Multi-source Coalition Transmissions with Fault Tolerant Routing for Data Centric WSNs

Lei Wang[1] Wei Lin[2] Jyh-Ming Huang[2]

[1] Department of Electrical Engineering, Feng Chia University
Taichung 407, Taiwan, ROC
leiwang@fcu.edu.tw

[2] Department of Information Engineering and Computer Science, Feng Chia University
Taichung 407, Taiwan, ROC
xberamode@gmail.com; jmhuang@fcu.edu.tw

Received 17 August 2013; Revised 18 September 2013 ; Accepted 21 September 2013

Abstract. For prolonging network lifetime, several energy-efficient routing schemes were proposed in wireless sensor networks (WSNs). Directed Diffusion (DD), a data-centric routing paradigm, is one of these schemes. In this protocol, every node has ability with sensing, collecting, and storing event data. Also, the best transmission path, called reinforcement path, will be determined by using exploratory packet information in DD. Transmitting data via reinforcement path can avoid blind transmission, and ensure that event data can quickly reach the base station. However, most of DD-like protocols are concentrated on finding an optimal routing path for single-source data transmission environment, and the paradigm is difficult to recover its faulty path when some fault occurs. In this paper, an integrated routing protocol is proposed, termed as Source Coalition Directed Diffusion (SCDD), to study multi-source data transmissions in WSNs. SCDD can not only efficiently merge two nearby data flows to reduce network traffics and the number of packet collisions, but also effectively create a spare path for fault tolerant routing. The simulation results show that SCDD can achieve about 2%~14% by comparing with original DD scheme. The fault tolerant routing method of DD, REEP, can also be improved by cooperating with SCDD to achieve 4%~13% improvements.

Keywords: wireless sensor network, directed diffusion, routing protocol, energy-efficient routing protocol, data-centric routing, fault-tolerant routing

1 Introduction

In wireless sensor networks (WSNs), sensor nodes are usually equipped with non-rechargeable batteries [1]. The network will lose its functionality as any node exhausts its energy. The truth makes the research about the techniques to minimize the total energy dissipation so as to extend the network lifetime to be a main concern for WSNs [2]. Most of the related researches focus on providing an energy-efficient routing protocol for network [3].

Directed Diffusion (abbreviated as DD) [4] is a data-centric, interest-based routing paradigm that usually be implemented for habitat monitoring applications [5]. It is a primary observer-initiated approach where the user retrieves information from the network by sending a query called an *interest* through a remote base station (BS). Since the application-aware feature of DD, sensor nodes store and interpret interests, rather than simply forwarding them along. Each sensor node that receives an interest maintains a table that contains which neighbor(s) sent that interest. To such a neighbor, it sets up a gradient. A gradient is used to evaluate the eligibility of a neighbor node as a next hop node for data dissemination. After setting up a gradient, the sensor node redistributes the interest by broadcasting. As interests travel across the network, sensors that match interests are triggered and the application activates its local sensors to begin collecting and sending data. Thus, the query propagates across the network, reaches nodes sensing the requested phenomenon, and responses return in the reverse direction over the same path to the observer.

Basically, the DD paradigm is easy to implement, since it only considers an event source to transmit its data on a single path, and hence need not to maintain other routes. However, the drawbacks arisen owing to its simplicity. For example, the heavy flooding overhead involved in interest propagation causes scalability issues when DD is applied for large scale, interactive and dynamically changing sensor networks [6]. Besides, DD lacks an efficient mechanism to quickly or locally recover its faulty path, while a node on the transmission route uses up its energy. Instead, it always initiates the BS to reconstruct an alternate new path. This would not only incur longer transmission latency, but also significantly waste network energy [7][8]. Moreover, this reinforcement doesn't ensure the total removal of faulty routes. Some nodes keep sending data through the defective path since they haven't received any failure detection or path recovery notification. This induces great amount of lost information and wasted energy.

In many DD applications (e.g. habitat monitoring), the desired events might simultaneously occur at different locations. With this situation, a large number of sensory traffics coming from different source nodes will be flooded all over the network to backward the BS, and thus increase packet collision ratio, in turns, prolong transmission latency and consume more energies. Furthermore, in the environments where the source nodes are close to one another and generate a lot of sensory data traffic with redundancy, transmitting all sensory data by separate nodes not only wastes wireless bandwidth, but also consumes a lot of battery energies.

In this paper, we propose an integrated routing scheme, termed as Source Coalition Directed Diffusion (SCDD), to resist such shortcomings, and thus promote network energy efficiency. SCDD can not only coalesce the data flows coming from two nearby source nodes to reduce network traffics, but also create a spare path for fault tolerant routing at the same time. Furthermore, SCDD can cooperate with other scheme, such as REEP [9], to locally recover its faulty path, and then make the network more efficient. Extensive simulation results produced by the study show that the proposed SCDD scheme can not only effectively reduce network traffics by integrating transmission flows, but also achieve 2%~14% and 4%~13% improvements on network energy efficiency, by comparing with the existing DD and REEP protocols, respectively.

The rest of this paper is organized as follows: some previous DD-based studies that are relevant to the focus of this study are first reviewed in section 2. In Section 3, the details of proposed SCDD scheme are described. For evaluating the efficiency of the SCDD, lots of simulations are made and the final simulation results are present in Section 4. Finally, we conclude this work in section 5.

2 Related Works

2.1 Two-Phase Pull Directed Diffusion [10]

This protocol is generally referred to as the traditional Directed Diffusion algorithm. Two-Phase Pull DD always consists of operational phases as stated below.

Phase-1: Interest Flooding

The main purpose of this phase aims at letting the end users (i.e. BS) send messages to tell all sensor nodes in the network "what event types they are interested in?", so that the sensor nodes can return their sensory data while the desired events occur in the near future. As a result, after the sensor nodes are deployed, the BS begins to broadcast an interest packet to all its neighboring nodes. The nodes receiving this packet will identify the packet's transmitter as its downstream node, and then repeat this process until all nodes in the network have received such packet. Certainly, the node receiving such packet with the same event content as before will not broadcast the packet again. The process is illustrated in Fig. 1(a).

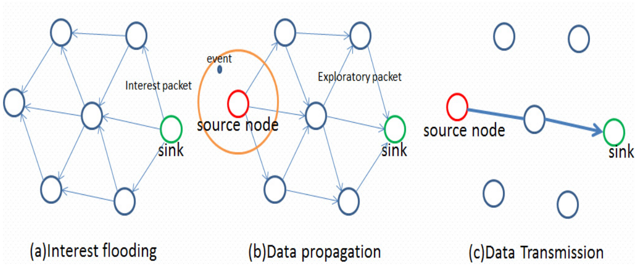


Fig. 1. The three phases in DD

Phase-2: Exploratory Data Propagation

The attempt in the second phase of Two-Phase Pull DD tries to find a designated path for later sensory data transmissions. As a desired event occurs, the node detecting the event will act as a source node, and immediately floods an exploratory packet to all its downstream nodes, along the paths built in the previous phase, in a reverse direction manner. Similarly, this process will also be repeated by each node until the BS received such packet as shown in Fig. 1(b).

By this process, each node in the network can log the ID of upstream node, which has the best gradient with transmission channel, to its neighbors table as a candidate node. The gradient in protocol is just a paradigm, it can be considered independently or together with the metrics of energy cost, transmission rate, or link quality [11][12][13]. After this step, the BS reversely sends a notification packet along the best transmission link to its candidate node, and the candidate node progressively repeats this process until this notification reaching the original source node. Eventually, an efficient transmission path, called as the reinforcement path, which is depicted in Fig. 1(c) can be formed.

When the reinforcement paths are determined, the source node can efficiently deliver its sensing data to the BS, along the reinforcement path established in the second phase. Periodically, the source sends additional exploratory data messages to adjust gradients in the case of network changes (due to node failure, energy depletion, or mobility), temporary network partitions, or to recover from lost exploratory messages.

According to the abovementioned phases, we conclude that the traditional DD protocol will work with following drawbacks. Firstly, since every node just locally records its last node's information, instead of globally considering the whole network transmission cost, to build up the reinforcement path. The traditional DD actually doesn't work well on energy conservation. Secondly, the reinforcement path might be blocked if an external force is imposed or some node's energy has drained out. In this case, the DD will re-send another notification packet again from the BS to create an alternate reinforcement path. Such behavior will consume lots of energies. Lastly, as many source nodes concurrently deliver their data packets to the BS, there are several data traffics coexist in the network, this might raise packet collision ratio. The retransmissions of data packets significantly consume a lot of energies, too. For reducing the wasting of energy produced by the method, we propose an improved DD scheme to resolve these shortcomings in this paper. The up-to-date information are used to decide a more efficient transmission path. At the same moment, several paths are integrated by the proposed scheme to reduce network flows and packet collision ratio, and thus promote the network energy usage. The detail algorithm of our proposed scheme is described in Section 3.

2.2 One-Phase Pull DD [14]

The basic routing principle in this scheme is similar to that in the Two-Phase Pull DD, except the exploratory data propagation phase is ignored. In One-Phase Pull DD, as a node received interest packets, it only records the information with the node which sends the interest packet first as its downstream node. The source node and all downstream nodes eventually formulate the reinforcement path for sensory data transmissions. In contrast to the Two-Phase Pull DD, One-Phase Pull DD although can run faster and save energy efficiently. The weakness of this method appeared in the consideration of gradient that is less flexible, the newest status about transmission paths cannot be attained to adjust the routing opportunely.

2.3 Mobile Agent-based Directed Diffusion (MADD) Protocol [15]

Recently, mobile agents (MA) have been proposed for efficient data dissemination in sensor networks. By considering mobile agents in multihop environments and adopts directed diffusion to dispatch MA, the gradient in DD gives a hint to efficiently forward the MA among target sensors. MA accounts for performing data processing and making data aggregation decisions at nodes rather than bring data back to a central processor (sink).

In DD, different data packets which are completely partially redundant each other are forwarded to the sink through multiple paths with a low probability to be aggregated. This aggregation technique can be considered as opportunistic aggregation. On the contrary, the MA aggregates individual sensed data when it visits each target source, then built the gradient for routing as DD does, and does not need more control overhead than DD: Once receiving a new task as requested by an application, the sink initially floods an interest packet to find out the sources which will perform the task. If the sources in the target region receive the interest packets, they flood exploratory data to the sink individually. Then, the sink will receive these exploratory data packets from various sources and decide the list of sources that will be visited by an MA. The MA-related operation begins at the point of the sink dispatching MA and ends when the MA returns to the sink with collected results.

In most cases, the MADD's performance in terms of energy consumption is better than that of DD. Thus, for the scenarios where energy consumption is of primary concern, MADD exhibits substantially longer network lifetime than DD. However, since MADD collect all sensed data via a single routing path, the scenarios restrain the transmission capability of a sensor network. The constraint lead the end-to-end delay of MADD is worse than that of DD.

2.4 REEP Routing Protocol [9]

REEP (Energy-Efficient and Reliable Routing Protocol) was proposed for improving the weakness of fault tolerant routing on traditional DD scheme [16]. In traditional DD, while a node on the reinforcement path exhausts its energy, it is difficult to quickly repair the broken path by only replacing the faulty node with its nearby nodes. Instead, it usually rebuilds a new reinforcement path from the BS. This reaction might waste lots of times and energies. REEP offsets this defect.

In REEP, every node records the sender's ID of exploratory packets in its RPQ (Request Priority Queue), according to the packet arriving sequence. While a node (for instance, node B in Fig. 2) on the transmission path is out of function, its downstream node (node C in Fig. 2) will try to find a replacing node, by visiting the nodes stored in RPQ in sequence, to replace it. This process is repeated in a reverse direction manner until a connection to the source node is found. If all the upstream nodes of the specific downstream node (i.e. node C) can't be used to create a backup path, the specific downstream node resumes the same process again from its successive node (i.e. node F in Fig. 2).

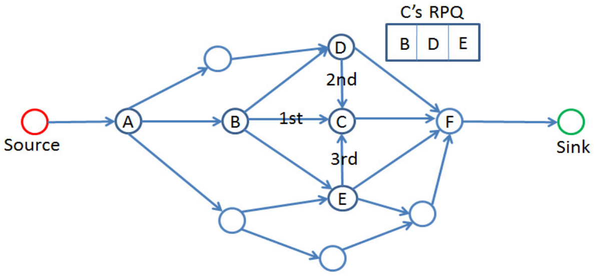


Fig. 2. RPQ in REEP scheme

REEP although can locally recover its transmission path around the faulty node, the selected backup path might be costly in energy consumption. In addition, the working times required to finding a backup path may be lengthen, and thus seriously increase the data transmission latency, if many path trials must be conducted. Consequently, in this paper, we also attempt to mitigate these defects by preparing a standby path for fault tolerant routing in advance.

2.5 Fault Tolerant Directed Diffusion (FaT2D) Protocol [17]

FaT2D is another fault tolerant protocol based on Directed Diffusion. It defines a technique which implements fast failure detection with a prompt path recovery regarding to nodes crash and topology changes. A failure detection timeout (denoted by T_{fd}) is defined in order to reduce the failure recovery time and consequently speed up the node failure detection and local path repair, while tolerating intermittent failures due to packet loss. If T_{fd} runs out, FaT2D immediately forwards a new message called *ExploreRequest* to notify the failure detection event and demand a new exploration for a reliable route replacing the faulty one. Therefore, each node in the faulty path deletes the corresponding gradient for failure elimination after receiving the *ExploreRequest* packet.

The *ExploreRequest* packet will be forwarded in order to reach the top source of data information relevant to the faulty route without invoking transmission loops or searching inappropriate nodes. When the source receives the exploration request it stops forwarding the *ExploreRequest* packet, then starts an exploration flooding as for the original DD. This will generate an early exploration round in order to find a new reliable path. The election of this path uses the same reinforcement rules as in DD. For each intermediate node receiving the *ExploreRequest* packet, FaT2D checks whether this node belongs to the corresponding faulty path. If so, it will negatively reinforce its gradient. This latter will be reelected by the next exploration sent by the top source of the corresponding route. Thus, every node runs a local negative reinforcement to its upstream neighbor in order to delete the broken path and stop sending wasted data.

According to the simulation results reported in [17], FaT2D shows significant enhancement for both recovery time and loss rate metrics than traditional DD. This allows a greater network lifetime, offers faster application stability, and assures a better data delivery. The most important influencing factor for the efficiency of FaT2D is the determination of T_{fd}. An inappropriate T_{fd} will extend the time for path recovery or create wrong estimates about failure detection.

3 SCDD

The basic operational procedures in our proposed SCDD scheme are similar to that in traditional DD. The main differences between them are summarized as follows: (1) The SCDD scheme is mainly used for multiple-source data-centric wireless sensor networks. (2) While flooding exploratory packets, the SCDD scheme tries to merge the data flows issued from distinct source nodes to reduce the packet collision ratio, and thus conserve the network energy depletion. (3) During the data flow merging step, a standby path is simultaneously built up for fault tolerant routing.

The basic running flow of each phase in SCDD scheme is described as follows:

3.1 Interest Packet Flooding Phase

The task conducted in this phase is the same as that in the first step of traditional DD. Interest packets are flooded from the BS to every node in the network to declare what kinds of event the end-users want. With this procedure, the flow directions of incoming exploratory data packets are determined.

3.2 Exploratory Data Packet Propagation Phase

This phase is the main core of SCDD scheme. The SCDD protocol adaptively merges two nearby exploratory data flows issued by different source nodes to find an optimal data transmission path in this phase. There are two cases may happen as described below:

Case 1: Packet goes through a normal node

While an event that be interested occurs, the nodes sensed the event will be identified as source nodes, and begin to flood respective exploratory packet to find their reinforcement paths. The exploratory packet format is list in Table 1. While a non-source node (i.e. normal node) received the exploratory packet, it takes down the packet information, and then broadcasts the packet with an updated transmission cost and distance again, if it is a downstream node of the packet sender. This process will be repeated until the exploratory packet reaching to the BS. Moreover, as a normal node received other exploratory packets with same data from other paths, it will replace its stored records with the lowest up-to-date cumulative transmission cost to select a most proper path for transmission.

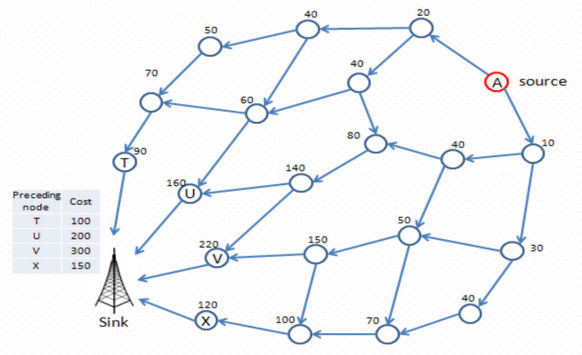
Table 1. The packet format of exploratory data

Packet ID	Source ID	Data type	Preceding node's ID	Cumulative transmission cost	Cumulative transmission distance

SCDD takes the metric of Eq. (1) as the cumulative transmission cost for the exploratory data packet. Where *Cost*(*j*) means the cumulative transmission cost calculated from source node to node *j*. $d(i, j)$ is the distance between node *i* and node *j*. *S* is the number of neighboring nodes around node *j*. E_{remain} and $E_{initial}$ represents the remaining energy and the initial energy of node *j*, respectively. δ is a weighting factor (it is set to be 1/3 in our simulations). The initial cumulative transmission cost of the source node is set to 0.

$$Cost(j) = Cost(i) + \frac{E_{initial}}{E_{remain}}\left(\delta \cdot d(i, j) + (1-\delta)\frac{1}{S}\right) \quad \textbf{(1)}$$

From Eq. (1), we know that more neighboring nodes closing to the transmission path involved, it is easier to locally recovering the faulty path, if a node on the transmission path is out of order. In addition, the nodes with fewer remaining energy would be precluded from the candidate transmission path.



Preceding node	Cost
T	100
U	200
V	300
X	150

Fig. 3. The cumulative transmission costs on different routes

Fig. 3 simply illustrates the scenario of an exploratory packet flooded by the source node A, and goes through different paths to the sink (i.e. BS). The value shown aside to each node represents the cumulative transmission cost summed up from the source node to the node. Eventually, the sink would receive many exploratory packets from different neighboring nodes (for instance, node T, U, V, and X in Fig. 3). The sink records the information with the format as exemplified in Table 2, and then determine the desired reinforcement path by means of the information. At last, the sink notifies the nodes locating on the reinforcement path for getting ready to transmit data packets.

Table 2. The information stored in the sink

Packet ID	**Source ID**	**Data type**	**Preceding node's ID**	**Cumulative transmission cost**	**Cumulative transmission distance**
171	A	car	T	100	80
171	A	car	U	200	90
171	A	car	V	300	120
171	A	car	X	150	115

Case 2: Packet goes past another source node

While an exploratory packet goes past another source node during transmission cycle, and the content of its data type field is consistent with that in the current source node, the ID of current source node, the cumulative transmission cost and distance between these two source nodes will be appended to the original exploratory packet (refers to Table 3), and sent out again. It should be noted that SCDD only takes down the information with the first source node the packet encountered. The reason is that the SCDD scheme only selects the flow of the nearest source node to perform path coalition.

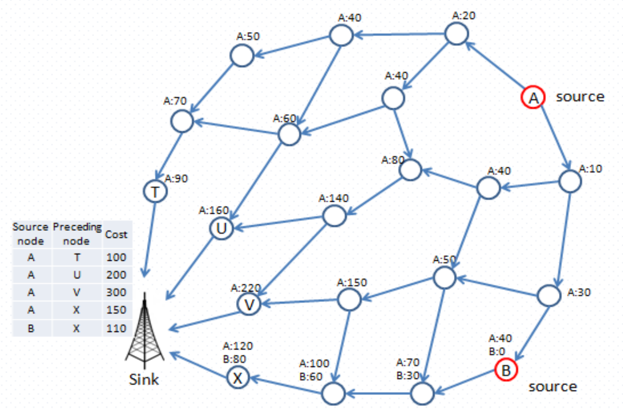
Table 3. The contents of exploratory packet after appended by another source node

Original packet contents	**The node ID of passed source**	**Cumulative transmission cost between two source nodes**	**Cumulative transmission distance between source nodes**
N/A	B	40	30

While exploratory packets arrived at the BS via different paths, the BS merges the contents to figure out the whole network transmission information as shown in Table 4. Hereafter, the SCDD begins to check whether the transmission paths of the two nearest source nodes can be coalesced together or not. Fig. 4 roughly displays the cumulative transmission costs recorded in each node and the sink.

Table 4. The source routing information stored in the BS

Packet's ID	**Source node's ID**	**Data type**	**Preceding node's ID**	**Cumulative transmission cost**	**Cumulative transmission distance**
135	A	car	T	100	80
246	B	car	X	110	75



Source node	Preceding node	Cost
A	T	100
A	U	200
A	V	300
A	X	150
B	X	110

Fig.4. Exploratory packet flows and cumulative cost

3.3 Strategy and Implementation of Path Coalition

Since one of two source nodes must forsake its original reinforcement path, and forward its data packet to the other after path coalition, the distance between the two source nodes therefore will consume extra energies in data transmission. This research establishes the path coalition based on the consideration: as the distance between the two source nodes is less than a certain rate of the minimum length of the paths from each source node to the sink, it should be helpful to energy conservation. Otherwise, path coalition will negatively induce more energy depletion.

SCDD checks whether two nearby transmission paths should be combined together or not by means of the evaluation rule list in Eq. (2). Where $d(i, j)$ denotes the distance between two source nodes i and j. α is a weighting factor. $d(i, sink)$ represents the cumulative transmission distance from source node i to the sink. If they are, it selects the best one (with the minimum cumulative transmission cost) as the candidate for later data transmission. From the simulation results, the study observed that α with value 1/2 performs better than other values.

$$d(i,j) < \alpha \times \min(d(i,sink), d(j,sink)) \tag{2}$$

When two source nodes decide to link up their paths, the sink will mark their node IDs to avoid making path coalition with other source nodes again. It means that the relevant information of the source node that has already made path coalition with other source will be ignored in later processes.

Fig. 5 demonstrates that the SCDD has decided to link up the path starting from source node B to the path starting from source node A, and the nodes (e.g. node C) on the route between sources A and B only stored the routing information with its upstream source node (e.g. source A).

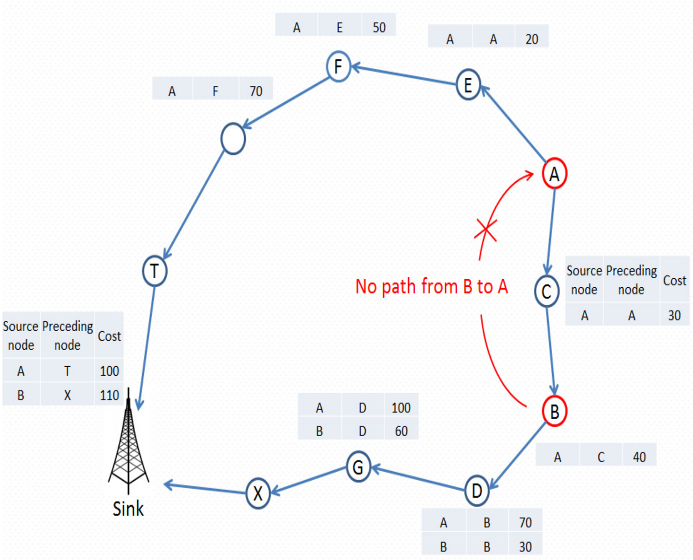


Fig. 5. Before integration of two source nodes

The sink must send a message in a reverse direction manner to notify the source node B to forward its sensory data to source A as shown in the example illustrated in Fig. 6. After the two-way transmission path between source nodes A and B was built up according to the cumulative transmission costs of these two paths, SCDD selects the better way as its transmission candidate.

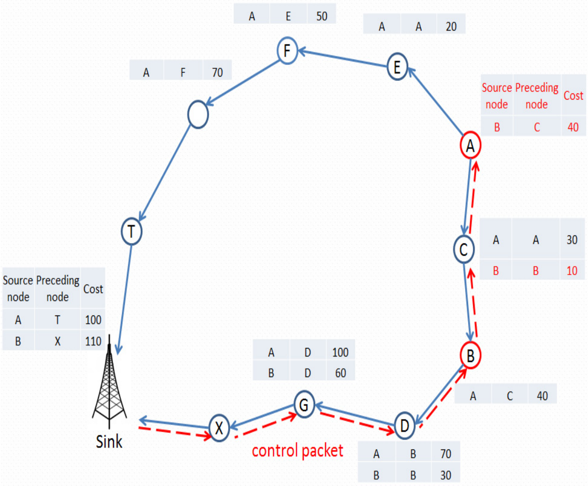


Fig. 6. The notification with path coalition Issued by the BS

Fig. 7 shows two transmission costs (100 and 110) cumulated from source nodes A and B to the sink before path integration, respectively. As a result, both source A and source B transmit their data packets through the same reinforcement path of B → C → A →…→sink. As shown in Fig. 8, the solid line depicted in this figure represents the final transmission path. The other alternative route, A → C → B →…→ sink that plotted by the dotted line becomes a spare path.

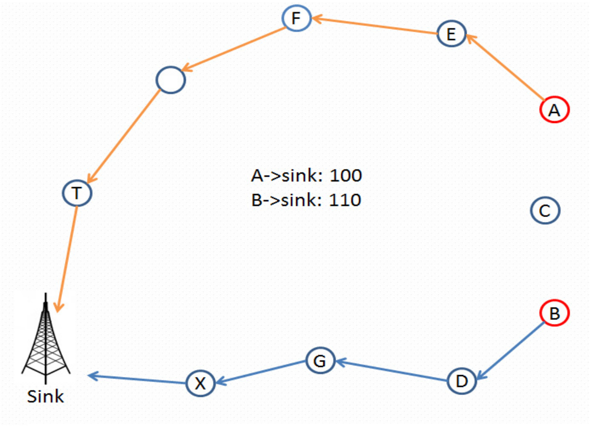


Fig. 7. Before path coalition

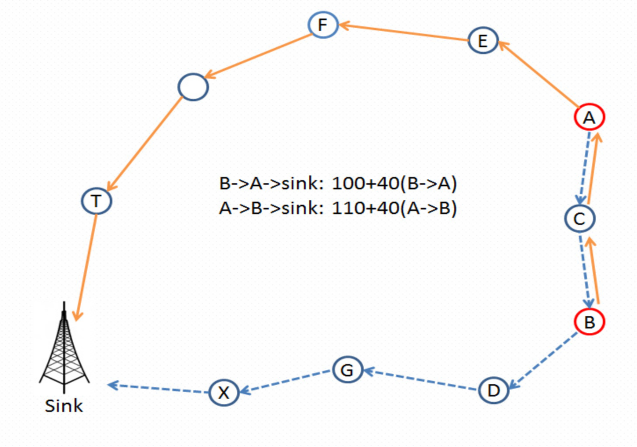


Fig. 8. After path coalition

After the coalesced path was created, the event data will be sent back to the sink via the designated routing path. Table 5 shows the format of data packet. The cost field in the packet is always cumulatively updated while data packet transmitted. Moreover, data aggregations are also performed to reduce the packet length and thus save node energy if two source nodes transmit their data packets at the same time. For simplifying the assumptions for the network design by ignoring other technique assumptions such as data compaction techniques to further reduce the cost of data aggregation. This research estimates the cost saved by combining two packets into one by counting the data bytes that can be saved because of duplicate fields. According to the data packet format list in Table 5, the data aggregation ratio is calculated to be 1.7. This ratio will be used to compute the energy consumption in our simulations.

Table 5. The packet format of event data

Packet ID	Source node's ID	Data type	Preceding node's ID	Cumulative transmission cost	Event payload

1) Transmission path switching

The energies of nodes on the transmission path will be gradually exhausted when data packets delivered, thus the BS always keeps up the up-to-date transmission cost while it received a data packet every time. When the up-to-date transmission cost is greater than that in the original spare path, the BS sends a negative message to notify the source node to change its transmission direction. At meanwhile, it also sends a reinforcement packet along the spare path to initiate the alternate transmission route. The BS can also switch to the spare path for continuing the data transmissions as any node on the reinforcement path is out of order. After the faulty path recovered, the BS will select the better path for delivering by comparing the transmission costs.

2) Path split

As one of these two source nodes on the coalesced path has not to transmit data packets anymore, the BS will be initiated to send a reinforcement packet to notify the other source using its original reinforcement path to transmit.

3) Partial path recovery

SCDD always select the better path of the two merged routes to transmit data even the transmission path is partially broken. The reason to do so is to avoid an unacceptable waiting time paid for path recovery. The proposed SCDD scheme can integrate with as REEP protocol to quickly recover its faulty path. Similarly, as recovery is done, the BS also compares the transmission costs of the original and the new backup paths, and then selects the better one to continue its data transmission.

4 Simulations and Performance Evaluation

To verify the practicability of our proposed SCDD scheme, we conduct many extensive simulations with different numbers of nodes (e.g. 100, 150, 200, and 250) randomly scattered over four different sensing areas (e.g. 100m*100m, 150m*150m, 200m*200m, and 250m*250m). The BS is assumed to locate on the bottom-left corner of sensing field. After the nodes deployed, we randomly generate 10 source nodes to periodically send 1000 data packets back to the BS. We evaluate the possibility of path coalition and the network energy efficiency, from the averaged results of 300 simulations experienced in different combined scenarios.

Since energy is the main comparing factor in our simulations, the study takes energy cost as the gradient in DD protocol. As a node on the transmission path exhausts its energy and thus causes path broken, DD will issue a new reinforcement packet from the BS to rebuild an alternate transmission path.

As for energy computation, the study adopts the following equations as our radio model. Eq. (3) and Eq. (4) represents the total energy consumed in receiving or transmitting a k-bit packet as defined in [18]. Where d is the distance away from two participating nodes, E_{elec} is the basic power consumption of the circuit, ε used for power amplifier. Table 6 lists the value for each parameter used in our simulations.

$$E_{Rx} = k \cdot E_{elec} \tag{3}$$

$$E_{Tx} = \begin{cases} k \cdot E_{elec} + k \cdot \varepsilon_{fs} \cdot d^2 \; for \;\; d \le d_0 \\ k \cdot E_{elec} + k \cdot \varepsilon_{mp} \cdot d^4 \; for \;\; d > d_0 \end{cases} \quad where\ d = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \tag{4}$$

Table 6. The value of energy parameter

Symbol	Parameter value
E_{elec}	50 *nJ/bit*
ε_{fs}	10 *pJ/bit/m*^{2}
ε_{mp}	0.0013 *pJ/bit/m*^{4}
node's initial energy	0.2 *Joule*

4.1 The Parameter of Path Coalition

In order to find out an optimal value α shown in (2) to perform path coalition, the study simulates the SCDD protocol with various values of α in different scenario to compare the energy efficiencies.

Figs. 9~13 are five results from our simulation scenarios that selected for exhibiting the characters of simulations. By those figures, we observe that: when α=1, the SCDD although can merge more flows as shown in Table 7, but it also spends a large amount of energy on the coalition path to transmit data. The reason is that a longer distance induced by those two coalesced paths, and thus results in heavy energy consumption. On the other hand, a smaller α will make path integration more difficult, but the condition is helpless to conserve energy. According to our simulation results, it seems good if α is set to be 1/2 or 1/3. From now on, we choose α = 1/2 as the value of path coalition parameter for further simulations.

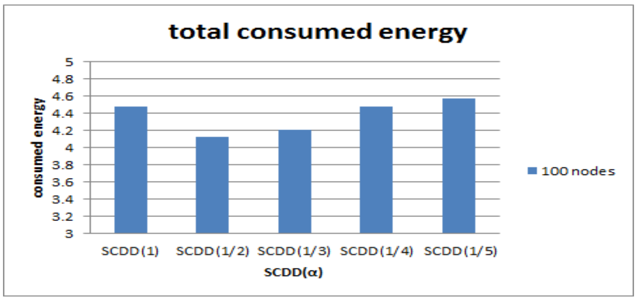


Fig. 9. Network energy consumption vs. different path coalition parameters (with 100 nodes, 100*100 network size)

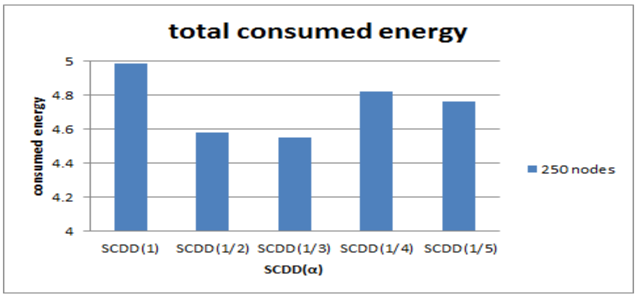


Fig. 10. Network energy consumption vs. different path coalition parameters (with 250 nodes, 100*100 network size)

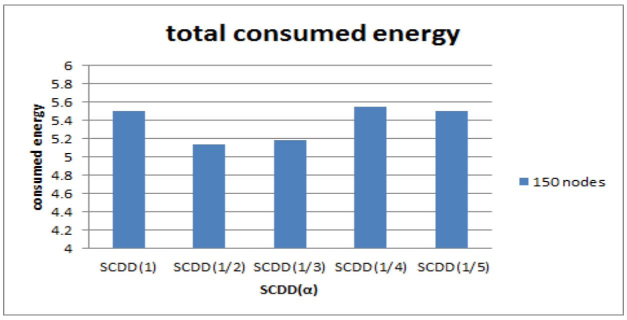


Fig. 11. Network energy consumption vs. different path coalition parameters (with 150 nodes, 150*150 network size)

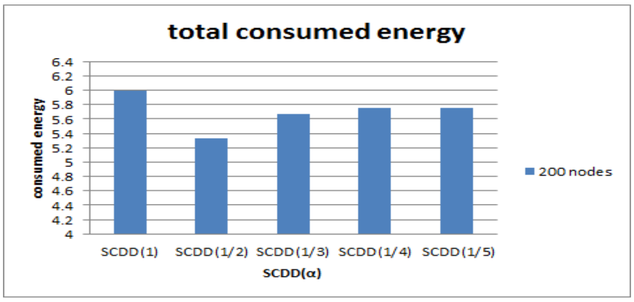


Fig. 12. Network energy consumption vs. different path coalition parameters (with 200 nodes, 200*200 network size)

Table 7. The average number of flows after path coalition

DD	SCDD(α=1)	SCDD(α=1/2)	SCDD(α=1/3)	SCDD(α=1/4)	SCDD(α=1/5)
10	7.451	8.173	8.841	9.145	9.194

4.2 The Calculation on Transmission Cost

As we have seen in (1), the cumulative transmission cost with a specific node is mainly dependent both on the transmission distance from its parent node to itself, and the number of neighboring nodes. Since the transmission distance will directly affect the energy consumed in data propagation, and the number of neighboring nodes implies how much the extra energies will be spent in recovering the faulty path, we therefore expect a bigger value of δ to be used in normal transmission state, and a smaller one is more favorable as faulty condition exists.

In order to realize the effect of δ on the network energy efficiency for normal transmission and path recovery states, and thus conclude a compromised value for δ, the study simulates the network energy consumption with 100 and 200 nodes under four different network sizes. In all simulation scenarios, we assume node's energy is unlimited, and periodically make some nodes that are located on the transmission path to die. The faulty path will be recovered immediately as faulty nodes appear.

Fig. 14 depicts the average simulation result for different δ values subject to various faulty rates of node. From the figure we can observe that as δ=1, i.e. only consider the distance between communicating nodes, the network costs fewer energy consumption with a lower faulty rate. On the contrary, energy depletion will be quickly lifted as the number of faulty nodes increase. As a result, for the condition that when nodes die frequently, it seems unwise to decide the path by only considering the distance factor in transmission cost computation. However, although the faulty path can be quickly recovered when a small value is assigned to δ, the network might also consume more energy because of long-haul transmissions.

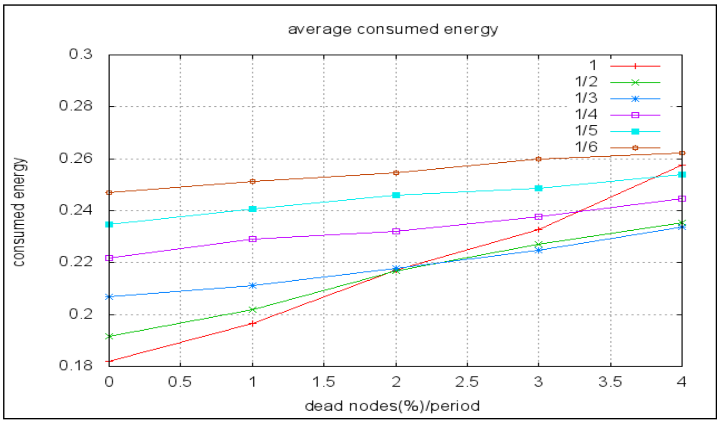


Fig. 14. The effect of various δ settings on the network energy under different node's faulty rates

By the observation from Fig. 14, we find that even the value of δ is less than 1/4, the SCDD scheme can still not mitigate the network energy consumption further. We also find that the SCDD can work well in different harsh environments, under different faulty rates and δ values, e.g. 1, 1/2, 1/3, and 1/4, respectively. We hence substitute these values into (1) to investigate the performance of SCDD scheme performed in normal transmission situation. Figs. 15~18 illustrate the total network energy consumption in different network sizes. We therefore conclude that the SCDD scheme can achieve a good compromise on transmission cost computation when δ = 1/3.

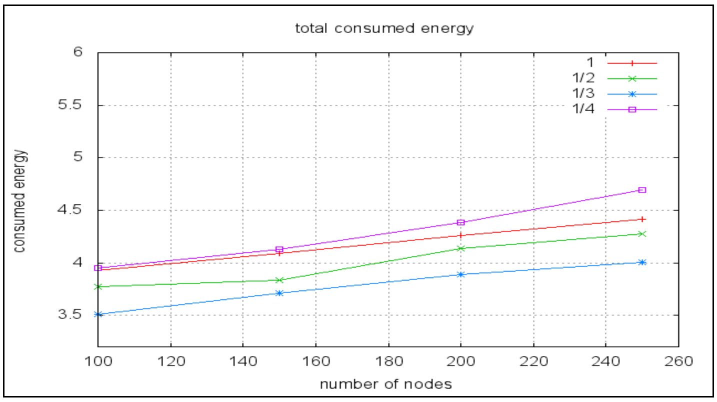


Fig. 15. The effect of various δ settings on the network energy (100*100 network size)

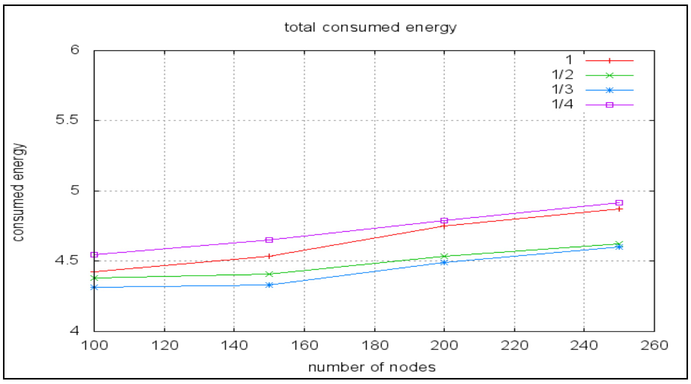


Fig. 16. The effect of various δ settings on the network energy (150*150 network size)

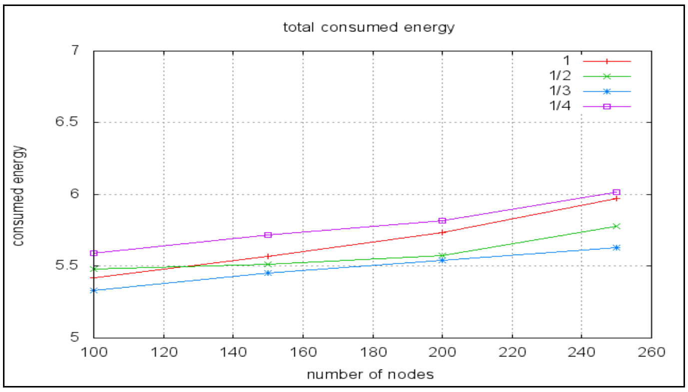


Fig. 17. The effect of various δ settings on the network energy (200*200 network size)

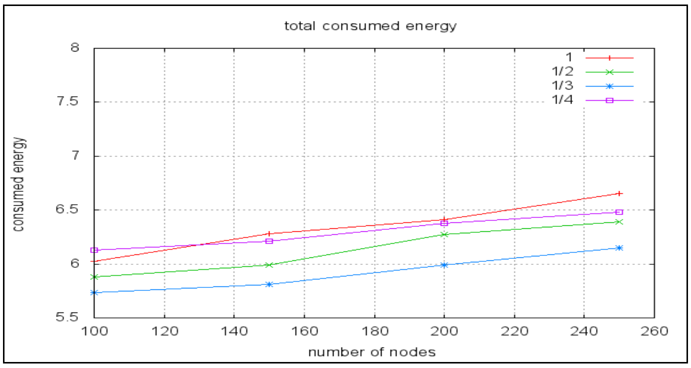


Fig. 18. The effect of various δ settings on the network energy (250*250 network size)

4.3 Comparisons of Total Energy Consumption

We simulate and compare the network energy efficiency with four different routing protocols such as DD, SCDD, REEP, and SCDD+REEP. We assume the REEP protocol begins to recover its transmission path, as the ratio of node's residual energy to the original energy is less than 1/3.

From the simulation results, we concluded that the SCDD scheme is superior to the DD protocol because of the effectiveness of path coalition. By examining the simulation results of DD and SCDD schemes, we found that the SCDD can achieve 2% ~ 14% reductions for total network energy consumption by comparing with the DD scheme. As for the comparison between SCDD and REEP protocols, since SCDD uses the cumulative transmission cost to decide its candidate transmission path, it can only switch to the spare path or rebuild a new alternate back up path from the BS when the original transmission route is blocked. On the other hand, the REEP protocol can partially repair its faulty path. As a consequence, the SCDD scheme can be costly in energy consumption by comparing with the simulations for REEP.

It is noted that although the REEP scheme behaves good in path recovery, it is unable to deal with the path coalition for multi-source data flows. SCDD+REEP integrate both schemes to benefit partial path recovery by REEP scheme and multi-path coalition in SCDD protocol. The simulation results show that SCDD+REEP can significantly reduce the traffic flows and data packet length, and thus mitigate the network energy cost. Based on the simulation results, 4% ~13% improvements on energy conservations are observed for the SCDD+REEP scheme over the original REEP method.

4.4 Comparison of Path Numbers with Different Number of Source Nodes

Preceding simulation results demonstrate our proposed SCDD scheme can effectively reduce energy consumption when 10 source nodes simultaneously transmit data packets in network. For the sake of studying SCDD is efficient with different source nodes, we also simulate other situation such as 2 sources and 6 sources in addition. The course of action is same as the simulation with 10 source nodes. Each source nodes' occurrence will delay a period of time, and sources' position are random. The values of parameters of equations (1) and (2) are set the same as with the simulation for 10 sources.

When only two source nodes transmit data packet, their coalition rate is 16%, we regard this value as the average ratio of each two sources that they have capabilities to do path coalition. With the increase of source nodes' amount, we find the ratio of complete path coalition will decrease. Because source nodes don't appear in the same time, and in SCDD's algorithm, if sink find two source nodes can be merged, it will immediately combine these two source nodes. The method makes source coalition in real time; but cause low coalition ratio.Table 8 lists the reduced path ratio by 2/6/10 sources, we find source coalition ratio is low when number of source nodes is few from the table. This ratio will sharply grow with the increase of source amount. The truth means that more source nodes can effectively provide SCDD more chances to achieve higher coalition ratio, then reduce more energy consumption.

Table 8. The average number of flows after path coalition

	Average Path	Reduced Path	Reduced Path Ratio
2 sources	1.84	0.16	8%
6 sources	5.04	0.96	16%
10 sources	8.173	1.827	18.27%

5 Conclusion

In this paper, a Multi-Source Coalition Directed Diffusion protocol named as SCDD for data-centric wireless sensor networks is proposed. In SCDD, two nearby transmission paths are coalesced to form an economic one for data transmissions. In addition, data aggregation is always conducted to reduce the energy consumption. Meanwhile, a spare path is also created in advance for real-time replacement as soon as the original path is failed. The SCDD can also further integrate some partial path recovery mechanisms to enhance its functionality.

To validate the proposed SCDD protocol, several scenarios with various conditions are simulated to evaluate the performance of SCDD. By comparing the simulated results with other famous DD-related protocols, the paper shows that the proposed SCDD protocol is superior to some specific schemes, in terms of network energy efficiency, especially for fault tolerant routing.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, Vol. 38, No. 4, pp. 393-422, 2002.

[2] S. J. Choi, K. H. Kwon, S. J. Yoo, "An Efficient Cross-Layer Based Flooding Algorithm with Retransmission Node Selection for Wireless Sensor Networks," in *Proceedings of International Conference on Advanced Information Networking and Applications*, Okinawa, pp. 941-948, 2008.

[3] J. N. Al-Karaki and A.E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, Vol. 11, No. 6, pp. 6-28, 2004.

[4] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of International Conference on Mobile Computing and Networking*, Boston, Massachusetts, pp.56-67, 2000.

[5] R. Szewczyk, A. Mainwaring, J. Polastre, J. A. D. Culler, "An Analysis of a Large Scale Habitat Monitoring application," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, pp. 214-216, 2004.

[6] J. Tang, S. Dai, J. Li, S. Li, "Gossip-based Scalable Directed Diffusion for Wireless Sensor Networks," *Int. J. Commun. Syst.*, Vol. 22, No. 11, pp. 1418-1430, 2011.

[7] J. Shi, K. Cai, C. H. He, G. Wei, Z. Shan, "An Energy-Adaptive Multiple Paths Routing Approach For Wireless Sensor Networks," *Journal of Mobile Multimedia*, Vol.8, No. 1, pp. 34-48, 2012.

[8] Y. M. Lu and W. S. Wong, "An Energy-efficient Multipath Routing Protocol for Wireless Sensor Networks," *Int. J. Commun. Syst.*, Vol. 20, No. 7, pp. 747-766, 2007.

[9] F. Zabin, S. Misra, I. Woungang, H. F. Rashvand, N.-W. Ma, M. Ahsan Ali, "REEP: Data-Centric, Energy-Efficient and Reliable Routing Protocol for Wireless Sensor Networks," *IET Communications*, Vol. 2, pp. 995-1008, 2008.

[10] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, "Directed Diffusion for Wireless Sensor Networking," IEEE/ACM Transactions on Networking, Vol. 11, No. 1, pp 2-16, 2003.

[11] Y. S. Chen, Y.W. Nian, J. P. Sheu, "An Energy-Efficient Diagonal-Based Directed Diffusion for Wireless Sensor Networks," in *Proceedings of Ninth International Conference on Parallel and Distributed Systems*, Taiwan, pp. 445-450, 2002.

[12] J. Liu, Y. Li, Q. Chen, Y. Kuang, L. Hu, K. Long, "Energy and Storage Efficient Directed-Diffusion for Wireless Sensor Networks," in *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, pp. 2460-2463, 2007.

[13] Z. Shousheng, Y. Fengqi, Z. Baohua, "An Energy Efficient Directed Diffusion Routing Protocol," in *Proceedings of Int. Conference on Computational Intelligence and Security*, Harbin, pp. 1067-1072, 2007.

[14] F. Silva, J. Heidemann, R. Govindan, D. Estrin, *Directed Diffusion*, Technical Report ISI-TR-2004-586, USC/Information Sciences Institute, 2004.

[15] M. Chen, T. Kwon, Y. Yuan, Y. Choi, and V.C.M. Leung, "Mobile Agent-Based Directed Diffusion in Wireless Sensor Networks," *EURASIP Journal on Advances in Signal Processing*, Vol. 2007, No. 1, pp. 1-14, Jan. 2007.

[16] J. Liu and Z. Ping, "Fault Tolerant and Storage Efficient Directed-Diffusion for Wireless Sensor Networks," in *Proceedings of IEEE International Conference on Information Theory and Information Security (ICITIS)*, Beijing, pp. 884 -887, 2010.

[17] F.Z. Benhamida and Y. Challal, "FaT2D: Fault Tolerant Directed Diffusion for Wireless Sensor Networks," in *Proceedings of IEEE International Conference on International Conference on Availability, Reliability, and Security*, Krakow, pp. 112-118, 2010.

[18] W. B. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, pp. 660-670, Oct. 2002.