideSourcePOS and SinkPOS are used to calculate neighboring nodes' virtual coordinates at each hop; Both θ_i and H_{s_i} are calculated by side source s_i and are not changed during path construction. In the fields for path fusion, MyID represent the identifer of current node; MyPOS is used for its neighbors to judge whether they are upstream node compared to current node; If SinkFlag is set to 1, all of the neighboring nodes will mark their FlowEntry to be set. The fields of NextHop, MyID and MyPOS will be changed at each hop, while SinkFlag is only set by the sink. And the other fields are fixed.

2.3.2 Example

As the example shown in Fig. 4, there are two source nodes, one is reference source s_r and the other is side source s_i. We assume that s_r has initiated the construction of reference path (s_r,1,2,3,t).

[1]if multiple neighbors use the same next hop node, which means that next hop has already been a fusion point, the neighbor whose distance is the closest to the sink will be selected.

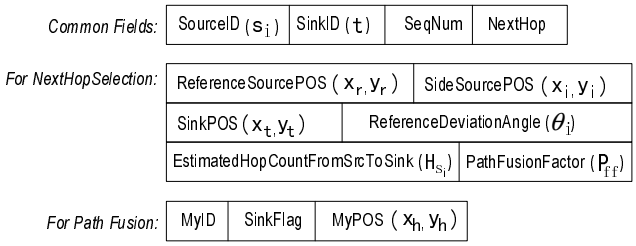


Figure 5: Illustration of Packet Format.

Since nodes $1, 2, 3$ are selected as the next hop nodes, their upstream neighbors will overhear the probe message broadcast by them. Thus, those upstream neighbors (e.g., nodes u, v and w) will know they have a neighbor (i.e., node 1, node 2, or node 3) with FlowEntry set. As shown in Fig. 4, the region of upstream neighbors usually forms a half circle.

We assume that s_i has constructed a partial path up to node h (i.e., $s_i \rightarrow h$). And we assume u, v and w are the neighbors of h. Both u and v have a neighbor 3 whose FlowEntry has been set while v has a neighbor 2 with FlowEntry set. When h checks its neighbor information table, it will find the next hop nodes in both u and v's FlowEntries are closer to the sink than w. At this moment, w is excluded from the next hop candidate list. Next, between u and v, h checks which one is closer to the sink. Finally, it selects u as its next hop and forward the probe message it. When u receives the message, since its FlowEntry has already set, it will terminates the construction of the side path which means P_i is fused with P_r.

2.4 Pattern Classification of DCF with Varying Multipath Fusion Factors

The path constructions initiated by side sources will halt in the proximity of reference path or other side path. These partially constructed "sub-side" paths are fused into previously built side paths which has already fused into reference path. Thus, the multipath fusion tree forms a hierachical formation pattern. If P_{ff} is a negative value, the side path will be expanded relative to the reference path, yielding thereafter expanding multipath fusion. With P_{ff} increasing, the formation pattern will be changed gradually from multipath-expanding to converging fusion, as shown in Fig. 6.

We categorize the patterns as follows:

- *large(-)*:[2] As shown in Fig. 6-(a), the inner side path expands widely, and thus blocks the outer side paths which are merged into those inner paths at the beginning of side paths' construction.

- *medium(-)*: In Fig. 6-(b), the side paths expand in parallel, which yields disjointed multiple paths.

- *close to 0*: The formation of fusion structure is shown in Fig. 6-(c). When P_{ff} = 0, the side path degrades to the shortest path pointing to the sink. It works similarly as opportunistic aggregation in Directed Diffusion [12].

- *medium(+)*: By tuning P_{ff} between 0 and a large positive value, aggregation points will be shifted from sink to the reference source (e.g., Fig. 6-(d)). Higher P_{ff} yields faster converging to the reference path than lower one. By such adjustment, DCF achieves an efficacy of finding optimal aggregation points similar to that of a heuristic approach (e.g., Oceanus [5]).

- *large(+)*: When P_{ff} is a large positive value, the side paths will be pulled towards the reference source at a fast convergence speed. Then, all of the side sources set up paths connecting the reference source, which makes the reference source work as a cluster head. Such multipath fusion formation is similar to that obtained by greedy incremental algorithms [3, 13]. Fig. 6-(e) shows an extreme case.

In summary, with P_{ff} changing from large(-) to large(+), the fusion formations almost include all existing formations presented in previous work (e.g., DD [12], Oceanus [5], greedy incremental [3, 13] algorithms). In DCF, P_{ff} provides a convenient control knob for easily trading off among the various scenarios. In addition, existing algorithms cannot generate the multipath-expanding structure. The multipath-expanding, introduced in DCF, takes advantage of both load balancing and data redundancy eliminating, increase the multipath aggregate bandwidth and efficiency.

3. PERFORMANCE EVALUATION

3.1 Simulation Methodology

We implement our protocols and perform simulations using OPNET Modeler [15, 16]. The network is uniformly deployed over a 1000m $\times$ 500m field. To verify the nice scaling property of DCF, we select large scale network scenario with 800 nodes.We let multiple source nodes be located at the left side of the field and one sink stay at the right side. The sensor application module consists of a constant-bit-rate source, which generates a sensed data every 100ms (1024 bits each). As in [12], we use IEEE 802.11 Distributed Coordinate Function as the underlying MAC, and the radio transmission range (R) is set to 60m. The data rate of the wireless channel is 1 Mb/s. All messages are 64 bits in length. We assume both the sink and sensor nodes are stationary. For consistency, we use the same energy consumption model as in [12]. The initial energy of each node is 5 Joules. The transmit, receive and idle power consumptions are 0.66 W, 0.395 W, and 0.035 W, respectively. We count all types of energy consumptions in the simulations, including transmission, receiving, idling, overhearing, collisions and other unsuccessful transmissions, MAC layer headers, retransmissions, and RTS/CTS/ACKs.

In this paper, the following three performance metrics will be evaluated:

- *Lifetime* - It's the time when the first node exhausts its energy.

- *Average Communication Energy*: the total communication energy consumption, including transmitting, receiving, retransmissions, overhearing and collision, over the total number of distinct reports received at the sink.

- *Average End-to-end Packet Delay* - It includes all possible delays during data dissemination, caused by queuing, retransmission due to collision at the MAC, and transmission time.

- *Integrated Performance*: For time-sensitive applications over energy constrained WSNs, it is important to consider lifetime, energy and delay simultaneously. We adopt $\eta = \frac{Delay \cdot Energy}{Lifetime}$ to evaluate the integrated performance. The higher the η, the better the composite performance provided by the WSN to energy-efficiently support time-constrained services for longer time.

[2]The notation "large(-)" means that P_{ff} is negative and its absolute value is relatively large, while "small(+)" means that P_{ff} has a positive small value, and so forth.

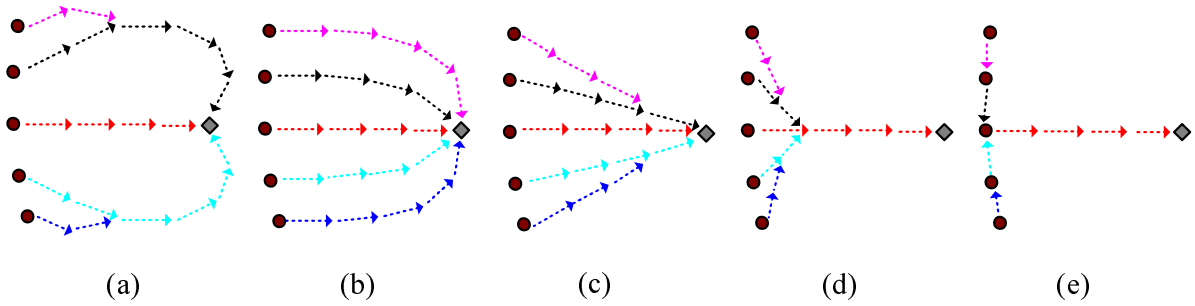
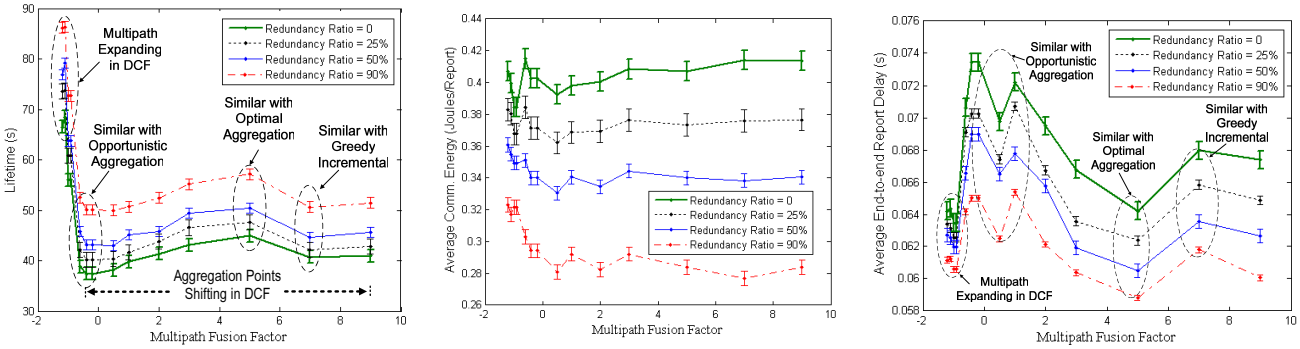


Figure 6: The Pattern Classification of DCF with varying P_{ff}: (a) large(-); (b) medium(-); (c) close to 0; (d) medium(+); (e) large(+).



(a) Lifetime (b) Average communication energy (c) Average end-to-end delay

Figure 7: The impact of P_{ff} and ρ on DCF performances: (a) Lifetime; (b) Average communication energy; (c) Average end-to-end delay.

3.2 Simulation Results

In the following sets of simulation results, P_{ff} is varied from -1.2 to 9, data redundancy ratio is changed by varying ρ (the definition is given in Section 2.1) from 0% to 90%. When $\rho = 0$, there is no redundancy in the sensory data. As ρ is increased, more data redundancy presents in sensory data, which can be eliminated at the path fusion points.

In Fig. 7, we present the impact of P_{ff} and ρ on the DCF performance. In Fig. 7(a), we find the lifetime is always the highest when P_{ff} is close to -1.2, which illustrates that multipath-expanding increases aggregate source-to-sink bandwidth and achieve better load balancing than traditional multipath-converging. Furthermore, the lifetime of multipath-expanding is insensitive to the changes of ρ, which indicates that the paths are more disjoint than multipath converging, and thus taking less advantage of redundancy eliminating. When P_{ff} is close to 0, the lifetime reaches its lowest points, which illustrates that opportunistic aggregation [12] makes the nodes in the proximity of the sink critical. When P_{ff} is increased, the fusion points are shifted from the sink to source nodes. When P_{ff} is around 5, the fusion points are scattered evenly around the middle between source and sink, which achieves highest lifetime among the multipath-converging cases.

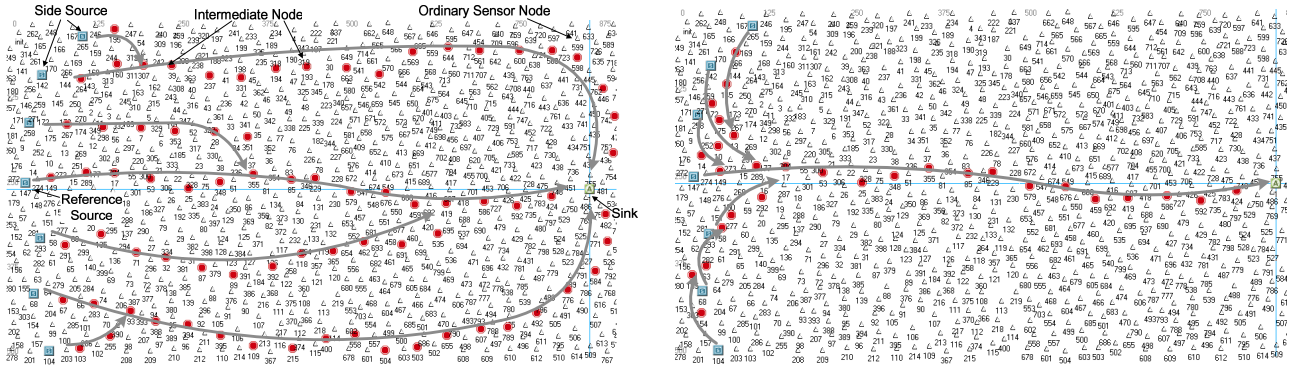
In Fig. 7(b), we find the energy consumption decreases for increased ρ. As shown in Fig. 7(c), the delay curves have two peaks at $P_{ff} = 0$ and $P_{ff} = 7$, respectively. This is because, in both opportunistic and greedy-incremental like schemes [3, 13], either the proximity of the sink or the reference sink become bottlenecks for multiple flows that compete for the limited bandwidth. However, congestion is not a problem for multipath expanding due to the load balancing effect of multipath routing, the delay stay low as similar to heuristic-based optimal aggregation tree algorithms (e.g., Oceanus [5]).

In Fig. 8, though the heuristic-based aggregation scheme achieves best integrated performance among all of the multipath-converging, the multipath-expanding even has higher performance since it takes advantage of both load balancing and data redundancy eliminating. This fact indicates that multipath-expanding is a new approach to achieve better performance rather than adjusting aggregation points in multipath-converging, especially for extending network lifetime.

Fig. 9 shows the snapshots of two OPNET simulations. The snapshots are for the multipath-expanding and multipath-converging in DCF's path construction and illustrate its flexibility to accommodate a wide range of applications.

4. CONCLUSION

For sensor data fusion, finding the optimal aggregation points in the network is known to be NP-Complete and is still an open area of research. In this paper, we propose a novel directional multipath fusion (DCF) scheme, which is fully distributed and only uses local information at each node for building the fusion structure. By tuning design parameter termed multipath fusion factor, DCF can achieve different fusion patterns, which can be divided into two categories, i.e., multipath-converging and multipath-expanding. While previous work can be categorized as multipath-converging, this paper first reveals that multipath-expanding achieves the efficacy of exploiting multiple paths to facilitate both load balancing and enlarging the aggregate bandwidth. Our extensive simulations demonstrate the efficacy of the proposed approach.



(a) Expanding Fusion in DCF

(b) Converging Fusion in DCF

Figure 9: Simulation Animation with Varying P_{ff}.

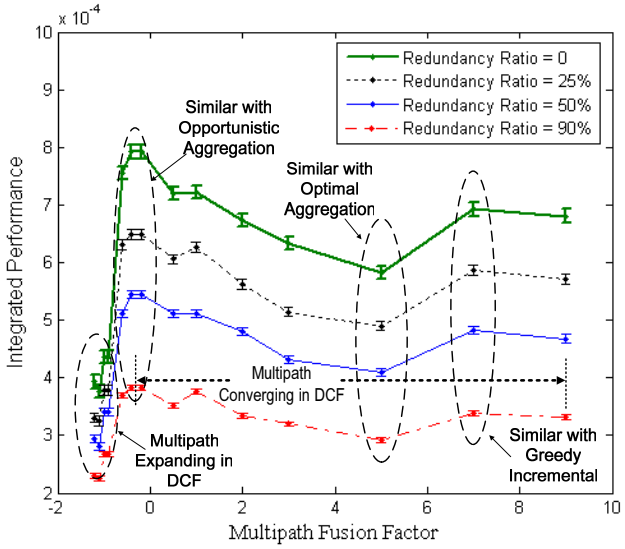


Figure 8: The impact of P_{ff} and ρ on integrated performance of DCF.

5. ACKNOWLEDGMENTS

This work was supported in part by the Canadian Natural Sciences and Engineering Research Council under grant STPGP 322208-05. Shiwen Mao's research has been supported in part by the National Science Foundation under Grant ECCS-0802113, and through the Wireless Internet Center for Advanced Technology (WICAT) at Auburn University.

6. REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-116, August 2002.

[2] I. Stojmenovic and S. Olariu, "Data-centric protocols for wireless sensor networks," in: *Handbook of Sensor Networks: Algorithms and Architectures*, Wiley, 2005, pp. 417-456.

[3] B. Krishnamachari, D. Estrin, and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," *Proc. of International Workshop on Distributed Event-Based Systems*, July 2002.

[4] B. Krishnamachari, D. Estrin, S.Wicker, "Modeling Data-Centric Routing in Wireless Sensor Networks," *IEEE INFOCOM'02*, New York, NY, 2002.

[5] A. Harris, R. Snader and I. Gupta, "Building trees based on aggregation efficiency in sensor networks," *Journal of Ad Hoc Networks*, No. 5, pp.1317-1328, 2007

[6] K. Akkaya, M. Demirbas and R. Aygun, "The Impact of Data Aggregation on the Performance of Wireless Sensor Networks," *Wireless Communications and Mobile Computing Journal*, Vol. 8, No. 2, pp.171-193, 2008.

[7] R. Snader, A. Harris and R. Kravets, "Tethys: a distributed algorithm for intelligent aggregation in sensor networks," *IEEE WCNC'07*, 2007.

[8] I. Stojmenovic, "Position-Based Routing in Ad Hoc Networks," *IEEE Comm. Magazine*, Vol.40, No.7, pp.128-134, July 2002.

[9] H. Frey and I. Stojmenovic, "On Delivery Guarantees of Face and Combined Greedy-Face Routing Algorithms in Ad Hoc and Sensor Networks," *ACM MOBICOM*, pp.390-401, 2006.

[10] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM Wireless Networks*, pp.609-616, November 2001.

[11] M. Chen, V.C.M. Leung, S. Mao and Y. Yuan, "Directional Geographical Routing for Real-Time Video Communications in Wireless Sensor Networks," *Elsevier Computer Communications*, Vol. 30, No. 17, pp.3368-3383, 2007.

[12] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," *Proc. of ACM MobiCom'00*, Boston, MA, August 2000.

[13] C. Intanagonwiwat, D. Estrin, R. Govindan and J. Hei, "Impact of network density on data aggregation in WSNs," *Proc. IEEE ICDCS*, 2002.

[14] M. Chen, X. Wang, V. Leung and Y. Yuan, "Virtual Coordinates Based Routing in Wireless Sensor Networks," *Sensor Letters*, Vol.4, pp.325-330, 2006

[15] Min Chen, "OPNET Network Simulation," *Press of Tsinghua University*, China, April 2004, ISBN 7-302-08232-4, 352 pages.

[16] OPNET Modeler, [online] Available: `http://www.opnet.com`.