EDGE: A ROUTING ALGORITHM FOR MAXIMIZING THROUGHPUT AND MINIMIZING DELAY IN WIRELESS SENSOR NETWORKS

Shuang Li, Alvin Lim, Santosh Kulkarni and Cong Liu

Auburn University, Computer Science and Software Engineering
Auburn, AL

ABSTRACT

Wireless sensor networks are distributed event-based systems with severe energy constraints, variable quality links, low data-rate and many-to-one event-to-sink flows. Communication algorithms for sensor networks, such as directed diffusion, are designed to operate efficiently under these constraints. However, directed diffusion is not efficient in more challenging domains, such as video sensor networks, because of the high throughput and low 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. Simulation results with CBR (constant bit rate) traffic show that our proposed distributed algorithm selects routes that give better throughput than those reinforced by standard directed diffusion, while maintaining low delay.

KEYWORDS

Wireless sensor networks, ETX, delay, throughput, directed diffusion, routing, greedy algorithm

INTRODUCTION

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. The successes of these sensor networks have motivated even more challenging applications, thus increasing the requirements for these sensor networks. For instance, video sensor networks requires larger amount of real-time multimedia data to be disseminated with low latency, high throughput and high delivery ratio. The main challenge is to develop a practical data-centric networking algorithm that can maximize throughput and minimize delay in wireless sensor networking environments.

Directed diffusion [1] generally selects routes with the lowest delay. Other ad hoc routing protocols, such as DSR [2] and DSDV [8], usually use a hop count metric. Throughput is considered in some recent ad hoc protocols [3]. The design of wireless sensor network protocols is often guided by two principles – self-detection of link quality and in-network processing. This is necessary because the variability in link quality, low bandwidth of wireless links and limited memory of sensor nodes. To quantify data transmission quality in sensor networks, ETX (Expected Transmission Count), a link layer metric, can be used by the network layer. In [3], the ETX of a route is computed by adding the ETX of all the links in the route. However, using this route ETX, a slower path that has fewer hops may be inappropriately selected (provided the best path has four or more hops).

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 [5]. Our improved method for computing ETX for a route measures intra-flow interference more accurately by considering the maximum total ETX of any three consecutive links in a route. Our algorithm also considers the delay metric in selecting the best route.

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 throughput much better than and with delay as good as standard directed diffusion.

RELATED WORK

Routing metrics in wireless ad hoc networks are important considerations due to the unpredictability and heterogeneity of link qualities [7]. Existing wireless ad hoc routing protocols typically select routes using minimum hop count, e.g. DSR [2] and DSDV [8]. Directed diffusion [1] selects routes in sensor networks with the least delay. Recently, many new link quality metrics have been proposed. [9] compares the performance of the following three metrics. Adya et al. [9] 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 [9]. Expected Transmission Count (ETX) [3] 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) [10] is used for selecting channel-diverse paths and accounts for the loss rate and bandwidth of individual links. Park et al. [5] 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.

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 [3]. Bader et al. [11] 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.

Throughput-delay trade-off in the Gupta-Kumar fixed network model [4] is theoretically analyzed in [6]. Our results also show similar trade-off between throughput and delay in practical sensor network algorithms.

PROBLEM STATEMENT

Routing protocols with minimum hop-count metrics do not consider wireless links that 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. In designing a metric to take into account delay, throughput and interference for sensor networks, 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.

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, a link's ETX 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.

Definitions, Notations and Formulae

The ETX of a link is the predicted number of data transmissions required to send a packet over that link [3].

$$ETX = \frac{1}{d_f \times d_r} \qquad (1)$$

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 jth hop. The number of bottleneck links may vary according to the network density.

$$\boldsymbol{ETX_p = \underset{i=0}{\overset{N-3}{Max}}\left(\sum_{j=i}^{i+2} ETX_j\right)} \qquad (2)$$

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.

$$Cost_p = ETX_p^{\alpha} \times delay_p^{\beta} \qquad (3)$$

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:

$$\boldsymbol{INTERVAL = ETX_i \times EXPLORE\_DELAY} \qquad (4)$$

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 [11]. 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. [12] 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 in Figure 1 illustrates this principle for the node placement.

We compute the maximum summation of ETXs 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. Total ETX exaggerates the intra-flow interference and will leads to a wrong route selection.

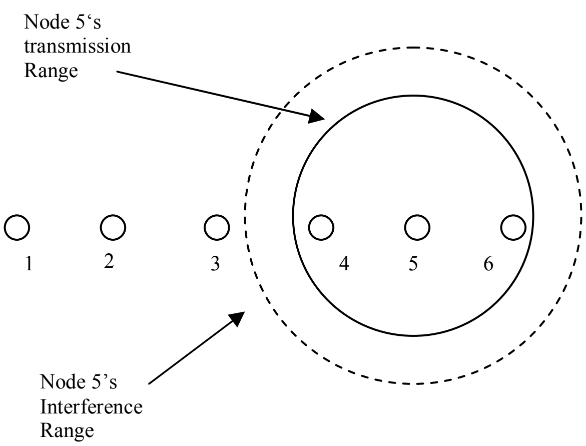


Figure 1: Transmission and interference range for a chain of nodes. Nodes within 3 hops interfere with each other.

Another reason for using our path ETX 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 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 [3] for more information.

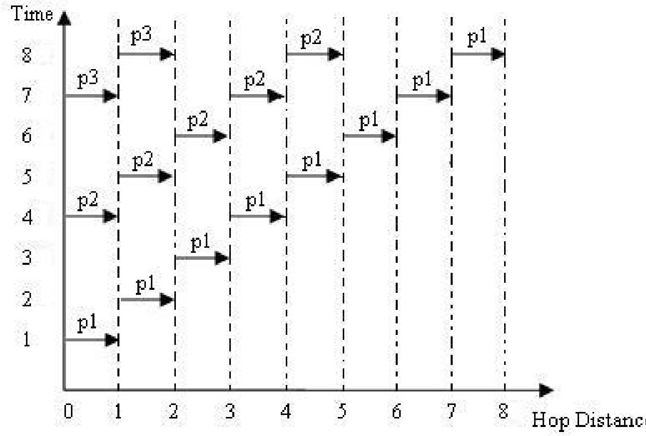


Figure 2: Pipelining mechanism of data transmission in sensor networks with intra-flow interference taken into account.

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 $Cost_p$, and the 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 = $\{(src,\ i)\}$, src is the source node; $\forall p \in P$, $dest(p) = i$, in which $p = (src, i)$.

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:

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

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

$$\forall j \in E, \qquad 1 \le ETX_j \le 8$$

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

$$\forall j \in E, \qquad 0 < d_f \le 1,\ 0 < d_r \le 1$$

The third constraint sets the minimum and maximum number of transmissions in wireless networks, which is based on the rule that the maximum transmission times in 802.11 is 7.

ALGORITHM DESIGN

In this section, we present centralized and distributed algorithms to compute the route which maximizes the throughput and minimizes the delay over lossy links in multi-hop wireless networks. We start by studying the centralized algorithm similar to that used in [3] for incorporating ETX into the initial route request in DSR. Then, we describe EDGE 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.

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. We assume that the ETX information of each link could be collected while the exploratory data are flooded. The sink node selects the route over which exploratory data are sent with the lowest $Cost_p$.

EDGE Algorithm

The previous sub-section briefly describes a centralized algorithm to find the best route in directed diffusion, which is impossible to be implemented in a real environment. 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 [13] problem.

As a result, we need to develop efficient heuristic algorithms to overcome the shortcomings of centralized algorithm. We propose **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. The packet format and local table format are shown in Figure 3 and Figure 4 respectively. ETX(0), ETX(1) and ETX(2) are used to compute ETX_p., Each link's ETX value is assigned to the ETX(n%3) field, where n is the hop number.

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. In this way, we reduce both the time complexity at the sink node and the minimum memory space of a sensor node 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.

Hop # n	ETX(0)	ETX(1)	ETX(2)	ETX_p	T1

Figure 3: Packet format. n is the hop number. T1 is the timestamp at which the source sent the packet.

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

Figure 4: Local table format. The third field "Report" indicates whether to report this flow to the next hop.

"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. $delay_p$ 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 selected at the next cross point even if other sub-paths have lower $Cost_p$ from 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.

IMPLEMENTATION

We conducted a packet level simulation study of EDGE, with $Cost_p$ metric, to systematically study its performance 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.

Simulation Methodology

We simulate the algorithms using a modification of directed diffusion release 3.2.0 in a process-level simulator.

Packets are sent as UDP packets. We use ISEE [14], a toolkit that we developed to provide graphic interface, for configuring and executing sensor networks.

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. In the current implementation, we assume ETX to be a non-varying, and statically-defined value for a given link. Since the probabilistic loss can be computed as a function of the distance based on the propagation profiles from Zhao *et al.* [15], delivery ratios of each links can be known beforehand if GPS devices are installed on all nodes.

Intermediate nodes in EDGE start 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.

Evaluation Metrics

We evaluate our algorithm with network sizes of 25 (5 by 5), 49 (7 by 7), 81 (9 by 9), 121 (11 by 11), 169 (13 by 13) nodes configured in regular grids where the left and bottom boundaries links are lossless and all other links are lossy.

We measure the performance of several simulation runs with varying $Cost_p$ formulae, link delivery ratios, timers and traffic. The distributed algorithms are executed on a network of ten computers. Each scenario is executed ten times, lasting 30 minutes each. The parameters are shown in Table 1. The performance metrics we use to compare the algorithms are throughput (packets per second) and end-to-end delay (second).

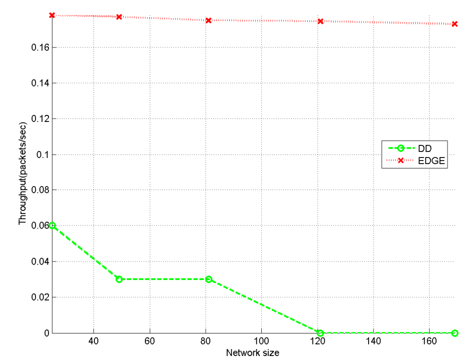
Table 1: Parameter List of Our Simulation

Data Packet Size	125 bytes
Number of nodes	25, 49, 81, 121, 169
Neighbors per node	8
Link delivery ratios	40%, 50%, 60%, 70%, 80%, 90%, 100%
Timeout values	50 msec, 100 msec, 150 msec, adaptive timeout
Traffic intervals	1s, 2s, 3s, 4s, 5s, 6s, 7s
Source	Node at the bottom right corner
Sink	Node at the upper left corner

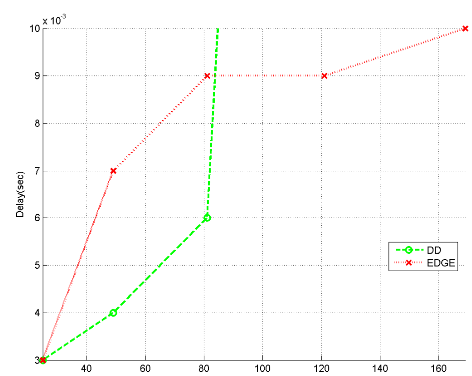
PERFORMANCE EVALUATION

We compare the performance of EDGE against the traditional implementation of directed diffusion, in terms of throughput and delay for all the scenarios described above. The first result is that the throughput of EDGE is much better than that of standard diffusion, in each of the five network sizes as shown in Figure 5(a). It also shows that as the network size increases, throughput of EDGE decreases more slowly than that of directed diffusion.

Figure 5(b) shows that the delay performance of EDGE is comparable to standard diffusion. However, standard diffusion could receive no packet at all for large network sizes of 121 and 169 because of severe flooding with large network sizes (EDGE is less affected by flooding). Thus, no delay or throughput could be computed for directed diffusion with network sizes of 121 and 169. In [6], 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. Since we do not use packet scheduling and directed diffusion rely on excessive flooding, we can hardly reach this optimal tradeoff.



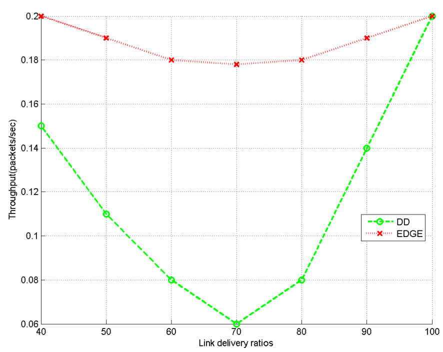
(a) Throughput with different network sizes.



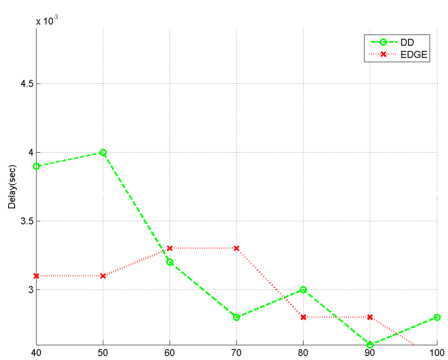
(b) End-to-end delay with different network sizes

Figure 5: Throughput and delay comparison between EDGE and directed diffusion in grid topologies with 70% as delivery ratio of the lossy links. Traffic rate is 0.2 packet per second. α=1, β=1 in $Cost_p$. Delay unit is second.

We also investigate the performance of EDGE with different delivery ratios of the lossy links. When the link quality is very poor, directed diffusion performs worse than EDGE because it may probabilistically select some lossy links which results in more retransmission and longer delays. When the lossy link's delivery ratio is above 50%, there is little difference in delay between EDGE and directed diffusion, as shown in Figure 6(b). EDGE is more likely to select the lossless boundary links for reinforcement and thus gets better throughput. As for higher delivery ratios, EDGE achieves higher throughput than directed diffusion regardless of whether perfect paths are reinforced. This means that EDGE is not as sensitive as directed diffusion to link quality deterioration as long as there exists at least one route with lossless links in the network. EDGE outperforms standard diffusion when the delivery ratios are low (40% or 50%), which shows its usefulness when wireless links are unstable.



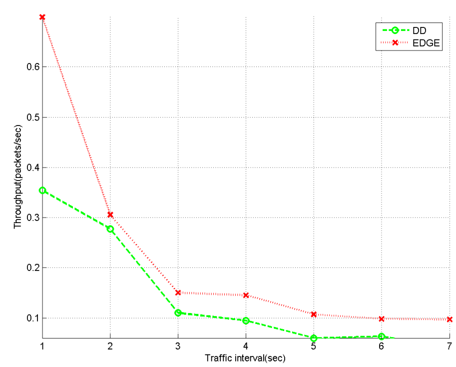
(a) Throughput with different delivery ratios of lossy links.



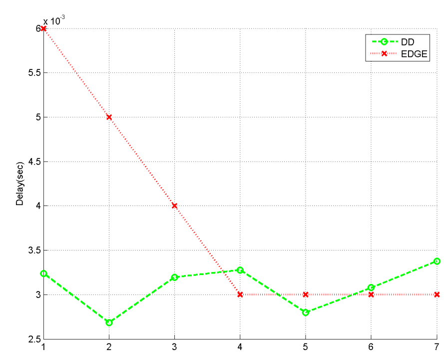
(b) End-to-end delay with different delivery ratios of lossy links.

Figure 6: Throughput and delay comparison between EDGE with an adaptive timer and directed diffusion in 5-by-5 grid with different delivery ratios of the lossy links. Traffic rate is 0.2 packet per second. α=1, β=1 in $Cost_p$.

To test EDGE's sensitivity to traffic rate, we use CBR (Constant Bit Rate) traffic. Figure 7 shows the tradeoff between throughput and delay. As the traffic rate decreases from 1 packet per second to 1/4 packet per second, EDGE achieves a drastic improvement in delay, while throughput decreases. Delay and throughput decrease slightly as the traffic rate decreases lower than 1/4 packet per second. The highest throughput can be achieved at 1 packet 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. However, high traffic rate causes congestion and results in the highest delay, as shown in Figure 7(b). Similar state transition could be found in [16]. Compared with standard diffusion, EDGE gains more than twice the throughput of standard diffusion. On the other hand, standard diffusion can always find the route with less delay than EDGE, especially when the traffic is high.



(a) Throughput with different traffic rates.



(b) End-to-end delay with different traffic rates.

Figure 7: Throughput and delay comparison between EDGE with an adaptive timer and directed diffusion in 5-by-5 grid with different traffic rates. The delivery ratio of lossy links is 70%. α=1, β=1 in $Cost_p$.

We investigate the effects of different $Cost_p$ formula on the performance (The results of adjusting α and β values are omitted due to page limit). When α=1 and β=0, only ETX_p is considered when computing the route metric, thus giving the best throughput, while delay is higher. When α=1 and β=1, both throughput and delay performance are good. Neither the case α=0, β=1 nor α=1, β=0 gives desirable delay performance. This indicates that ETX_p also contributes to improving delay.

In general, EDGE outperforms standard directed diffusion in throughput and it is also comparable to standard directed diffusion in delay in almost all scenarios. By altering the timeout value and $Cost_p$ formula, EDGE could achieve even better performance. Our results also show the tradeoff between throughput and delay. In actual application domains, the parameters in the $Cost_p$ formula should be selected appropriately based on whether the application places higher priority on throughput or delay.

CONCLUSION

We have described EDGE – a greedy algorithm for selecting routes based on link costs that include both ETX and delay. The advantage of EDGE is that it can maximize throughput and minimize delay. Our simulation results show that the throughput performance of EDGE is on the average 4 times better than the standard directed diffusion while its delay is only 20% higher (on the average) than directed diffusion for the scenarios we investigated. EDGE can also be easily implemented in the standard directed diffusion software to support demanding applications, such as video sensor networks.

Several aspects of the algorithm could still be improved in our future work. For example, we plan to investigate the utility of the algorithms for multimedia applications with multi-path transmission and evaluate the quality of service.

REFERENCES

[1] C. Intanagonwiwat, R. Govindan, and D. Estrin, *Directed diffusion: A scalable and robust communication paradigm for sensor networks*. Proc. of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00), August 2000, Boston, Massachussetts.

[2] D. Johnson, D. Maltz, and Y. Hu. *The Dynamic Source Routing Protocol for mobile ad hoc networks (DSR)*. Internet draft (work in progress), IETF, April 2003. http://www.ieft.org/internet-drafts/draft-ietf-manet-dsr-09.txt.

[3] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris. *A high-throughput path metric for multi-hop wireless routing*. Proc. of MobiCom, 2003.[4] P. Gupta and P. Kumar. *The Capacity of Wireless Networks*. IEEE Transactions on Information Theory, 46(2):388-404, March 2000.

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

[6] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, *Throughput-delay trade-off in wireless networks*. Proc. of IEEE INFOCOM, March 2004.

[7] D. Couto, D. Aguayo, B. Chambers, and R. Morris, *Performance of multihop wireless networks: Shortest path is not enough*. Proc. of the First Workshop on Hot Topics in Networks (HotNets-I), Princeton, New Jersey, October 2002. ACM SIGCOMM.

[8] C. Perkins and P. Bhagwat. *Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers*. Proc. of ACM SIGCOMM Conference (SIGCOMM '94), pages 234-244, August 1993.

[9] R. Draves, J. Padhye, and B. Zill, *Comparison of Routing Metrics for Static Multi-Hop Wireless Networks*. Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), pages 133–144, ACM Press, 2004.

[10] R. Draves, J. Padhye, and B. Zill, *Routing in multi-radio, multi-hop wireless mesh networks*. Proc. of the 10th Annual International Conference on Mobile Computing and Networking, 2004.

[11] A. Bader and E. Ekici, *Throughput and delay optimization in interference-limited multihop networks*. Proc. Of MobiHoc06, Florence, Italy, May 22-25, 2006.

[12] J. Li, C. Blake, D. D. Couto, H. Lee, and R. Morris, *Capacity of Ad Hoc Wireless Networks*. ACM MobiCom'01, pp. 61-69, July 2001.

[13] J. Kleinberg and E. Tardos, *Algorithm Design*. Pearson-Addison Wesley, 2005.

[14] Mark Ivester and Alvin Lim, *Interactive and Extensible Framework for Execution and Monitoring of Wireless Sensor Networks*, First Int. Conf. on Communication Systems Software and Middleware, IEEE/ACM, New Delhi, India, January 8-12, 2006.

[15] Jerry Zhao and Ramesh Gvindan. *Understanding packet delivery performance in dense wireless sensor networks*. Proc. of the First International Conference on Embedded Networked Sensor Systems, pages 1-13. ACM Press, 2003.

[16] Yagesh Sankarasubramaniam, Ozgur B. Akan and Ian F. Akyildiz. *ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks*. Proc. of MobiHoc, 2003.