Mobile Netw Appl
DOI 10.1007/s11036-008-0133-6

Directional Controlled Fusion in Wireless Sensor Networks

Min Chen · Victor C. M. Leung · Shiwen Mao

Abstract Though data redundancy can be eliminated at aggregation point to reduce the amount of sensory data transmission, 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 notation of “multipath expanding” to solve the above problems by jointly considering data fusion and load balancing. We propose a novel directional-controlled fusion (DCF) scheme, consisting of two key algorithms termed as 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 simulations that verify the effectiveness of the proposed scheme.

Keywords data fusion · image sensor networks · multipath routing · wireless sensor networks

M. Chen (✉) · V. C. M. Leung
Department of Electrical & Computer Engineering,
University of British Columbia, Vancouver,
British Columbia, Canada V6T 1Z4
e-mail: minchen@ece.ubc.ca

S. Mao
Department of Electrical & Computer Engineering,
Auburn University, Auburn, AL 36849-5201, USA

1 Introduction

Wireless sensor networks (WSNs) [3] 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 problem. Building an optimal aggregation tree (e.g. Weighted Steiner Tree [10, 11], or schemes approximating Steiner Tree [7]) requires centralized control; before the algorithm starts, complete knowledge of all the data flows should 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 [2]. 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 [7, 15] has considered both aggregation efficiency and aggregation cost in building efficient aggregation trees, contension 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 the existing 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 paper, we focus on eliminating data redundancy 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) or a localization algorithm [13]. In our system, several source nodes equipped with cameras take images of a certain target object and transmit them to a sink node. Since the cameras are located in the same target region, the captured images may be highly correlated, e.g., they may have a similar background. We aim at finding an appropriate tree formation for eliminating such redundancy in sensory data while considering other quality of service (QoS) requirements.

An example is shown in Fig. 1, in which three source nodes capture and transmit images via different disjoint 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. Assume that the image flow of camera-3 merges 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.[1] 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.

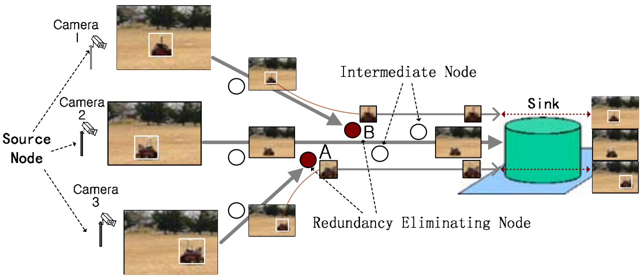


Fig. 1 Example of data redundancy eliminating in directional-controlled fusion

We propose a novel directional-controlled fusion (DCF) scheme. DCF is a combination of two algorithms: (i) directional control algorithm for controlling path directions, and (ii) path fusion algorithm for building the multipath fusion tree with the hierarchical structure. Our main contributions are two-olds: (i) introducing a new concept of "multipath-expanding" which increases the multipath aggregate bandwidth and efficiency by taking advantage of both load balancing and data redundancy elimination, and (ii) achieving flexible trade-offs between multipath-converging and multipath-expanding.

As opposed to the existing data fusion solutions, the DCF scheme has the following unique 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, 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.

[1] CMOS cameras have been successfully incorporated into I-Mote2 sensor nodes [1]; each I-Mote2 nnode has 32 MB of SDRAM and a PXA271 XScale CPU running at 416MHz and is capable of performing in-node image-processing [14].

The rest of the paper is organized as follows. Section 2 presents related work. We describe the DCF algorithm in Section 3. Simulation model and experiment results are presented in Section 4. Section 5 concludes the paper.

2 Related work

Data aggregation (also referred to as in-network processing) is a common technique for reducing the amount of data in WSNs. In opportunistic aggregation (e.g., Directed Diffusion [9]), when data flows meet in an intersection point through shortest paths from the sources to the sink, possible data aggregation is performed to decrease the communication energy consumption. Since scattered data flows originated from difference source nodes shrink to the same sink along the shortest paths, the opportunity of multiple flows' meeting at the same node is higher at the proximity of the sink. Thus, the aggregation points in opportunistic aggregation are more likely close to the sink.

In contrast, some heuristic-based aggregation tree algorithms use greedy incremental aggregation [8, 12]. In such methods, one source initiates a shortest path flow to the sink. Then, the other sources connect to that path via shortest paths, which generally results in aggregation points closer to the sources. However, the efficiency of the greedy incremental method is entirely determined by the first path and can result in very inefficient aggregation trees. Thus, finding the optimal aggregation points in the network is still an open area of research. A starting point is the work by Krishnamachari et al. [10, 11], which first reduce finding the optimal aggregation tree to a Minimum Steiner Tree problem. This problem is known to be NP-Complete [2], and possibly can be approximately solved by heuristic-based approaches (e.g., Oceanus [7]). In addition, data aggregation can be performed by mobile agent while it visits each target source [4].

3 Directional controlled fusion

3.1 Architecture overview

As the example in Fig. 1, 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, 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 target region, or closeness to the sink, etc.) using the following mechanism. Initially, every source node starts a so-called Reference-Source-Selection-Timer (RSS-Timer). A random value is set to each timer based on a specific criterion. A smaller value indicates a source has higher eligibility as the reference source. The source whose RSS-Timer expires first will then 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 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 and redundancy in image data will be eliminated.

The common background image is defined as $B = I_{i \bigcap j}$ $(\forall i, j \in V)$. 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., 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 $\rho = 1 - |R_i|/|I_i|$.

3.2 Directional control

3.2.1 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 [6]. The virtual coordinates of node s_i's neighbor j is denoted by (x_j^v, y_j^v), which can be calculated by Eq. 1. In Eq. 1, γ is the angle between

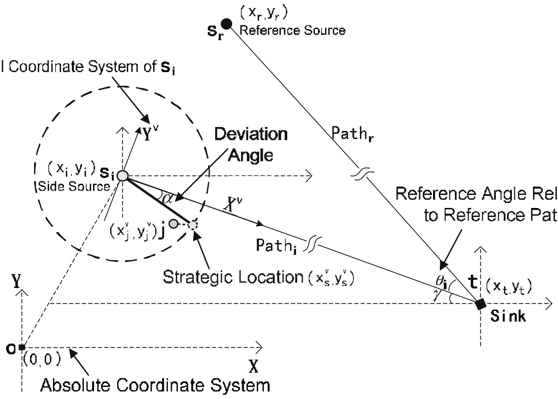


Fig. 2 Illustration of the related calculations in DCF directional control

absolute coordinate system's X-axis with the line connecting side source s_i and the sink.

$$\begin{cases} x_j^v = \cos(\gamma) \cdot (x_j - x_i) + \sin(\gamma) \cdot (y_j - y_i) \\ y_j^v = \cos(\gamma) \cdot (y_j - y_i) - \sin(\gamma) \cdot (x_j - x_i), \end{cases} \tag{1}$$

where $\gamma = \arctan[(y_t - y_i)/(x_t - x_i)]$.

3.2.2 Calculating 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. 2. 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 compute θ_i and the distance between any pair of the nodes as

$$\theta_i = \arccos\left[\frac{(D_t^r)^2 + (D_t^i)^2 - (D_r^i)^2}{2 \cdot D_t^r \cdot D_t^i}\right], \tag{2}$$

where $D_t^r = \sqrt{(y_t - y_r)^2 + (x_t - x_r)^2}$, $D_t^i = \sqrt{(y_t - y_i)^2 + (x_t - x_i)^2}$, and $D_r^i = \sqrt{(y_r - y_i)^2 + (x_r - x_i)^2}$.

3.2.3 Calculating 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 [5]. 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 $\beta \cdot R$, 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 Eq. 3.

$$\alpha_h^i = \begin{cases} \min[90,\ P_{ff} \cdot \theta_i \cdot f(h)], & \theta_i \geq 0 \\ \max[-90,\ P_{ff} \cdot \theta_i \cdot f(h)], & \theta_i < 0. \end{cases} \tag{3}$$

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

3.2.4 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 3.2.3, (x_s^v, y_s^v) can be calculated by Eq. 4.

$$\begin{cases} x_s^v = \cos(\alpha_h^i) \cdot R, \\ y_s^v = -\sin(\alpha_h^i) \cdot R. \end{cases} \tag{4}$$

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

$$\Delta D_j = \sqrt{(x_s^v - x_j^v)^2 + (y_s^v - y_j^v)^2} \tag{5}$$

3.2.5 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 will calculate its (x_s^v, y_s^v) and the virtual coordinates of its neighbors. DCF will then 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 1 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 source-sink path will be established.

3.2.6 Polarity map in DCF

Exploiting the adjusting function in Eq. 3, the deviation angle becomes larger when the probe makes 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

Table 1 Pseudo-code for selecting the neighbor with the minimum D_j as *NextHop*

```
procedure NextHopSelection($V_h$)
  $h$ is the $h$th hop node along the side path $i$;
  $V_h$ is the set of node $h$'s neighbors in the
    forwarding area;
begin
 calculate $f(h)$ based on current hop counts;
 calculate $\alpha_h^i$ based on $\theta_i$, $P_{ff}$ and $f(h)$;
 calculate $(x_s^v, y_s^v)$ based on $\alpha_h^i$;
for each neighbor $j$ in $V_h$
   calculate $(x_j^v, y_j^v)$;
   calculate $\Delta D_j$ according to Eq. 5;
endfor
for each neighbor $j$ in $V_h$
  if $\Delta D_j = min\{\Delta D_k | k \in V_h\}$
     select $j$ as NextHop;
     break;
  endif
endfor
```

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 2, 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 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 2, 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.

3.3 Multipath fusion

Basically, when the current node receives a probe message, it will performs next-hop-selection as described in Section 3.2.5. 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.[2] 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).

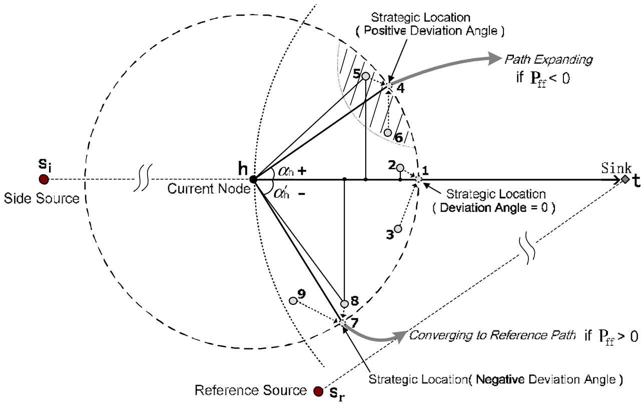


Fig. 3 Illustration of polarity calculation

Table 2 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

As the example 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). Since nodes 1, 2, 3 are selected as the next hop nodes, their upstream neighbors will overhear the probe message broadcast by them.[3] 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.

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

[2] 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.

[3] The upstream neighbors mean the neighbors whose distances to the sink are larger than the distance between current node and the sink. Generally, the region of upstream neighbors forms a half circle, as shown in Fig. 4.

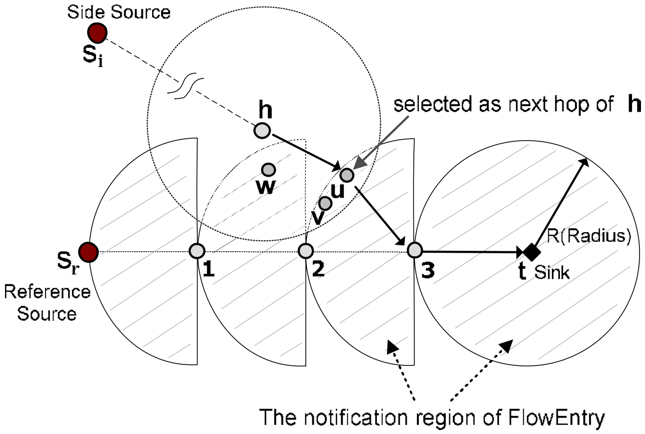

Side Source
S_i
h
selected as next hop of h
w
u
v
S_r
Reference Source
1
2
3
R(Radius)
t Sink
The notification region of FlowEntry

Fig. 4 Illustration of the multipath merging protocol

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.

3.4 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. 5.

We categorize the patterns as follows:

- *large(-)*:[4] As in Fig. 5a, 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. 5b, the side paths expand in parallel, which yields disjointed multiple paths.
- *close to 0*: The formation of fusion structure is shown in Fig. 5c. 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 [9].
- *small(+)*: The formation of fusion structure is shown in Fig. 5d. It also can be deemed as "directed-path fusion (DPF)" for existing schemes, which generally set up paths to involve nodes that are closer to the sink.
- *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. 5e and Fig. 5f). 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 [7]).
- *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 [8, 12]. Fig. 5e shows an extreme case.
- *too large(+)*: The formation of fusion structure is shown in Fig. 5h. The side paths are expanded towards the opposite direction, which means P_{ff} is set to too large a value inefficiently.

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 [9], Oceanus [7], greedy incremental [8, 12] algorithms).

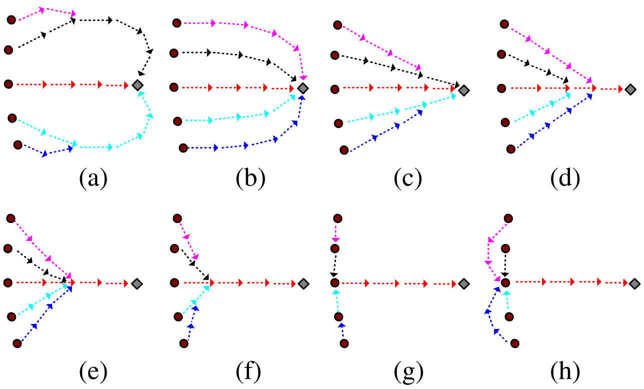
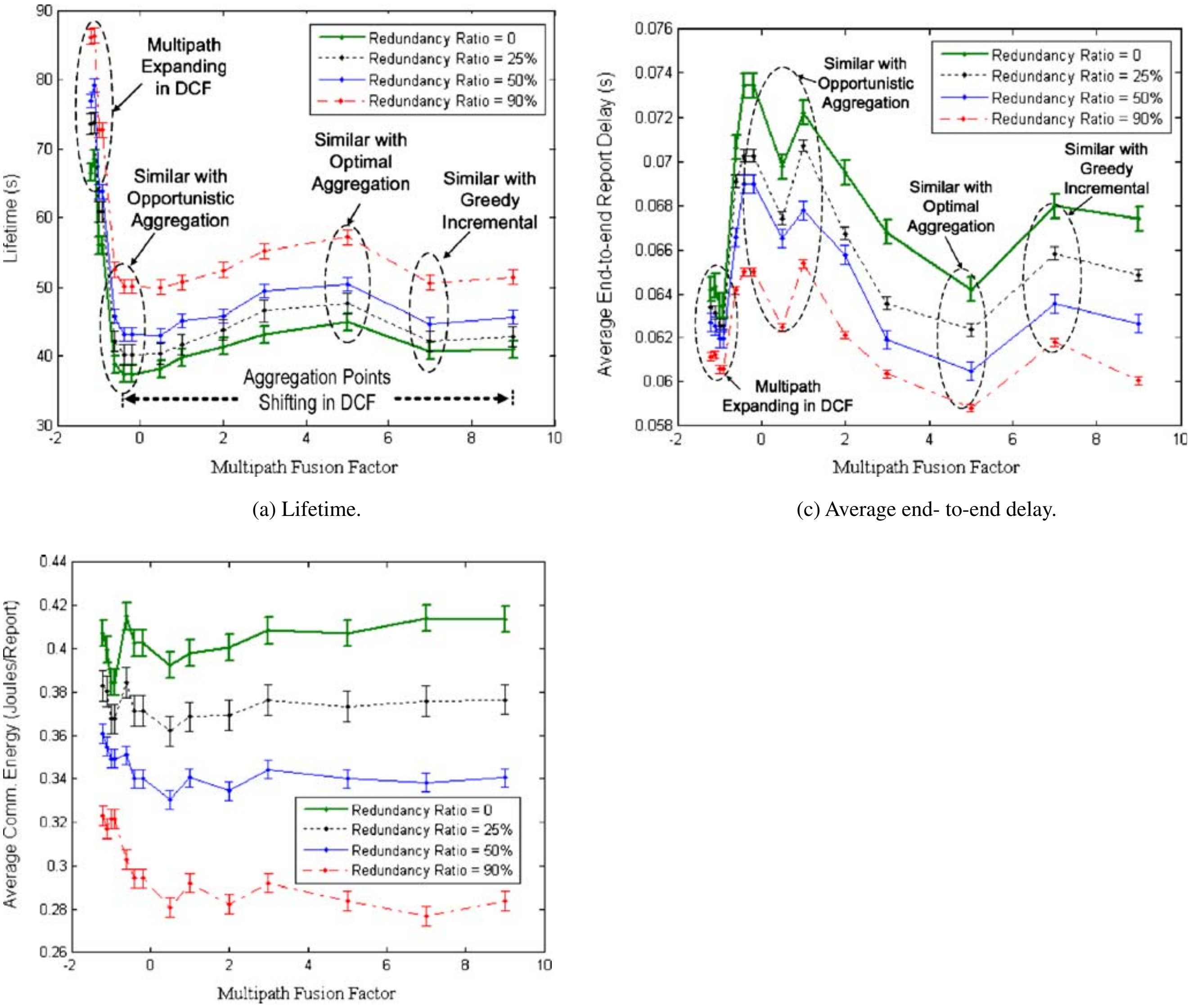


Fig. 5 The pattern classification of DCF with varying P_{ff}: **a** large(-); **b** medium(-); **c** close to 0; **d** small(+); **e** medium(+); **f** medium(+); **g** large(+); **e** too large(+)

[4] 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.



(a) Lifetime.

(c) Average end- to-end delay.

(b) Average communication energy.

Fig. 6 The impact of P_{ff} and ρ on DCF performances (**a**–**c**)

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.

4 Performance evaluation

We implement our protocols and perform simulations using OPNET Modeler. The network is uniformly deployed over a 1000 m × 500 m field. To verify the scaling property of DCF, we select a 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 1024 bits every 100 ms. As in [9], we use IEEE 802.11 Distributed Coordinate Function as the underlying MAC, and the radio transmission range (R) is set to 60 m. 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 [9]. The transmit, receive and idle power consumptions are 0.66 W, 0.395 W, and 0.035 W, respectively. The initial energy of each node is 5 Joules. We count all types

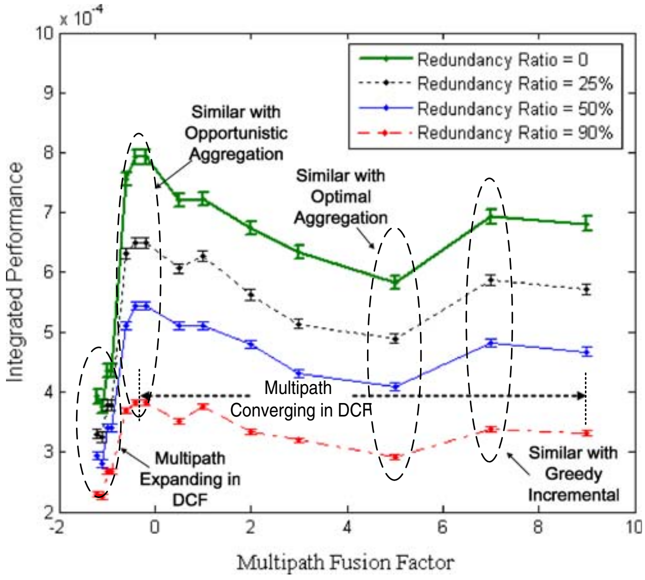


Fig. 7 The impact of P_{ff} and ρ on integrated performance of DCF

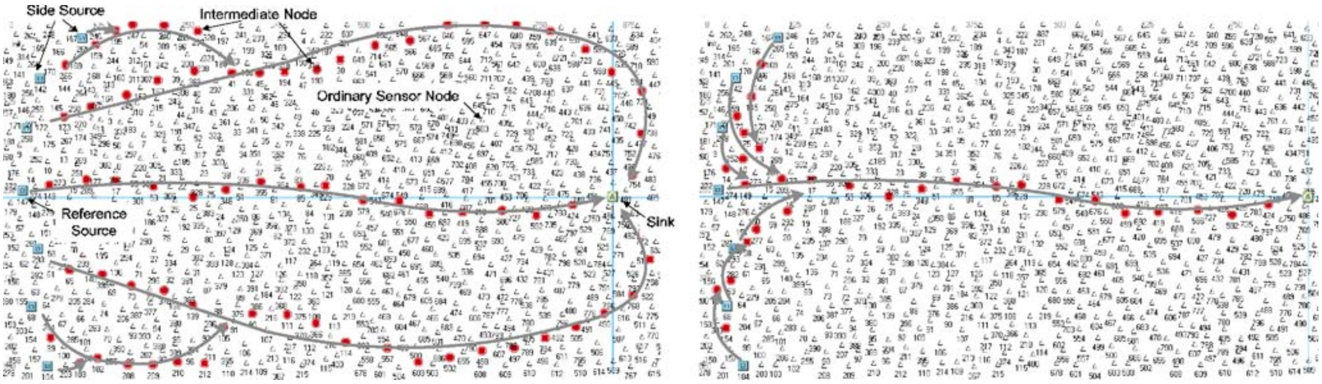
of energy consumptions in the simulations, including transmission, receiving, idling, overhearing, collisions and other unsuccessful transmissions, MAC layer headers, retransmissions, and RTS/CTS/ACKs.

We consider the following three performance metrics:

- *Lifetime*: 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*: including 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 define $\eta = \frac{Delay \cdot Energy}{Lifetime}$ as the *integrated performance metric*. The higher the η, the better the overall WSN performance.

In the following simulation results, P_{ff} is varied from -1.2 to 9, and data redundancy ratio is changed by varying redundancy ratio ρ (the definition is given in Section 3.1) 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. 6, we present the impact of P_{ff} and ρ on the DCF performance. In Fig. 6a, 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 [9] makes the nodes in the proximity of the sink critical.



(a) Expanding fusion in DCF

(b) Converging fusion in DCF.

Fig. 8 Simulation animation with varying multipath fusion factor (**a**, **b**)

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. 6b, we find the energy consumption decreases for increased ρ. As shown in Fig. 6c, 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 [8, 12], either the proximity of the sink or the reference source 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 [7]).

In Fig. 7, although the heuristic-based aggregation scheme achieves the best integrated performance among all the multipath-converging schemes, the multipath-expanding scheme achieves even higher performance since it takes advantage of both load balancing and data redundancy eliminating. This fact indicates that the novel multipath-expanding approach can achieve better performance rather than merely adjusting aggregation points in multipath-converging, especially for extending network lifetime.

Figure 8 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.

5 Conclusion

For sensor data fusion, finding the optimal aggregation points in the network is known to be NP-Complete and is still an open problem. 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, 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 increasing the aggregate bandwidth. Our simulations demonstrate the efficacy of the proposed approach.

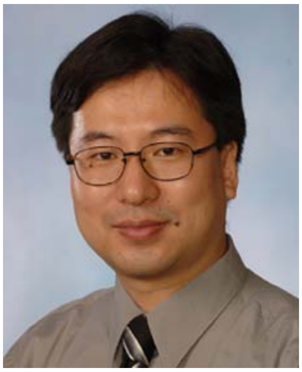
Acknowledgements This work was supported in part by the Canadian Natural Sciences and Engineering Research Council under grant STPGP 322208-05. S. Mao's research has been supported in part by the U.S. National Science Foundation under Grant ECCS-0802113, and through the Wireless Internet Center for Advanced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

References

1. Crossbow Technology (2008) BehaviorScope—Using WSN to recognize human behavior. http://blog.xbow.com/xblog/2007/06/behaviorscope_u.html.
2. Akkaya K, Demirbas M, Aygun R (2008) The impact of data aggregation on the performance of wireless sensor networks. Wirel Commun Mob Comput J 8(2):171–193
3. Al-Karaki J, Kamal E (2004) Routing techniques in wireless sensor networks: a survey. IEEE Pers Commun 11(6):6–28
4. Chen M, Kwon T, Yuan Y, Choi Y, Leung V (2007) Mobile agent-based directed diffusion in wireless sensor networks. EURASIP J Appl Signal Process 2007. Article ID 36871. doi:10.1155/2007/36871
5. Chen M, Leung V, Mao S, Yuan Y (2007) Directional geographical routing for real-time video communications in wireless sensor networks. Comput Commun 30(17):3368–3383
6. Chen M, Wang X, Leung V, Yuan Y (2006) Virtual coordinates based routing in wireless sensor networks. Sens Lett 4(3):325–330
7. Harris A, Snader R, Gupta I (2007) Building trees based on aggregation efficiency in sensor networks. J Ad Hoc Netw 5(8):1317–1328
8. Intanagonwiwat C, Estrin D, Govindan R, Heidemann J (2002) Impact of network density on data aggregation in wireless sensor networks. In: Proc. IEEE ICDCS'02. Vienna, Austria
9. Intanagonwiwat C, Govindan R, Estrin D, Heidemann J, Silva F (2003) Directed diffusion for wireless sensor networking. IEEE/ACM Trans Netw 11(1):2–16
10. Krishnamachari B, Estrin D, Wicker S (2002) Modeling data-centric routing in wireless sensor networks. In: Proc. IEEE INFOCOM'02. New York, NY
11. Krishnamachari B, Estrin D, Wicker S (2002) The impact of data aggregation in wireless sensor networks. In: Proc. international workshop on distributed event-based systems. Vienna, Austria, pp 575–578
12. Krishnamachari B, Estrin D, Wicker S (2002) The impact of data aggregation in wireless sensor networks. In: Proc. international workshop of distributed event based systems. Vienna, Austria
13. Li J, Jannotti J, Couto DD, Karger D, Morris R (2000) A scalable location service for geographic ad hoc routing. In: Proc. ACM MobiCom'00. Boston, MA, pp 120–130
14. Rowe A, Goel D, Rajkumar R (2007) FireFly mosaic: a vision-enabled wireless sensor networking system. In: Proc. IEEE RTSS'07. Tucson, AZ, pp 459–468
15. Snader R, Harris A, Kravets R (2007) Tethys: a distributed algorithm for intelligent aggregation in sensor networks. In: Proc. IEEE WCNC'07. Hong Kong, pp 4117–4122

Min Chen received the Ph.D degree in Electrical Engineering from South China University of Technology in 2004, when he was 23 years old. Since Mar. 2006, he is Post-Doctoral Fellow in Department of Electrical and Computer Engineering at University of British Columbia. Before joining UBC, he has been a Post-Doctoral Fellow in School of Computer Science and Engineering at Seoul National University for one and half years. Dr. Chen's research interests include algorithmic, optimization and performance issues in wireless ad hoc and sensor networks and multimedia communications over wireless networks. He was interviewed by Chinese Canadian Times where he appeared on the celebrity column in 2007. He is the author of a textbook *OPNET Network Simulation* (Tsinghua Univ. Press, 2004). Dr. Chen received the *Best Paper Runner-up Award* from The Fifth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine) 2008.

Victor C.M. Leung received the B.A.Sc. (Hons.) and PhD degrees, both in electrical engineering, from the University of British Columbia (UBC) in 1977 and 1981, respectively. He was the recipient of many academic awards, including the APEBC Gold Medal as the head of the 1977 graduate class in the Faculty of Applied Science, UBC, and the NSERC Postgraduate Scholarship. From 1981 to 1987, Dr. Leung was a Senior Member of Technical Staff and satellite systems specialist at MPR Teltech Ltd. In 1988, he was a Lecturer in Electronics at the Chinese University of Hong Kong. He returned to U.B.C. as a faculty member in 1989, where he is a Professor and holder of the TELUS Mobility Research Chair in Advanced Telecommunications Engineering in the Department of Electrical and Computer Engineering. His research interests are in mobile systems and wireless networks. Dr. Leung is a Fellow of IEEE and a voting member of ACM. He is an editor of the IEEE Transactions on Wireless Communications, an associate editor of the IEEE Transactions on Vehicular Technology, and an editor of the International Journal of Sensor Networks.



Shiwen Mao received the Ph.D. degree in Electrical and Computer Engineering (ECE) from Polytechnic University, Brooklyn, NY in 2004. He was a Research Scientist at Virginia Tech, Blacksburg, VA from December 2003 to April 2006. Currently, he is an Assistant Professor in ECE at Auburn University, Auburn, AL. Dr. Mao's research interests include modeling and optimization of wireless networks, cognitive networks, and multimedia communications. He is on the Editorial Board of the Hindawi Advances in Multimedia Journal and the Wiley International Journal of Communication Systems. Dr. Mao received the *2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems* and the *Best Paper Runner-up Award* from The Fifth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine) 2008. He is the co-author of a textbook *TCP/IP Essentials: A Lab-Based Approach* (Cambridge Univ. Press, 2004).