A PRIORITY-BASED MULTI-PATH ROUTING PROTOCOL FOR SENSOR NETWORKS

LIU, YUZHE

NATIONAL UNIVERSITY OF SINGAPORE

2003

A PRIORITY-BASED MULTI-PATH ROUTING PROTOCOL FOR SENSOR NETWORKS

LIU, YUZHE

(*B.Eng., NWPU, P.R.China*)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

Acknowledgements

I would like to express my deepest appreciation to my supervisor Dr. Seah, Khoon Guan Winston for his inspiration, excellent guidance, support and encouragement. His erudite knowledge, the deepest insights have been the most inspiration and made this research work a rewarding experience. I owe an immense debt of gratitude to him for having given me the curiosity about the sensor network technology and the most invaluable guidance and support about this research work. His rigorous scientific approach and endless enthusiasm have influenced me significantly. Without his kindest help, this thesis and many other works would have been impossible.

Thanks also go to the faculties in the Institute for Infocomm Research (I2R) and the Electrical & Computer Engineering Department, the National University of Singapore (NUS), for their constant encouragement and valuable advice.

I sincerely acknowledge the help from all members in the *New Student Cluster*, I2R, for their kind assistance and friendship which have made my life in Singapore easy and colorful.

Acknowledgement is extended to I2R and NUS for awarding me the research scholarship and providing me the research facilities and challenging environment during my study in Singapore.

Last but not least, I would thank all my family members, especially my sister and my parents, for their constant support, understanding, and patience in my pursuit of a M.Eng. This thesis, thereupon, is dedicated to them for their infinite love.

Contents

List of Figures

List of Tables

Summary

In this thesis, a priority-based multi-path routing protocol (*PRIMP*) is proposed for sensor networks to provide extended network lifetime and reliable transmissions, under the contexts of stringent energy constraint and dynamic environmental conditions. To address the primary issue in sensor networks — stringent energy constraint on sensors, a novel *on-demand virtual source* technique is adopted reactively by *PRIMP*. This technique aims to explore source region or re-establish the data paths from sources to sinks, whenever it is necessary. It facilitates the subsequent directional maintenance of the data paths from sources to sinks, and minimizes the transmission overhead from interest dissemination. Thus, significant energy conservation and extended network lifetime are achieved. Due to the vulnerability of sensors to the physical environment, poor network fault tolerance proves to be another key issue in sensor networks. To address this issue, *PRIMP* periodically maintains multiple *braided* data paths from sources to sinks through directional interest dissemination toward sources. Thus the candidate data paths from sources to sinks are constantly kept alive and refreshed. Data events will then be probabilistically and simultaneously routed over multiple candidate paths, in a priority-based approach depending on the energy resource conditions of all candidate paths. This load-balanced routing strategy renders a reliable data delivery performance to *PRIMP*. Moreover, compelled by time-sensitive applications, *PRIMP* addresses the *slow startup* problem left unexplored in existing routing protocols for sensor networks so that different sinks initiating identical interests will be able to retrieve corresponding data events without being “discriminated” in application startup phase. Finally, the performances of both *PRIMP* and its comparable routing protocols are evaluated through extensive simulations and analysis, and the advantages of *PRIMP* in energy conservation and the provisioning of reliable transmissions are validated.

Chapter 1

Introduction

Wireless sensor networks is currently an active research topic in the fields of information gathering and processing. With the recent technical advances in distributed micro-sensing [1], *in-network* information processing [2, 3, 4, 5], and wireless communication, a wide range of applications have been made viable based on the collaboration of a large number of networked sensors deployed in the target area.

1.1 Introduction to Sensor Networks

Sensor networks are composed of a collection of untethered and unattended sensors or actuators within a target area. Sensors are usually small in size, of low cost, and battery-powered. Each sensor is also chip-embedded and has sensing, data processing and computation capabilities. The recent technical advances in micro-electro-mechanical systems (MEMS) [6, 7], wireless communications and digital electronics have enabled the development of such multi-functional sensors. Sensor network applications are fulfilled through the collaboration of these self-organized sensors through multi-hop wireless communications.

Figure 1.1 gives the logical communication architecture of sensor networks. Sensors are densely deployed in the target area to retrieve desired *data* information from within the

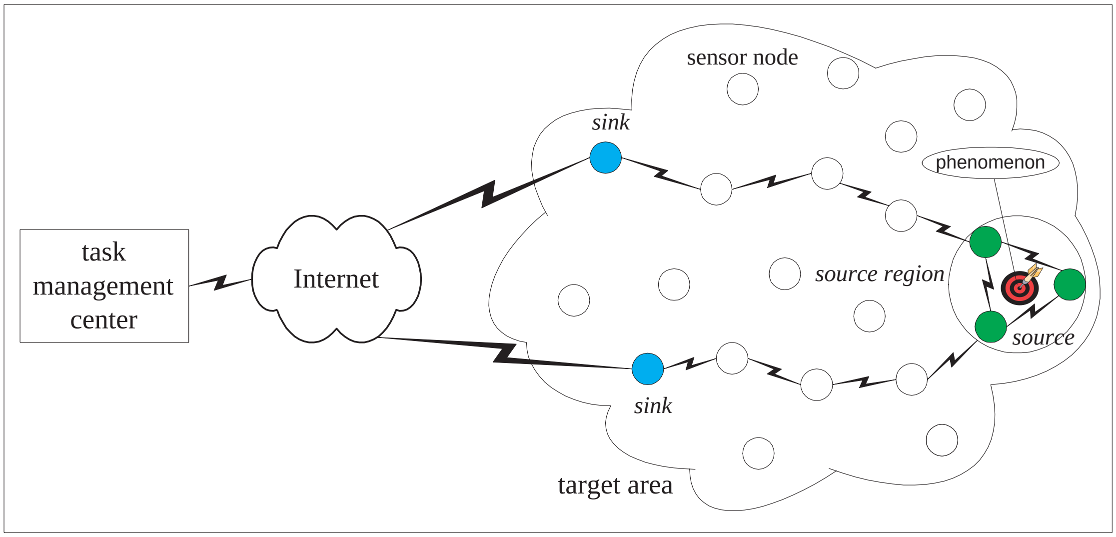


Figure 1.1: Communication architecture of sensor networks

target area. Firstly, tasking information — enquiry describing the task to be fulfilled, will be dispatched into the network from some sensors called *sinks*. This tasking information will finally arrive at some *sources* which are sensors in the *source region* that is inside or close to the phenomenon. Since these sources can provide the data information desired by sinks, they will process the sensed data events, generate corresponding data messages, and send these messages back to the task management center via sinks through an infrastructureless multi-hop networking architecture.

The deployment of sensors can be quite flexible, and is usually conducted in an ad hoc fashion. Sensors can either be placed in a target area manually, or randomly scattered by planes, robots or mini-rockets. The deployed sensors can be of heterogeneous types. Sensors of several basic types are illustrated as follows:

- light sensor;
- temperature sensor;
- humidity sensor;
- heat sensor;

- acoustic sensor;
- seismic sensor.

These various kinds of sensors can be applied in a wide range of application domains used in a variety of conditions. The sensor network applications can be generalized into the following three categories:

- Home and office use: such as smart environment;
- Business use: such as conferencing, inventory;
- Clinic and military use: such as target surveillance and monitoring.

1.2 Research Challenges

Research challenges mainly arise from the constraints on hardware designs, as well as the unique characteristics of sensor networks, and they serve as guidelines for the design of sensor network protocols.

1.2.1 Unique Features of Sensor Networks

As shown in Figure 1.1, wireless ad hoc networking techniques are needed for the multi-hop communications in sensor networks. Therefore, routing protocols should be carefully designed to support a robust infrastructureless networking architecture. Routing protocols for mobile ad hoc networks (MANET) appear to fit this need. However, routing protocols for MANET are unsuitable for adoption in sensor networks due to the unique features of sensor networks, which are listed as follows:

- Compared to MANET, sensor networks are much more densely deployed. The number of sensors in the network can be several orders of magnitude higher than that of nodes in MANET. Depending on the application, node density within sensor networks can range from a few sensors to a few hundred sensors in a region less than 10 m in diameter [8].

- The transmission range of a sensor is typically much smaller than that of a MANET node, and is usually limited to within tens of meters;
- Sensors have only limited computation capability, memory storage, and battery power, while nodes in MANET are assumed to be more resource-abundant;
- An address-centric wireless communication paradigm is adopted in MANET, while in sensor networks, communications are data-centric;
- Sensor networks and MANET employ different addressing techniques. Address structure adopted in sensor networks is usually application-dependent, in contrast to the application-independent addressing in MANET;
- Sensors are much more vulnerable to the dynamic environmental conditions than nodes in MANET. Therefore, transmission reliability (fault tolerance issue) in sensor networks are much more critical than that in MANET.

Taking these unique characteristics of sensor networks into account, corresponding factors or research issues should be carefully explored in order to design novel and robust routing protocols to meet the special requirements in sensor networks.

1.2.2 Key Research Issues

Unlike cellular networks and MANET, protocol design in wireless sensor networks generally does not focus on quality of service (QoS) issues. Instead, the major concerns in the design of sensor network protocols are how to extend the network lifetime and how to provide robust network fault tolerance. The key research issues are outlined as follows:

Energy-efficiency:

Energy conservation is always the primary concern in sensor networks. Sensors randomly deployed in a target area are usually small micro-electronic devices. This implies that a sensor node only can be battery-powered and power replenishment is almost impossible. Thus, the energy depletion of individual nodes will not only cause the failure of the nodes themselves, but also shorten the lifetime of networks. Each layer of protocol stack therefore should work energy-efficiently so that network lifetime can be maximally extended. According to [17], energy-efficiency serves as a good indicator of network lifetime.

Fault Tolerance:

In sensor networks, node failure can be caused by various factors. Besides battery power depletion, sensors may also frequently disfunction arising from dynamic environmental conditions. For instance, the operation of sensors may be interrupted when they are stuck or blocked by the terrain, or when they are displaced from the target area by wind or rain. Such frequent node failures will consequently lead to changes in the network topology. However, in sensor networks, individual node failures should not affect the overall application. That is, sustained services should be provided smoothly without any interruption by such failures. Sensors in the network are therefore required to self-configure and reorganize in face of such frequent network dynamics. Generally, fault tolerance capability is indicated by the reliability of the communications in sensor networks.

It must be noted that the network density is an important factor that influences the energy-efficiency and fault tolerance. Sensor networks are usually "densely" deployed [8, 15]. The high density of sensor networks aims to ensure sustained functionality of sensor networks in face of frequent node failures through sensor node redundancy. That is, by densely deploying sensors within the target area, it is hoped that the fault tolerance issue can be addressed quantitatively. According to [9], network density can be defined as:

$$\mu(R) = \frac{(N\pi R^2)}{A} \tag{1.1}$$

where N is the number of sensor nodes deployed in the target area, R is the radio transmission range of sensor nodes, and A denotes the size of the target area. However, the interpretation of "densely" is rather vague in sensor networks, and network density varies greatly with different application scenarios. Protocols should therefore be designed to scale well so that they work with the increasing network densities, as well as increasing target area sizes.

1.3 Ongoing Research on Sensor Networks

With the developments in battery technology and energy scavenging [10, 11] techniques, as well as the recent technical advances in IC design and MEMS techniques, strengthened efforts have been dedicated to sensor network technology by researchers all over the world. Generally, among the numerous projects and programs for sensor networks, two notable efforts are Wireless Integrated Network Sensors (WINS) [7] by University of California, Los Angeles, and PicoRadio networks [11, 12] by University of California, Berkeley Wireless Center.

Sensor nodes in WINS combine micro-sensor technology, low power signal processing, computation and low cost networking capability in a compact system. In WINS networks, sensors are networked to provide various kinds of embedded system applications. In-network information processing is supported in WINS. In a WINS node, the micro-power components — low power sensor interface, and signal processing architecture & circuits, operate continuously for constant monitoring of events in the environment, while the micro-power RF interface runs at a very low duty cycle for energy conservation. The radio interface parameters are used for the simulations in our study.

The PicoRadio project focuses on ultra-low power techniques. Compelled by the "last-meter" problem, this technique aims to support ad hoc wireless sensor networks composed of an assembly of self-contained heterogeneous nodes. Sensors in such networks, called *Pico Nodes*, are of extreme low power, light weight, and low cost. According to [11, 12], the Pico Node is smaller than one cubic centimeter, weighs less than 100 grams, and costs substantially less than one dollar. The power dissipation level in PicoRadio is even more aggressive — below 100 microwatts. This strategy aims to eliminate battery replacement, and will enable Pico Nodes to scavenge and harvest energy from environment. Compared with WINS nodes, Pico Nodes have a much smaller transmission range, limited to within only a couple of metres. The data reporting rate of Pico Node is also much lower, usually less than 1 Hertz, with an active cycle typically less than 1%. Constrained by such harsh requirements on energy consumption, the PicoRadio technique is expected to be more suitable for location-aided sensor network applications. Though the radio interface features of PicoRadio are not employed in our study to evaluate the performance of *PRIMP*, we expect a successful adoption of *PRIMP* in this technique due to the located-aided nature of *PRIMP*.

1.4 Main Contributions of the Thesis

In this thesis, a new routing scheme *PRIMP* is proposed to address the key issues in sensor networks — stringent energy constraint and poor fault tolerance capability so that long and reliable services can be provided. The main contributions are listed as follows:

- For energy conservation, a novel *on-demand virtual source* technique is designed to explore and update the location information of sources whenever necessary in a reactive manner. This technique is also used for data path re-establishment in the event of failures of all current paths between a sink-source pair. It significantly reduces the communication overhead from the dissemination of *interest* messages;

- *PRIMP* only maintains the directional data paths from sources to sinks through periodic interest dissemination towards the source region once the location information of the source region is obtained. Such directional path maintenance aims to keep alive the data paths from sources to sinks, under the context of unreliable transmissions in sensor networks;

- *PRIMP* routes data traffic over multiple *braided* paths simultaneously and probabilistically in a priority-based approach, based on the energy resource conditions of the paths. Such routing strategy effectively balances the traffic, and contributes to long-term energy-efficiency;

- *PRIMP* addresses the *slow startup* problem for time-sensitive sensor network applications. This allows the different sinks that initiate identical interest to retrieve data nearly simultaneously, without being *discriminated*. This is of special importance for integrating various kinds of collected data messages at multiple sinks in time-critical missions.

1.5 Organization of the Thesis

The thesis is organized as follows.

In this chapter, the basic concept and communication architecture of sensor networks are introduced, the characteristics and unique features of sensor networks are described, and the key research issues and challenges in sensor networks are presented.

In Chapter 2, techniques in the design of sensor network protocols are presented in the architecture level, followed by a brief overview on the existing key protocols in the medium access control (MAC) sub-layer and network layer. Some definitions and terminologies used throughout the thesis are also defined.

In Chapter 3, the basic ideas and design motivations of *PRIMP* are presented, based on the investigation of the existing routing protocols discussed in chapter 2.

In Chapter 4, the design of *PRIMP* is described in detail. The virtual source technique employed in interest dissemination is discussed, followed by the gradient paths setup procedure and the priority-based probabilistic routing approach.

In Chapter 5, simulation results for the performance evaluations of *PRIMP* and other comparable routing protocols are presented, and corresponding analysis is given accordingly.

In Chapter 6, this thesis concludes and discusses the future work on this research area.

Chapter 2

Protocol Design Guidelines and Preliminary Remarks

In this chapter, system features of sensor networks are presented. These features serve as important guidelines to the protocol designs in sensor networks. Some existing work on protocol designs are also investigated, followed by some definitions and terminologies which will be used throughout this thesis.

2.1 Protocol Design Considerations

As mentioned earlier in section 1.1, sensor networks have a special communication architecture. As part of the build-up of such communication architecture, protocol designs in sensor networks should therefore take the influencing system features of sensor networks into consideration. The key system features that impact protocol design include application-dependent attribute-based low-level naming, and data-centric communications. Protocols in sensor networks should therefore be designed to address key issues, based on the knowledge of these features.

2.1.1 Sensor Network Protocol Stack

With the considerations mentioned above, protocol stack embedded in sensor nodes is envisaged to be as shown in Figure 2.1 [15]. The protocol stack designed for sensor networks is different from the classic seven-layer Open System Interconnection (OSI) model [32]. It consists of only five layers and three management planes [15]. The respective functions of these layers are outlined as follows:

- The top layer — application layer provides a platform to build services in a sensor network application;
- The transport layer maintains the flows of data messages obtained from within the network, and ensures error-free data message deliveries and proper arriving sequence;
- The network layer is responsible for routing data messages from sources back to sinks;
- Data link layer deals with the data transmission between sensors over an unreliable medium;
- The physical layer is concerned with unstructured bit stream transmission over physical links.

The three management planes take care of the monitoring work of a sensor node. The instant power, movement and task allocation situations of a sensor node should be integrated with the protocol in each layer. For example, power condition of a sensor node can be incorporated into the network layer to help routing protocol make routing decision; the mobility of sensors can be integrated into the application layer to change the coordinating strategies employed by these sensors.

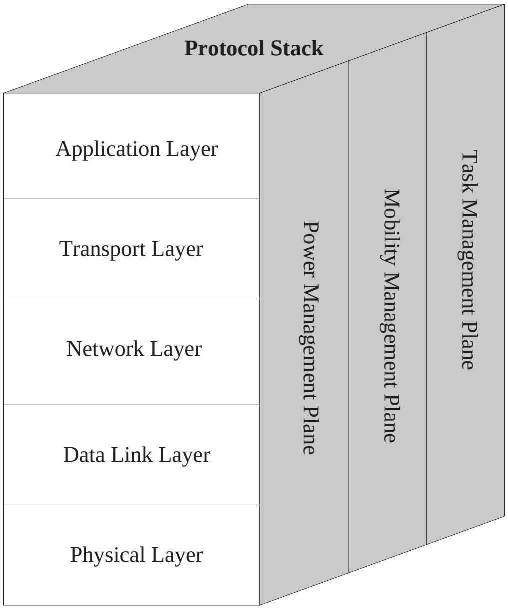


Figure 2.1: The protocol stack adopted by sensor nodes

Network Category	MANET and Cellular Networks	Sensor Networks
Major Concern	QoS issue	energy issue
Information based on	Topological location of nodes	application-dependent attributes
Support in-network processing	Not supported	supported
Naming binding resolution	Necessary	Do not need
Routing protocols adopting such naming strategy	DSR[13], AODV[14]	SPIN, Directed Diffusion

Table 2.1: Major distinctions of different wireless networks

2.1.2 Data-centric Communication Paradigm of Sensor Networks

Application-dependent attribute-based low-level naming strategy

As mentioned earlier in section 1.2.1, the communication paradigm for sensor network is different from that of MANET, and is also different from that of cellular networks. Table 2.1 provides an insight into their major distinctions. As shown in Table 2.1, low-level naming of th nodes in MANET or cellular networks leverages topological location, such as globally unique IP address. Such low-level naming is independent of any concrete application launched. However, in sensor networks, assigning a globally unique identification to each sensor is impossible due to their large numbers in the target area. Instead, low-level communication relies on attribute-based names which are *external* to the network topology and *relavant* to the application. Moreover, low-level naming in MANET and Cellular Networks is mapped to high-level naming. Communication built upon such naming primitives therefore requires the address resolution, as shown in Table 2.1. Address resolution is not an issue in MANET or cellular networks where energy resource is assumed to be abundant. However, in sensor networks, the overhead introduced by naming binding resolution is unaffordable for sensors with only scarce and unreplenishable power resource. The attribute-based low-level naming strategy for sensor networks can be illustrated by the following example — naming in a target surveillance mission:

Naming in tasking information:

Target = VX nerve gas truck

Surveillance__Scope__Min__Latitude = $33^\circ 10'$

Surveillance__Scope__Max__Latitude = $33^\circ 58'$

Surveillance__Scope__Min__Longitude = $44^\circ 20'$

Surveillance__Scope__Max__Longitude = $44^\circ 58'$

Task__Duration = 24 hrs //for the next 24 hours from from on

Naming in detected data event:

Target = VX nerve gas truck

Target__Location__Latitude = $33^\circ 23'$

Target__Location__Longitude = $44^\circ 47'$

Target__Detection__Time = $16 : 15 : 53$ //at a time which is about 16.26 hours later

Data-centric routing

In cellular networks or MANET, each node is named a globally unique identification, and IP-based communication is employed. Routing in these networks is therefore address-centric. As described in Table 2.1, such low-level communication primarily aims to achieve QoS performances, such as throughput and delay requirements. In contrast, a sensor network application is usually more interested in querying a phenomenon rather than a specific node. Messages transmitted within the network, such as task descriptions or data events are therefore named based on their respective attributes. That is, routing in sensor networks is actually data-centric in nature. Such data-centric routing is essential for sensor networks where power issue is the primary concern. With such “self-identifying” data-centric naming strategy, in-network processing is possible for dynamic task allocation, data aggregation and collaborative signal processing. This in-network processing can significantly conserve dissipated energy, e.g. by aggregating different messages and suppressing duplicate messages.

2.2 Related Work on Sensor Network Protocols

Due to the unique features of sensor networks, especially the stringent resource constraint and poor network fault tolerance, robust and energy-efficient routing and MAC protocols are desired. Here, some key existing work on MAC and routing protocols are briefly investigated to illustrate the basic design principles.

2.2.1 MAC Protocols

In the MAC layer, protocol should be designed to be energy-efficient and self-organized. From the view of the overall task, per-node fairness or latency in contending for the shared media are basically less important. They can be traded off for energy-efficiency, as long as the *end-to-end* (source-to-sink) fairness and latency performances are still acceptable. Currently, most of the existing MAC protocols are proposed for cellular networks and MANET, they are however unsuitable to be used in sensor networks.

MAC for Cellular Networks

Firstly, MAC protocols designed for cellular networks can not be adopted for sensor networks. Cellular network is a one-hop communication system: each mobile node communicates with base-stations directly. These base-stations are static and form the wired-backbone of the whole network. MAC for cellular networks focuses on QoS issues. It centrally controls the access of mobile nodes to the media resource through base-station to achieve certain QoS performance. Energy-efficiency, in contrast, is less important in such infrastructure-based communication system, because of the abundant power resource on the backbone and the replenishable power at mobile nodes.

MAC for Ad Hoc Networks

Similarly, MAC designed for MANET also aims at provision of a high QoS. Since power can be replenished or replaced at each node, energy-efficiency in MANET is also of secondary importance. All nodes in MANET are peers and physically similar, end-to-end multi-hop QoS is therefore achieved through the strategy taken by MAC at each single hop.

Despite of the trivial importance of the energy issue in MANET, it is noticed that an interesting idea about energy conservation in MAC is still proposed in *PAMAS* [19] for MANET. Based on the original *MACA* [21] protocol, *PAMAS* adds one separate channel for signaling. The unique feature of *PAMAS* is that it conserves the battery power by intelligently powering off nodes that are not actively transmitting or receiving. This insight gives an inspiration to the MAC protocol designs for sensor networks. It is reported in [19] that the energy conservation manner does not influence the delay or throughput characteristics of multi-access protocols, and can be easily built into CSMA-based routing protocols for energy conservation.

MAC for sensor networks

As mentioned earlier, power conservation is the primary concern in MAC protocol designs for sensor networks. A sensor network MAC protocol therefore should be energy-efficient firstly, so that network lifetime can be extended maximally. Secondly, traffic patterns in sensor networks are distinct from that in MANET or cellular networks. In MANET or cellular networks, the occurrence of the packet transmission is assumed to follow a stochastic distribution; while in sensor networks, traffic tends to be highly correlated and periodical. Another point needs to be considered is that sensor networks are data-centric, and operate as a collective structure; while in MANET or cellular networks, traffic flows are independent and point-to-point.

Traditionally CSMA protocols are considered to be unsuitable for sensor networks due to its full-time channel sensing. However, some CSMA protocols also support energy conservation. For example, in IEEE 802.11 [16], radios can be turned off if the virtual carrier sense — Network Allocation Vector (NAV) [16] finds the medium is not free.

A transmission control scheme for media access is proposed in [17] based on the insights on the traffic characteristics of sensor networks. This CSMA-based MAC finds that constant sensing periods and introduction of random delay prior to transmission can provide robustness against collision, and are the most energy-efficient for CSMA schemes. It is also reported in [17] that fix window and binary exponential decrease backoff scheme should be incorporated with the above listen and delay strategies to help maintain proportional fairness to original traffic and route-thru traffic. A simple adaptive rate control scheme is also proposed for achieving multi-hop fairness. Additionally, introduction of phase change in application level is also advised to get over any *capturing effects* [17].

A MAC protocol which contains a variant of TDMA is proposed in [18] in order to suppress the idle power dissipation. Two algorithms are employed in this protocol: *SMACS* and *EAR*. *SMACS* algorithm aims to achieve network start-up and link-layer organization, and *EAR* algorithm provides seamless connection of mobile nodes in the network. *SMACS* is a distributed infrastructure-building protocol that enables nodes to discover their neighbors and establish transmission or reception schedules distributively. Power conservation is achieved by using a random wake-up schedule during the connection phase and by turning off the radio during idle time slots. *EAR* attempts to offer continuous service to the mobile nodes under both mobile and stationary conditions. *EAR* is transparent to the *SMACS*, so that *SMACS* will function until mobile nodes is introduced into the network. The design of this protocol is based on the assumption that most of the sensors are static, with only a small fraction of nodes are mobile, i.e., every mobile node can find a number of stationary nodes in its vicinity.

S-MAC proposed in [20] provides an energy-efficient MAC protocol for sensor networks. *S-MAC* expects that individual sensors remain largely inactive for long periods of time, but then suddenly become active driven by the sensed phenomena. As described in Table 2.2, *S-MAC* treats energy conservation and self-configuration as the primary design goals, while per-node fairness and latency are considered less preferable. In *S-MAC*, three novel techniques are used to reduce the energy consumption and to support network self-configuration. Firstly, to reduce the power dissipation from idle listening of sensor nodes, sensor nodes are put to sleep periodically. Neighboring sensors self-organize to form virtual clusters to auto-synchronize on their sleep schedules. Secondly, an in-channel signaling technique is used to switch off the radio at appropriate time for overhearing avoidance. Thirdly, *message passing* technique is applied

MAC protocol	design goals design techniques employed
IEEE 802.11 with power conservation	suppress idle power dissipation switching radio power based on NAV status
MAC protocol proposed in [17]	fair bandwidth allocation to the infrastructure and energy efficiency constant listen period; introduction of random delay prior to transmission; an adaptive rate control scheme; introduction of phase change in application level
MAC protocol proposed in [18]	suppress the idle power dissipation two algorithms: *SMACS* and *EAR*
S-MAC	energy conservation and self-configuration three novel techniques: periodic listen-sleep schedule; overhearing-avoiding in-channel signaling; *message passing*

Table 2.2: MAC alternatives for sensor networks

to reduce application-perceived contention latency for sensor network applications, i.e., a long message is fragmented into many small parts which are transmitted in a burst. This helps to reduce the costly retransmission of long messages due to transmission corruption.

2.2.2 Routing Protocols

In sensor networks, ad hoc networking technique is required to route data packets back to the task management center through multi-hops. Firstly, the unique communication paradigm indicates that the routing protocol for sensor networks must be data-centric. Secondly, due to the scarce and irreplenishable battery power, routing protocol must be energy efficient so that network lifetime can be extended maximally. Thirdly, fault tolerance issue must be addressed to provide reliable deliveries of data events under the context of frequent node failures.

Here, several existing routing schemes proposed for sensor networks are briefly discussed to highlight the design principles.

Flooding: *Flooding* is the simplest approach for routing. Whenever a sensor node receives a data packet, it will simply broadcast the packet. The broadcasting will continue until the TTL (time to live) of a data packet times out. It turns out to be a deficient protocol, because a lot of duplicate traffic will be generated by the immediate nodes. The repeated transmission and reception of duplicate traffic leads to the famous *data implosion* problem in *flooding*. It is simple, robust, but too expensive in terms of energy dissipation.

SPIN: Sensor Protocols for Information via Negotiation (*SPIN*) [22] is a family of adaptive data-centric protocols. *SPIN* protocol family rests upon two basic ideas. The first idea is: several applications carried out can operate efficiently and conserve energy by communicating with each other about what data they already have and what they still need to obtain respectively. Since exchanging meta-data is more energy-efficient than exchanging data, energy can be conserved. *SPIN-1* is a simple three-stage (ADV-REQ-DATA) handshake protocol using such technique to disseminate a newly-obtained data message at a node. The second idea is: a routing protocol adaptive to the energy resource of nodes helps extend the network lifetime. Based on this idea, *SPIN-2* protocol adds a simple energy-conservation heuristic to the *SPIN-1* protocol. When the energy resource of a node is plentiful, *SPIN-2* works just like *SPIN-1*; when a node observes that its energy resource approaches a low-energy threshold, *SPIN-2* will work in an adaptive, conservative manner such that the node's participation in the *SPIN-1* protocol will be reduced.

SAR: Sequential Assignment Routing (*SAR*) proposed in [18] tried to improve the energy-efficiency in low-mobility sensor networks through a table-driven multi-path approach. Multiple paths are established from each sensor node to the sink with these multiple paths only necessarily to be disjoint inside the one-hop neighborhood of the sink. This is achieved through the building of multiple trees, each is rooted from a one-hop neighbor of the sink and grows outwards from the sink by successively branching to neighbors at higher hop distances from the sink while avoiding nodes with very low QoS and energy reserves. The advantage of this structure is that it allows each sensor to directly control which one-hop neighbor of the sink will relay a message. When data messages are to be routed back to the sink, path selection will be made by the data events initiator based on three considerations: energy resource estimated by maximum number of packets that can be routed without energy depletion if it has exclusive use of this path; additive QoS metric where a higher metric implies low QoS; and the priority level of a packet.

LEACH: Low-energy adaptive clustering hierarchy (*LEACH*) [23] is a clustering-based protocol that tries to distribute the high energy dissipation in communication with the base station to all sensor nodes in the network. That is, different cluster-heads are selected in each periodic setup phase through the random number generation [23]. Once a cluster-head is selected, leadership and membership of the cluster-head and the cluster-members will be set up through advertising message. TDMA approach is used by cluster-head to assign times slots to cluster-members for them to send to the cluster-head in the steady phase.

Directed Diffusion: *Directed Diffusion* [24] is data-centric communication paradigm for sensor networks. In *Directed Diffusion*, all the sensors in the networks are application-aware, and are collaborated to obtain the named data. Generally, four stages are required to draw the desired data from within the network. Firstly, interest is flooded into the whole network periodically for the named data. Interest and desired data are all named by a list of attribute-value pairs. After sources receive the interests, exploratory data will be sent back along all the existing gradients. By the time the exploratory data arrives at sinks, positive reinforcement will be initiated by sinks to set up one shortest-delay path from itself to sources. Finally, after path reinforcement is finished, data message will be sent back to sinks along this path. Path exploration and path reinforcement will also be conducted periodically for discovering new empirically shortest-delay reinforced paths. Negative Reinforcement is used as another local rule to aggressively truncate the path from sources sending duplicate data traffic, or to serve as memory-saving alternative to message caching technique for loop removal.

2.3 Definitions and Terminologies

The following terminologies and definitions are used throughout this thesis.

2.3.1 Sensor Networks Terminologies

1. *interest*: An *interest* is a querying message that describes the task to be fulfilled, i.e., tasking information.

2. *data*: *Data* is a message that replies to an interest, it describes the events sensed by sensor nodes inside or beside phenomenon.

3. *sink*: A *sink* is a sensor node where interests are generated before they are disseminated into the network, it is a network entering point of interests;

4. *source*: A *source* is a sensor node where data is generated after events are sensed and obtained from phenomenon;

5. *gradient*: A *gradient* is a direction state which is set up in the cache of a sensor node when the node receives an interest. The direction of a gradient is set toward the neighboring node from which the interest is received.

6. *source region*: *Source region* is a small geographic area where sources are located. It is located inside or near the phenomenon.

2.3.2 Definitions

1. *accumulated hop count* — weighted average hop count from a certain node to a certain sink. It is repeatedly updated as an interest packet traverses the network hop by hop.

2. *remaining power resource* — weighted average amount of power resource from a certain node to a certain sink. It is repeatedly updated as an interest packet traverses the network hop by hop.

3. *priority* tag — information tagged to a gradient cached at a node. It indicates the predicted energy resource condition of the data paths from the node to a sink along this gradient.

4. *group id* tag — information tagged to a gradient cached at a node. It indicates which sink(s) can be reached if the data event is routed through this gradient.

5. *slow startup* problem — Due to the information about each specific sink is transparent to each source, identical interest initiated by different sinks will not evoke the sending-back of data events from a source more than once. This makes some sinks suffer long delay between the initiation of interest and the arrival of the first replying data message.

Chapter 3

Ideas and Design Motivations of *PRIMP*

In sensor networks, two important issues that have to be primarily considered, while routing, are stringent constraint on power and vulnerability of sensors to dynamic environmental conditions. This creates a demand for an energy efficient, robust routing protocol. In this thesis, a priority-based multi-path routing protocol (*PRIMP*) is proposed to address these two key issues. In an effort to maximize energy efficiency and to provide a robust network fault tolerance, *PRIMP* aims in offering long and reliable services for sensor network applications. Besides, *PRIMP* also solves the slow startup problem that may occur in other data-centric routing schemes.

3.1 Design Motivations of *PRIMP*

In data-centric sensor networks, all the messages transmitted within the network except data are considered as overhead. Based on the investigation conducted on existing routing protocols, as mentioned in chapter 2, it is observed that large communication overhead is incurred in most of the data-centric routing protocols, such as *directed diffusion*. In *directed diffu-*

sion, the overhead mainly comes from the flooding of interest messages which are periodically *refreshed* [24] to overcome the unreliable transmission in sensor networks. Periodic propagation of exploratory data, along with positive and negative reinforcement messages also contribute to considerable energy dissipation. Geographical Energy Aware Routing (*GEAR*) [25] can significantly improve the energy-efficiency of *directed diffusion*. This is achieved by establishing a single path between source region and each sink. However, information about sources is necessary. Based on the above insight, *PRIMP* aims at improving energy-efficiency by suppressing all possible communication overhead so as to achieve maximum network lifetime. *PRIMP* realizes this through its novel on-demand virtual source technique and the convergence of data paths to different sinks.

Communication reliability (fault tolerance) is another critical issue in sensor networks. However, not much work has been done so far, to address this issue on a satisfactory level. For instance, in *directed diffusion*, for each round of path exploration, only one empirically lowest delay path is reinforced. This leads to a potential poor reliability of the data transmission on the reinforced path. Though different reinforced data paths may be set up over times due to MAC dynamics and changing environmental conditions, the achieved effect is trivial. This is because the data rate in sensor networks tends to be extremely low. Therefore, in the absence of any obstruction, the empirically low delay paths reinforced are likely to overlap with the shortest path. Unlike *directed diffusion* and *GEAR*, multiple paths are used in the routing scheme proposed in [26] to improve network fault tolerance. However, it has the following limitations: firstly, multiple paths of same minimum hop count from a sensor to a base-station (sink) may not exist due to the uneven network density; secondly, it aims at delivering the data traffic from a sensor to only one base station (the nearest sink); thirdly, in multi-base-station scenario, if the multiple paths to different base stations share some common *gradients*, data destined to the nearest base station may not be delivered to it actually. Moreover, the scheme still resorts to flooding to propagate poll messages (interests). *SAR* (section 2.2.2) is expected to provide a good network fault tolerance capability. It creates multiple paths from each node to the sink by building multiple trees, each rooted from a one-hop neighbor of the sink. At the end of the tree-building procedure, most nodes will belong to multiple trees, and thus have multiple paths disjoint inside the one-hop neighborhood of the sink. However, metric update requires periodic update of these multiple trees. The maintenance of such multiple tree structure is too costly in terms of energy consumption. This is especially true in the multi-sink scenario where tree structures originating from neighbors of different sinks are expected to be independent of each other. Moreover, *SAR* uses only one path selected by the data-initiator (source) to maximize the weighted QoS for packets of different priorities. Therefore, transmission along the selected single path is not reliable in sensor networks. This is also the case for the routing protocol proposed in [26]. In *PRIMP*, to achieve energy-efficiency as well as fault tolerance robustness, multiple paths of different lengths are established explicitly in *braided* (mesh) structure. These multiple paths are used simultaneously to draw data from the network. The selection of these multiple paths is conducted dynamically by the nodes at each hop of the data paths. Such path selection manner further helps in balancing the traffic load, without compromising the energy-efficiency of the protocol too much.

3.2 Key Ideas of *PRIMP* and Assumptions

PRIMP is proposed under an application scenario where sources information is absent. The compelling reason for our study in such scenario is that it represents a broad spectrum of applications, for example, target surveillance or area monitoring for military or civilian use.

3.2.1 Assumptions for *PRIMP*

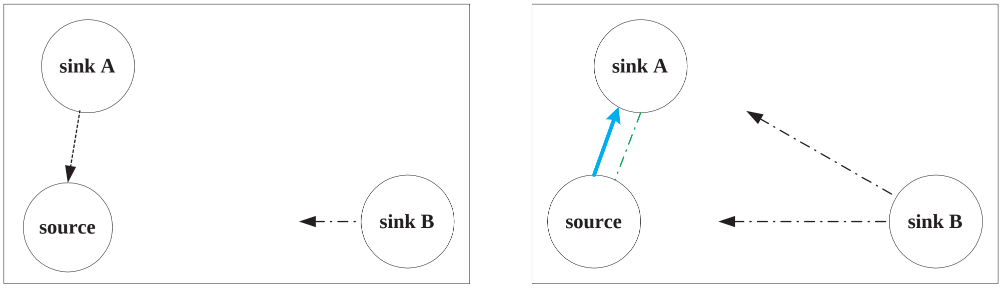
In this study, the existence of a localization system [27, 28] at each sensor is assumed, as it enables each sensor to obtain its current geographic position. Also, since wireless sensor networks are largely application dependent, the target area where an application is to be fulfilled through the collaboration of sensors has to be designated by the human-operated task management center before the application starts. Therefore it is reasonable to assume that the rough geographic information about the boundary of the target area is available.

3.2.2 Key Ideas of *PRIMP*

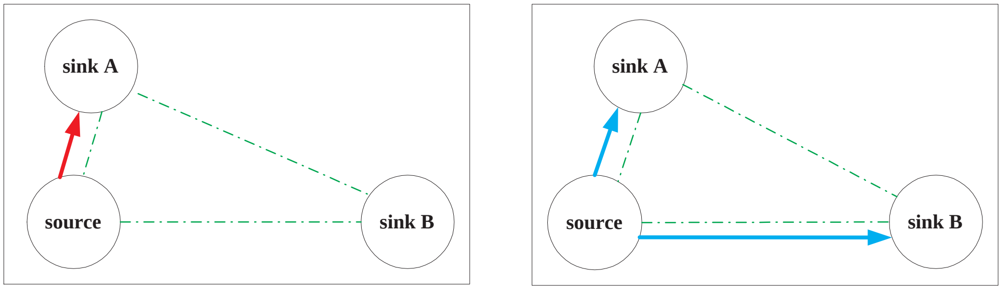
Key features of *PRIMP* include:

- Employing a novel on-demand virtual source technique either to update sinks' knowledge of the whereabout of source region whenever necessary, or to re-establish data paths from sources to sinks when all the paths to a particular sink are corrupted;
- Directionally maintaining multiple braided data paths from sources to sinks at the interest dissemination stage. As a result, multiple data paths to different sinks will be combined to the largest possible extent;
- Attaching priority tag and group id tag to each gradient cached at nodes when setting up multiple data paths. Priority tag can be either of these two types: accumulated hop count, and remaining power resource. The energy level of each node can be classified into two phases (i.e., *good* or *poor*) according to its residual battery power;
- Routing data messages over multiple paths simultaneously *on the fly*. In other words, when data message traverses from sources to sinks in a hop by hop fashion, multiple gradients will be selected in a priority-based probabilistic approach at each hop.

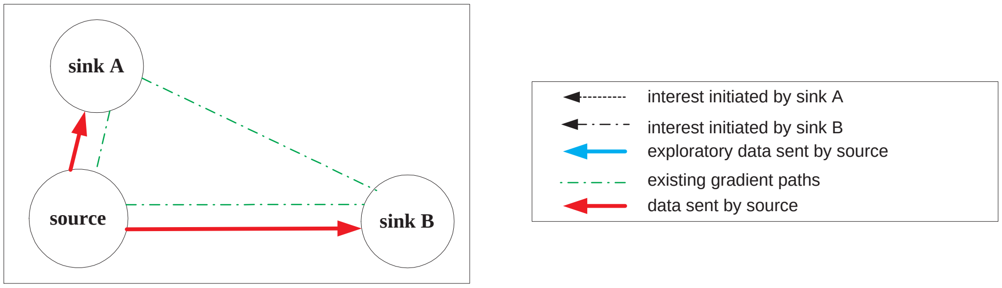
Additionally, *PRIMP* also addresses the *slow startup* problem that occurs in *directed diffusion* in multi-sink scenario. Here we define the *startup time* to be the time duration from the launching of a sensor network application to the moment when every sink begins to receive data message successfully. In time-critical sensor network applications, such as battlefield



(a) ***interest*** **from** ***sink*** **A arrives at** ***source*** **first** **(b)** ***exploratory data*** **is sent to** ***sink*** **A only**



(c) ***data*** **is sent to** ***sink*** **A after positive reinforcement** **(d) second round** ***exploratory data*** **sent to** ***sink*** **A and** ***sink*** **B**



(e) ***data*** **is sent to** ***sink*** **A and** ***sink*** **B after positive reinforcements**

Figure 3.1: *Slow startup* problem

or rescue missions, short *startup time* can be very critical. If more than one kind of data messages are required to be drawn from the network instantaneously, slow startup problem can be even severe. For example, several kinds of data messages need to be aggregated with the information that are available at these sinks so that meaningful information can be derived. For every kind of these data messages, if there are some sinks suffering from slow startup problem, eventually the time for these sinks to obtain all the desired messages will be severely delayed.

The *slow startup* problem can be illustrated in a simple two-sink one-source scenario, as shown in Figure 3.1. Suppose that two *sinks* A and B initiate identical interests. If the interest initiated by A arrives earlier at the source than that initiated by B, exploratory data would be sent back from the source via all the established gradients from the source. However, since the gradients from the source towards B have not been set up yet, exploratory data cannot reach B directly. It is noticed that exploratory data may also reach B via A. However, due to the short one-way latency [24] for data traffic in *directed diffusion*, the gradient paths from A towards B would not be established by the time the exploratory data arrives at A. Therefore, this exploratory data cannot reach B via A. Later on, the source will not send exploratory data in response to the received interest initiated by B, because it would find that the reply to such interest type has already been sent back. Since sink B cannot receive any exploratory data, it would not reinforce a path to draw data messages from the source. This situation will last until the next round of exploratory data invocation at the source. Since propagating exploratory data is energy-consuming, it is only conducted infrequently (i.e., dispatching cycle of exploratory data is long). Therefore, long startup time is experienced by sink B.

Chapter 4

Scheme Design of *PRIMP*

The design of *PRIMP* is based on two basic principles: suppressing communication overheads from all possible sources, and balancing traffic load for the provisioning of robust network fault tolerance capability.

In general, *PRIMP* operates in two separate stages:

- Interest Dissemination Stage
- Priority-based Path Selection Stage

At the interest dissemination stage, interest messages are dispatched to the network from sinks. When sources receive the interests, corresponding data will be send back at the priority-based path selection stage.

4.1 Interest Dissemination Stage

At the interest dissemination stage, interest messages are generated by sinks and dispatched to the network. Interest dissemination initiated by sinks will be refreshed periodically due to unreliable transmission in sensor networks. It aims to fulfill either of the following two functions:

- Function 1: Updating sinks' knowledge of the whereabout of source region, or re-establishing the multiple data paths from sources to sinks in response to network dynamics;
- Function 2: Directionally maintaining multiple paths from sources to respective sinks for transmission reliability.

At this stage, virtual source technique is invoked reactively in the interest dissemination to update sinks' knowledge of the sources (Function 1). This technique is triggered in the event that sources information is absent or it is necessary to re-establish all the data paths from sources to a particular sink. As regards function 1, this technique is simply used by sinks for interest dissemination. In case of the latter, when new sources (maybe a new source region) find something that can reply to the cached interest that is still alive, a short control message *REQ_UPDATE_MSG*, initiated by sources will be flooded out for some time. The time period of flooding, which is a protocol design parameter, should be long enough to make sure that the sinks can receive it and invoke the virtual source technique for interest dissemination accordingly. In case of the latter, the virtual source technique may also be triggered due to failures of all the current data paths from sources to a certain sink.

When virtual source technique is used, an interest will be directionally propagated towards the virtual sources. This aims at disseminating the interest to all the nodes within the network. By this way, the interest finally arrives at sources. Then the sources send back matched data events, piggybacking the geographic information of source region (e.g. the geographic center of source region). Once the sinks obtain the piggy-backed information, the interest packets that are generated subsequently are directionally disseminated to the source region (Function 2). Thus, only the data paths from the source region to sinks are periodically maintained against unreliable transmission. The *confirm flag* bit in the subsequently generated interest packets is set to acknowledge the sources about the location information.

Due to unreliable transmission in sensor networks, sinks constantly employ virtual source technique for interest dissemination until the the location information of the source region is obtained. Similarly, sources constantly send data containing the piggy-backed location information of the source region until they detect that the confirm flag in the received interest packet is set.

If virtual source technique is used, at the end of the interest dissemination stage, multiple data paths are set up from every node within the network to each sink. If the knowledge of source region has already been obtained by sinks, only the data paths from sources to each sink are established.

4.1.1 Virtual Source Technique

As shown in Figure 4.1, the topology of the target area in which a task is to be performed can always be considered as rectangular. This rectangular area can be further divided into four sub-areas (marked by sub-areas 1, 2, 3 and 4 in Figure 4.1) according to the sink's position.

When virtual source technique is used for interest dissemination, an interest packet generated at the sink is simply broadcast. When the interest arrives at the immediate neighbors of the sink, virtual sources are selected by these neighbors. The virtual sources selected by a certain neighbor are always located at the corners of the current sub-area in which this neighbor resides. Once the virtual sources are determined, the interest packet will be disseminated towards them within the current sub-area. An intermediate node within the current sub-area forwards the received packet by the approach illustrated below after it has received an interest. Since currently the sink has no idea of where the real sources are located, such interest dissemination approach makes sure that the interest will be disseminated throughout the network to nearly every sensor node. This is because: The sub-area in which an intermediate node resides (could be any of the four sub-areas: sub-area 1, 2, 3 and 4) can be further devided

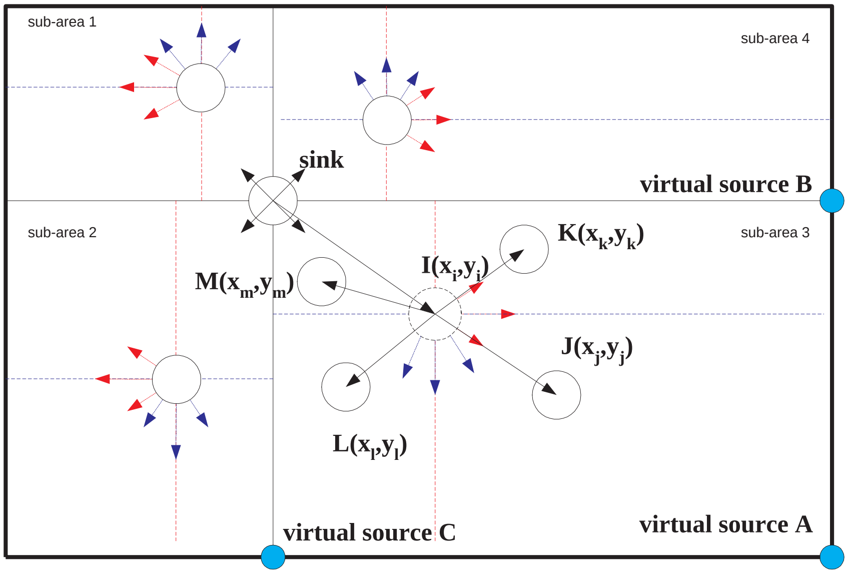


Figure 4.1: Interest dissemination with virtual source technique invoked

into four smaller regions using two lines that are parallel to the two borders of the target area and cross at the geolocation of this intermediate node. Thus, *all* the downstream neighbors of this intermediate node will fall into one of these four regions without exception. At the same time, the three selected virtual sources will fall into one of the three regions (except the region where the sink resides) respectively. Therefore, a downstream neighor will be forced to forward the interest packet for sure as long as this downstream neighbor does not "step back" (geographically closer to the sink, and farther from the three selected virtual sources) into the region where the sink resides. Since an interest packet always traverses the network in the direction from the sink towards the borders of the target area, this virtual source technique makes sure that all the sensor nodes qualified for directional interest-flooding will not be missed out, while at the same time contributes to the energy conservation (suppress the flooding explosion at every single relay of an interest packet).

As shown in Figure 4.1, when an interest packet initiated by the sink arrives at node I, an immediate neighbor of the sink in sub-area 3, node I selects virtual sources for this received interest. In *PRIMP*, virtual sources are selected to be at the corners of the current sub-area. Therefore, *virtual sensors A, B* and *C* at the corners of the sub-area 3 where node *I* resides are selected. Then node *I* continues to relay this interest. As Figure 4.1 shows, when the interest arrives at the downstream neighbors of node *I*, say, nodes *J, K, L* and *M*, which also reside in the sub-area 3, these downstream neighbors judge whether they are qualified to forward the received interest packet. If we further divide a sub-area into 4 smaller regions using two lines that are parallel to the two borders of the target area and cross at the geo-location of a sensor node, each neighbor of the node will be located in one of the four smaller regions. For instance, when an interest arrives at downstream neighbor node *J*, node *J* finds that it is located at the bottom right region of node *I*. Node *J* then checks if there is any virtual source located in this small region. Since node *J* finds that there is a virtual source, i.e., *A* is in this region, it decides that it is qualified to relay this interest packet.

Similarly, the downstream neighbors *K* and *L* find that they are also qualified to relay the received interest, while node *M* finds that it is not. As regards node *K*, it finds that virtual source *B* is located in the same smaller region (upper right region of node *I*) in which it is also located and for node *L*, it shares the same smaller region (bottom left region of node *I*) with the virtual source *C*. Since there is no virtual source in the upper left region of node *I*, node *M* simply drops the received interest packet.

The effect of this virtual-source-based interest dissemination is that interests go out towards the four borders of the current sub-area. Such forwarding strategy can be represented by the simple algorithm described in Figure 4.2.

After the piggy-backed location information of the source region is obtained by all the sinks, virtual source technique is no longer be used. The subsequent interest dissemination follows a different rule: an interest-receiving node will forward the packet within the current sub-area, only if this node is qualified according to the algorithm described in Figure 4.3.

```
Algorithm interest_forwarding_employing_virtual_source_technique
    if (current node is sink) then
        broadcast the interest pakcet
    else
        geo_target_area <-- obtain geo-info about the target area from the received interest
                            pakcet; this is part of tasking information
        geo_sink <-- obtain geo-info about the sink from the received interest packet
        geo_self ($x_j$, $y_j$) <-- obtain the geo-info of current node
        geo_up ($x_i$, $y_i$) <-- obtain the geo-info of the upstream neighbor i that forwarded this
                          interest to the current node j
        sub_area_id <-- judge which sub-area the current node is in using geo_target_area,
                       geo_sink and geo_self ($x_j$, $y_j$)
        switch (sub_area_id) {
        case sub_area_1:
            if ($x_j < x_i \,||\, y_j > y_i$) then
                forward this interest after processing
            else
                drop this interest
            end
        case sub_area_2:
            if ($x_j < x_i \,||\, y_j < y_i$) then
                forward this interest after processing
            else
                drop this interest
            end
        case sub_area_3:
            if ($x_j > x_i \,||\, y_j < y_i$) then
                forward this interest after processing
            else
                drop this interest
            end
        case sub_area_4:
            if ($x_j > x_i \,||\, y_j > y_i$) then
                forward this interest after processing
            else
                drop this interest
            end
        }
    end
end algorithm interest_forwarding_employing_virtual_source_technique
```

Figure 4.2: Interest forwarding algorithm with virtual source technique invoked

```
Algorithm interest_forwarding_after_obtaining_source_region_knowledge
    if (current node is sink) then
        obtain the piggybacked geo-location of the source region from recevied data message
        put source region geo-location information into the newly initiated interest
        broadcast this newly initiated interest pakcet
    else
        geo_source_region <-- obtain geo-info about the source region from the received
                              interest pakcet; this is part of tasking information
        geo_sink <-- obtain geo-info about the sink from the received interest packet
        geo_self ($x_j$, $y_j$) <-- obtain the geo-info of current node
        geo_up ($x_i$, $y_i$) <-- obtain the geo-info of the upstream neighbor i that forwarded this
                          interest to the current node j
        sub_area_id <-- judge which sub-area the current node is in using geo_target_area,
                        geo_sink and geo_self ($x_j$, $y_j$)
        switch (sub_area_id) {
        case sub_area_1:
            if ($x_j < x_i$ && $y_j > y_i$) then
                forward this interest after processing
            else
                drop this interest
            end
        case sub_area_2:
            if ($x_j < x_i$ && $y_j < y_i$) then
                forward this interest after processing
            else
                drop this interest
            end
        case sub_area_3:
            if ($x_j > x_i$ && $y_j < y_i$) then
                forward this interest after processing
            else
                drop this interest
            end
        case sub_area_4:
            if ($x_j > x_i$ && $y_j > y_i$) then
                forward this interest after processing
            else
                drop this interest
            end
        }
    end
end algorithm interest_forwarding_after_obtaining_source_region_knowledge
```

Figure 4.3: Directional interest forwarding algorithm

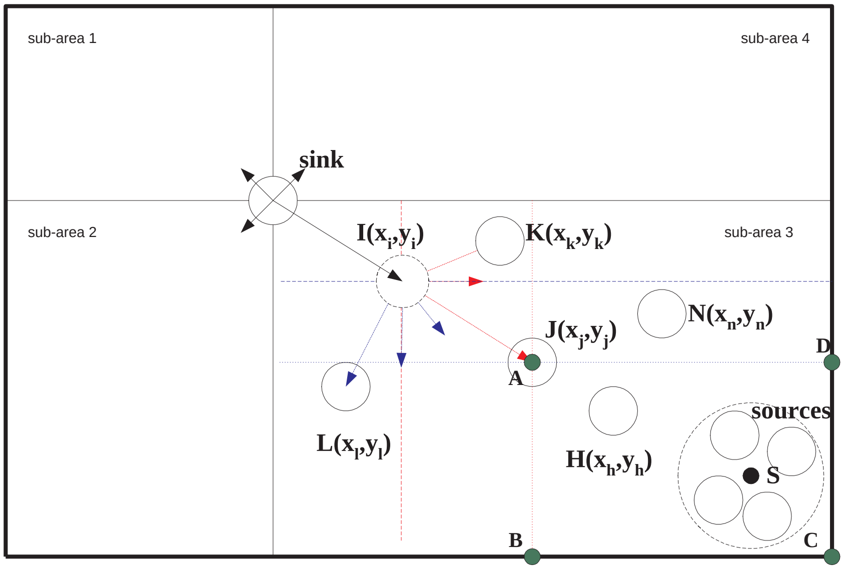


Figure 4.4: Directional interest dissemination

This forwarding algorithm can be further illustrated in the following example. Here, the sources are located in the sub-region 3 (Figure 4.4), with S denoting the geographic center of the source region. Once the geographic information of the source region (the geographic location of S) has been obtained by the sink, this information is put into the subsequent interest messages generated by the sink. By the time such an interest arrives at the downstream neighbors of node I via I, decision on forwarding this received interest message is conducted at each of these downstream neighbors respectively. Suppose the interest arrives at the downstream neighbor J, node J finds that it is located in sub-area 3 with the source region. The algorithm described in Figure 4.3, takes the decision to forward the interest message. Similarly, node K and node L find that they are not allowed to forward the received interest message, and simply drop it after necessary processing.

As described in Figure 4.3, similar rules are adopted for other three sub-areas (sub-area 1, 2 and 4). Such interest forwarding rule actually decides that a downstream neighbor of an intermediate node can forward the received interest message only if it is located in the same smaller region with the sources (source region). Corresponding to the cases where the sources reside in the sub-areas 1, 2, 3 and 4, the smaller regions of an intermediate node where the sources are located are the upper-left region, bottom-left region, bottom-right region, and upper-right region, respectively.

Note that no matter the virtual source technique is invoked for interest dissemination or not, the judgement on interest forwarding qualification of a certain node requires the location information of its upstream neighbors from which it receives the interest packet. This information is contained in the interest packet. In fact, whenever an interest message is to be forwarded by a node, the position information contained in the packet will be updated first. The current position of the node will replace the existing position information of the upstream neighbor that forwarded this packet to the node.

4.1.2 Setting Up Gradient Paths

During the dissemination of interest, a sensor node may receive multiple copies of an interest packet identified by a unique packet id from different upstream neighbors. This is due to the high node density in sensor networks. However, only one interest copy will be forwarded by the node, although all the interest copies will be used to update the gradient cache. The node *may* create and cache a new gradient entry for this interest type if the packet is from a new neighbor. If the packet is received from a neighbor whose corresponding gradient already exists in the gradient cache, the node simply updates the time-stamp of the corresponding cached gradient entry. However, not every interest packet from a new neighbor necessarily invokes the setup of a new gradient entry in the cache. Instead, only a certain number (α) of gradients are allowed to be set up and cached. The *qualification* of gradient setup will be discussed later in this section.

In this way, multiple ($\leq \alpha$) gradients are set up and cached at the interest receiver at each hop when an interest packet traverses the network in a hop by hop fashion, at the interest dissemination stage. Since several downstream neighbors may respectively cache a gradient entry that points a same upstream node, the gradients cached at the downstream neighbors is in a braided (or mesh) structure. Therefore, at the end of the interest dissemination stage, multiple braided paths will be established from the sources to sinks.

At every node in the network, each cached gradient is tagged with two pieces of tagging information: ***group id***, and ***priority***.

Group Id Information:

Group id tagging information is generated together with the tasking information at a sink, and is attached to the interest packet to specify the interest generator (that sink). Since the number of sinks in a sensor network application is usually small, it is possible to assign a unique group id for each sink. If a gradient entry needs to be set up when an interest packet arrives at a node, the group id information in the packet will be used to tag this gradient. A gradient tagged with a group id indicates that the corresponding sink can be reached via this gradient. As will be shown later in section 4.2.3, the introduction of group id leads to energy saving in routing data traffic. It is also employed to perform loop avoidance.[1]

Priority Information:

Priority tagging information can be either one of the two types: *accumulated hop count* and *remaining power resource.* Every time before a node forwards a received interest packet, this information will be computed (the detailed computing approaches will be discussed later in section 4.1.4) to update the older priority information contained in the packet. After the packet that is sent out by this node, arrives at one of its downstream neighbors, a gradient towards this node will probably be set up and cached at that neighbor. And this tagging

---

[1] In multi-sink scenario, gradients established by the identical interests initiated by different sinks may have various directions. Therefore, they may form loops.

information in the packet will be used by that neighbor to tag the established gradient.

As mentioned earlier in this section, the gradient setup at a node is conducted based on sets of rules when an interest arrives at the node. Here, *qualification* of the gradient setup is discussed in details. Generally, *PRIMP* aims in setting up only a certain number (α) of most energy-efficient gradient paths at each hop.[2] When an interest packet arrives at a node, the priority information contained in the packet will be used for the gradient setup. In fact, the qualification of the gradient setup needs to be judged using this priority information based on the rule described in Figure 4.5. Precisely, if the number of the cached gradient of this interest type is less than α, a new gradient is set up as long as the interest is from a new upstream neighbor (no cached gradient corresponds to this neighbor). If there are already α gradients of this interest type in the cache, the value of the accumulated hop count contained in the on-coming interest packet will matter in the qualification decision.

Finally, it is noticed that in *PRIMP*, the value of design parameter α should be chosen carefully to trade off energy-efficiency for increased network reliability under a certain network density.

4.1.3 Determining Priority Tagging Information Type

As mentioned in previous section, the type of priority tagging information needs to be determined, and its value be computed and enclosed before an interest packet can be forwarded. This section describes the algorithm used in choosing tagging information type. That is, a node has to decide whether the accumulated hop count or the remaining power resource should be chosen as the priority information. By the time the packet arrives at the downstream neighbors of the node, such priority information contained in the packet will be used to tag the established gradient towards this node, if the setup of this gradient is qualified. In *PRIMP*, a gradient tagged with **accumulated hop count** is considered as a **high priority**

---

[2] Paths with shorter hop count tend to be more energy efficient, besides they are most likely to be the paths with shorter delay.

```
Algorithm Gradient_setup_at_a_node (for each round of new interest that arrives at the node)
    if (priority information contained in the packet is accumulated hop count ) then
        if (number of cached gradients of the matched interest type < alpha) then
            if (interest packet from a neighbor that corresponds to a cached gradient) then
                update the timestamp of the cached gradient
            else    //interest from a new neighbor
                set up a new gradient and cache it
            end
        else    //number of cached gradients of the matched interest type is alpha
            if (interest packet from a neighbor that corresponds to a cached gradient) then
                update the timestamps and value of accumulated hop count
            else    //interest from a new neighbor
                if (value of accumulated hop count < value of the accumulated hop count of
                  some cached gradient) then
                    set up a new gradient corresponding to this new neighbor and
                    replace that cached gradient with this new gradient
                else
                    drop the received interest packet
                end
            end
        end
    else    // priority information contained in the packet is remaining power resource
            if (interest packet from a neighbor that corresponds to a cached gradient) then
                update the timestamp of the cached gradient
            else    //interest from a new neighbor
                set up a new gradient and cache it
            end
    end
end algorithm Gradient_setup_at_a_node (for each round of new interest that arrives at the node
```

Figure 4.5: Gradient setup algorithm

gradient, while a gradient tagged with **remaining power resource** is considered as of **low priority**.

Figure 4.6 demonstrates all possible cases in choosing priority information type. To decide which type of priority information should be computed and enclosed in a to-be forwarded packet at node A, both the energy level of A (marked by "+" or "−") and the priority tags of the cached gradients at A (marked by h or l) should be considered. Symbols "+" and "−" represent the node's residual power is either above or below the self-configured energy threshold, symbolized "*good*" and "*poor*", respectively. Symbols h and l stands for high and low priority respectively denoting the cached gradient towards an upstream neighbor ($\in UP\_N$). The arrows in Figure 4.6 represent the directions of gradients. UP_N denotes all the neighbors that correspond to the gradients cached at node A. As a part of the build-up of data paths, choosing priority information at each hop must take the energy resource condition of data paths into consideration. This is because the established braided data paths

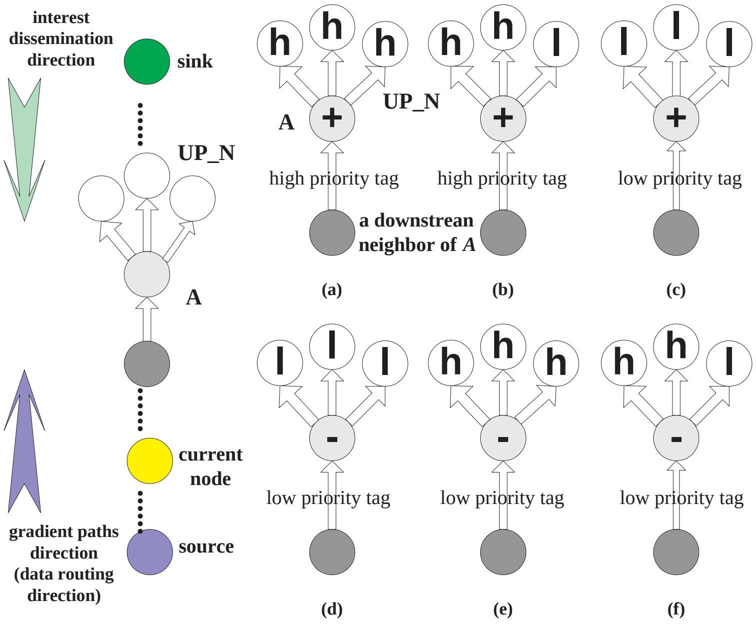


Figure 4.6: Choosing priority tagging information

will be used for delivering data traffic in the path selection stage. Thus, accumulated hop count will be chosen only if "energy-sufficient data paths" from *current node* (Figure 4.6) to the sink exist. In other words, at each hop of the data paths from current node to the sink (e.g. the hop from node A to UP_N), the energy level of the node (A) is good ("+"), and gradients tagged with accumulated hop count (h) are cached by the node. This is the case demonstrated in Figure 4.6 (a) and (b). When no such energy-sufficient paths exist, as in the cases shown in Figure 4.6 (c)(d)(e)(f), remaining power resource (l) will be chosen instead.

4.1.4 Computing Priority Tagging Information

For brevity, the term "gradient" hereafter in this thesis refers to the cached gradient corresponding to the the interest type of the received interest packet, i.e., the cached gradient of a matched interest type.

Accumulated hop count

At a node A (Figure 4.6), the value of accumulated hop count is denoted as $H(A)$ and is defined as follows:

$$H(A) = H^{'}(A) + 1 \tag{4.1}$$

$$H^{'}(A) = \frac{\sum_{\forall n \in HC\_set} H(n)}{N} \tag{4.2}$$

$H(n)$ is the value of the accumulated hop count tagging information of the cached gradient towards n; HC_set denotes the corresponding neighbor set of all the cached high priority gradients. N is the size of set HC_set. Since the number of the gradients allowed to be cached at each hop is limited ($\leq \alpha$), $N \leq \alpha$.

Remaining power resource

If the energy level of the node A is poor, it will first converts the values of the accumulated hop count tagging information of the cached high priority gradients (for the cases in Figure 4.6 (e)(f)) to an equivalent power value $R^{''}(A)$:

$$R^{''}(A) = H^{'} * Energy\_Thresh \tag{4.3}$$

Then the value of remaining power resource information is computed as follows:

$$R(A) = R_A + R^{'}(A) \tag{4.4}$$

$$R^{'}(A) = \frac{\sum_{(\forall k \in P\_set) \bigcup R^{''}(A)} R(k)}{M} \tag{4.5}$$

Here, $Energy\_Thresh$ is the self-configured energy threshold; R_A is the current energy level of node A; P_set denotes the corresponding neighbors of all the cached low priority gradients. M is the size of set P_set. If all the cached gradients are tagged with remaining power resource (Figure 4.6 (d)), then $R^{''}(A) = 0$.

The equations above show that remaining power resource, like accoumulated hop count, is also computed in an approach that reflects the accumulated effect of the energy resource condition of data paths. However, it needs to point out that the conversion from accumulated hop count to remaining power resource is necessary when computing remaining power resource. Because even if all the upstream neighbors of a sensor node are in good energy condition, the sensor node itself is low in power resource, the data paths via this sensor node should still not be encouraged to use in a preferable manner, as that will lead to the death of this node very soon. Therefore, remaining power resource metric should still be used instead of accumulated hop count under this circumstance. And one way of discouraging the data to flow through this sensor node is to convert all the cached accumulated hop counts into an average remaining power resource value, as shown in Equation 4.3.

There may exist other metrics that also reflect the energy resource condition of the data paths. The reason we choose accumulated hop count and remaining power resource is that they indicate the energy resource condition of the data paths within the different areas ahead of the current node towards the sink. Since these two metrics show the accumulated effects of the energy resource condition of the data paths from the sink to the current node, they help choose paths to route data packets in an energy-aware approach more wisely.

4.2 Priority-based Path Selection Stage

As illustrated in section 4.1.3, the accumulated hop count or remaining power resource tag value computed at a node indicates the energy resource condition of the data paths from this node to the sink. The proposed priority-based routing approach is based on this view. Here, the principle of path selection algorithm with a single-sink single-source scenario is demonstrated first. Then, the extensions to multi-sink and multi-source scenarios are presented.

As specified in Figure 4.6, after an interest initiated by a sink arrives at a source, matched data events are sent back immediately. For each data event to be delivered, several data paths have to be selected and and used simultaneously. At each hop, the selection of data paths can be interpreted as gradients selection. Gradients are selected based on the priorities of the cached gradients. In *PRIMP*, high priority gradients are preferred to low priority gradients, if both kinds exist in the cache; low priority gradients are used only when no high priority ones are cached.

4.2.1 High Priority Gradient Selection

If multiple high-priority gradients are cached by a node, the gradients with smaller accumulated hop count tag values (shorter paths) are preferred for energy-efficiency. Here, we choose the gradients by a probabilistic approach. Inspired by the probability assignment to forwarding entries in [29], a weight is assigned to each candidate gradient in our probabilistic selection strategy. The weight assignment to each cached gradient is based on its accumulated hop count tag value:

$$w_n^{hc} = \frac{[\frac{1}{H(n)}]^\beta}{\sum_{m \in HC\_set}[\frac{1}{H(m)}]^\beta} \qquad n \in HC\_set \tag{4.6}$$

From Equation 4.6, it can be inferred that the bigger the weight is, the more likely the corresponding gradient is used to forward data traffic. β is a weight factor (a design parameter) that decides extent to which the shorter paths should be favored. It should be adjusted according to network density.

4.2.2 Low Priority Gradient Selection

As mentioned in section 4.1.3, when this selection approach is used, there exists no energy-efficient paths from the current node (Figure 4.6) to the sink. Therefore, the scarce remaining power resource of the paths from the current node to the sink should be dealt with greater care. In other words, data paths with more residual power to the sink will be preferred to the paths with shorter lengths. Similarly, each gradient will be assigned a weight w_n^{rp} by the current node:

$$w_n^{rp} = \frac{R^\gamma(n)}{\sum_m R^\gamma(m)} \qquad m, n \in NBR\_SET \tag{4.7}$$

In the Equation 4.7, NBR_SET denotes the neighbor set corresponding to all the cached gradients. γ is a weight factor (design parameter) that indicates the sensitivity of routing decision to the energy resource conditions of distinct gradient paths. The choice of γ should be adjusted depending on the data traffic load, because different amount of traffic influences the energy resource condition of paths.

4.2.3 Gradient selection in Multi-sink Scenario

In multi-sink scenario, when a data packet arrives at a node, the node will route this packet along the cached gradients to all the sinks whose group ids exist in the matched interest entries in the cache. In such scenario, cached gradients may be tagged with multiple group ids, i.e., when paths from this node to multiple sinks exist. In this case, the following gradient selection approach will be employed: instead of selecting a certain number (η) of gradients (because multiple gradient paths should be selected at each hop) for each sink and sending η copies of a data event along them, the gradients that have been used for η times will not be used for the routing of this data event any more. For a certain sink, the selected η gradients must be distinct. The gradient selection will continue until no sink is missed out for η times. In this way, much fewer copies of data packets will be transmitted from the node, leading to great energy conservation.

As illustrated in Figure 4.7, five gradients toward upstream neighbors *UP_N_1*, . . . , *UP_N_5* are cached by node A. The different numbers within the colored boxes represent the group ids specifying different sinks (sink *1*, *2*, *3*, and *4*), respectively. Here we assume that two gradients ($\eta = 2$) are selected for each sink. For instance, let the gradients towards *UP_N_2* and the gradient toward *UP_N_3* be selected for sink *1*. By the time the gradients are selected for sink *2*, no work needs to be done at all, because sink *2* has already been allotted with two gradients unintentionally when the gradient paths were selected for sink *1*. Now gradient selection is conducted for sink *3*. Since sink *3* has already been allotted a gradient, when

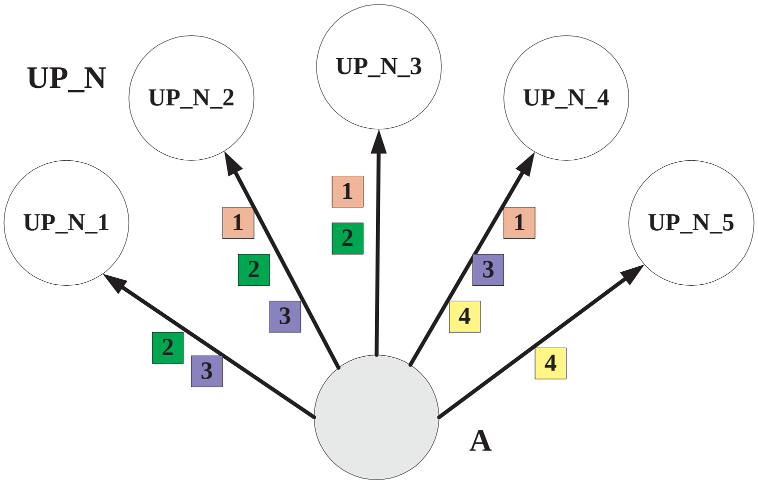


Figure 4.7: Gradient selection in multi-sink scenario

gradients were selected for sink *1*, only one more gradient needs to be selected. Suppose gradient toward *UP_N_1* is selected. Here, two gradients will be selected to deliver data traffic to sink *4*. Since no gradients have been selected for sink *4* yet, two distinct gradients are selected among the *unused* (times being selected $< \eta$) gradients (in this case, they are gradients toward *UP_N_1*, *UP_N_4* and *UP_N_5*) that are tagged by the specifier of sink *4*. Thus, gradients toward *UP_N_4* and *UP_N_5* are selected.

4.2.4 Data Aggregation of *PRIMP*

In *PRIMP*, if data events sent out from multiple sources, they will be suppressed if identical or aggregated if supplementary to each other at the intermediate nodes near the source region.

Besides the in-network processing of the data events from multiple sources, data aggregation is also important for energy conservation in the multi-path routing strategy employed in *PRIMP*. Since a data event is sent multiple times at each hop, the duplicate event copies that arrive later at an intermediate node will be dropped. Thus, redundant data traffic is effectively suppressed. The next chapter shows that data aggregation significantly helps in suppressing the duplicate data traffic, and makes our multi-path strategy achieve satisfactory energy-efficiency.

Chapter 5

Simulation and Analysis

In this chapter, the performance of *PRIMP* is evaluated and compared with that of *directed diffusion* and *flooding* through extensive simulations and analysis.

5.1 Performance Metrics

In our study, four metrics are employed to evaluate the performance of a routing protocol: **average dissipated energy**, **average forwarded data**, **distinct-data delivery ratio** and **application startup speed**.

- **Average dissipated energy** measures the average power consumption of a node for every distinct data event delivered to sinks. It is defined as the ratio of total dissipated energy per node to the total number of distinct data events successfully received by sinks, and it serves as an indicator of the lifetime of the network;

- **Average forwarded data** measures the capability of a routing protocol to distribute data traffic load. It is defined as the ratio of the total number of distinct data events relayed by a data-forwarding node to the number successfully received by sinks. It implicates the potential fault tolerance capability and the long-term energy-efficiency of a routing scheme;

- **Distinct-data delivery ratio** measures the reliability of packet transmission within a network. It is defined as the ratio of total number of the distinct data events successfully received by sinks to the number originally sent out by sources;
- **Application startup speed** records the number of distinct data events delivered to each sink after a sensor network application is launched. This metric tracks the impact of *slow startup* problem on the data collection at different sinks.

5.2 Methodology Employed In Simulation and Simulation Parameters

In our study, simulations are implemented in *ns-2* [30] simulator. The performance evaluations and analyses are conducted among three routing protocols: *PRIMP*, *directed diffusion* and *flooding*.

It is found that although sensor networks are assumed to be *densely* deployed, the interpretation of the word "densely" is rather vague, to our best knowledge. In different application scenarios, sensor networks may have various densities. Besides, even the densities in different regions within a network may not be same. Therefore, it would be meaningful to find out the impact of network density on the routing performance. In this study, routing performances under various specific network densities are evaluated to achieve this goal. Within a fixed 150x150 m^2 target area, 50 to 110 sensor nodes are randomly deployed in the increment of 20 nodes, representing each specific network density scenario.

The performance evaluation in our simulation is conducted under four-sink four-source scenarios with $\alpha = 3$, $\beta = 2$, $\gamma = 1$ and $\eta = 2$. All sources are located in a fixed small source region of 50x50 m^2 near the center of the target area. All sinks are uniformly scattered across the target area.

In our simulation, each sensor node is assumed to have a constant transmission range of 40 m. The energy model for sensor radio device is adopted such that the power consumption for transmission, reception and idle state is 660 mW, 395 mW, and 35 mW, respectively. At the sinks, 32-byte interest packets are initiated every 5 seconds; at the sources, 64-byte data events are generated every 0.5 seconds. For *directed diffusion*, 64-byte exploratory data events are also generated every 50 seconds. All these messages are transmitted at a constant rate of 1.6 Mbps.

5.3 MAC Dynamic Discussion

It is observed that MAC protocol employed in the protocol stack may influence the routing performance in various ways. For example, channel access given to two outgoing packets at two neighboring nodes may lead to collisions. This can affect *end-to-end* (i.e., sinks-to-sources) throughput. Likewise, the energy conservation strategy, which can be distinct in different MAC protocols influences the energy-efficiency evaluation of a routing protocol differently.

Therefore, to minimize the variations on routing performance evaluation resulting from MAC layer interaction, we firstly adopts a modified version of IEEE 802.11, with no idle power conservation strategy introduced. In this modified version of 1.6 Mbps 802.11 MAC, the realistic sensor network radio parameters are used to evaluate the routing performances. [31] The reason for not applying energy conservation in MAC is that we would like to obtain the most conservative evaluations on the advantages that can be achieved by *PRIMP*.

However, 802.11 which is a typical contention-based MAC protocol, is not completely satisfactory, because contention-based channel access is deemed unsuitable for sensor networks due to the requirement to monitor the channel at all times [15]. That is, the key problem for contention-based MAC protocol in sensor networks — energy inefficiency, needs to be attended. Therefore, in our simulation, routing performances are re-evaluated with idle power dissipation in MAC being suppressed. Through such energy conservation strategy, we demonstrate the impact of energy conservation in MAC layer on the energy-efficiency performance measurement. It is also reported in [15] that IEEE 802.11 standard for WLANs support further power conservation by switching off radio depending on NAV status. With this technique, energy consumed in packet overhearing can be further saved.

As mentioned above, by suppressing the idle power dissipation in MAC layer, we tend to measure the most conservative advantage of *PRIMP* over other comparable routing protocols. The reason lies: aiming to minimize all possible transmission overheads, the duty cycle of the radio in *PRIMP* is smaller than that in other routing protocols. Therefore, at most comparable amount of idle power is dissipated in MAC layer when *PRIMP* is used, compared to the cases when *directed diffusion* or *flooding* is used. We validate that even with this comparable but unnegligible amount of energy consumption accounted in the metric evaluation, *PRIMP* still performs noticeably better. Thus, the advantage of *PRIMP* will reasonably be even more evident when MAC protocol is completely energy-efficient.

5.4 Simulation Results

Figure 5.1 shows the energy-efficiency comparison among *PRIMP*, *directed diffusion* and *flooding* when no energy conservation strategy is adopted in MAC protocol. It is observed that *directed diffusion* performs much better than *flooding*, the bench-mark scheme. The dissipated energy of *directed diffusion* is only 45% – 67% that of *flooding*, varying with the network density. Unlike *directed diffusion*, the average dissipated energy of *PRIMP* is quite *insensitive* to network density, as shown in Figure 5.1. Even with two data paths ($\eta = 2$) used for carrying data traffic simultaneously, *PRIMP* still outperforms *directed diffusion* by 20% – 60% in energy-efficiency. This is mainly due to the updating and maintaining ap-

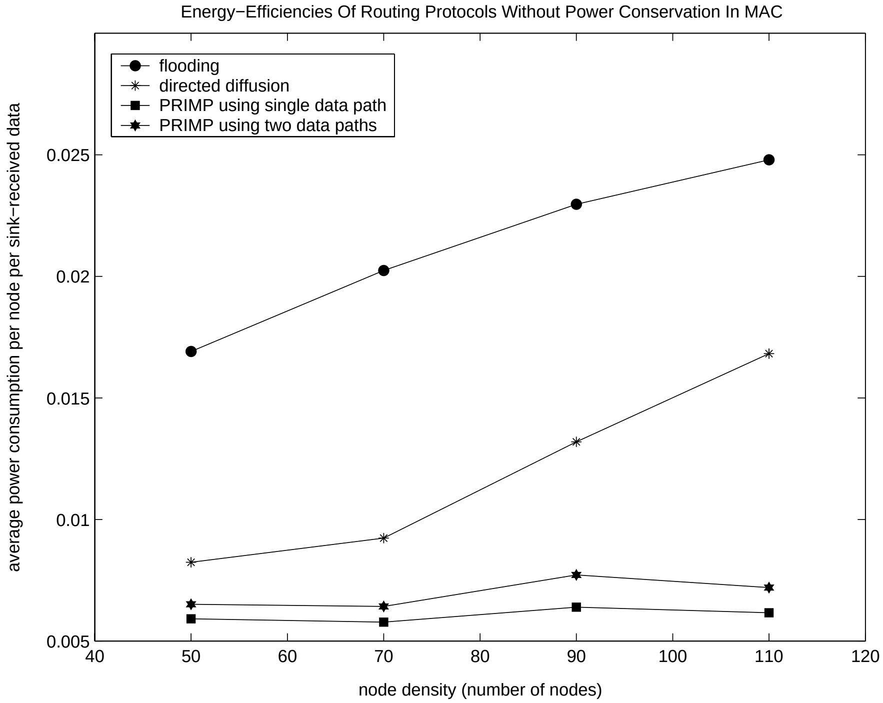


Figure 5.1: Energy-efficiency measurement with no power conservation in MAC

proaches adopted by *PRIMP*. In *directed diffusion*, interest messages are periodically flooded throughout the network to keep the data paths from source region to sinks alive and updated. *PRIMP*, however, only reactively updates data paths invoked by complete path failure or updated knowledge of source region, and directionally maintain the gradient paths from source region to sinks. Therefore, transmission and overhearing of interest packets are significantly reduced. The intermediate node suppression of the identical interests initiated from different sinks also contributes to the energy-efficiency of *PRIMP*. It is also observed that energy-efficiency is not worsen too much when multiple data paths are used in *PRIMP* to enhance the transmission reliability. As Figure 5.1 shows, only $10\% - 20\%$ more energy is consumed when two paths are used, compared to the single path strategy. This is because data events are only carried by the cached gradient paths at data-forwarding nodes, and the number of the cached gradient paths at a node is limited ($\leq \alpha$). Although each data-forwarding node uses two cached gradient paths to forward a received data event simultaneously, mostly

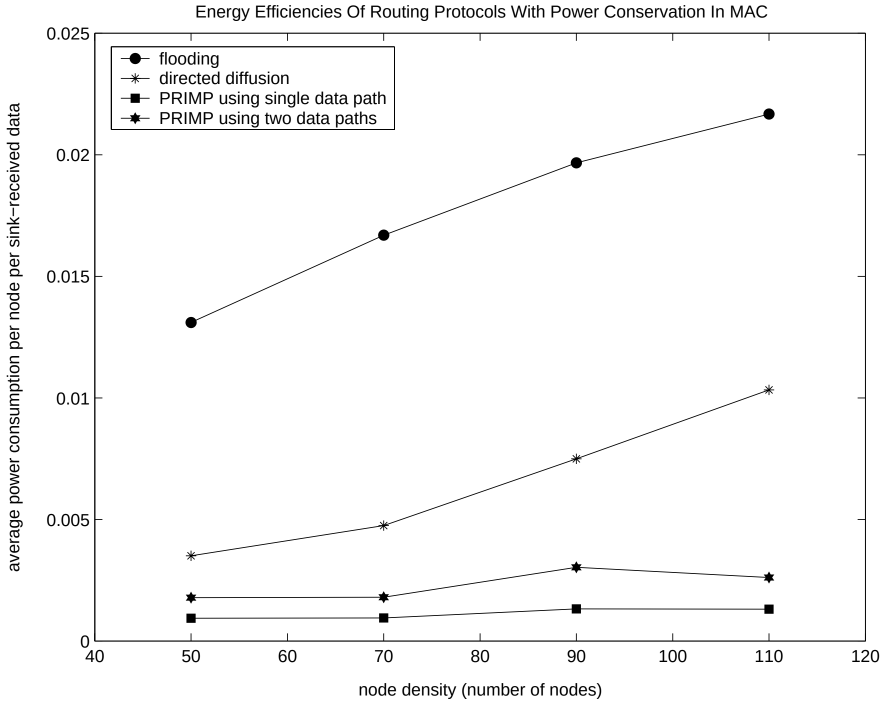


Figure 5.2: Energy-efficiency measurement with idle power conservation in MAC

duplicate data events transmitted at nodes will be suppressed within short period when they go upstream toward the sinks.

When energy conservation strategy is employed in MAC protocol to suppress the idle power dissipation, the advantage of *PRIMP* in energy conservation is more evident. As shown in Figure 5.2, the energy consumption of the three routing protocols are all greatly decreased. By the comparison between Figure 5.1 and 5.2, it is observed that comparable amount of idle power is dissipated for all three protocols. However, the influence of this power conservation in MAC protocol is unnegligible: with comparable amount of idle power dissipation being conserved, *PRIMP* using two paths outperforms *directed diffusion* by 2–4 times. That is, the idle power dissipation in MAC layer actually makes the advantage of *PRIMP* in energy-efficiency less obvious. Therefore, the evaluations given in Figure 5.1 tends to demonstrate the energy efficiency advantage of *PPRIMP* rather conservatively.

Figure 5.3 shows the load balancing capability of three routing protocols. Due to the avail-

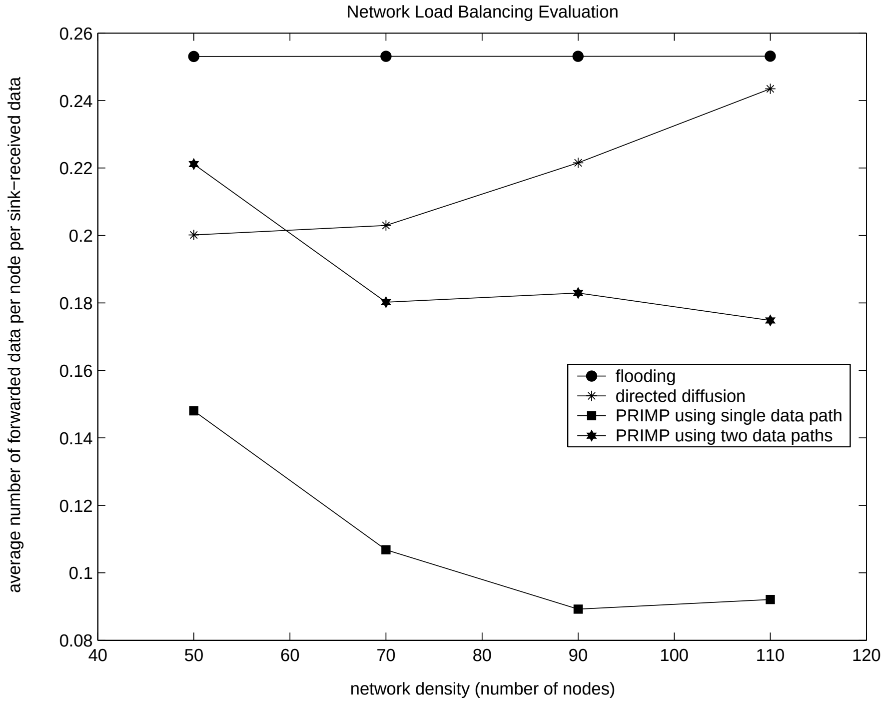


Figure 5.3: Load-balancing capability of different routing protocols

ability of multiple data paths, *PRIMP* performs noticeably better than *directed diffusion* in distributing traffic load. With *directed diffusion*, each sink only aims to set up one shortest delay path to draw data from the sources to that sink for each round of path exploration. This data path will be constantly reinforced and used if there is no MAC dynamics or transmission blocks. With MAC dynamics, as is the case we simulated, *directed diffusion* is able to deliver data traffic through multiple reinforced paths that are established in multiple rounds of path exploration. However, the reliance on the MAC dynamics is proved to be rather trivial (as will be shown later in this section). Therefore, as a routing protocol, *directed diffusion* did little efforts in distributing the traffic load in the network layer. Thus the potential danger from the overuse of shortest delay paths is high. In comparison, in *PRIMP*, multiple gradients ($\eta = 2$ for $\alpha = 3$ in simulation) are selected at each hop of data paths. By carefully selecting the gradient path candidates, data traffic are balanced over a lot more nodes without compromising the energy-efficiency severely, as previously demonstrated in Figure 5.1 and 5.2. Figure 5.3 shows that when single path is used in *PRIMP*, *PRIMP* performs more than two times better in balancing the traffic load in most of the network density scenarios. It is noticed that when multiple data paths are employed to deliver data events, the load balancing of *PRIMP is not* worsen *actually*, though seemingly it is as appeared in Figure 5.3. The reason is that in the multi-path strategy in *PRIMP*, each data-forwarding node only forwards more duplicated data events. It is also observed in Figure 5.3 that the ability of *directed diffusion* to balance the traffic load becomes less evident, *if there is any*, when network density increases; on the contrary, *PRIMP* shows increasingly better performance in distributing data traffic ("PRIMP using single data path" curve in Figure 5.3). The reason behind is quite interesting: with the increase of network density, the delay characteristic of all data paths from the source region to a sink becomes less distinct, i.e., all the data paths will have potentially similar congestion possibility. MAC dynamic is therefore less influential to the path reinforcement in *directed diffusion*. This explains why the load balancing performance of *directed diffusion* becomes worse when network density increases. For *PRIMP*, a high network density only helps build more data paths with shorter lengths. Thus the energy-efficiency performance can be further enhanced.

Figure 5.4 shows the measurement on distinct-data delivery ratio performed by three routing protocols in face of node failures. We realize that distinct-data delivery ratio is directly related to the number and frequency of node failures happened in a sensor network. In our study, we deliberately turn off some nodes on the shortest path between sinks and sources. The intent is to create node failures in the paths that are mostly likely used by both *PRIMP* and *directed diffusion*. The sensor nodes are turned off repeatedly for 20 seconds.

For *directed diffusion*, since the path exploration activity initiated by sources is infrequent due to its energy-consuming nature, new path establishment between a sink and a source only works when failed nodes are detected in a path exploration activity. However, in many cases, sensors are not permanently dead; rather, temporary transmission blockage happens much more frequently due to dynamic environmental conditions. This causes transmission

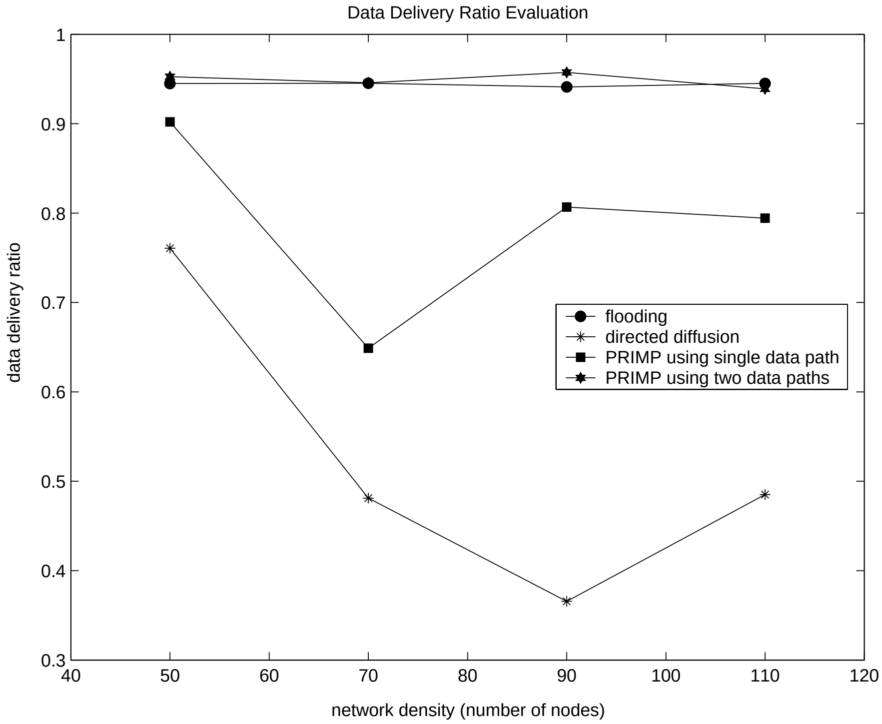
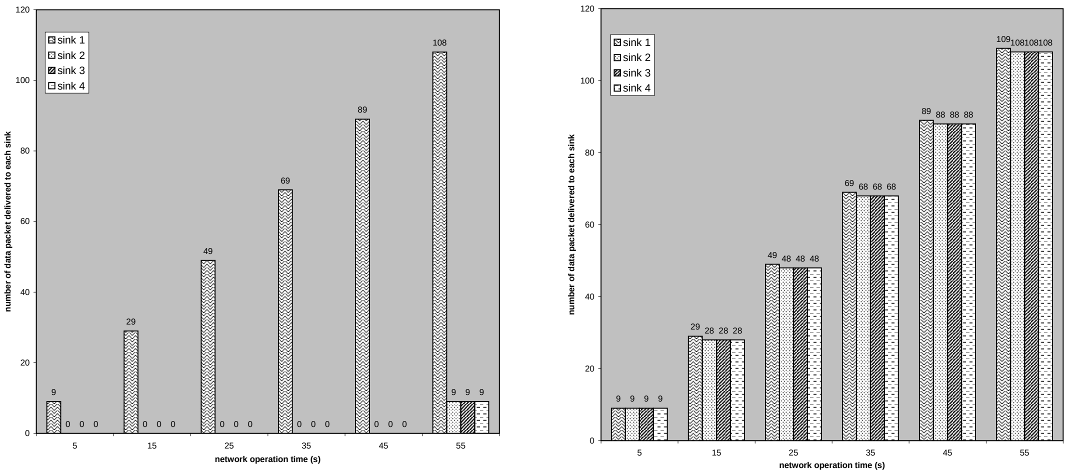


Figure 5.4: Distinct-data delivery ratio of different routing protocols

uncertainties and dynamics within sensor networks important issues to be handled. Under this context, path exploration in *directed diffusion* may fail as long as node failures happen before the occurrence of a path exploration. As shown in Figure 5.4, when the node density is relatively low, the data delivery ratio achieved by *directed diffusion* is high due to the existence of alternative paths. However, in high density scenarios, the delivery ratio is very low. In contrast, *PRIMP* performs much better in delivering data events. Figure 5.4 shows that even if only one data path is used, it still outperforms *directed diffusion* by at least 18% in data traffic delivery. In such case, however, whenever data events arrive at failed nodes, they are dropped. Though *PRIMP* periodically maintains data paths from source region to sinks through directional interest dissemination, the temporary and frequent node failures still leads to an unsatisfactory delivery ratio compared to the case when multi-path ($\eta = 2$) strategy is used in *PRIMP*. As shown in Figure 5.4, the use of multiple paths significantly improves the transmission reliability. Since there are always more than one gradient paths carrying



(a) *Directed Diffusion* (b) *PRIMP*

Figure 5.5: Impact of slow startup problem on the data collection at different sinks

the same data even when a data event is lost at failed nodes, the overall transmission of data events is necessarily affected. Thus, a constantly high data delivery ratio is achieved in *PRIMP* through multi-path delivery strategy. For *flooding*, since broadcasting is used at each single hop for data delivery, it still proves to be the most robust routing scheme, if energy-efficiency is not a concern.

In this thesis, a short introduction to the *slow startup* problem has been given in section 3.2.2. Here, the impact of the slow startup problem on the data collection at different sinks is evaluated under a four-sink-four-source scenario in *directed diffusion* and *PRIMP* .

Figure 5.5 demonstrates the number of data events delivered to each sink within a time period after the launch of a sensor network application. As shown in Figure 5.5(a), for *directed diffusion*, no data events are received by sinks 2, 3 and 4 within a long period of time after the application is launched. This shows that the application starts up very slowly when *directed diffusion* is employed. When *PRIMP* is used (5.5(b)), nearly the same number of data events (packet numbers above the bars in Figure 5.5) are delivered to each sink within the same time period. That is, the four sinks begin to retrieve data events almost simultaneously and instantly. In *directed diffusion*, *slow startup* problem occurs because the information about sinks is transparent to sources. In other words, the early arrival of the interest message from sink 1 invokes the exploration data to be propagated before the gradient paths to sink 2, 3, and 4 are established. As shown in the figure, this situation lasts until the next round of path exploration (after a path exploration cycle, 50 seconds in simulation) is invoked at the sources. In *PRIMP*, the *slow startup* problem is prevented with the aid of *group id* tagging information. That is, sources rely on the interests (each contains a *group id* specifying its initiator) from different sinks respectively and directly (no exploratory data is needed). Different sinks therefore do not influence their respective data collection.

5.5 Design Parameters Discussion

Last, it is worthwhile to mention that complexity of a routing protocol is also an important issue, due to the limitation in computational capability and memory capacity of sensor nodes. In *PRIMP*, complexity is also related to the design parameters α, β, γ and η.

Although Moore's law predicts that hardware for sensor networks will inexorably become smaller, cheaper, and more powerful, technological advances will never prevent the need to make tradeoffs. *PRIMP* is proposed to achieve the performance enhancement in energy efficiency, fault tolerance and load balancing capabilities, while at the same time allow some tradeoffs in local computation and storage. Such motivation in *PRIMP* lies: In most cases, sensor node can only afford a limited energy in battery. And transmitting or receiving a bit wirelessly is much more expensive than processing a bit in local CPU [33]. According to the example described in [34], the energy cost of transmitting 1 KB a distance of 100 m is approximately the same as that for executing 3 million instructions by a 100 million instructions per second (MIPS)/W processor, assuming Rayleigh fading and fourth power distance loss. Hence, local data processing is crucial in minimizing power consumption in multi-hop sensor network, and it is much beneficial to take advantage of the higher computational power in smaller and smaller processors, with the understanding that the processing unit of a sensor is still a scarce resource. Since computational complexity of *PRIMP* is also related to the design parameters α, β, γ and η, these parameters should be carefully tuned so that complex computation will not be incurred. In *PRIMP*, for every incoming interest packet, only simple geo-location coordinates comparisons are needed (Figure 4.2, Figure 4.3); for each incoming data event, α, β, γ and η are set to be small to avoid complex computation (Equations 4.1 - 4.7).

Storage complexity issue in *PRIMP* is also carefully handled. A typical cubic-centimeter battery stores about 1,000 milliamp-hours, so centimeter-scale devices can run almost indefinitely in many environments. However, low-power microprocessors have limited storage, typically less than 10 Kbytes of RAM for data and less than 100 Kbytes of ROM for program storage, about 10,000 times less storage capacity than a PC has. This limited amount of memory consumes most of the chip area and much of the power budget. Designers typically incorporate larger amounts of flash storage, perhaps a megabyte, on a separate chip. For example, Berkley smart dust note prototype contains a microcontroller with 8KB instruction flash memory, 512 bytes RAM and 512 bytes EEPROM [35]. Off-chip flash memory provides storage to hold both the program while it transfers through the network and the data buffering beyond the on-chip RAM. Compared to *directed diffusion*, which have already been implemented and ported to multiple platforms including WINSng 2.0 nodes, USC/ISI PC/104 nodes, and Motes, *PRIMP* does not significantly increase the storage burden of sensor nodes: only a few dozen of extra bytes are needed to store the information of the cached gradients towards its corresponding upstream neighbors, whose number is still dependent on the tunable design parameters.

Chapter 6

Conclusions and Future Works

In this thesis, a new routing protocol *PRIMP* is proposed to address the key issues in sensor networks — stringent energy constraint and network fault tolerance capability, as well as the slow startup problem that occurs in other data-centric routing schemes. Based on the characteristics of communication architecture and the unique system features of sensor networks, *PRIMP* achieves its design goals in the approaches highlighted as follows:

(a.) Invoked by the updated information of sources or the failures of the current data paths from sources to a certain sink, on-demand virtual source technique is employed reactively to re-explore the multiple braided data paths from sources to sinks. This novel technique greatly suppresses the overhead from interest dissemination;

(b.) Each sink directionally maintains the data paths from sources to it through periodic dissemination of interest messages, after the knowledge of source region is obtained. Together with the on-demand virtual source technique, such directional interest dissemination not only makes *PRIMP* energy-efficient, but also provides network fault tolerance robustness against the transmission unreliability in sensor networks;

(c.) After data paths from sources to sinks are established in the interest dissemination stage, multiple paths will be selected probabilistically in a priority-based approach to

> route data traffic. The routing decision is made by data-forwarding nodes at every hop of the data paths. That is, gradients are selected at each hop based on the energy resource conditions of the paths led by the current node to a sink. The multiple gradient paths selected at each hop will be used to deliver data traffic simultaneously.

In this study, extensive simulations are implemented to validate the performance advantages of *PRIMP* over other comparable routing schemes. In the performance evaluations, the influence of network density is originally explored through quantitative discussions on the "high density" of sensor networks.

PRIMP is proved to be capable of extending the network lifetime significantly and providing noticeable better transmission reliability. Moreover, for time-sensitive sensor network applications, *PRIMP* addresses effectively the slow startup problem encountered by *directed diffusion*.

We also render an interesting clue for the adoption of CSMA-based MAC protocols in sensor networks. It is found that a high network density seems to entitle the different paths stretching out from sinks with a more *isotropic* delay characteristic, while the delays on such paths tend to be more *anisotropic* if the target area is more scarcely-deployed.

We have evaluated the performance of *PRIMP* under the scenarios of different network densities and fixed network size, and validates that *PRIMP* scales well network density. Our future work is oriented toward the scalability study under the scenarios of changing network size and fixed network density. In this study, design parameters of *PRIMP* are set such that $\alpha = 3$, $\beta = 2$, $\gamma = 1$ and $\eta = 2$. However, it is observed that the performance of *PRIMP* is a function of α, β, γ and η. The influence of these parameters will be further explored so that the best performance can be achieved in different scenarios by tuning them carefully.

Bibliography

[1] Chih-fan Hsin, Mingyan Liu. “A distributed monitoring mechanism for wireless sensor networks”. In *Proceedings of the ACM workshop on Wireless security*, Atlanta, GA, USA, 2002.

[2] John Heidemann, Fabio Silva, Shalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, Deepak Ganesan. “Building Efficient Wireless Sensor Networks with Low-Level Naming”. In *Proceedings of the 18th ACM Symposium on Operating System Principles*, 2001.

[3] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden. “A survey of active network research”. *IEEE Communications Magazine*, 35(1): 80–86, Jan. 1997.

[4] E. Amir, S. McCanne, and R. H. Katz. “An active service framework and its application to real-time multimedia transcoding”. In *Proceedings of the ACM SIGCOMM Conference*, pages 178–189, Vancouver, Canada, Sept. 1998. ACM.

[5] S. Michel, K. Nguyen, A. Rosenstein, L. Zhang, S. Floyd, and V. Jacobson. “Adaptive web caching: Towards a new global caching architecture”. In *Proceedings of the 3rd International World Wide Web Conference*, Manchester, England, June 1998.

[6] B. Warneke, M. Last, B. Leibowitz, K.S.J. Pister. "Smart Dust: Communicating with a Cubic-Millimeter Computer", *Computer Magazine*, Jan. 2001. IEEE, Piscataway, NJ. pp. 44–51.

[7] G. Asada, M. Dong, T.S. Lin, F. Newberg, G. Pottie, W.J. Kaiser, "Wireless Integrated Network Sensors: Low Power Systems on a Chip", *Proceedings of the 24th IEEE European Solid-State Circuits Conference* (Den Hague, The Netherlands, Sept. 21–25). Elsevier, 1998, 9–12.

[8] S. Cho, A. Chandrakasan, "Energy-efficient protocols for low duty cycle wireless microsensor", *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Maui, HI Vol.2(2000), p.10.

[9] N. Bulusu, D. Estrin, L.Girod, J. Heidemann, "Scalable coordination for wireless sensor networks: self–configuring localization systems", *International Symposium on Communication Theory and Applications (ISCTA 2001)*, Ambleside, UK. July 2001.

[10] Shenck, N.S., Paradiso, J.A., "Energy Scavenging with Shoe-mounted piezoelectrics", *Micro, IEEE*, Volume: 21, Issue: 3, pp. 30–42, May–June, 2001.

[11] Rabaey, J.M.; Ammer, L.; Karalar, T; Suetfei Li; Otis, B.; Sheets, M.; Tuan, T., "PicoRadios for wireless sensor networks: the next challenge in ultra-low power design", *Solid-State Circuits Conference, 2002. Digest of Technical Papers.* pp. 200–201, *ISSCC. 2002 IEEE International*, Volume: 1, 2002.

[12] Jan M. Rabaey; M. Losie Ammer; Lulio L. Da; Silva Jr. Danny Patel; Shad Roundy, "PicoRadio Support Ad Hoc Ultra-Low Power Wireless Networking", *2000 issueof Computer Magazine*, pp. 42–48, July 2000.

[13] D. B. Johnson and David A. Maltz, "Dynamic source routing in ad hoc wireless networks", *Mobile Computing*, Kluwer, 1996, pp. 153–181.

[14] C. Perkins and E. Royer, “Ad hoc On demand distance vector routing”, P̂roc. 2^{nd} IEEE Wksp Mobile Comp. Sys. & Apps., Feb 1999.

[15] I.F. Akyildiz, W. Su, Y. Sankarasubramamiam and E. Cayirci, “Wireless Sensor Networks: A Survey”, *Computer Networks*, Vol. 38, No. 4, pp. 393–422, Match 2002.

[16] LAN MAN Standards Committee of the IEEE Computer Society, *Wireless LAN medium access control (MAC) and physical layer (PHY) specification*, IEEE, New York, NY, USA, IEEE Std 802.11–1997 edition, 1997.

[17] A. Woo, D. Culler, “A transmission control scheme for media access in sensor networks”, *Proceedings of ACM MobiCom 01*, Rome, Italy, July 2001, pp. 221 235.

[18] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie, “Protocols for self-organization of a wireless sensor network”, *IEEE Personal Communications*, October 2000, pp. 16 27.

[19] S. Singh and C.S. Raghavendra, “PAMAS: Power aware multi-access protocol with signalling for ad hoc networks”, *ACM Computer Communication Review*, vol. 28, no. 3, pp. 5 26, July 1998.

[20] Wei Ye; Heidemann, J.; Estrin, D. “An energy-efficient MAC protocol for wireless sensor networks”, *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Volume: 3 , 2002, pp. 1567–1576 vol. 3.

[21] P. Karn, “MACA — a New Channel Access Method for Packet Radio”, in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–140, 1990.

[22] W.R. Heinzelman, J. Kulik, H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks”, *Proceedings of the ACM MobiCom 99*, Seattle, Washington, 1999, pp. 174–185.

[23] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", IEEE Proceedings of the Hawaii International Conference on System Sciences, January 2000, pp. 1–10.

[24] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva, "Directed Diffusion for Wireless Sensor Networking", *ACM/IEEE Transactions on Networking, 11* (1), pp. 2–16, February, 2002.

[25] Yan Yu, Ramesh Govindan and Deborah Estrin. "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks". *UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023*, May 2001.

[26] Xiaoyan Hong; Gerla, M.; Hanbiao Wang; Clare, L., "Load balanced, energy-aware communications for Mars sensor networks", *Aerospace Conference proceedings, 2002. IEEE*, Volume: 3 , 2002, Page(s): 3–1115 vol. 3.

[27] A. Savvides, C. C. Han, M. B. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors", *In Proceedings of MOBICOM 01*, 2001.

[28] Nirupama Bulusu, John Heidemann, and Deborah Estrin. "GPS-less low cost outdoor localization for every small devices". *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.

[29] Rahul C. Shah and Jan Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", *IEEE Wireless Communications and Networking Conference (WCNC)*, March 17–21, 2002, Orlando, FL.

[30] S. Bajij, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Haldar, M. Handley, A. Helmy, J. Heidemann, P. Huang, S. Kumar, S. McCanne, R. Rejaie, P. Sharma, K. Varadhan, Y. Xu, H. Yu, and D. Zappala, "Improving simulation for network research", Univ. Southern California, Los Angeles, Tech. Rep. 99–702b, 1999.

[31] W. J. Kaiser, "WINS NG 1.0 transceiver power dissipation specifications", *Sensoria Corporation*, San Diego, CA.

[32] *Information processing systems - Open Systems Interconnection - Basic Encoding Rules for Abstract Syntax Notation One* (ASN. 1). International Standard, no. 8825. International Organization for Standards, May 1987.

[33] V. Raghunathan, C. Schurgers, S. Park, and M.B. Srivastava, "Energy-Aware Wireless Microsensor Networks", *IEEE Signal Processing Magazine*, vol.19, no.2, March 2002, pp.40-50.

[34] G.J. Pottie, W.J. Kaiser, "Wireless integrated network sensors", *Communications of the ACM*, 43(5):551-558, May 2000.

[35] A. Perrig, R. Szewczyk, V. Wen, D. Culler, J.D. Tygar, "SPINS: security protocols for sensor networks", *Proceedings of ACM MobiCom'01*, Rome, Italy, 2001, pp. 189-199.