# PORT: A Price-Oriented Reliable Transport Protocol for Wireless Sensor Networks

Yangfan Zhou and Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong, China
{yfzhou, lyu}@cse.cuhk.edu.hk

Jiangchuan Liu
School of Computing Science
Simon Fraser University
Vancouver, Canada
jcliu@cs.sfu.ca

Hui Wang
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong, China
hwang3@ie.cuhk.edu.hk

## Abstract

*In wireless sensor networks, to obtain reliability and minimize energy consumption, a dynamic rate-control and congestion-avoidance transport scheme is very important. We notice that reporting packets may contribute to the sink's fidelity of its knowledge on the phenomenon of interest to different extents. Thus, reliability cannot simply be measured by the sink's total incoming packet rate as considered in current schemes. Also, communication costs between sources and the sink may be different and may change dynamically. Based on these considerations, we propose PORT (Price-Oriented Reliable Transport protocol) to facilitate the sink to achieve reliability. Under the constraint that the sink must obtain enough fidelity for reliability purpose, PORT minimizes energy consumption with two schemes. One is based on the sink's application-based optimization approach that feeds back the optimal reporting rates. The other is a locally optimal routing scheme according to the feedback of downstream communication conditions. PORT can adapt well to the communication conditions for energy saving while maintaining the necessary level of reliability. Simulation results in an application case study demonstrate the effectiveness of PORT.*

## 1. Introduction

In wireless sensor networks (WSNs) [2, 6, 13], a number of in-situ sensor nodes are deployed to collect data about some physical phenomena of interest. The sensor nodes form an ad hoc multi-hops wireless network through which the collected data are conveyed to the sink nodes. There are a great variety of proposed applications of WSNs such as environmental monitoring, object tracking, surveillance, *etc*. [2, 12]. Although different WSNs have different task-specific requirements, they all require a sensor-to-sink data transport scheme[1] to take account of two important issues. The first is reliability assurance, which means we must guarantee that the sink can obtain enough information about the phenomenon of interest. The second is energy-efficiency, as recharging the sensor nodes is usually impractical [7]. Therefore, a sensor-to-sink data transport scheme should aim to minimize energy consumption under the constraint that the sink can collect enough information on the phenomenon of interest.

The notion of reliability on sensor-to-sink communication was first introduced in [14], where the authors notice that, unlike existing WSN transport schemes (*e.g.*, PSFQ [18] and RMST [17]) that focus on end-to-end reliable data transferring, absolute end-to-end reliable data transport is usually not needed when transmitting sensor reporting packets. Packet loss within a certain limit can usually be well tolerated in most application scenarios. This notion is important to the design of a reliable sensor-to-sink data transport protocol; however, there are still several unconsidered problems in current approaches.

First, we notice that the packets from different sources may make a different contribution to improve the sink's information on the phenomenon of interest. We regard the contribution of a source node as being how much it reduces the sink's uncertainty on the data about the phenomenon.

[1] The sensor-to-sink data transport scheme refers to the data transport scheme that transfer the desired information collected by the in-situ sensors to the sink.

Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)
1071-9458/05 $20.00 © 2005 IEEE

Thus reliability cannot simply be measured by the total incoming packet rate, as considered in current approaches, *e.g.*, ESRT (Event-to-Sink Reliable Transport) [14]. Instead, it should be assured with the cooperation of a reliable sensor-to-sink data transport scheme and network applications.

Second, to achieve reliability, ESRT adjusts the report rates of sources in an undifferentiated manner. But, as the communication cost from different sources to the sink may be different and may change dynamically, and also the contributions of packets from different sources are also different, adjusting the report rates of the sensor nodes in an undifferentiated manner is not the most energy-efficient way to increase the knowledge of the phenomenon. It is therefore necessary to bias the reporting rates of the sources.

Third, to minimize energy consumption, we must avoid links with high communication costs. Congestion always results in an increase in communication cost, and so congestion control is vital to minimize energy consumption. ESRT proposes to avoid congestion in an end-to-end manner by reducing reporting rates. CODA [19] also proposes to avoid congestion by slowing down sending rates. However, slowing down sending rates may cause the sink to receive fewer packets, which may yield in insufficient information on the phenomenon of interest. In this case, the sink will ask for higher reporting rates and that may cause congestion again if a reliability control mechanism like what ESRT proposes is employed. Therefore, besides an end-to-end congestion-avoidance mechanism, an in-network congestion-avoidance mechanism is also necessary.

In this paper, we aim to address these problems by providing a Price-Oriented Reliable Transport protocol (PORT). PORT is based on the following assumptions.

- The sensor reporting traffic lasts for a considerable duration.
- The sink is aware of the sources of the data packets; *i.e.*, the sink can identify where a packet originates.
- The sink is aware of the information a packet carries.

The first assumption means source sensor nodes would keep reporting data on the phenomenon of interest for a long period of time. It is generally valid because in most application scenarios such as environmental monitoring, object tracking, surveillance, *etc.*, WSNs are employed to provide continuous data streaming about the phenomenon of interest.

The second assumption is also reasonable in most application scenarios. This is because of two reasons. First, it is usually necessary for the sink to know the physical location of the phenomenon. Where a packet originates provides information on where the phenomenon of interest is taking place. Second, the sink should usually fuse the data packets it has received. Each source node should be identified in order to provide information on how to fuse the packets. Note that PORT does not require a heavy-weighted address-based approach. It only requires the sink can identify different sources which are reporting data on the same phenomenon. This can be achieved, for example, by randomly generating an identifier and embedding it in reporting packets when a node is sensing and reporting the phenomenon of interest.

The third assumption means that the sink knows how a packet can improve its knowledge on the phenomenon of interest. It is true as the sink is where data packets are interpreted.

PORT employs *node price*, which is defined as the total number of transmission attempts across the network needed to achieve successful packet delivery from a node [2] to the sink, to measure the communication cost from a node to the sink.

Under the constraint that the sink must obtain enough information, PORT dynamically feeds back the optimal reporting rate to each source according to the current contribution of the packets from each source and the node price of each source.

Based on the neighboring nodes' feedback of their node prices and the loss rates of the links between the neighbors and the node, an in-network node dynamically allocates its outgoing traffic to avoid high loss rate paths (which are probably caused by congestion). PORT, in this way, alleviate congestion in an in-network manner. Also, congestion will increase the node price of the sources. The source reporting rate control mechanism of PORT is aware of node prices of the sources, and can decide to adjust the source reporting rates (it might slow down one with a high node price and speed up one with a low node price) with a guarantee that the sink can still obtain enough information. Hence, with this in-network congestion-avoidance mechanism and this end-to-end reporting-rate adjustment mechanism, PORT provides a good congestion-avoidance mechanism.

The rest of this paper is organized as follows. Section 2 briefly surveys the related work. Section 3 discusses the requirements that a reliable sensor-to-sink data transport scheme should fulfill. In Section 4, we provide the design considerations of PORT. In Section 5, we elaborate the implementation of PORT. Section 6 evaluates our mechanism with NS-2 in an application case. We conclude the paper in Section 7.

## 2. Related Work

In traditional TCP/IP networks, data transport mechanism is implemented with an address-based end-to-end data communication concept. But, this is not appropriate for WSNs not only because of the need to simplify the implementation, but also to save energy [11]. This consideration

[2] Successful packet delivery from a node means that the packet from the node arrives at the sink successfully.

has led to the development of a well-received concept: data-centric communication [1, 11].

Data-centric schemes deliver sensor data throughout the network with an application specific naming scheme for the data. Routing paths are constructed on-demand based on the specific task and the data packets are routed according to the data they carry. A representative example of data-centric routing is directed diffusion [11]. In directed diffusion, the sink requests its data of interest by broadcasting its *interests*. The interest packets are flooded throughout the network and the nodes set up *gradients* to save the data-centric routing information. Using the gradient filter, directed diffusion conveys sensor reporting data to the sink. The reporting packets might be delivered to the sink along multiple paths. The sink determines the best path and increases the desired reporting rate along this path. This process is called path *reinforcement*. Directed diffusion can periodically reinitiate the reinforcement process to find the new best path. Many other schemes (*e.g.*, [4, 5, 10, 15, 16]) have been proposed based on directed diffusion or similar concepts (work in [1] provides a good survey on routing protocols for WSNs). But, most of the work does not consider the notion of *reliability* of sensor-to-sink communication within the design of the protocols.

PSFQ [18] and RMST [17] are concerned with reliable data transport protocol over WSNs. However, they aim at providing 100% reliable data transport for WSNs. In [14], the authors argue that absolute end-to-end reliable data transport is usually not needed for transmitting the sensor reporting packets. They propose ESRT (Event-to-Sink Reliable Transport) to address the reliable sensor-to-sink communication problem. They measure the reliability of the event features achieved in terms of total packet receiving rate. The communication is considered to be reliable if the number of the received packets is not less than the desired number of packets per unit time. ESRT ensures that the total incoming packet rate of the sink stays within the desired range by providing a mechanism to feed back the required reporting rate directly to the source nodes. But the reporting rate of each source is adjusted in an unbiased manner.

Congestion control for WSNs is studied in [14] and [19]. In [19], congestion is detected by sampling wireless channel utilization. In [14], congestion is detected according to the buffer utilization of the in-network nodes. These studies avoid congestion by slowing down the sending rate, regardless of what the node is reporting.

## 3. Protocol Requirements

Because WSNs are employed to sense and convey information of some physical phenomenon of interest, the reliability of sensor-to-sink data transport should be considered as the fidelity[3] of the knowledge obtained by the sink on the physical phenomenon. Based on this notion, we define that a sensor-to-sink data transport is *reliable* when the transport mechanism can assure that the sink is able to collect enough information; *i.e.*, *the sink can obtain enough fidelity of the knowledge on the phenomenon of interest*.

Specifically, we consider the sensor-to-sink data transport is *reliable* when the following inequation holds.

$$u = f(t_1, t_2, ..., t_m) \geq u', \tag{1}$$

where $m$ denotes the number of the sources, $u'$ denotes the required minimum fidelity on the phenomenon of interest and $u$ denotes the current fidelity obtained, which is a function of the incoming packet rate $t_i$ $(i = 1, 2, ..., m)$ from each of the sources. Note that we adopt incoming packet rates of the sink rather than the reporting rates of the sources, as packet loss along the sensor-to-sink paths would cause that the reporting rates do not well indicate the fidelity obtained by the sink [14].

As each data packet sent by the source sensor node obviously contains some information of the phenomenon of interest and therefore contributes to the sink's fidelity on the phenomenon of interest, $u$ is an increasing function of the incoming packet rate $t_i$ $(i = 1, 2, ..., m)$, *i.e.*,

$$f(t_1, ..., t_i + 1, ..., t_m) > f(t_1, ..., t_i, ..., t_m) \quad \forall i = 1, 2, ...m. \tag{2}$$

In case that the incoming packet rate from each source is $t_j$ $(j = 1, 2, ..., m)$, if we increase the reporting of the $i$th source by one, the additional fidelity obtained, denoted by $\delta_i$, is computed as follows:

$$\delta_i = f(t_1, ..., t_i + 1, ..., t_m) - f(t_1, ..., t_i, ..., t_m) \quad \forall i = 1, 2, ...m. \tag{3}$$

In existing work (*e.g.*, [14]), reliability (*i.e.*, the fidelity on the phenomenon of interest) is often measured in terms of the ratio of the achieved total incoming packet rate to the desired incoming packet rate regardless of the sources of the incoming packets, which can be modeled in terms of $f(t_1, t_2, ..., t_m)$ as follows:

$$f(t_1, t_2, ..., t_m) = \gamma \sum_{i=1}^{m} t_i, \tag{4}$$

where $\gamma$ is a constant.

But this consideration is not adequate. Total incoming packet rate is not a good indicator of how reliable the sensor-to-sink data transport is. Take the scenario in Figure 1 as an example. The sink is interested in the physical phenomenon $P$. The sensor nodes $A$, $B$ and $C$, which

[3] *Fidelity* means how certain the phenomenon value obtained by the sink is. We also use the word 'uncertainty' as its opposite.

can detect $P$, are instructed to collect information about $P$ and report data on $P$ to the sink. In this scenario, node $A$ is nearer to the physical phenomenon than node $B$ and node $C$. In most application cases, measurement error is a monotonically increasing function of the distance between the sensor and the phenomenon. Node $A$ may thereby measure the phenomenon data with less error and provide higher certainty of the phenomenon value than node $B$ and node $C$. In this scenario, if the sink receives a given number of packets, its fidelity on the phenomenon is related to the proportions of the packets sent by different sources.

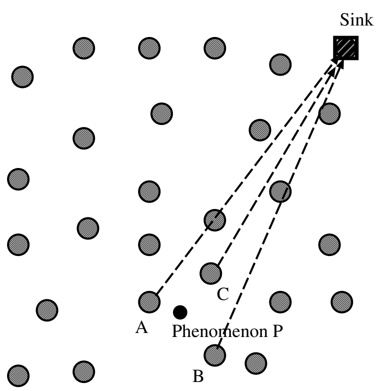


**Figure 1. A scenario of the WSN.**

Moreover, according to Equation (3) and Equation (4), $\delta_i$ is considered constant, which is not true in most application cases. $\delta_i$ is usually a decreasing function of $t_i$. The reporting packets from source $i$ in one time unit contain redundant information on the measuring phenomenon. The higher $t_i$ is, the higher the source reporting rate is required, and as a result, the more the redundant information packets from source $i$ contain, which consequently causes $\delta_i$ to decrease.

According to the above considerations, obviously, to save energy, it is better for WSNs to bias packet reporting rates of source sensor nodes according to their current contributions to improve the sink's fidelity on the phenomenon of interest. Therefore, a reliable sensor-to-sink data transport scheme should provide the sink with a mechanism to adjust the reporting rate of each data source dynamically in a discriminative manner.

But, is the contribution of each source node the only factor that influences the decision about source reporting rates? Again, consider the example scenario in Figure 1. If the current fidelity $u$ of the phenomenon is lower than the acceptable fidelity $u'$, we should increase the source reporting rates so that the sink can obtain higher fidelity.

Assume increasing the packet reporting rate of node $A$ by $r_1$ or increasing the packet reporting rate of node $C$ by $r_2$ ($r_2 > r_1$) can make the fidelity higher than the acceptable fidelity. Although the sink needs to increase the packet rate from node $A$ by less than that from node $C$ to make the fidelity acceptable (because, say, packets from $A$ have higher contribution to decrease uncertainty), increasing node $A$'s reporting rate to reduce uncertainty may not be a better solution in terms of minimizing energy consumption. This is because increasing the reporting rate of node $A$ may counter-intuitively require more energy consumption than increasing the reporting rate of node $C$ if the communication cost from node $A$ to the sink is much higher than that from node $C$ to the sink. Especially when the path from node $A$ to the sink suffers from high packet loss rate, *e.g.*, due to congestion, much energy will be consumed to convey packets along this path.

We propose that source-reporting rates should be decided based on an optimization approach. This sink should determine the reporting rates of sources so that the energy consumption of the WSN is minimized, subject to the constraint that the fidelity of the phenomenon knowledge cannot exceed a given tolerable minimum value. Formally,

$$\begin{aligned} &minimize && \sum_{i=1}^{m}(t_i \times p_i) \\ &subject\ to && u = f(t_1, t_2, ..., t_m) \geq u' \end{aligned} \quad (5)$$

where $p_i$ is the communication cost (*i.e.*, the energy consumed to successfully deliver a packet) for each source $i$ to the sink.

As $f(\cdot)$ is application-related, how to determine it and how to solve the optimization problem are beyond the scope of our protocol design. Note that only the sink (*i.e.*, where the application runs) is required to solve this optimization problem. It would not cause any energy overhead at in-network sensor nodes.

Although solving the above optimization problem is the task of applications, it is vital for a reliable sensor-to-sink data transport protocol to provide information about the communication cost $p_i$ from each source to the sink, so that the sink can properly decide the reporting rates.

Another important merit of providing end-to-end communication cost is that it can offer a congestion control mechanism. As congestion causes high communication costs, it can be alleviated with a discriminative source-rate control mechanism provided by a reliable sensor-to-sink data transport scheme. The sink can slow down sources that cause congestion and speed up sources with lower communication costs. In the meanwhile, enough fidelity can still be obtained based on the optimization approach discussed above.

In summary, to assure that the sink can obtain enough fidelity of the knowledge on the phenomenon of interest and achieve energy-efficiency, it is necessary for a reliable sensor-to-sink data transport protocol to provide two mechanisms which is listed as follows:

- A dynamic and discriminative source reporting rate feedback mechanism, allowing the sink to adjust the


1071-9458/05 $20.00 © 2005 IEEE

reporting rate of each data source.

- A mechanism to provide the sink with the current end-to-end communication cost from each source to the sink.

Note that we intend to violate the common *layering* network-protocol principle by somewhat coupling data transport protocol and applications (*i.e.*, let applications solve an optimization problem and feed back required reporting rates of sources). This is based on the features of WSNs. A WSN is usually employed to conduct one or a few specific tasks; *i.e.*, only one or a few specific applications are running at the sink. Traditional layering concept aims at general purpose protocol design. In transport layer design, it aims to provide data transport service for various applications. However, strict layering is not necessary in WSN because applications of a network are always deterministic before a network is set up (whereas, they are not deterministic in traditional networks). Moreover, it is even worse as it would cause much protocol overhead. Violating layering principle and utilizing application information in data transport protocol design can let the application facilitate the data transport protocol to save energy, which is an important merit of this paper.

## 4. Design Considerations

### 4.1. The concept of node price

As wireless communication consumes most of the energy in WSNs, the energy consumption of local computation at each node can be ignored [7]. Also, even though the packet size of each packet may be dynamic, the inevitable large overhead of the physical layer implementation of traditional wireless communication schemes makes the energy consumption of each packet transmission attempt nearly constant. So, we consider the total number of transmission attempts of the nodes required to successfully deliver a packet as the metric to evaluate the energy cost of the communication. The formal definition is as follows.

*The price of a node $n$ is, the total number of transmission attempts all in-network nodes have made to successfully deliver a packet from node $n$.*

We denote the node price of node $n$ as $NP(n)$. Obviously, node price is determined by the price of its downstream neighbors, the link loss rate between the node and its downstream neighbors, the end-to-end packet loss rate from its downstream neighbors to the sink, and the proportion of the outgoing traffic allocated to each downstream neighbor. Table 1 describes the symbols employed during our following discussions.

Now we derive the node price of each in-network node in a recursive way. Consider node $n$ sends out $\mathcal{N}$ packets via its downstream neighbors to the sink. The number of packets that can successful reach neighbor $n_i$ is:

| | |
|---|---|
| $n_i$ | The $i$th downstream neighbor of node $n$ |
| $NP(n_i)$ | The node price of node $n_i$ |
| $\omega(n, n_i)$ | The proportion of node $n$'s outgoing traffic that is routed to its downstream node $n_i$ |
| $p(n)$ | End-to-end packet loss rate from node $n$ to the sink |
| $p(n_i)$ | End-to-end packet loss rate from node $n_i$ to the sink |
| $h(n_i, n)$ | Link packet loss rate from node $n$ to its downstream node $n_i$ |

**Table 1. The descriptions of the symbols**

$$\mathcal{N} \cdot \omega(n, n_i) \cdot (1 - h(n_i, n)), \tag{6}$$

in which the number of packets that can successfully reach the sink, denoted by $\mathcal{N}_i$, is:

$$\mathcal{N}_i = \mathcal{N} \cdot \omega(n, n_i) \cdot (1 - h(n_i, n)) \cdot (1 - p(n_i)). \tag{7}$$

Therefore, according to the definition of the node price, the total number of transmission attempts that all in-network nodes have made to successfully deliver $\mathcal{N}_i$ packets from node $n$ via the path along node $n_i$ is:

$$\mathcal{N}_i \cdot NP(n_i) + \mathcal{N} \cdot \omega(n, n_i). \tag{8}$$

The total number of packets that can successfully reach the sink is:

$$\sum_{\forall i} \mathcal{N}_i = \sum_{\forall i} \{\mathcal{N} \cdot \omega(n, n_i) \cdot (1 - h(n_i, n)) \cdot (1 - p(n_i))\}. \tag{9}$$

The total number of transmission attempts that all in-network nodes have made to successfully deliver $\sum_{\forall i} \mathcal{N}_i$ packets is:

$$\sum_{\forall i} [\mathcal{N}_i \cdot NP(n_i) + \mathcal{N} \cdot \omega(n, n_i)]. \tag{10}$$

According to Equations (6)−(10), we can calculate $NP(n)$ as follows:

$$\begin{aligned} NP(n) &= \frac{\sum_{\forall i} [\mathcal{N}_i NP(n_i) + \mathcal{N} \omega(n, n_i)]}{\sum_{\forall i} \mathcal{N}_i} \\ &= \frac{\sum_{\forall i} \{\omega(n, n_i)[(1 - p(n_i))(1 - h(n_i, n)) NP(n_i) + 1]\}}{\sum_{\forall i} [\omega(n, n_i)(1 - p(n_i))(1 - h(n_i, n))]}. \end{aligned} \tag{11}$$

The end-to-end loss rate from node $n$ to the sink $p(n)$ is:

$$\begin{aligned} p(n) &= 1 - \frac{\sum_{\forall i} \mathcal{N}_i}{\mathcal{N}} \\ &= 1 - \sum_{\forall i} \{\omega(n, n_i) \times [(1 - p(n_i)) \cdot (1 - h(n_i, n))]\} \end{aligned} \tag{12}$$

As the traffic ends at a sink, the sink always has $NP(sink) = 0$ and $p(sink) = 0$.

If the link packet loss rates along all the paths of the sensor-to-sink traffic can be obtained with a hop-by-hop feedback mechanism along the reverse direction of the sensor-to-sink traffic, any node $n$ along the path can calculate its $NP(n)$ and $p(n)$ according to Equation (11) and Equation (12) based on $NP(n_i)$ and $p(n_i)$ fed back by its downstream nodes $n_i$ and its outgoing traffic allocation scheme $\omega(n, n_i)$.

Because of the dynamic nature of the WSN traffic, the link loss rate is a dynamic variable. Accurate and up-to-date hop-by-hop loss rate estimation is necessary to ensure that the price of a node represents the real downstream communication conditions. We will discuss how to obtain the link loss rate in Subsection 4.2 and the routing scheme that determines $\omega(n, n_i)$ in Subsection 4.3.

### 4.2. Link loss rate estimation

There are three situations in which the communication load may change. The first one is that a new task is assigned and the responsible sensor nodes begin to report packets. The second one is that the sink requests the source nodes to change their reporting rates. The last one is that some in-network nodes decide to change their routing scheme, *e.g.*, a node may begin to send more packets to a downstream neighbor when it finds that the price of the neighbor has become smaller. According to the discussion in [9], link loss rate will increase smoothly as the traffic load gradually increases. PORT ensures that these three situations will cause only a gradual change of traffic load (we defer the discussion of the mechanism to Subsection 4.3 and Section 5). As a result, the packet loss rate will not change quickly in our scheme. It is therefore reasonable to estimate link loss rate based on an EWMA (Exponentially Weighted Moving Average) approach.

We base our link loss rate measurements upon the sequence of the arrival packets' serial numbers (SN). Every node $n$ sends packets to each downstream neighbor $n_i$ with consecutively increasing SN. The receiver, *i.e.*, node $n_i$ can measure the link loss rate according to the missing SN. Then we can calculate the link loss rate with an EWMA approach. Formally,

$$h(n_i, n) = \alpha \times h_{-1}(n_i, n) + (1 - \alpha) \times h'(n_i, n) \quad (13)$$

where $h_{-1}(n_i, n)$ is the previous estimate of link loss rate; $h'(n_i, n)$ is the link loss rate according to current sampling result; and $\alpha$ is a weighting factor, whose value is selected empirically according to the traffic features of WSNs. Generally speaking, $\alpha$ should be set close to 1 when we have *a priori* knowledge that the traffic of the WSN is stable.

As congestion will increase packet loss rate, the link loss rate gives a good indication of the congestion condition. The communication cost metric, *i.e.*, node price, calculated from the packet loss rates, is therefore also influenced by congestion. As a result, reporting rate control and routing based on node price can provide a good congestion avoidance mechanism.

### 4.3. Routing scheme

As the price of a node determines the energy efficiency of the communication between the node and the sink, the nodes can make a local optimal decision on where to route packets to minimize their prices. If the in-network node finds that its outgoing traffic is not fully allocated to the current best downstream path (*i.e.*, the traffic is not 100% sent to the preferred downstream neighbor to achieve the smallest local NP), the node will shift the outgoing traffic that is currently allocated to the other downstream neighbors to the best downstream neighbor. Obviously, such an optimization approach always allocates all the traffic to one best path.

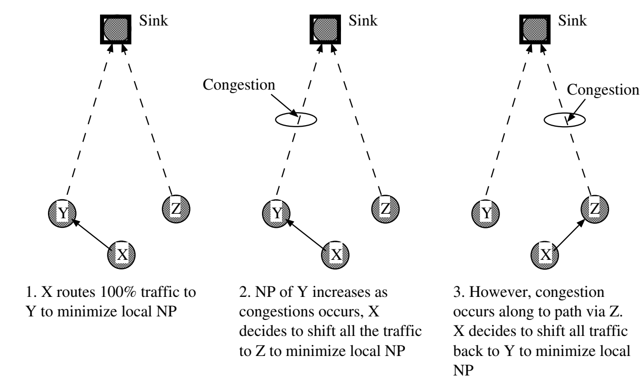


**Figure 2. An example showing oscillation**

However, the price of a node's downstream neighbors might vary with the change of the node's outgoing traffic allocation. In the worst case, the dynamics of the downstream neighbors' price caused by the node's outgoing traffic allocation change will result in fast routing oscillation. An example is shown in Figure 2. When node $X$ routes all traffic via its neighbor $Y$ (scenario 1 in the figure), the path to the sink via $Y$ may get congested. The packet loss rate along the path will increase and so the price of $Y$ will increase (scenario 2 in the figure). Node $X$, to minimize its own price, will then shift all the outgoing traffic to Node $Z$. As a result, the path via $Z$ will get congested and the price of $Z$ will increase (scenario 3 in the figure). Node $X$ has to shift all the traffic back to node $Y$ to achieve minimal price. Oscillation is inevitable in this example scenario.

Such oscillation caused by interaction of traffic loads and path cost (in our protocol, it is node price) is a notorious routing problem in data networks [3]. Fortunately, since we can have more than one outgoing path at a time, we can

Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)
1071-9458/05 $20.00 © 2005 IEEE

avoid such a fast oscillation by shifting the traffic to the new detected best downstream path in a gradual manner. Let us denote the current proportion of outgoing traffic allocated to the bad downstream node (a 'bad' downstream neighbor means that routing through it causes high node price comparing to routing through a 'good' one) with a higher price $NP_{high}$ as $\omega(n, high)$, and denote the price of the good downstream node as $NP_{low}$. The proportion of traffic that will be shifted from the bad node to the good node in each decision interval is:

$$\omega(n, high) \times \frac{NP_{high} - NP_{low}}{NP_{high}} \tag{14}$$

This scheme assures that the more the difference between the prices of the downstream nodes, the more traffic would be shifted each time. The network can thus adapt to the communication condition changes and avoid fast oscillation with a proper decision period.

If congestion of one selected path occurs, the node price of the neighbor in that path will increase. The node will gradually shift outgoing traffic to a new best path. This scheme could result in an increase of the node's price. If the new best path never gets congested because of the traffic shift, the node will locally avoid congestion by eventually allocating all traffic to the new best path. Otherwise, because the price of the node will eventually influence the node price of the source that sends packets via this node, the sink can decide to slow down the source that keeps sending packets to the congested path and speed up another source, using the rate control scheme provided by PORT.

## 5. Protocol Description

When a new task is assigned, PORT employs a similar routing information establishment mechanism to directed diffusion [11] by flooding the task description packet (called *interest* in [11]) to achieve the in-network nodes' neighborhood information. After the task assignment phase, the nodes in the WSN begin to report data packets to the sink if the physical phenomenon of interest can be sensed. The outgoing traffic allocation of a node can be dynamically adjusted during the reporting period according to the feedback about downstream communication conditions sent by its downstream neighbors. The sink also feeds back new reporting rate requirements to source nodes. We elaborate our detailed protocol implementation as follows.

### 5.1. Task initialization

We employ a reactive routing approach: the sink initiates a task by flooding its interest on some physical phenomenon. The nodes' neighborhood information is initialized as the interest packet travels throughout the network. A node's price is initially set to be the hop number between the node and the sink, and all the loss rates are considered to be zero. The nodes that are responsible for reporting data begin to report at the desired rate described in the interest packet. In order to ensure that the traffic pattern is changed in a gradual manner, the initial desired reporting rate is cautiously set to a very small value in the interest packet. After initialization, further adjustment will be conducted by the sink as described in the following subsection.

### 5.2. Feedback of newly desired source reporting rates

A source node encapsulates its node price in its reporting data packets. In this way PORT provides the node price of a data source to the application. If the application at the sink finds that the packets received per unit time provide more or less information on the physical phenomenon of interest than it desires, it will adjust the reporting rates based on an optimization approach. The new desired reporting rate of each source node is fed back to PORT by applications.

The feedback information is sent to the sources by PORT along the reverse path of the sensor-to-sink traffic. The rate control packets are inserted at the head of the sender nodes' queues and sent out with the highest priority. Such rate control packets can also be sent back directly to individual source nodes as implemented in ESRT [14] if the wireless interface of the sink is powerful enough.

### 5.3. Feedback of wireless communication condition

The sink, and the in-network nodes that are conveying the sensor-to-sink packets, estimate the link loss rate from each of their upstream neighbors to themselves. The link loss rate and their prices, as well as their end-to-end path loss rates (from them to the sink), are checked in a given time interval. If they find that these values have changed, the new values are fed back to their upstream neighbors. These feedback packets are inserted to the head of the nodes' queues and sent out with the highest priority.

Upon receiving a communication condition feedback packet from a downstream neighbor, a node will re-allocate its outgoing traffic as discussed in Subsection 4.3 if it finds that the current traffic allocation cannot achieve the local lowest price. The new price and path loss rate are calculated according to Equation (11) and Equation (12).

### 5.4. Fault tolerance and scalability considerations

In the case that a node dies (silently quitting the task), its upstream neighbor should shift the traffic routed via this node to other nodes immediately. We employ a timer on each node to detect the quitting of its downstream nodes. For each timeout occurrence, if a node fails to receive any feedback information from a downstream neighbor, it considers the downstream neighbor has failed and set the price of the neighbor as infinite to avoid routing packets to it.

If a new node (which could be a newly awakened node, a newly deployed node, or a node that recovered from a previous failure) detects an ongoing task, it might decide to join the routing task. In this case, the node will broadcast to its neighbor nodes to inform them that it is up. Neighboring nodes will send it their prices. The node selects some nodes with the lowest prices as its downstream neighbors and sends its own calculated price and the path loss rate to those neighboring nodes with larger prices. Upon receiving this information, those neighbors with larger prices will consider the node as a possible downstream neighbor. In this way, the new node joins the routing task.

## 6. Protocol Evaluation: A Case Study

To verify PORT, we code it over NS-2 [8]. As discussed above, PORT is employed to facilitate the sink to achieve reliability. To perform simulations, an application model should be specified. Although we verify PORT in a given application case, note that more sophisticated models could be employed in real world applications. The performance of PORT is surely influenced by the application, as it is the application that determines the reporting rates of the sources. The aim of our simulations is to show that with a proper decision on source reporting rates, PORT can effectively facilitate the sink to achieve energy-efficiency and maintain reliability.

Without loss of generality, PORT can be applied in many application scenarios for energy saving. The prerequisite is that the application should determine reporting rates of sources dynamically according to the data reported by the sources and the communication cost reported by PORT.

### 6.1. Simulation model

In our application scenario when conducting simulations, the sink is interested in a phenomenon with physical position $(x, y)$. $m$ nodes that are close to the phenomenon measure the physical value of that phenomenon and report each measurement value with a packet sent to the sink. For simplicity, assume that the $j$th measurement value of node $i(i = 1, 2, ..., m)$, denoted by $s_{i,j}$, is one-dimensional. The measurement model is

$$s_{i,j} = X + e_{i,j} \quad (15)$$

where $X$ is the true value of the phenomenon parameter; $e_{i,j}$ is the error of the $jth$ measurement of node $i$. Assume $e_{i,j}$ $(j = 1, 2, ...)$ are Gaussian-distributed with zero mean and with standard deviation $v_i$. $v_i$ is related to the physical distance $d$ between node $i$ and $(x, y)$. For simplicity, we set it as follows, which means that the uncertainty of each measurement is directly proportional to the square of the distance $d$.

$$v_i = 0.0001 \times d^2 \quad (16)$$

The sink fuses the data received from node $i$ in one second by calculating the mean of them (we denote the incoming packet rate from node $i$ as $t_i$). The sink then calculates the average of the fused result of each node as the value of the phenomenon.

$$\frac{1}{m} \cdot \sum_{i=1}^{m} (\frac{1}{t_i} \cdot \sum_{j=1}^{t_i} s_{i,j}) \quad (17)$$

Thus, the sink's uncertainty $\upsilon$ on the value of the phenomenon is calculated as the standard deviation of the error:

$$\upsilon = \sqrt{\frac{1}{m^2} \cdot \sum_{i=1}^{m} \frac{\theta_i^2}{t_i}} \quad (18)$$

where $\theta_i$ is the standard deviation of $t_i$ measurements (*i.e.*, $s_{i,j}, \forall j = 1, 2, ..., t_i$) of source $i$, obtained statistically.

In our simulations, we compare two sensor-to-sink data communication protocols: one is a directed-diffusion-based shortest path routing scheme with an ESRT-like unbiased report rate control approach (denoted as scheme 1); the other is PORT (denoted as scheme 2).

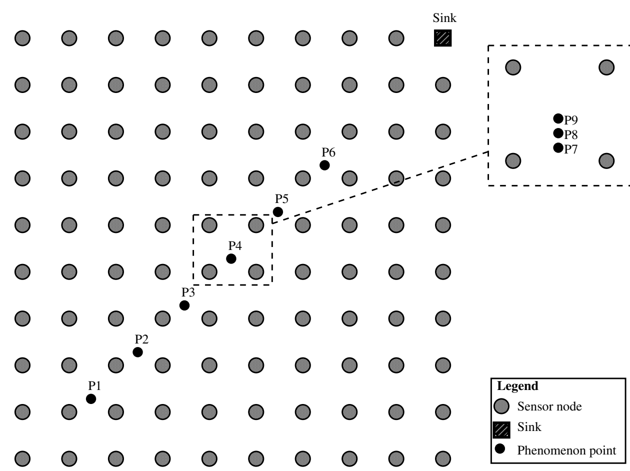


**Figure 3. The simulation network.**

The original locations of the sensor nodes are in a grid-like way shown in Figure 3. For each simulation (*i.e.*, for each location of the phenomenon point), we change the location of each sensor node (except the source nodes and the sink) *randomly in a uniform manner* in a $100\times100m$ square which centers on its original location shown in Figure 3 for 20 times. We average the simulation results for each settings of node locations.

The sink is in the top-right corner of the network. The wireless parameter settings are the same as the study of directed diffusion [11]. Detailed settings of the simulation network are shown in Table 2.

| Area of sensor field | 1350m*1350m |
|---|---|
| Number of sensor nodes | 100 |
| MAC | IEEE 802.11 without CTS/RTS and ACK |
| Radio power | 0.2818 W |
| Packet length | 36 bytes |
| Transmit power | 0.660 W |
| Receive power | 0.395 W |
| IFQ length | 50 packets |
| Simulation time at each setting | 500 seconds |
| Feedback / decision period | 1 second |

**Table 2. Simulation network settings.**

## 6.2. Energy consumption comparison

To study the total energy consumptions, we set the phenomenon at six different positions marked by $P1 - P6$ in Figure 3. For each setting, a set of different uncertainty values are required by the sink. For each uncertainty requirement, the four nearest nodes report their measurements to the sink in 500 seconds. Figure 4 shows the total energy consumptions of the whole network under these two protocols given different uncertainty requirements when the phenomenon points are at different positions.

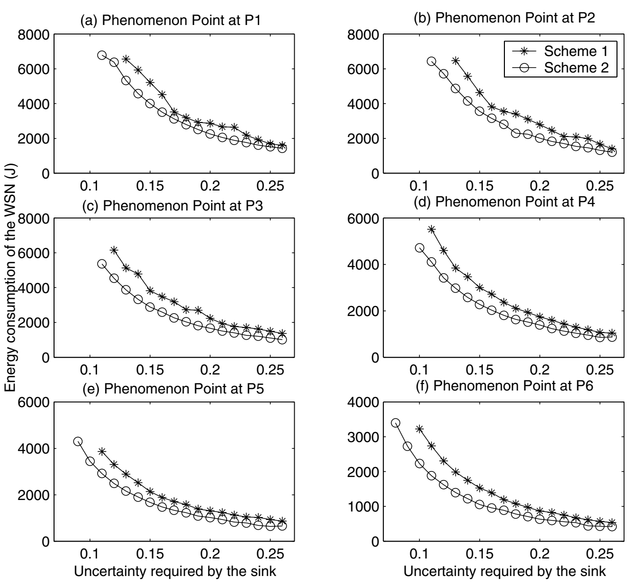


**Figure 4. Energy consumption comparisons.**

The results show that PORT can save 10% to 30% of the energy consumption, compared to an existing scheme which employs unbiased source reporting rate control. This is not surprising, as PORT biases the reporting rates of the sources according to their contributions to reduce the uncertainty of the phenomenon value and their prices, which is a more energy-efficient approach.

PORT saves more energy when a smaller uncertainty is required. This is because, when a small uncertainty is required, large source reporting rates are needed. As a result, traffic load is high. Packet loss rate along the sensor-to-sink path is then also high. PORT can allocate traffic to alleviate congestion. In this case, PORT saves much more energy than the existing scheme.

Moreover, the results show that PORT can satisfy a smaller uncertainty requirement (uncertainty requirements less than $0.12$ in Figure 4(a) and Figure 4(b), $0.11$ in Figure 4(c), $0.10$ in Figure 4(d) and Figure 4(e), and $0.09$ in Figure 4(f)). In the very small uncertainty requirement cases, large source reporting rates of the sources overload the network capacity. The network severely congests and thus cannot provide the sink with enough packets. The uncertainty requirement cannot then be fulfilled. As PORT can alleviate congestion by routing via different paths, it allows higher reporting rates than existing schemes and hence it can fulfill a smaller uncertainty requirements. It shows that PORT provides a better congestion avoidance scheme.

## 6.3. The impact of reporting sensors' uncertainty distribution

To study the impact of the reporting sensors' uncertainty distribution, we set the phenomenon point at three different places in the network grid marked by $P7 - P9$ in Figure 3. Also, four nodes in the corners of the grid are reporting their measurements. Note that the closer the phenomenon point to the center of the grid, the more similar are the contributions of the four sources. Figure 5 shows the total energy consumptions of the whole network under these two protocols given different uncertainty requirements and different phenomenon positions.

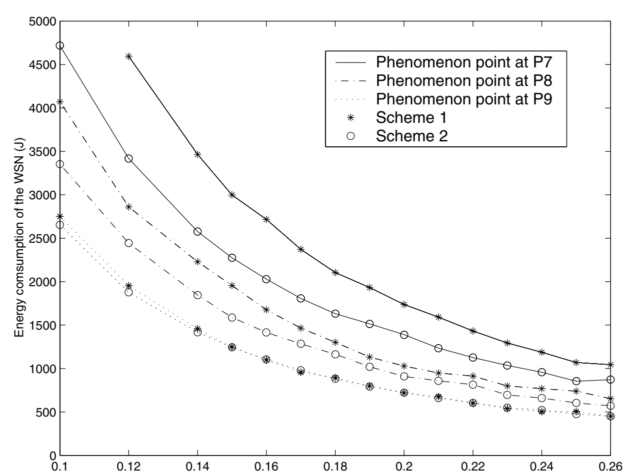


**Figure 5. Energy consumption comparisons: different phenomenon positions in a grid**

These results show that PORT can save more energy if the contributions of the sources are more different. PORT achieves little improvement when the sources have the same contributions. This is not surprising, as PORT biases the reporting rates of sources according to the sources' contributions to reduce the sink's uncertain of the phenomenon value. When the sources' contributions are almost the same, PORT will adjust the reporting in an almost unbiased manner, like existing schemes. Their energy consumptions, as a result, are almost the same.

## 7. Conclusion

This paper proposes PORT, a price-oriented sensor-to-sink data transport protocol for wireless sensor networks. Under the constraint that the sink must obtain reliable information on the phenomenon of interest, PORT minimizes the energy consumptions using two schemes. One is based on the sink's application-based optimization approach that feeds back the optimal reporting rate of each source according to the contribution of the sources and the energy consumption of the sensor-to-sink communication from each source to the sink. The other is a locally optimal routing scheme for in-network nodes according to feedback of downstream communication conditions. The communication conditions estimation is based on an estimation of link loss rate along the sensor-to-sink traffic path. PORT can obtain the sensor-to-sink communication condition such as congestion and weak link which cause packet loss, and thus it adapts well to network dynamics caused by these factors.

We code PORT on the NS-2 network simulation tool. Simulation results in an application case study demonstrate that PORT is an effective transport protocol for reducing energy consumption comparing to existing schemes. Thus, it can prolong the life time and reliability of wireless sensor networks.

## Acknowledgement

The work described in this paper was substantially supported by two grants, RGC Project No. CUHK4205/04E and UGC Project No. AoE/E-01/99, of the Hong Kong Special Administrative Region, China. J. Liu's work is supported in part by a Canadian NSERC Discovery Grant 288325, an NSERC Research Tools and Instruments (RTI) Grant 613276, and an SFU President's Research Grant.

## References

[1] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Elsevier Ad Hoc Network Journal*, 2(3):325–349, 2005.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, April 2002.

[3] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Upper Saddle River, 1992.

[4] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *Proc. of the First Workshop on Sensor Networks and Applications*, Atlanta, GA, October 2002.

[5] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16(3), August 2002.

[6] D. Estrin, D. Culler, and K. Pister. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1(1), January-March 2002.

[7] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proc. of the 5th MobiCom*, Seattle, Washington, August 1999.

[8] K. Fall and K. Varadhan. *The ns manual*, Dec. 2003. http://www.isi.edu/nsnam/ns.

[9] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla. The impact of multihop wireless channel on tcp throughput and loss. In *Proc. of the 22nd IEEE Infocom*, San Francisco, CA, March 2003.

[10] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM Mobile Computing and Communications Review (MC2R)*, 1(2):295–298, January 2002.

[11] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. of the 6th MobiCom*, Boston, Massachusetts, August 2000.

[12] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proc. of the ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[13] G. Pottie and W. Kaiser. Wireless integrated network sensors. *Communications of ACM*, 43(5), May 2000.

[14] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz. Esrt: Event to sink reliable transport in wireless sensor networks. In *Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, June 2003.

[15] C. Schurgers and M. Srivastava. Energy efficient routing in wireless sensor networks. In *Proc. of the 2001 MILCOM*, volume 1, pages 357–361, 2001.

[16] R. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proc. of the IEEE Wireless Communications and Networking Conference*, Orlando, FL, March 2002.

[17] F. Stann and J. Heidemann. Rmst: Reliable data transport in sensor networks. In *Proc. of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[18] C. Wan, A. Campbell, and L. Krishnamurthy. Psfq: A reliable transport protocol for wireless sensor networks. In *Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, September 2002.

[19] C. Wan, S. Eisenman, and A. Campbell. Coda: Congestion detection and avoidance in sensor networks. In *Proc. of the the 1st ACM Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, November 2003.