AN INTERFERENCE-AWARE ROUTING ALGORITHM FOR MULTIMEDIA STREAMING OVER WIRELESS SENSOR NETWORKS

Shuang Li[1], Raghu Kisore Neelisetti[2], Cong Liu[3], Santosh Kulkarni[2],

and Alvin Lim[2]

[1]Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio, USA
li.908@osu.edu
[2]Department of Computer Science and Software Engineering, Auburn, Alabama, USA
neelira@auburn.edu, kulkasa@auburn.edu, lim@eng.auburn.edu
3Department of Computer Science and Engineering, The University of North Carolina at Chapel Hill, Chapel Hill, NC, USA
cong@email.unc.edu

ABSTRACT

Wireless sensor networks are originally designed as distributed event-based systems that differ from traditional communication networks in several ways. These networks typically have nodes with severe energy constraints, variable quality links, low data-rate and many-to-one event-to-sink flows. Recently, Wireless Multimedia Sensor Networks (WMSNs) have been developed due to the availability of low-cost cameras, microphones, and other sensors producing multimedia data. The applications, accordingly, are extended to video surveillance and notification, video and computer assistance in video-assisted living and healthcare. The stringent requirements of real-time multimedia applications include end-to-end delay, bandwidth and loss during data transmission. Communication algorithms for WMSN must therefore be specially designed to operate efficiently under these constraints. Directed diffusion is a data-centric protocol designed for wireless sensor networks. However, it is not efficient in more challenging domains, such as video sensor networks, because of its inability to satisfy the throughput and delay requirements of multimedia data. Instead, we propose EDGE — a greedy algorithm based on directed diffusion that reinforces routes with high link quality and low latency, thus maximizing throughput and minimizing delay. ETX (Expected Transmission Count) is used as the metric for measuring link quality. This paper presents an improved method for computing aggregate ETX for a path that increases end-to-end throughput. NS-2 simulation results with video data as CBR (constant bit rate) traffic show that our proposed distributed algorithm selects routes that give better throughput and jitter than those reinforced by standard directed diffusion, while maintaining low delay, thereby improving the performance of wireless sensor network for multimedia data transmission.

KEYWORDS

Wireless Sensor Networks, Multimedia, ETX, Delay, Throughput, Jitter, Directed Diffusion, Routing, Greedy Algorithm.

1. INTRODUCTION

Current Wireless Sensor Networks (WSNs) support a wide range of applications, such as target tracking, home automation and environmental monitoring. Some of these applications may be reinforced or augmented with the transmission of multimedia data over WSNs. Existing WSNs, nevertheless, have restrictions in supporting these video/audio streaming applications due to the hardware such as cameras small enough to be installed on sensor nodes (SNs), bandwidth of the network, and power supplies of SNs. Fortunately, recent advances of wireless technologies, embedded systems, multimedia source coding techniques and inexpensive hardware such as CMOS cameras, microphones, etc. have fostered the development of Wireless Multimedia Sensor Networks (WMSNs), over which multimedia data streams are transmitted. Accordingly, new applications are created, such as: multimedia surveillance, storage of potentially relevant activities, traffic avoidance, advanced health care delivery, automated assistance for the elderly and family monitors, etc. They usually have a set of stringent QoS requirements, such as, end-to-end delay, bandwidth, and jitter guarantees.

To meet these requirements, a more efficient routing protocol needs to be designed for WMSNs. Data-centric networking, such as directed diffusion [1], has been commonly used for wireless sensor networks because of its energy efficiency and scalability. It enables sensor data to be disseminated from data sources to sinks with low delay. WMSNs require larger amount of real-time multimedia data to be disseminated with low latency and high delivery ratio. In transmitting multimedia data traffic, additional quality of service constraints must be satisfied. The main challenge is to develop a practical data-centric networking algorithm that can maximize throughput, minimize delay and meet other QoS constraints as much as possible in wireless sensor networking environments.

Directed diffusion uses a publish/subscribe communication model whereby a sink node requests data by sending interests for a named data. As the interest is flooded through the network, each intermediate node establishes a gradient with its neighbors and enables data that match the interest to be “drawn” towards the sink. Sensor nodes with data that matches the interest will forward an “exploratory data” that is propagated by intermediate nodes through established gradients to the sink. The sink sends a reinforcement message to the node that first forwarded the new data to it. Intermediate nodes use the same rule to reinforce their upstream neighbor. After the reinforcement stage, the source node continues to send data through the reinforced path.

Based on the above rule, directed diffusion [1] generally selects routes with the lowest delay. Other ad hoc routing protocols, such as DSR [2] and DSDV [3], usually use a hop count metric. Throughput is considered in some recent ad hoc protocols [4]. The design of wireless communication protocols in sensor networks is often guided by two principles — self-detection of link quality and in-network processing. This is necessary because of the variability in link quality, low bandwidth of wireless links and limited memory of sensor nodes.

To quantify data transmission in sensor networks, two models [5] for successful reception of a transmission over one hop were proposed - the Protocol Model and the Physical Model. SNR (Signal-to-Noise Ratio) is an indicator of link quality in the Physical Model. Since SNR is computed at the physical layer, it is inaccessible to the network stack. On the other hand, ETX (Expected Transmission Count) is a link layer metric that can be used by the network layer in a cross-layer design. In [4], the ETX of a route is computed by adding the ETX of all the links in the route. Their results show that using this route ETX, their algorithm may inappropriately choose a slower path that has fewer hops (provided the best path has four or more hops). This is because the longer the path, the larger the sum of ETX.

We propose an improvement on computing the appropriate route ETX to rectify the above problem by taking into account bottleneck links in paths that may cause higher delay. The use of ETX has been criticized because of its deficiency in modeling transmission interference [6]. Our improved method for computing ETX for a route measures intra-flow interference more accurately since it considers the maximum of any three consecutive links in a route that are within interference range. Our algorithm also considers the delay metric in selecting the best route.

In WSNs, especially video sensor networks, transmitting multimedia data requires the selection of paths that ensure high throughput and low latency. As pointed out by Gupta and Kumar [5], the fundamental reason leading to the degradation of the performance as the number of nodes increases is the fact that each node has to share the radio channel with its neighbors. Subsequently, our main motivation is to enable the interference-aware selection of the best route for maximizing throughput and minimizing delay using an integrated metric. Our simulation results show a trade-off between throughput and delay which is comparable to the optimal theoretical trade-off as analyzed in [7].

Delay is the only metric used in directed diffusion for reinforcing a path with the shortest delay. On the other hand, greedy algorithms that consider both throughput and delay may not always find the best route since they do not have sub-solution optimality property. However, the best route can be determined if the source node memorizes all the possible routes. This is then a PSPACE problem. Since sensor nodes have limited memory, such centralized algorithm is not practical. In practice, greedy algorithms can produce reasonable good performance. Our results show that our greedy algorithm can find routes with much better delay and throughput than standard directed diffusion with retransmission.

2. Related Work

The latest series of TelosB motes [8], the ZigBee motes [9] with improved abilities, or PC104 [10] may be used for applications in WSNs which require intensive memory and bandwidth. Most of the sensors used in research for audio/video streaming are found to use embedded microprocessors which have higher computing abilities [11].

Many world-wide universities and research companies have been conducting research projects of video sensor networks or WMSNs, such as self-configuring video-sensor networks for healthcare at Imperial College London [12], large scale video sensor networks for distributed surveillance at Palo Alto Research Center (PARC) [13], "Video Web" at University of California — Riverside [14], a video-based sensor network architecture for video surveillance and environment monitoring proposed by Feng *et. al.* [15], maximizing the life of wireless video sensor networks at Virginia Tech [16], the Distributed Interactive Video Array (DIVA) system at Spawar Systems Center (SSC) San Diego [17], WMSNs research at Ohio State University [18], video sensor network for autonomous coastal sensing at Boston University [19], and Quality of Service (QoS) research for vision-based WSNs at Purdue [20].

The network layer of WMSN needs to address QoS issues of multimedia streams. [21] considers the bandwidth constraints for multimedia mobile medical calls. Distributed image sensing with QoS-based geographic routing is used in [22] for network localization, dynamic routing and load balancing. Other papers are more concerned with real time streaming issues, e.g. RAP [23], SPEED [24] and its extension MMSPEED [25]. They prioritize packets based on their delivery speed, computed from geographic information and elapsed time, either at the source, hop-by-hop or every few hops. MMSPEED also performs route selection in reliability domain. Although they are generic protocols for real time data transmission over ad hoc or sensor networks, real time protocols for WMSN could be developed by extending their framework.

Routing metrics in wireless ad hoc networks are important considerations due to the unpredictability and heterogeneity of link qualities [26]. Existing wireless ad hoc routing protocols typically select routes using minimum hop count, e.g. DSR [2] and DSDV [3]. Directed diffusion [1] selects routes in sensor networks with the least delay. Recently, many new link quality metrics have been proposed. [27] compares the performance of the following three metrics. Adya et al. [27] measures the round trip delay of unicast probes between neighboring nodes and proposes Per-hop Round Trip Time (RTT). Per-hop Packet Pair Delay (PktPair) measures the delay between a pair of back-to-back probes to a neighbor node [27]. Expected Transmission Count (ETX) [4] measures the loss rate of broadcast packets between pairs of neighboring nodes and estimates the number of retransmissions required to send unicast packets. Weighted Cumulative Expected Transmission Time (WCETT) [28] is used for selecting channel-diverse paths and accounts for the loss rate and bandwidth of individual links. Park et al. [6] presented a new metric, Expected Data Rate (EDR), for accurately finding high-throughput paths in multi-hop ad hoc wireless networks based on a new model for transmission interference.

Unfortunately, none of these metrics can be directly applied to wireless sensor network that simultaneously take into account delay, throughput and interference. Furthermore, none of previous papers proposed a combined metric for sensor networks with all those considerations. In [4], ETX was incorporated into DSR and DSDV to improve throughput with little consideration of delay or interference. WCETT [28] is more suitable in multi-radio wireless mesh networks. EDR [6], unlike ETX, cannot be computed dynamically. More space and computation are required by EDR when it is incorporated into DSR and AODV.

Interference-aware protocols have recently been explored in multi-hop wireless networks. [29] studies routing problems in a multihop wireless network using directional antennas with dynamic traffic and presented new definitions of link and path interference. In their other paper [30], they present routing algorithms to compute interference-optimal cost-bounded paths and an optimal bandwidth allocation algorithm to allocate timeslots. We have not given detailed analysis, computation and implementation for limiting interference yet because we are currently exploring the full use of ETX information. [31] and [32] give the throughput bounds and capacity for interference-aware routing in wireless networks respectively. We could use them to test our protocol by observing the throughput performance. [33] derives an interference aware metric NAVC based on the information collected from 802.11 MAC. In [34], an interference aware routing scheme is designed to alleviate the near-far problem at the network level for cellular systems. EIBatt et. al. [35] address the problem of interference-aware routing by coupling the lower three layers of the ISO Open Systems Interconnection (OSI) protocol stack. We only use ETX, the link layer indicator, to measure the link quality as well as interference to simplify the problem. Nguyen et. al. [36] consider radio interference and modify OLSR routing protocol for bandwidth reservation and interferences. Our paper modifies directed diffusion, a routing protocol for wireless sensor networks, to take into account throughput, interference and delay.

In sensor networks, each node has limited memory and requires in-networking processing. Link quality is highly variable and delay metrics may not be able to measure the variation. Most sensor network nodes are equipped with one omni-directional radio and use one channel at a time. Thus there is more interference than in multi-radio or multi-channel nodes. Taking the summation of ETX in a route penalizes routes with more hops and assumes that this will lower throughput due to interference between different hops of the same path [4]. It is not true that all the hops in a path will interfere with each other. Bader et al. [37] discovered the optimal packet injection in linear networks and they found that the first packet has outpaced the rest of the packets when the fourth packet is to be injected. Based on this result, we modify the computation of ETX for a path to more accurately quantify intra-flow interference. With this change, Dijkstra's algorithm can no longer be utilized and greedy algorithm is used instead. Inter-flow interference is also considered in Dynamic Codeword Routing (DCR) [38].

Throughput-delay trade-off in the Gupta-Kumar fixed network model [5] is theoretically analyzed in [7]. Our results also show similar trade-off between throughput and delay in practical sensor network algorithms.

3. Problem Statement

WMSNs have urgent needs for new protocols which meet the stringent QoS requirements of multimedia streaming. For example, the data rate of H.264 varies between 64 kbps and 240

Mbps depending on different levels [39]. Both throughput and delay requirements should be embodied in the new protocols. The shortcomings of minimum hop-count as a metric have been widely recognized. Routing protocols with minimum hop-count metrics assume that all links have identical properties. In practice, wireless links often do not have the same quality, due to different antenna power, background noise and interference. None of the other metrics can be directly used in directed diffusion to take into account all the delay, throughput and interference constraints. Interference affects throughput which is a highly emphasized need for multimedia data to a large extent. In designing a metric to take into account delay, throughput and interference for WMSNs, the key challenge here is to find an effective way to combine them so that we can compute the cost of each route and find a route with the minimum cost that satisfy our goals for multimedia data.

3.1. Assumptions and Goals

We begin by listing the assumptions we made about the networks.

- All nodes in the network are stationary.
- Each node is equipped with one 802.11 radio.
- There are one source and one sink in the network.

Based on these assumptions, we have three main goals. First, the protocol should take both end-to-end delay and ETX of a route into account. Since the 802.11 MAC implements an ARQ (retransmission) mechanism, the ETX of a link can be computed. Second, the path metric should not decrease when one more hop is added to the route. Third, the method for computing the path ETX must consider intra-flow interference.

3.2. Definitions, Notations and Formulae

The ETX of a link is the predicted number of data transmissions required to send a packet over that link [4].

$$ETX = \frac{1}{d_f \times d_r} \tag{1}$$

The forward delivery ratio, d_f, is the probability that a data packet successfully arrives at the recipient; the reverse delivery ratio, d_r, is the probability that the ACK packet is successfully received.

Definition of ETX_p: The path ETX is the maximum of the sum of the *ETX*s of any three successive hops in a route. This computes the amount of bottleneck. *N* is the number of hops. ETX_j is the ETX value of the *j*th hop. The number of bottleneck links may vary according to the network density.

$$ETX_p = \operatorname*{Max}_{i=0}^{N-3}\left(\sum_{j=i}^{i+2} ETX_j\right) \tag{2}$$

Definition of $delay_p$: The end-to-end delay of a packet in a network is the time it takes the packet to reach the sink from the time it leaves the source.

Definition of *Cost_p*: The path cost is the combined metric of a route. α and β are non-negative integers.

$$COST_p = ETX_p^{\alpha} \times DELAY_p^{\beta} \tag{3}$$

Definition of *decision interval* **(INTERVAL)**: We start an *adaptive timer* at each node (except the source) when the node receives the first exploratory packet. After an **INTERVAL** period, the timer expires and it selects the route with the lowest $Cost_p$. *EXPLORE_DELAY* is a constant with the basic timeout value. ETX_i is the *ETX* value of the upstream link on which the first exploratory data arrive. Different **INTERVAL** may be computed at different nodes based on the following formula:

$$INTERVAL = ETX_i \times EXPLORE\_DELAY \quad (4)$$

3.3. Computing Path Metric

Our path metric is called $Cost_p$ which conforms to the three goals we set earlier. First, it takes both end-to-end delay ($delay_p$) and ETX of a route (ETX_p) into account. By adjusting the values of α and β, we are able to set different weights to each factor. If throughput is more important for an application, α should be greater than β and vice versa. The way we compute ETX for a path is based on the theoretical analysis and experimental demonstration in [37]. Bader et al. employed the Packet Decoupling property to conclude that the first packet has outpaced the rest of the packets when the fourth packet is to be injected. Li. et al. [40] examined the capacity of a chain of nodes and they found that an ideal MAC protocol could achieve chain utilization as high as 1/3. The example below illustrates this principle for the node placement in Figure 1.

We compute the maximum summation of *ETX*s in every three successive hops and regard it as the bottleneck. This is a more accurate indicator of the worst bottleneck in the entire path. Assuming that 2, 2, 2, 2, 2, 3 are the *ETX* values for the six links in Figure 1. Then ETX_p is 7. If we change the *ETX* values in Figure 1 to 1, 1, 1, 3, 3, 3, the new ETX_p becomes 9. According to the definition of ETX_p, the latter path is worse. If path *ETX* is computed using the total *ETX* of a path, we get 13 for the former path and 12 for the latter. Then the latter path is better. Total *ETX* exaggerates the intra-flow interference and will lead to a wrong route selection.

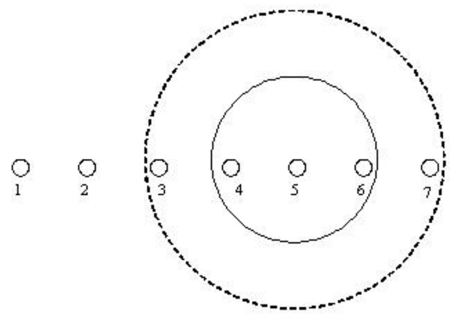


Figure 1. Transmission range and interference range for a chain of nodes. The solid line circle is Node 5 transmission range while the dotted line circle shows the interference range. Nodes within 3 hops interfere with each other.

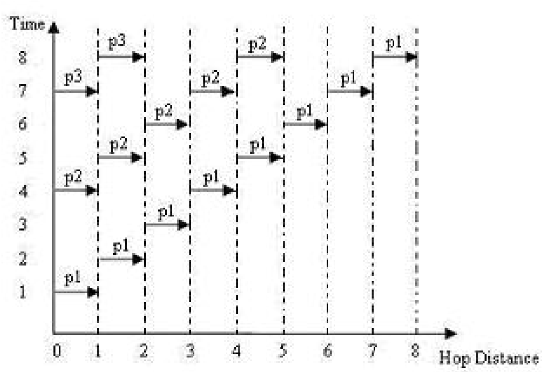


Figure 2. Transmission Pipelining mechanism for data transmission in sensor networks that takes into account intra-flow interference.

Another reason for using our path ETX metric is the impact of intra-flow interference in the pipeline of packet transmission (Figure 2). A packet is injected at Hop 0 every unit time interval. *p1* is the first packet transmitted. Suppose that each packet takes the same time to transmit on each hop, say, 30ms. When *p1* finishes transmission on Hop 1, *p2* is injected into the network. *p2* has to wait till *p1* is transmitted on Hop 3 due to the intra-flow interference. The delay here should be 60ms. The combined metric also satisfies the second goal that it does not decrease when one more hop is added to the route.

We consider intra-flow interference in the third rule by adding the *ETX* values of three successive hops together. Refer to [4] for more information.

3.4. Problem Formulation

Our routing algorithm with metric $Cost_p$ can be formulated as a cross-layer combinatorial optimization problem, where the objective is minimizing metric $Cost_p$ in order to meet QoS requirements of multimedia data. In this formulation, constraints include connectivity, link stability, and retransmission times. The solution space consists of combinations of all possible routes that provide a connection from the source to the sink. We now present the NLP (Nonlinear Programming) formulations for our routing algorithm.

We model the network as a directed graph *G(V,E)* and a collection of sub-paths from the source to any other node in the network. Let *P* denotes the set of all sub-paths from the source to any other node in the network. Thus, $\forall i \in V \setminus \{src\}$, $P = \{(src, i)\}$ and $\forall p \in P$, $dest(p) = i$, where *src* is

the source node and $p = (src, i)$.

With such path models, we want to minimize both the ETX_p and $delay_p$. The mathematical formulation is as follows:

$$\underset{p \in P}{Min}(ETX_p^{\alpha} \times delay_p^{\beta})$$

The above objective function is subject to:

$$\forall p \in P, \quad ETX_p = \underset{i=0}{\overset{dest(p)-3}{Max}}\left(\sum_{j=i}^{i+2} ETX_j\right)$$

$$\forall p \in P, \quad delay_p = t_{dest(p)} - t_{src}$$

$$\forall j \in E, \quad 1 \leq ETX_j \leq 8$$

$$\forall j \in E, \quad ETX_j = \frac{1}{d_f \times d_r}$$

$$\forall j \in E, \quad 0 < d_f \leq 1, \; 0 < d_r \leq 1$$

The first constraint defines ETX_p based on the *ETX* value for the sub-path. The second constraint is the definition of $delay_p$ for the sub-path. The third constraint sets the minimum and maximum number of transmissions in wireless networks, which is based on the rule that the maximum retransmission times in 802.11 is 7. The fourth constraint computes the *ETX* value for each link from the forward and backward delivery ratios.

4. Algorithm Design

In this section, we present centralized and distributed algorithms to compute the route which more accurately estimates interference, maximizes the throughput and minimizes the delay over lossy links in multi-hop WMSNs. [36] discards all nodes whose local available bandwidth is smaller than the requested bandwidth and all nodes in the interference area of such transmitter nodes if the shortest route provided by OLSR does not provide the requested bandwidth. In our current protocol, none of the specific multimedia QoS requirements guides the routing decision process because we assume every node has the same bandwidth and lack of global information prevents us from getting the similar interference area of a certain node. We only try to maximize throughput and minimize delay in order to meet the requirements. Packets are not prioritized anywhere because we only reinforce one route and every packet has the same deadline for end-to-end delay. We assume packets which are sent earlier by the source are added into the queue at each node earlier, which means they are processed earlier. Admittedly, packet scheduling helps achieve the hard deadline requirement. For sensor nodes, however, they may drain their energy to do the scheduling work. Transport protocols like MRTP [41] may work together with EDGE to make full use of the QoS requirements of the client so that EDGE could re-flood the interest in order to find a new robust route based on the mechanism of periodic QoS reports in MRTP. If we use level 1 of H.264 (Max macro blocks per second: 1485; Max frame size: 99 macro blocks; Max video bit rate: 64 kbps) [39], then we get 533 bytes (64k/1485*99/8) as the frame size. Suppose we do not decompose frame into several packets, the packet size is still 533 bytes. The throughput required, in this way, is 15 packets per second. We start by studying the centralized algorithm similar to that used in [4] for incorporating ETX into the initial route request in DSR. Then, we describe EDGE (ETX-Delay GrEedy algorithm) which is a distributed version and explain why it finds better routes than directed diffusion with respect to our goals, although it does not always find the best route that satisfy these goals.

4.1. Optimum Algorithm

We first introduce the simple optimal algorithm by enumerating all the routes and find the one with the best metric value. This is a centralized algorithm processed by the sink node. Each flow is labelled in the order of their arrival. *System time* is the time obtained from the system clock. We assume that the ETX information of each link could be collected while the exploratory data are flooded. *T1* is the timestamp when the packet is generated at the source node.

Table 1 Pseudo code of Centralized Algorithm.

T2 ← *System time*
MinCost ← *MAXIMUM*
Flowlabel ← 0
for each flow **do**
    $ETX_p = \underset{i=0}{\overset{N-3}{Max}}\left(\sum_{j=i}^{i+2} ETX_j\right)$
    $Cost_p = (ETX_p)^{\alpha} \times (T2 - T1)^{\beta}$
    if $Cost_p < MinCost$
        MinCost ← $Cost_p$
        Minlabel ← *Flowlabel*

We illustrate the centralized algorithm with the example below in Figure 3. In this example, there are four routes with $Cost_p$ metric of 42, 81, 120 and 80 $\alpha=1$ and $\beta=1$). This algorithm will choose the best route src → x → y → sink with the $Cost_p$ metric of 42.

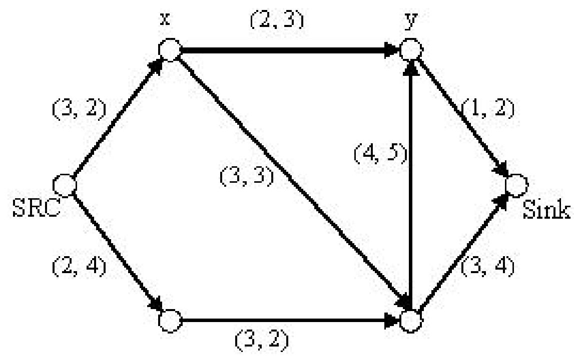


Figure 3. Illustration of the centralized algorithm. In each tuple, the first element is the ETX value of the link and the second element is the delay of the link which includes the packet queuing delay at the upstream node. Gradients are indicated by arrows.

4.2. EDGE Algorithm

In the previous sub-section, we present a centralized algorithm to find the best route in directed diffusion. It is impossible, however, for us to implement the algorithm in a real environment. There are two reasons. First, directed diffusion is a data-centric routing protocol and no global trace is recorded. Second, sensor networks may be composed of hundreds of nodes which have limited memory space. The number of routes increases exponentially with the number of the nodes, which becomes a PSPACE [42] problem. The definition of PSPACE is {x: x requires exponential memory}. Figure 4 shows one of the worst cases in which the number of routes is $O(A^n)$. A is a constant and n is the number of nodes.

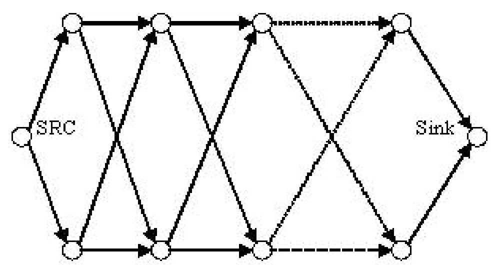


Figure 4. The worst case routes. Every 2 nodes except the source and sink are put in a vertical line. Every node in a line has two gradients pointing to both nodes in the next line. The number of routes is $O(2^{n/2})$. If we put B nodes in each vertical line, the number becomes $O(B^{n/B})$.

Since this is a PSPACE problem, we need to develop efficient heuristic algorithms to overcome the shortcomings of centralized algorithm. We propose an **E**TX-**D**elay **G**r**E**edy (EDGE) algorithm, which is based on directed diffusion. We let each node maintain a table that records the information about each sub-path from which it could receive exploratory data packets. Only the flow from the best sub-path is allowed to propagate to the next hop. Each node except the source runs the pseudo code in Table 4. The packet format and local table format are shown in Table 2 and Table 3 respectively. $ETX(0)$, $ETX(1)$ and $ETX(2)$ are used to compute ETX_p. At each link, its ETX value will be assigned to the $ETX(n\%3)$ field, where n is the hop number.

Table 2. Packet format. *n* is the hop number. *T1* is the timestamp at which the source sent the packet.

n	$ETX(0)$	$ETX(1)$	$ETX(2)$	ETX_p	$T1$

Table 3. Local table format. The third field *Report* indicates whether to report this flow to the next hop.

Last hop ID	$Cost_p$	Report (T/F)

EDGE is a distributed algorithm that dynamically selects suitable sub-path. The final route is determined at the sink node. The number of candidate routes at the sink node is *O(M)*, where *M* is the maximum number of neighbours for each node. In this way, we reduce both the time complexity at the sink node and the minimum memory space a sensor node needs from $O(A^n)$ to *O(M)*.

Unfortunately, this algorithm may not always find the best route since it does not have sub-solution optimality property. Despite this, the performance of the algorithm shows significant improvements over existing algorithms. EDGE may eliminate the sub-path which constitutes the best path at certain cross points. This is illustrated in an example in Figure 5.

Table 4. Pseudo code of EDGE Algorithm.

$MinCost \leftarrow MAXIMUM$
for each flow coming within $INTERVAL$
Set Last hop ID
$ETX(n\%3) = ETX_i$
$ETX_p = max(ETX_p, ETX(0) + ETX(1) + ETX(2))$
$T2 \leftarrow Systemtime$
$Cost_p = (ETX_p)^\alpha \times (T2 - T1)^\beta$
Record Last hop ID, $Cost_p$ to local table
if $Cost_p < MinCost$
Set Report to $TRUE$
Set $Report$ of previous $MinCost$ to $FALSE$

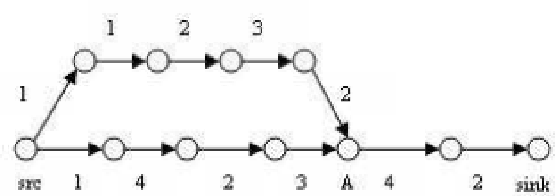


Figure 5. An example showing that $Cost_p$ does not have sub-solution optimality property. The numbers denote the ETX values of those links. Delay for each link is 1. Suppose $\alpha = 1$ and $\beta = 1$.

In Figure 5, Node *A* makes a decision whether the top flow or the bottom flow should continue. $Cost_p$ of the top flow is 35, less than that of the bottom flow which is 36. According to EDGE, only the top flow is sent to the next hop. However, if the entire route from source to sink is considered, $Cost_p$ of the bottom one is 54, less than that of the top flow which is 56. This means we would have made the wrong decision at Node *A*.

4.3. "Look-ahead" Algorithm

In order to improve the performance of EDGE and solve the above problem, we propose a "look-ahead" algorithm which helps to predict $Cost_p$ of the current sub-path. When interests are diffused, neighbour nodes exchange ETX information. Each node keeps the ETX information of its neighbours within *C* hops. *C* is a positive integer. The detailed algorithm is illustrated in Table 5. ETX_i is the ETX value of the current link.

Table 5. Pseudo code of "Looking-ahead" Algorithm

$MinCost \leftarrow MAXIMUM$
for each flow coming
$ETX(n\%3) = ETX_i$
$ETX_p = max(ETX_p, ETX(0) + ETX(1) + ETX(2))$
$T2 \leftarrow System \quad time$
$MinCost'' \leftarrow MAXIMUM$
for each downstream till the C-hop neighbour
$ETX_p'' = max(ETX_p, \max_{k=1}^{C} (\sum_{j=i+k-2}^{i+k} ETX_i))$
$delay_p'' = \frac{T2-T1}{n} \times (n + C)$
$Cost_p'' = (ETX_p'')^{\alpha} \times (delay_p'')^{\beta}$
if $Cost_p'' < MinCost''$
$MinCost'' \leftarrow Cost_p''$
$Cost_p \leftarrow MinCost''$
Record Last hop ID, $Cost_p$ to local table
if $Cost_p < MinCost$
Set $Report$ to $TRUE$
Set $Report$ of previous $MinCost$ to $FALSE$

$delay_p$ of the sub-path with *C* more hops is predicted by projection from the current sub-path. We assume that delay changes gradually from hop to hop. At a cross point, the sub-path with *C* more hops which has the lowest predicted $Cost_p$ is determined. This predicted sub-path is more likely to be selected at the next cross point than other sub-paths that share the same sub-path between the source and this cross point.

This look-ahead algorithm still cannot guarantee that the best route is always selected due to the absence of the sub-solution optimality property. However, it predicts the trend of the cost variation, which makes the sub-path selection at each cross point more accurate and robust. The overhead of computing all the sub-routes to all *C*-hop neighbours is high, especially in the framework of the data-centric protocol, such as directed diffusion.

5. SIMULATION STUDY

We did both packet level simulation study [43] and NS2 simulation to demonstrate the effectiveness of EDGE, with $Cost_p$ metric, in achieving much higher throughput and lower delay relative to the conventional implementation of directed diffusion. We evaluate the performance on different topologies with lossy links. The following sub-sections describe our methodology and the evaluation metrics.

5.1. Simulation Methodology

Our simulation setup consists of a sender, a receiver, and a traffic generator. Nodes are evenly distributed in a grid topology, in which the sender is at the bottom right corner and the receiver is at the top left corner. We simulate the algorithms using a modification of directed diffusion release 3.2.0 in ns2.29.

We use the IEEE 802.11 protocol as the MAC layer. The channel has a bandwidth of 2Mb/s. The transmission range is 250m and the interference range is 550m. The distance between the closest pair of nodes is 123m. The maximum number of link layer retransmissions is seven, after which the packet is dropped. Packets are sent as UDP packets in the transport layer.

CBR traffic is generated to simulate video streams. Most encoding schemes are CBR-based algorithms, such as MPEG-2 TM5 or MPEG-4 VM Q2 [44]. Even if the encoder generates VBR (variable-bit-rate) traffic, there are methods to optimize its transmission on CBR channel [45]. Usually, it is difficult for sensor nodes to implement the sophisticated video coding techniques used in the MPEGx or H.26x series [46]. Encoding/compression schemes suitable in WMSNs are divided into three categories [11]: JPEG with Change/Difference Coding, Distributed Source Coding and Multi-layer Coding with Wavelet Compression. We omit the encoder and the decoder in this simulation by only generating CBR packets with the size of 500 bytes (the packet size 533 bytes is discussed in earlier sections. We use 500 to make it simple) to simulate one frame so that we could design a generic protocol for video data transmission in WMSNs. The minimum traffic rate is 40 packets per second, which exceeds 15 packets per second, the lowest throughput requirement of H.264 as mentioned above.

As defined earlier, *ETX* is the predicted number of data transmissions required to send a packet over a given link. Thus *ETX* measures the link loss ratios, asymmetry in the loss ratios between the two directions of each link and interference among successive links of a path. Typical implementations like MintRoute, the standard routing protocol within TinyOs, use packet based snooping to compute the ratio of number of packets received to the total number of packets transmitted over a link and use this value to compute *ETX*. Since the quality of the link varies over time, *ETX* needs to be updated periodically and may take historical variations into account. In the current implementation, however, we assume the *ETX* of a given link to vary only a little. In ns2, error model simulates link-level errors or loss and errors are generated from a simple model with packet error rate in our simulation. We insert an error model whose error rate follows uniform distribution with a certain ratio (from 0.2 to 0.7, step 0.1 in our simulation) over outgoing wireless channels to each node which are not in the top or right boundaries and keep incoming channels with no error. Consider a link from node A to node B ($A \rightarrow B$). T_A and R_A are the error rates of node A's outgoing channel and incoming channel, respectively. T_B and R_B of node B are defined the same as node A. Link delivery ratios can be computed as follows.

$$d_f = (1 - T_A) \times (1 - R_B),\ d_r = (1 - T_B) \times (1 - R_A) \qquad (5)$$

Unlike traditional implementation of directed diffusion where intermediate nodes select the first link on which the exploratory data arrived, intermediate nodes in EDGE starts a timer on receiving the first exploratory data packet. It then buffers all incoming exploratory data packets until the *INTERVAL* timer expires. On expiration of the timer, the node computes the cost for each exploratory packet received and selects the link from which the exploratory data with least cost arrived. It then forwards that packet to all the neighbours who had earlier expressed an interest for the named data. While forwarding the exploratory data, the node also updates the ETX_p field in the packet header. When the sink reinforces the link from which the exploratory data with least cost arrived, the reinforcement propagates all the way back to the source through the least cost links recorded by all the intermediate nodes. Data from the source is then drawn towards the sink on this reinforced path. Periodically the source marks one of its data packets as exploratory and floods it to the sink, in order to discover a better path if it exists.

Our algorithm is implemented without synchronization although its use could simplify the implementation. To do this, each node scales the time period between the first and the last exploratory data within timeout. $delay_p$ is computed with a relative timestamp, which is an integer between an arbitrary number C_0, such as 10, and *INTERVAL*. Suppose there are N flows reaching a certain node when the timer expires. The timestamps are T_1, T_2, ... , T_N respectively ($T_1 < T_2 < ... < T_N$). *Flow 1* (T_1) is assigned C_0 as $delay_p$ in the $Cost_p$ formula, *Flow N* (T_N) is assigned the $delay_p$ value of *INTERVAL* and *Flow i* (T_i) is assigned the $delay_p$ computed by the following formulas:

$$INTERVAL - \frac{(T3 - Ti)(INTERVAL - C_0)}{(T3 - T1)} \tag{6}$$

or

$$C_0 + \frac{(Ti - T1)(INTERVAL - C_0)}{T3 - T1} \tag{7}$$

5.2. Evaluation Metrics

We evaluate our algorithm with network sizes of 100 (10 by 10), 144 (12 by 12), 196 (14 by 14), 256 (16 by 16), 324 (18 by 18), 400 (20 by 20) nodes configured in regular grids where the top and right boundaries links are lossless and all other links are lossy (Figure 6).

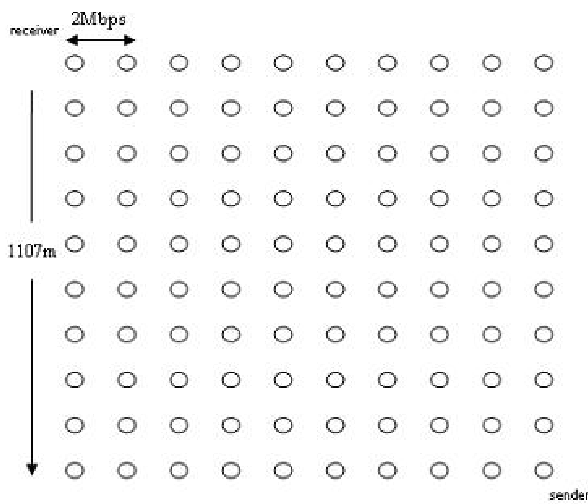


Figure 6. A 10-by-10 Grid Topology

Table 6. Parameter List of Our Simulation

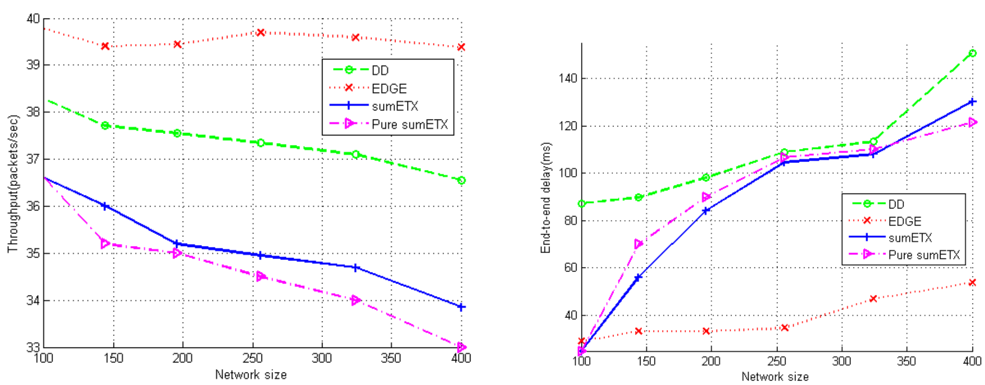
Data Packet Size	500 bytes
Number of nodes	100, 144, 196, 256, 324, 400
Neighbours per node	20
Outgoing channel error rate	0.2, 0.3, 0.4, 0.5, 0.6, 0.7
Traffic intervals	25ms, 17ms, 12ms, 10ms, 8ms, 6ms, 5ms
Network density	1 node per $(123m)^2$

We measure the performance of several simulation runs with varying link delivery ratios, traffic rates and packet sizes. The distributed algorithms are executed in ns2. Each scenario is executed 11 times by setting random seeds to generate different start time points. The simulation time of each run is 360s in ns2. The parameters are shown in Table 6. The performance metrics we use to compare the algorithms are throughput (packets per second), end-to-end delay (ms), and jitter (ms). Jitter is defined as the difference between the delay that a packet has experienced and the target delay for the flow [47]. In this paper, we use the end-to-end delay of the first packet of a synchronization unit [48] as the target delay. The synchronization unit used in our simulation is 360s, the whole simulation time. We take the maximum of each packet's jitter as the jitter of the current run and we take the average of each run's jitter as the jitter of current setting. $jitter_{ji}$ is the jitter of the ith packet in the jth run. r is the number of runs for each setting. p is the number of packets received in each run. The average jitter we measure is formulated as follows.

$$\left(\sum_{j=1}^{r}\left(\mathop{Max}_{i=1}^{p}(jitter_{ji})\right)\right)/r \tag{8}$$

6. Performance Evaluation

We compare the performance of EDGE against the traditional implementation of directed diffusion, original *ETX*, and the original *ETX* with delay. The performance metrics are throughput, delay for all the scenarios described above and jitter for different delivery ratios. The first result is that the throughput of EDGE is much better than that of standard diffusion, in each of the six network sizes as shown in Figure 7a. The original *ETX* with and without delay performs the worst since the sum of *ETX*s on each path is exaggerating the ETX metric of the path, which leads to the wrong path selection. It also shows that as the network size increases, throughput of EDGE decreases more slowly than the other protocols.

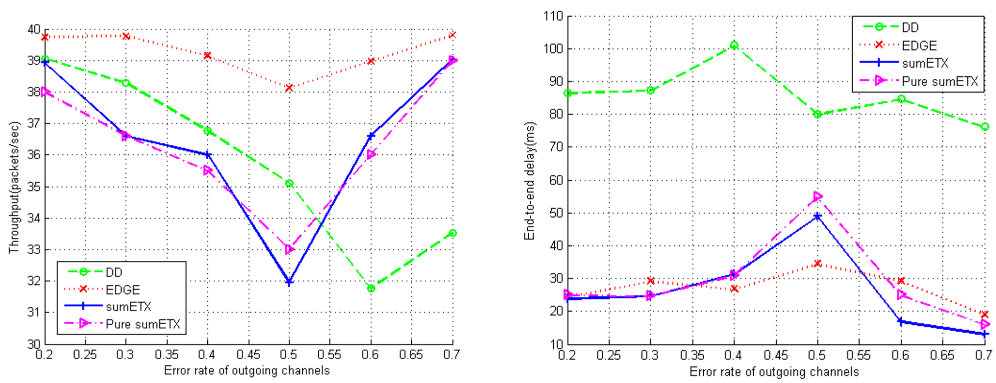


(a) (b)

Figure 7. Throughput and delay comparison between EDGE and directed diffusion in grid topologies. The error rate of outgoing channels follows uniform distribution with expected value of 0.3. $\alpha=1$, $\beta=1$ in $Cost_p$. Packet size is 500 bytes.

Figure 7b compares their end-to-end delay and shows that the delay performance of EDGE is also much better than the other three. Besides, delay of diffusion rises faster than that of EDGE. The sum of *ETX* with delay and pure sum *ETX* are in between. They do not differ much because the sum of ETX is a large number compared to the normalized delay which is between 10 and 60. Both differences of delay and throughput between EDGE and directed diffusion can be explained by the better (border) links EDGE reinforces. Lossy links not only tend to drop packets; they also incur longer delay due to retransmissions. The larger the network size, the more throughput and delay will be degraded. In [7], the optimal throughput-delay tradeoff is given by $D(n)=\Theta(nT(n))$, where $T(n)$ and $D(n)$ are the throughput and delay respectively. We can see $D(n)/T(n)$ increases in Figure 7 with the growth of network. However, we can hardly reach this optimal tradeoff even for EDGE since we do not use packet scheduling and EDGE still relies on excessive flooding.

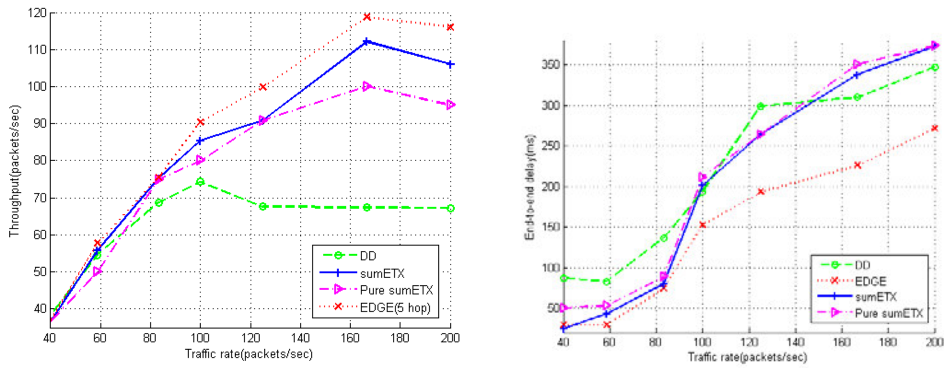
We also investigate the performance of EDGE with different delivery ratios of the lossy links. In Figure 8a, EDGE achieves higher throughput and lower delay than directed diffusion with all delivery ratios. The throughput of EDGE outperforms that of directed diffusion most when the delivery ratios of lossy links are rather low (error rate of 0.6 and 0.7) because EDGE tends to select the route with lossless links. The throughput of EDGE drops as the error rate decreases from 0.7 to 0.5, increases as the error rate decreases from 0.5 to 0.3, and becomes stable from then on. On the other hand, the trend of delay is generally reversed — it increases as the error rate increases from 0.2 to 0.5 and then decreases for error rates higher than 0.5. The reason why most of these protocols perform the worst at the moderate error rate of 0.5 scenario compared to 0.6 and 0.7 is that exploratory data which flow through those lossy links (error rate 0.5) has a higher probability of being reinforced. When the error rate goes below 0.5, throughput is not affected much by lossy links because they are closer to lossless links in delivery ratios than when the error rate is 0.5.



(a) (b)

Figure 8. Throughput and delay comparison between EDGE and directed diffusion in 10-by-10 grid with different delivery ratios of the lossy links. Traffic rate is 40 packets per second. $\alpha=1$, $\beta=1$ in $Cost_p$. Packet size is 500 bytes.

To test EDGE's sensitivity to the change in traffic rate, we use *CBR* (Constant Bit Rate) traffic. In Figure 9a, we plot 5-hop *ETX* (maximum sum of *ETX*s of five consecutive hops) instead of the standand 3-hop *ETX* in order to get the best performance. The number of hops we need to consider depends on the interference range and the distance between node pairs. In Figure 9b, as the traffic rate increases from 40 to 58.8 packets per second, the delay of EDGE gradually increases. When the traffic continues to increase, delay jumps up drastically, probably due to congestion that occurs at 58.8 packets per second. EDGE always performs better than directed diffusion in both delay and throughput. The throughput of EDGE increases with the growth of traffic rate until 166.7 packets per second while directed diffusion stops rising at 100 packets per second. There is a turning point at 100 packets per second when the throughput of EDGE increases rapidly, but after that the rate of increase of the throughput is reduced a little. It means EDGE is more tolerant of congestion than directed diffusion. The highest throughput can be achieved at 166.7 packets per second because the sink is able to receive enough packets to maintain reliability although a large number of data packets are injected within one time unit. When the congestion is less severe (between 58.8 and 166.7 packets per second), the throughput of EDGE keep rising because the packets dropped by congestion are compensated by packets transmitted at a higher traffic rate. As the congestion increases, the throughput drops. Compared with standard diffusion, EDGE achieves almost 1.5 times higher throughput than standard diffusion when the traffic rate is 166.7 packets per second.



(a) (b)

Figure 9. Throughput and delay comparison between EDGE and directed diffusion in 10-by-10 grid with different traffic rates. The error rate of outgoing channels follows uniform distribution with expected value of 0.3. $\alpha=1$, $\beta=1$ in $Cost_p$. Packet size is 500 bytes.

We compare the network delivery ratios of EDGE and DD at three traffic rates: 125, 166.7 and 200 packets per second in Figure 10. It further shows that the percentage of packets received decreases when the traffic rate increases. EDGE always has higher delivery ratios than DD. When the traffic rate increases, nevertheless, the difference gets smaller because both EDGE and DD suffer from severe congestion.

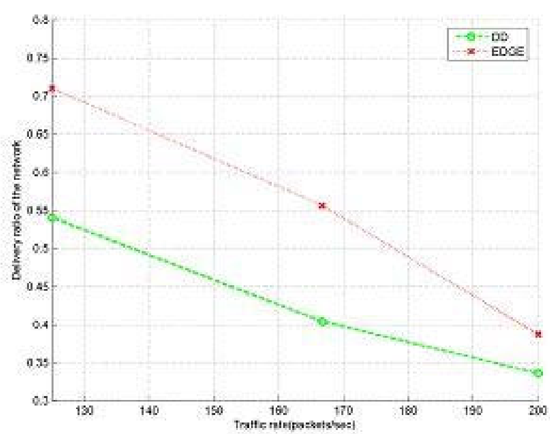


Figure 10. Network delivery ratio comparison between EDGE and directed diffusion in a 10-by-10 grid network. The error rate of outgoing channels follows uniform distribution with expected value of 0.3. $\alpha=1$, $\beta=1$ in $Cost_p$. Packet size is 500 bytes

We compare EDGE and diffusion in jitter. As mentioned earlier, jitter, in our simulation, is defined as the difference between the delay that a packet has experienced and the delay of the first packet for the flow. EDGE suffers from jitter much less than directed diffusion, as shown by the result that jitter of directed diffusion is 8 times larger than that of EDGE (on the average). EDGE achieves best performance when the links are very good (error rate less than 0.2) and very bad (error rate more than 0.7), which can be explained by the lossless links being reinforced in preference to very bad links. For links with medium error rate (0.3, 0.4 and 0.5), links with moderate error may have a higher probability to be selected than worse links (error rate: 0.6 or 0.7), which contributes to the worst jitter.

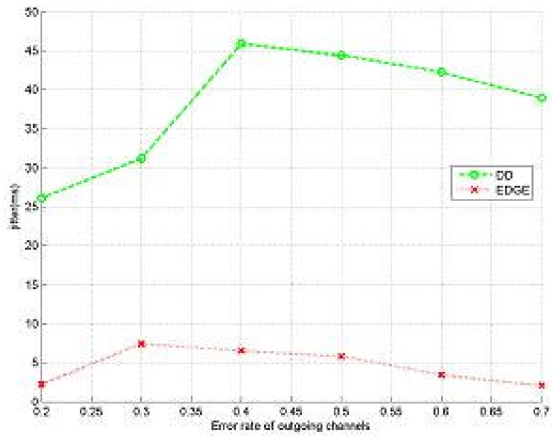
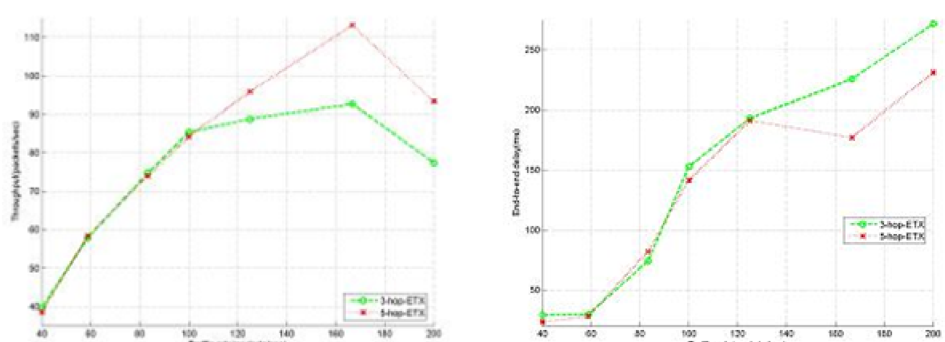


Figure 11. Jitter comparison between EDGE and directed diffusion in a 10-by-10 grid network with different delivery ratios. $\alpha=1$, $\beta=1$ in $Cost_p$. Traffic rate is 40 packets per second. Packet size is 500 bytes.

To improve the performance of EDGE, we use a more accurate ETX_p formula based on the distance between the closest pair of nodes of 123m and the interference range in ns2 of 550m. Since $\lceil 550/123 \rceil = 5$, nodes within 5 hops interfere with each other. We then modify the ETX_p formula as follows:

$$ETX_p = \underset{i=0}{\overset{N-3}{Max}}\left(\sum_{j=i}^{i+4} ETX_j\right) \tag{9}$$

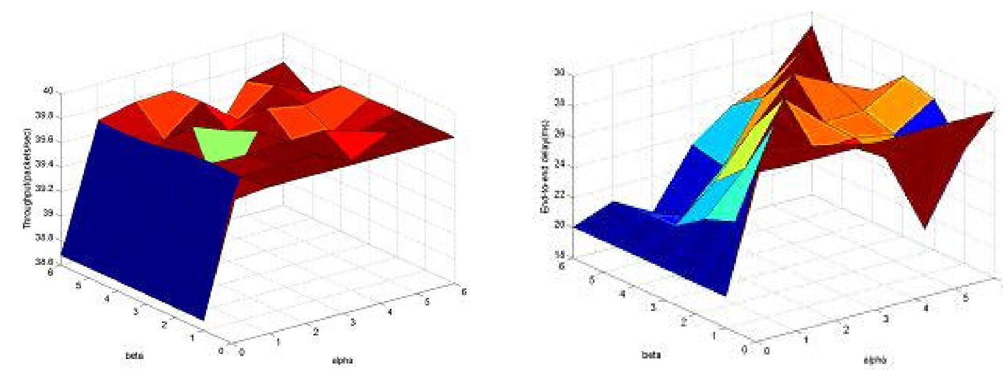
We compare throughput and delay between EDGE with different ETX_p formulae, where the 5-hop case is closer to previous experimental setup. It is observed that 5-hop EDGE has higher probability of reinforcing lossless links than its counterpart. As shown in Figure 12, there is little difference in the throughput and delay at low traffic rate. When the traffic rate increases, the benefits of 5-hop EDGE becomes evident.



(a) (b)

Figure 12. Throughput and delay comparison between 3-hop EDGE and 5-hop EDGE in a 10-by-10 grid network with different traffic rates. The error rate of outgoing channels follows uniform distribution with expected value of 0.3. $\alpha=1$, $\beta=1$ in $Cost_p$. Packet size is 500 bytes.

We adjust the α and β values in $Cost_p$ formula trying to improve throughput or delay of our algorithm. In Figure 13a, throughput increases rapidly when ETX is taken into account α from 0 to 1). For other combinations, there is no significant improvement in throughput. It achieves the highest throughput when only ETX is considered β=0). Likewise, delay is lowest when α=0. It is different from directed diffusion because EDGE has an adaptive timer at each node. ETX somewhat helps decrease delay, as we can see in Figure 13b that peaks at α=3 or 4 instead of 6. However, pure ETX (β=0) worsens delay to a large extent.



(a) (b)

Figure 13. Throughput and delay comparison with different α and β values in $Cost_p$ of EDGE in a 10-by-10 grid. The error rate of outgoing channels follows uniform distribution with expected value of 0.3. α=1, β=1 in $Cost_p$. Packet size is 500 bytes. Traffic rate is 40 packets per second.

In general, EDGE outperforms standard directed diffusion in throughput, delay and jitter, which at least satisfies the basic QoS requirements of video data transmission. For example, when we use 14 by 14 as the network size, 500 bytes for a packet, 40 packets per second as the traffic rate, and 70% uniform distribution for lossy links, the throughput of EDGE is 38.86 packets per second, which is equal to 155 kbps, much larger than 64 kbps. The end-to-end delay with the same setting above is 59.4 ms. Delay tolerance of real-time multimedia streaming depends on the client. 200 ms is used in [49], which is much longer than our experimental delay. Jitter is also one of the QoS requirements of the client. The average jitter, in our simulation, is within 10 ms, which is much smaller than the maximum jitter of 32 ms in [50]. We plan to test our protocol in a PC104 network test-bed and observe whether EDGE is still able to satisfy the QoS requirements mentioned earlier in video data transmission. Our current results also demonstrate the tradeoff between throughput and delay.

7. CONCLUSIONS

We have described EDGE - a greedy algorithm for selecting routes based on link costs that include both ETX and delay which enable multimedia data to be transmitted over sensor networks by increasing throughput and decreasing jitter and end-to-end delay. Our ns2 simulation results show that the throughput performance of EDGE is on the average 15% higher, jitter is 8 times lower and delay is 20% lower than directed diffusion for the scenarios we investigated. The main reason is EDGE has a higher probability of selecting routes with better links, which reduces retransmission (resulting in longer delay), packet dropping (giving rise to lower throughput) and link instability (related with jitter). Not only does EDGE provide better performance, the algorithm can also be easily implemented in the standard directed diffusion software to support demanding applications, such as video sensor networks or WMSNs.

Several aspects of the algorithm could still be improved in our future work. Currently, we only consider intra-flow interference when computing ETX for a path in single source and single sink scenarios. In future, we will develop algorithms that take into account interflow interference with multiple sources and multiple sinks. We are working on a multi-path version of this protocol to support multiple description (MD) video coding. Variable amount of workload, which happens in real multimedia transmission system, will be used to test EDGE's sensitivity to system workload. We will also consider incorporating other cost factors, such as SNR (Signal to Noise) which is a physical layer metric and mobility. Random topologies with random error rates of outgoing channels will be further investigated.

REFERENCES

[1] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, 2003.

[2] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, vol. 353, 1996.

[3] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*. New York, NY, USA: ACM, 1994, pp. 234–244.

[4] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wirel. Netw.*, vol. 11, no. 4, pp. 419–434, 2005.

[5] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, Mar 2000.

[6] J. Park and S. Kasera, "Expected data rate: an accurate high-throughput path metric for multi-hop wireless routing," in *Proc. of second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 2005.

[7] A. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in wireless networks," *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. –475, 7-11 March 2004.

[8] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN '05*: *Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005, p. 48.

[9] "Crossbow technology," http://www.xbow.com/, accessed July 24, 2007.

[10] "Pc104.com," http://www.32.com/., accessed July 24, 2007.

[11] M. R. Satyajayant Mishra and G. Xue, "A survey of multimedia streaming in wireless sensor networks," under submission.

[12] J. L. W. D. Agathangelou, B. P. Lo and G.-Z. Yang, "Self-configuring video-sensor networks," in *Proceedings of the 3rd International Conference on Pervasive Computing,* 2005.

[13] M. Chu, J. Reich, and F. Zhao, "Distributed attention in large scale video sensor networks," *Intelligent Distributed Surveilliance Systems*, *IEE*, pp. 61–65, 23 Feb. 2004.

[14] "Video sensor network laboratory at UCR receives federal funding," http://www.newsroom.ucr.edu/cgibin/display.cgi?id=1298, accessed July 24, 2007.

[15] W.-c. F. C. P. Wu-chi Feng, Jonathan Walpole, "Moving towards massively scalable video-based sensor networks," in *Workshop on New Visions for Large-Scale Networks: Research and Applications*, 2001.

[16] "Maximizing the life of wireless video sensor networks," http://www.ece.vt.edu/news/ar04/video.html., accessed July 24, 2007.

[17] B. M. Kogut, G. and H. Everett, "Using video sensor networks to command and control unmanned ground vehicles," in *AUVSI Unmanned Systems in International Security 2003 (USIS 03)*, 2003.

[18] "Wireless multimedia sensor networks," http://www.ece.osu.edu/ ekici/, accessed July 24, 2007.

[19] J. K. T.D.C. Little, P. Ishwar, "A wireless video sensor network for autonomous coastal sensing," in *Proc. Conference on Coastal Environmental Sensing Networks*, *(CESN 2007)*, 2007.

[20] "Quality of service(QoS) for wireless sensor networks," http://cobweb.ecn.purdue.edu/RVL/Projects/SensorNetworksQoS/index.htm.

[21] S. K. F. Hu, "Multimedia query with QoS considerations for wireless sensor networks in telemedicine," in *Proc. of Society of Photo-Optical Insrumentation Engineers ?Intl. Conf. on Internet Multimedia Management Systems*, 2003.
[22] L. Savidge, H. Lee, H. Aghajan, and A. Goldsmith, "Qos-based geographic routing for event-driven image sensor networks," *Broadband Networks*, *2005 2nd International Conference on*, pp. 991–1000 Vol. 2, 3-7 Oct. 2005.
[23] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "Rap: A real-time communication architecture for large-scale wireless sensor networks," Charlottesville, VA, USA, *Tech. Rep*., 2002.
[24] Tian He, John A. Stankovic, Chenyang Lu and Tarek F. Abdelzaher, "A spatiotemporal communication protocol for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 10, pp. 995–1006, 2005.
[25] Emad Felemban, Chang-Gun Lee and Eylem Ekici, "Mmspeed: Multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006,.
[26] D. S. J. D. Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Performance of multihop wireless networks: shortest path is not enough," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 83–88, 2003.
[27] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *SIGCOMM '04*: *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2004, pp. 133–144.
[28] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in MobiCom '04: *Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2004, pp. 114–128.
[29] C. C. J. Tang, G. Xue, "Interference-aware routing in multihop wireless networks using directional antennas," in *Proceedings of IEEE INFOCOM*, 2005.
[30] J. Tang, G. Xue, and C. Chandler, "Interference-aware routing and bandwidth allocation for QoS provisioning in multihop wireless networks: Research articles," *Wirel. Commun. Mob. Comput.*, vol. 5, no. 8, pp. 933–943, 2005.
[31] C. D. T. Y. Z. Chiranjeeb Buragohain, Subhash Suri, "Improved throughput bounds for interference-aware routing in wireless networks," in *Proc. 13th Computing and Combinatorics Conference*, 2007.
[32] G. H. B. W. E. Weiss, O. Klein, "Capacity and interference aware ad hoc routing in multi-hop networks," in *Proceedings of 12th European Wireless conference*, 2006.
[33] F. A. Liran Ma, Qian Zhang and X. Cheng, "Diar: A dynamic interference aware routing protocol for ieee 802.11-based mobile ad hoc networks," in *International Conference on Mobile Ad-hoc and Sensor Networks (MSN) 2005*, LNCS, 2005.
[34] H. Mahmood and C. Cornaniciu, "Interference aware routing for cdma wireless ad hoc networks," Military Communications Conference, 2005. *MILCOM 2005*. IEEE, pp. 1059–1063 Vol. 2, 17-20 Oct. 2005.
[35] T. ElBatt and T. Andersen, "Cross-layer interference-aware routing for wireless multi-hop networks," in *IWCMC '06: Proceedings of the 2006 international conference on Wireless communications and mobile computing*. New York, NY, USA: ACM, 2006, pp. 153–158.
[36] D.-Q. Nguyen and P. Minet, "Interference-aware QoS OLSR for mobile ad-hoc network routing," in *SNPD-SAWN '05*: *Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 428–435.
[37] A. Bader and E. Ekici, "Throughput and delay optimization in interference-limited multihop networks," in *MobiHoc '06*: *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2006, pp. 274–285.
[38] Y. Fang and B. McDonald, "Dynamic codeword routing (dcr): A cross-layer approach for performance enhancement of general multi-hop wireless routing," in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks 2004 IEEE SECON 2004*, 2004.
[39] "H.264/mpeg-4 avc," http://en.wikipedia.org/wiki/H.264.

[40] J. Li, C. Blake, D. S. D. Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2001, pp. 61–69.
[41] S. Mao, D. Bushmitch, S. Narayanan, and S. Panwar, "MRTP: a multiflow real-time transport protocol for ad hoc networks," *Multimedia, IEEE Transactions on*, vol. 8, no. 2, pp. 356–369, April 2006.
[42] J. Kleinberg and E. Tardos, Algorithm Design. *Pearson-Addison Wesley*, 2005.
[43] S. Li, A. Lim, S. Kulkarni, and C. Liu, "EDGE: A routing algorithm for maximizing throughput and minimizing delay in wireless sensor networks," *Military Communications Conference, 2007. MILCOM 2007*. IEEE, pp. 1–7, 29-31 Oct. 2007.
[44] J. Sun, W. Gao, and Q. Huang, "A novel fgs base-layer encoding model and weight-based rate adaptation for constant-quality streaming," in *ICIG '04: Proceedings of the Third International Conference on Image and Graphics*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 373–376.
[45] M. K. R.I Chang, M. Chen and J. Ho, "Optimizations of stored vbr video transmission on cbr channel," in *SPIE VVDC*, 1997.
[46] C.-F. Chiasserini and E. Magli, "Energy consumption and image quality in wireless video-surveillance networks," Personal, Indoor and Mobile Radio Communications, 2002. *The 13th IEEE International Symposium on*, vol. 5, pp. 2357–2361 vol.5, 15-18 Sept. 2002.
[47] D. Ferrari, "Client requirements for real-time communication services," *IEEE Communications*, 1990.
[48] Z. Wang and J. Crowcroft, "Analysis of burstiness and jitter in real-time communications," in *SIGCOMM '93: Conference proceedings on Communications architectures, protocols and applications*. New York, NY, USA: ACM, 1993, pp. 13–19.
[49] A. Begen, Y. Altunbasak, and O. Ergun, "Fast heuristics for multi-path selection for multiple description encoded video streaming," *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 1, pp. I–517–20 vol.1, 6-9 July 2003.
[50] A. La Corte, A. Lombardo, S. Palazzo, and G. Schembra, "A feedback approach for jitter and skew enforcement in multimedia retrieval services," Global Telecommunications Conference, 1995. *GLOBECOM '95., IEEE*, vol. 2, pp. 790–794 vol.2, 14-16 Nov 1995.

Authors

Shuang Li is a PhD student in the department of Computer Science and Engineering at the Ohio State University. She received her M.S. degree from Auburn University, Auburn, Alabama in 2008. Her research interests are routing metrics in wireless sensor networks, power control and resource allocation in MANETs and VANETs.

Cong Liu is a PhD student in the department of Computer Science at the University of North Carolina – Chapel Hill. He received his Master's degree from the department of Computer Science and Software Engineering at Auburn University in Dec. 2007. His current research interests are real-time systems and scheduling theory on real-time multicore platforms.

Alvin Lim is currently an associate professor of computer science and software engineering at Auburn University. He received his Ph.D. degree in computer science from University of Wisconsin at Madison in 1993. His research interests include self-organizing sensor networks, mobile and pervasive computing, network security, wireless networks, reliable and dynamically reconfigurable distributed systems, complex distributed systems, mobile and distributed databases, distributed operating systems, and performance measurement and analysis. He has published widely in journals and conferences in these networking and distributed systems areas. He is a subject area editor of the International Journal of Distributed Sensor Networks. His work had been supported by the National Science Foundation, the DARPA SensIT program, U.S. Air Force Research Lab and the U.S. Army.