Improving Throughput of Video Streaming in Wireless Sensor Networks

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Shuang Li

Certificate of Approval:

Xiao Qin
Assistant Professor
Computer Science and
Software Engineering

Alvin S. Lim, Chair
Associate Professor
Computer Science and
Software Engineering

Wei-Shinn Ku
Assistant Professor
Computer Science and
Software Engineering

George T. Flowers
Interim Dean
Graduate School

Improving Throughput of Video Streaming in Wireless Sensor Networks

Shuang Li

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
August 9, 2008

Improving Throughput of Video Streaming in Wireless Sensor Networks

Shuang Li

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

VITA

Shuang Li, daughter of Fu'an and Derong Li, was born October 21, 1983, in Chongqing, China. She earned her Bachelor's degree in Computer Science from Wuhan University of Technology, Hubei, China in 2005.

THESIS ABSTRACT

IMPROVING THROUGHPUT OF VIDEO STREAMING IN WIRELESS SENSOR NETWORKS

Shuang Li

Master of Science, August 9, 2008
(B.S., Wuhan University of Technology, China, 2005)

115 Typed Pages

Directed by Alvin S. Lim

Wireless sensor networks are originally 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 with 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 living, etc. 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 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. And we present an improved method for computing aggregate ETX for a path that increases end-to-end throughput. Multiple-path is one of the ways how QoS routing issues are dealt with in wired environment. Besides, some existing ad hoc routing algorithms also provide multi-path routing. Directed diffusion has been commonly used for wireless sensor networks because of its energy efficiency and scalability. However, the basic protocol only routes packets through a single path, which barely meets the throughput requirement of multimedia data. Instead, we propose a multi-path algorithm based on directed diffusion that reinforces multiple routes with high link quality and low latency. In directed diffusion, many routing messages are propagated unnecessarily and may cause different interference characteristics during route discovery phase and in the actual application data transmission phase. We propose a routing protocol that uses ID-free epidemic flooding to limit interference in conjunction with metrics for increasing throughput and reducing delay.

Acknowledgments

I would like to express my appreciation to Dr. Lim for the guidance he has provided throughout my study at Auburn. I would also like to express my gratitude to the advisory committee members, Dr. Xiao Qin and Dr. Wei-Shinn Ku.

I would also like to thank several fellow students including Raghukisore Neelisetti and Santosh Kulkarni for their significant contributions to this research.

Above all, I would like to thank my parents who helped me come to U.S. and pursue graduate study at Auburn.

Style manual or journal used Journal of Approximation Theory (together with the style known as "aums"). Bibliograpy follows van Leunen's *A Handbook for Scholars*.

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `aums.sty`.

TABLE OF CONTENTS

LIST OF FIGURES xi

1 INTRODUCTION 1

2 RELATED WORK 6
  2.1 Hardware Support 6
  2.2 On-going Video Sensor Network Research 7
  2.3 QoS issues in WMSN 7
  2.4 Routing Metrics in Wireless Sensor Networks 8
  2.5 Interference-aware Protocols 9
  2.6 Short-comings in ETX 10
  2.7 Theoretical Analysis of Throughput-Delay Tradeoff 11
  2.8 Multi-path Routing 11
  2.9 Multistream Video Coding 12
  2.10 Message Redundancy in Routing Protocols 12

3 MOTIVATIONS AND APPLICATIONS 14
  3.1 Motivation 14
  3.2 Applications 15
    3.2.1 Multimedia Surveillance 15
    3.2.2 Data-centric Storage 15
    3.2.3 Traffic Conditions and Collision Avoidance 16

4 PROBLEM STATEMENT 17
  4.1 Single-path metric $Cost_p$ 17
    4.1.1 Assumptions and Goals 17
    4.1.2 Definitions, Notations and Formulae 18
    4.1.3 Computing Path Metric 19
    4.1.4 Problem Formulation 21
  4.2 Multi-path Protocol DCHT 23
  4.3 Interference Limiting 26
    4.3.1 Overview of Directed Diffusion and Its Variants 26
    4.3.2 Route Selection Problems 27

5 PROTOCOL DESIGN 29
  5.1 EDGE Design 29
    5.1.1 Optimum Algorithm 30

5.1.2 EDGE Algorithm 31
5.1.3 Look-ahead Algorithm 34
5.2 DCHT Design 35
5.2.1 Route discovery 36
5.2.2 Parameter settings 40
5.3 Epidemic Algorithm in Diffusion 44
5.3.1 Epidemic Interest Flooding 45
5.3.2 Epidemic Exploratory Data Propagation 46

6 IMPLEMENTATION 49
6.1 EDGE implementation 49
6.1.1 Simulation Methodology 49
6.1.2 Evaluation Metrics 52
6.2 Multi-path Protocol Implementation 54
6.3 Implementation of Epidemic Interest Flooding 56

7 PERFORMANCE EVALUATION 58
7.1 EDGE 58
7.2 DCHT 62
7.2.1 Performance 62
7.2.2 Discussion 66
7.3 Epidemic flooding 67

8 CONCLUSIONS AND FUTURE WORK 91

BIBLIOGRAPHY 94

List of Figures

2.1 Scenario for MD source coding with two channels and three receivers [1] . . 12

4.1 Transmission range and interference range for a chain of nodes. The solid line circle is a nodes transmission range while the dotted line circle shows the interference range. Nodes within 3 hops interfere with each other. 20

4.2 Pipelining mechanism of data transmission in sensor networks with intra-flow interference taken into account. 21

4.3 Transmission and interference ranges of Node 3. The solid circle is the transmission range and the dashed circle is the interference range. $R = 2r$. We assume homogeneous nodes. 27

4.4 A simple topology of triangular tessellation with 19 nodes. Each node has the same distance to its nearest neighbors. 28

5.1 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. 31

5.2 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 $O(2^{\frac{n}{2}})$. If we put B nodes in each vertical line, the number becomes $O(B^{\frac{n}{B}})$. 32

5.3 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 $\beta = 1$. 34

5.4 How to avoid the node already reinforced in another path. 38

5.5 How to avoid loops. 39

5.6 Unavoidable bottleneck nodes. Nodes connected by a line are able to communicate with each other. Or else, they are out of range. 42

5.7 Overlapping of A's and B's transmission ranges. 43

5.8 The relation of $P(x > 1)$ and N. 44

5.9 Probabilistic interest dropping reduces interference level during exploratory data phase. The dotted circle is the interference range of Node 1. 46

5.10 Comparison of two gossip-based algorithms. The left is GOSSIP and the right is Fireworks. 48

6.1 A 10-by-10 Grid Topology 52

6.2 Topology with equilateral triangular tessellations. There are 77 nodes in this graph. 55

6.3 Topologies of triangular, grid and hexagon tessellation, from left to right. Every node has 6, 4, and 3 neighbors, respectively. 57

7.1 Throughput and delay comparison between EDGE and directed diffusion in grid topologies. The error rate of outgoing channels follows uniform distribution with average error rate of 0.7. $\alpha = 1$, $\beta = 1$ in $Cost_p$. Packet size is 500 bytes. 68

7.2 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. . . . 69

7.3 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 average error rate of 0.7. $\alpha = 1$, $\beta = 1$ in $Cost_p$. Packet size is 500 bytes. 70

7.4 Network delivery ratio comparison between EDGE and directed diffusion in 10-by-10. The error rate of outgoing channels follows uniform distribution with average error rate of 0.7. $\alpha = 1$, $\beta = 1$ in $Cost_p$. Packet size is 500 bytes. 71

7.5 Jitter comparison between EDGE and directed diffusion in 10-by-10 grid with different delivery ratios. $\alpha = 1$, $\beta = 1$ in $Cost_p$. Traffic rate is 40 packets per second. Packet size is 500 bytes. 72

7.6 Throughput and delay comparison between 3-hop EDGE and 5-hop EDGE in 10-by-10 grid with different traffic rates. The error rate of outgoing channels follows uniform distribution with 0.7. $\alpha = 1$, $beta = 1$ in $Cost_p$. Packet size is 500 bytes. 73

7.7 Throughput and delay comparison with different α and β values in $Cost_p$ of EDGE in 10-by-10. The error rate of outgoing channels follows uniform distribution with average error rate of 0.7. $\alpha = 1$, $\beta = 1$ in $Cost_p$. Packet size is 500 bytes. Traffic rate is 40 packets per second. 74

7.8 Throughput of DCHT, EDGE and basic diffusion with different network sizes. 75

7.9 Goodput of DCHT, EDGE and basic diffusion with different network sizes. 75

7.10 End-to-end delay of DCHT, EDGE and basic diffusion with different network sizes. 76

7.11 Number of loss packets per frame in DCHT. There are 29 packets in a large video frame. 76

7.12 Number of loss packets per frame in EDGE. There are 29 packets in a large video frame. 77

7.13 Delivery ratio and goodput ratio of DCHT with different numbers of descriptions/paths. The network size is 77 nodes. 77

7.14 Throughput and goodput of DCHT with different numbers of descriptions/paths. The network size is 77 nodes. 78

7.15 End-to-end delay of DCHT with different numbers of descriptions/paths. The network size is 77 nodes. 78

7.16 Average energy consumption of a single node in the network of DCHT with 3 paths (descriptions), EDGE and basic diffusion with different network sizes. 79

7.17 Average energy consumption of a single node in the network of DCHT with different number of paths (descriptions). The network size is 77 nodes. . . . 79

7.18 Extra routing overhead (we only consider $NEG\_RESPONSE$) of the whole network of DCHT with different number of paths (descriptions) in different network sizes. 80

7.19 Throughput of triangular tessellation topology with different sizes. 81

7.20 End-to-end delay of triangular tessellation topology with different sizes. . . 81

7.21 Percentage of zero throughput, no path built and no interest reaching source of triangular tessellation topology with 77 nodes. 82

7.22 Throughput of grid topology with different sizes. 83

7.23 End-to-end delay of grid topology with different sizes. 84

7.24 Throughput of hexagon tessellation with 32, 50, 72 nodes. 85

7.25 End-to-end delay of hexagon tessellation with 32, 50, 72 nodes. 86

7.26 Throughput of random topology with 60, 80, 90, 100 nodes. 87

7.27 End-to-end delay of random topology with 60, 80, 90, 100 nodes. 88

7.28 Throughput comparison of 4 topologies. 88

7.29 End-to-end delay comparison of 4 topologies. 89

7.30 Throughput comparison of 3 topologies with full flooding. 89

7.31 End-to-end delay comparison of 3 topologies with full flooding. 90

Chapter 1

Introduction

In September 1999, networked microsensors technology was heralded as one of the 21 most important technologies for the 21st century by Business Week [2]. Cheap, smart devices with sensors, connected by wireless links and deployed in large numbers, make instrumenting and controlling homes, cities, and the environment feasible [3]. Networked microsensors is a subcategory of sensor networks that use multiple distributed sensors to collect information on entities of interest. Applications of sensor networks include: military sensing, physical security, air traffic control, traffic surveillance, video surveillance, industrial and manufacturing automation, distributed robotics, environment monitoring, and building and structures monitoring [3].

Early research focused on military sensor networks. During the Cold War, the Sound Surveillance System (SOSUS), a system of acoustic sensors on the ocean bottom, was deployed at strategic locations to detect and track quiet Soviet submarines. Modern research on sensor networks started around 1980 with the Distributed Sensor Networks (DSN) program at the Defense Advanced Research Projects Agency (DARPA). Researchers at Carnegie Mellon University (CMU), Pittsburgh, PA, Massachusetts Institute of Technology (MIT), Cambridge, University of Massachusetts, Amherst were working on different sensor network testbeds. Talking about sensor network research in the 21st century, we cannot ignore the microelectromechanical system (MEMS) [4] technology, wireless networking, and inexpensive low-power processors. The recently concluded DARPA Sensor Information Technology (SensIT) program [5] pursued two key research and development thrusts. First, it developed new networking techniques. The second thrust was networked information processing, i.e., how to extract useful, reliable, and timely information from the deployed sensor network.

We are focusing on one of the applications, video surveillance. Recent advances of wireless technologies, embedded systems, multimedia source coding techniques and inexpensive hardware such as CMOS cameras (e.g. Stargate board interfaced with a medium resolution camera and Acroname GARCIA [6]), 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, and automated assistance for the elderly and family monitors [6]. They usually have a set of stringent QoS requirements, such as, end-to-end delay, bandwidth, and jitter guarantees. For example, the data rate of H.264 varies between 64 kbps and 240 Mbps depending on the level [7].

To meet them, a more efficient routing protocol needs to be designed for WSNs. Data-centric networking, such as directed diffusion [8], 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.

Multipath transport provides higher available bandwidth for a session by splitting traffic and achieving better load balancing. This technique has long been used in wired networks. Heuristics-based solutions to find the set of paths that minimizes the cost or maximizes throughput are proposed in [9], [10]. For ad hoc networks, DSR and AODV are modified to support multiple paths [11][12][13] by sending back multiple REPLYs from the destination. Disjoint paths are preferred [11][12][13] since the node shared by more than one path is not able to send packets from different paths simultaneously. Shared nodes increase queuing delay and end-to-end delay. On the other hand, contention and interference cause packets to be dropped which leads to lower throughput.

Data-centric networking, such as directed diffusion [8], enables sensor data to be disseminated from data sources to sinks with low delay. In addition to low delay, multimedia dissemination also requires high bandwidth and delivery ratio. This throughput requirement cannot be guaranteed by a single path reinforced in basic directed diffusion. Thus enhancement to directed diffusion is needed.

Multiple Description Coding (MDC) is a coding technique which generates multiple equally important descriptions [14]. The descriptions refer to n independent sub-streams ($n >= 2$). The packets of each description are transmitted over multiple paths. The decoder reconstructs the video clip from any combination of descriptions received, including a single description. MDC is error resilient to media streams in that packet loss or network congestion will not interrupt the stream but only cause a temporary loss of quality. MDC is better for wireless links than the layered coding as used in MPEG-2 and MPEG-4. Layered coding mechanisms generate a base layer and n enhancement layers. If the base layer is missing, which is quite possible for the time-varying wireless links, media streams are interrupted. MDC matches multipath routing very well in that different descriptions generated by MDC are transmitted over different paths.

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 [15], 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. Standard NS-2 uses primitive propagation models, including Free Space, Two Ray Ground and Shadowing which set a signal strength threshold to determine whether one frame is received correctly by the receiver. To provide a more accurate error model that reflects real BER (bit error rate), Wu [16] added SNR and BER models into NS-2 and model interference accurately. Thus other frames received by a receiver simultaneously are also modeled. We use their model in our simulation.

Routing protocols that require location information, such as LAR [17], GPSR [18], and DREAM [19], do not need to flood routing requests. Others, such as DSR [20], AODV [21], ZRP [22], and TORA [23], suffer from the effects of flooding, even with some optimizations, since nodes do not know their locations. Flooding causes many routing messages to be propagated unnecessarily. To reduce the number of routing messages sent and to guarantee reliable data dissemination, epidemic algorithms, which was first used in replicated databases [24], has begun to be used in the context of wireless sensor networks, such as GOSSIP [25] and Fireworks [26]. These protocols require the sender to have knowledge of the potential receivers, which is achieved by exchanging beacons. Anonymous Gossip (AG) [27] overcomes this problem by attempting to send a gossip message and waiting until the other node sends back a gossip reply, incurring higher overhead.

Directed diffusion, though regarded as an epidemic algorithm in [28] since it avoids broadcast storm, does not perform well with interest flooding. No matter what metrics are used in selecting a route (basic directed diffusion uses delay), the route which performs best during route discovery phase may not perform well during the actual data transmission phase due to differences in interference levels caused by the different traffic patterns. Interest flooding increases traffic in the network and causes maximum level of interference. Exploratory data are flooded to determine the best path, which follows gradients established in the interest propagation phase. Actual application traffic only flows through the reinforced path, which is not affected by inter-path traffic at all, assuming there is no other data transmission at that time. Every node has an interference range. Interference set [29] and conflict graph [30] are used to schedule network traffic or theoretically analyze the impact of interference on wireless networks. However, no routing protocol could have the prior knowledge about which path the actual data traffic will go through and what the traffic pattern will be like before the route is determined. Our goal is to design solutions which make more accurate routing decisions by reducing the interference level during the route discovery phase and making it more similar to that during the actual data transmission phase.

Chapter 2 gives background information about research related to our protocols for video streaming over wireless sensor networks. In Chapter 3, we discuss the motivations for our work. We describe the problem statement in Chapter 4. Protocol design is presented in Chapter 5. Chapters 6 and 7 explain the implementation and performance of the protocols. We conclude our research and discuss future work in Chapter 8.

CHAPTER 2

RELATED WORK

2.1 Hardware Support

The latest series of TelosB motes [31], the ZigBee motes [32] with improved abilities, or PC104 [33] may be used for applications in WSNs which require intensive memory and bandwidth.

Telos is an ultra low power wireless sensor module (”mote”) for research and experimentation [30]. The latest in a line of motes developed by UC Berkeley to enable wireless sensor network (WSN) research is called Telos. It is a new mote design built from scratch based on experiences with previous mote generations. Its new design consists of three major goals to enable experimentation: minimal power consumption, easy to use, and increased software and hardware robustness.

ZigBee [32] is an important standardized approach to the control of sensor and actuator networks in home/building/industrial automation applications that use low rate wireless PAN network technology. ZigBee stack includes Network and Application Layer proposed by ZigBee Alliance and IEEE 802.15.4 MAC and Physical Layer for both mesh and tree-based communication.

PC104 [33] is a popular standardized form-factor for small computing modules typically used in industrial control systems or vehicles. It gets the name from the popular desktop personal computers initially designed by IBM called the PC, and from the number of pins used to connect the cards together (104). PC104 cards are much smaller than ISA-bus cards found in PC’s and stack together which eliminates the need for a motherboard, backplane, and/or card cage. Power requirements and signal drive are reduced to meet the needs of an embedded system. Because PC104 is essentially a PC with a different form factor, most of the program development tools used for PC's can be used for a PC104 system. This reduces the cost of purchasing new tools and also greatly reduces the learning curve for programmers and hardware designers.

Most of the sensors used in research for audio/video streaming are found to use embedded microprocessors which have higher computing abilities [34].

2.2 On-going Video Sensor Network Research

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 [35], large scale video sensor networks for distributed surveillance at Palo Alto Research Center (PARC) [36], Video Web at University of California at Riverside [37], a video-based sensor network architecture for video surveillance and environment monitoring proposed by Feng et. al. [38], maximizing the life of wireless video sensor networks at Virginia Tech [39], the Distributed Interactive Video Array (DIVA) system at Spawar Systems Center (SSC) San Diego [40], WMSNs research at Ohio State University [41], video sensor network for autonomous coastal sensing at Boston University [42], Quality of Service (QoS) research for vision-based WSNs at Purdue [43], etc.

2.3 QoS issues in WMSN

The network layer of WMSN needs to address QoS issues of multimedia streams. [44] considers the bandwidth constraints for multimedia mobile medical calls. Distributed image sensing with QoS-based geographic routing is used in [45] for network localization, dynamic routing and load balancing. Other papers are more concerned with real time streaming issues, e.g. RAP [46], SPEED [47] and its extension MMSPEED [48]. They prioritize packets based on their delivery speed, computed from geographic information and time elapsed, at the source, hop-by-hop or every a 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.

2.4 Routing Metrics in Wireless Sensor Networks

Routing metrics in wireless ad hoc networks are important considerations due to the unpredictability and heterogeneity of link qualities [49]. Existing wireless ad hoc routing protocols typically select routes using minimum hop count, e.g. DSR [20] and DSDV [50]. Directed diffusion [8] selects routes in sensor networks with the least delay. Recently, many new link quality metrics have been proposed. [51] compares the performance of the following three metrics. Adya et al. [51] 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 [51]. Expected Transmission Count (ETX) [52] 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) [53] is used for selecting channel-diverse paths and accounts for the loss rate and bandwidth of individual links. Park et al. [54] 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 [52], ETX was incorporated into DSR and DSDV to improve throughput with little consideration of delay or interference. WCETT [53] is more suitable in multi-radio wireless mesh networks. EDR [54], unlike ETX, cannot be computed dynamically. More space and computation are required by EDR when it is incorporated into DSR and AODV.

2.5 Interference-aware Protocols

Interference-aware protocols have recently been explored in multi-hop wireless networks. [55] 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 [56], they present routing algorithms to compute interference-optimal cost-bounded paths and an optimal bandwidth allocation algorithm to allocate timeslots. We do not give detailed analysis, computation and implementation for interference because we try to make full use of the ETX information. [57] and [58] give the throughput bounds and capacity for interference-aware routing in wireless networks respectively. We may use them to test our protocol by observing the throughput performance. [59] derives an interference aware metric NAVC based on the information collected from 802.11 MAC. In [60], an interference aware routing scheme is designed to alleviate the near-far problem at the network level for cellular systems. EIBatt et. al. [61] 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. [62] 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.

2.6 Short-comings in ETX

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 [52]. It is not true that all the hops in a path will interfere with each other. Bader et al. [63] 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, Dijkstras algorithm can no longer be utilized and greedy algorithm is used instead. Inter-flow interference is also considered in Dynamic Codeword Routing (DCR) [64].

2.7 Theoretical Analysis of Throughput-Delay Tradeoff

Wireless nodes are composed of nodes which communicate with each other over a wireless channel. Its capacity is different from wireline networks. Gupta-Kumar [65] presents a network model and analyzes the capacity of wireless networks. The conclusion is that when n identical randomly located nodes, each capable of transmitting at W bits per second and using a fixed range, form a wireless network, the throughput obtainable by each node for a randomly chosen destination is $\Theta(\frac{W}{nlogn})$ bits per second under a noninterference protocol.

Throughput-delay trade-off in the Gupta-Kumar fixed network model [65] is theoretically analyzed in [66]. Gamal et. al. [66] focuses on characterizing the delay and determining the throughput-delay trade-off. They show that the optimal throughput-delay trade-off is given by $D(n) = \Theta(nT(n))$, where $T(n)$ and $D(n)$ are the throughput and delay respectively.

Our results also show similar trade-off between throughput and delay in practical sensor network algorithms.

2.8 Multi-path Routing

Multipath routing has long been researched in wired networks. One of the earliest papers [13] proposed an extension of the distance-vector algorithm to find multiple disjoint paths, one of which is the shortest path to a destination. Alternate path routing (APR) [12] provides load balancing and failure prevention by distributing traffic among a set of diverse paths in mobile ad hoc networks. Geographic proximity of candidate paths is avoided in the algorithm. Split Multipath Routing (SMR) [11] is another multipath protocol for ad hoc networks that allows paths to share nodes when no disjoint paths can be found. They also use a per-packet allocation scheme to distribute packets over multiple paths instead of splitting traffic only at the source.

2.9 Multistream Video Coding

In a framework for supporting multipath video transport over ad hoc networks, multistream video coding is one of the essential components. Mao et. al. [10] combines multipath transport with multiple description coding (MDC) in ad hoc networks. Other efficient coding schemes that were proposed are feedback-based reference picture selection (RPS) [67], layered coding (LC) with selective ARQ [68] and multiple description motion compensation (MDMC) [15].

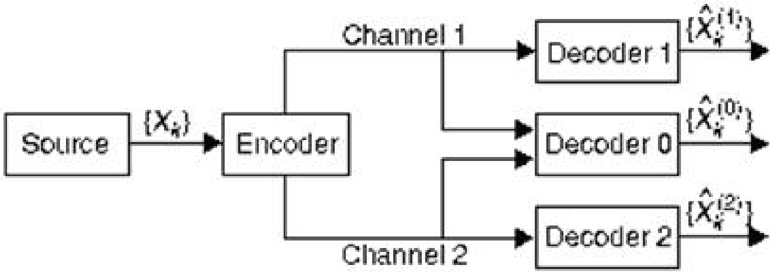


Figure 2.1: Scenario for MD source coding with two channels and three receivers [1]

2.10 Message Redundancy in Routing Protocols

Message redundancy caused by excessive flooding has been realized in recent years. The taxonomy of the major proposed solutions is described in [26]: Probabilistic-based schemes, area-based methods and neighbor knowledge methods. GOSSIP [25], Anonymous gossip [27] and Fireworks [26] fall under the first category. In GOSSIP, when a node first receives a route request, it broadcasts the request to its neighbors with probability p and it discards the request with probability $1-p$. In Fireworks, a node re-broadcasts the message to all its neighbors with probability p and it sends it to only c randomly selected neighbors with probability $(1-p)$. The Fireworks protocols results in higher reliability given the same number of links over which the broadcast packet is transmitted. Anonymous gossip (AG) does not require any member to know the other members of the multicast group because the sender sends a gossip message and it will gossip until the other node sends back a gossip reply. The other two categories require location awareness or two hop neighborhood knowledge, both of which incur more overhead than the first category.

Chapter 3

Motivations and Applications

In this chapter we discuss the motivations for our protocols in video sensor networks as well as its potential applications.

3.1 Motivation

Although directed diffusion is a widely used in sensor networks, it has several weaknesses when applied to Wireless Multimedia Sensor Networks (WMSN). It only considers delay as the routing metric to select best path. As a result, high throughput may not be achieved. Furthermore, directed diffusion is a single-path protocol, which causes throughput degradation for multimedia applications, considering the limited bandwidth and energy constraint of sensor nodes. Besides, its failure to consider deadline during the route selection lowers the percentage of packets which can meet the deadline of real-time traffic. Finally, its reliance on flooding gives rise to differences in interference levels during the exploratory data phase and the actually data transmission phase, which makes the route decision even more inaccurate. We propose three routing protocols to augment and extend directed diffusion to handle these issues. First, we propose EDGE (**E**TX-**D**elay **G**r**E**edy algorithm) where a single-path metric which considers throughput, delay and intra-flow interference. Secondly, we further address the throughput and deadline issues in multimedia streaming over sensor networks by proposing multi-path protocol DCHT (Delay-Constrained High Throughput). Lastly, we present the improvement on QoS-based routing by limiting interference in lossy wireless sensor networks.

3.2 Applications

Wireless Sensor Networks (WSNs) can support a wide range of applications such as target tracking, home automation and environmental monitoring. They serve to explore the requirements, constraints and guidelines for general sensor network architecture design. Some of the applications may be reinforced or augmented with transmission of multimedia data over WSNs so that we may have multimedia surveillance, storage of potentially relevant activities from networked cameras, traffic conditions and collision avoidance.

3.2.1 Multimedia Surveillance

With the improvement of video technology and sensor networks, the infrastructure for new generations of multimedia surveillance systems was proposed. Many different media streams, such as audio, video, images, textual data and sensor signals, will be collected and analyzed automatically to interpret the controlled environment in real-time [69]. Distributed architectures with fixed and active cameras are devised as new solutions in order to enlarge the view of traditional surveillance systems. Their view is enhanced with other sensed data and multi-resolution views are explored with zooming and omnidirectional cameras. Multimedia surveillance applications involve both indoor and outdoor areas and people surveillance gains its most popularity. Biometric technology is also used to enrich multimedia surveillance systems.

3.2.2 Data-centric Storage

Habitat and environmental monitoring is a new class of sensor network applications which has enormous potential benefits for scientific communities and the whole society [70].

Long-term data collection in different scales and resolutions can be enabled by equipping natural spaces with networked microsensors. It is hard to obtain the localized measurements and detailed information through traditional instrumentation; while sensor has intimate connection with its immediate physical environment, which allows each sensor to win over traditional methods. Sensor nodes communicate with each other and cooperate in performing more complex tasks.

3.2.3 Traffic Conditions and Collision Avoidance

Vehicular network is composed of cars with GPS devices and, in a more advanced systems, with video cameras. Cameras may acquire information about the environment, such as traffic signs, congestions, traffic accidents, and road merging information, and either report them to the base station or broadcast them to their neighbors. Although cars have more power than sensor nodes, the bandwidth limit due to mobility is similar to that of sensor networks which is caused by constrained power of sensors.

CHAPTER 4

PROBLEM STATEMENT

4.1 Single-path metric $Cost_p$

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 [7]. 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 critical requirement need for multimedia data. 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.

4.1.1 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 802.11b 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 802.11b MAC implements an ARQ (retransmission) mechanism, a links 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.

4.1.2 Definitions, Notations and Formulae

The ETX of a link is the predicted number of data transmissions required to send a packet over that link [52].

$$ETX = \frac{1}{d_f * d_r} \tag{4.1}$$

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 ETXs 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.

$$ETX_p = \max_{i=0}^{N-3} (\sum_{j=i}^{i+2} ETX_j) \tag{4.2}$$

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.

$$Cost_p = {ETX_p}^{\alpha} \times {delay_p}^{\beta} \tag{4.3}$$

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:

$$INTERVAL = ETX_i \times EXPLORE\_DELAY \tag{4.4}$$

4.1.3 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 [63]. 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. [71] 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 4.1.

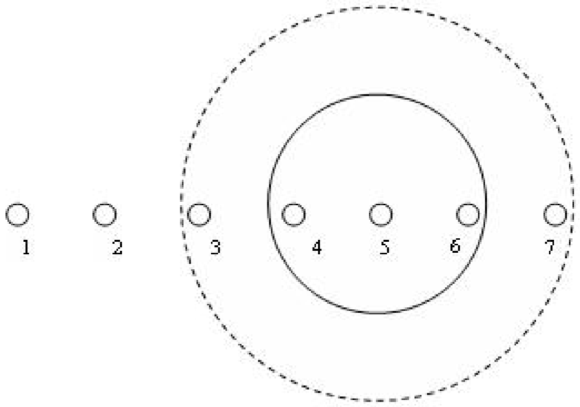


Figure 4.1: Transmission range and interference range for a chain of nodes. The solid line circle is a nodes transmission range while the dotted line circle shows the interference range. Nodes within 3 hops interfere with each other.

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. Assuming that 2, 2, 2, 2, 2, 3 are the ETX values for the six links in Figure 4.1. Then ETX_p is 7. If we change the ETX values in Figure 4.1 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 leads to a wrong route selection.

Another reason for using our path ETX is the impact of intra-flow interference in the pipeline of packet transmission (Figure 4.2). 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.

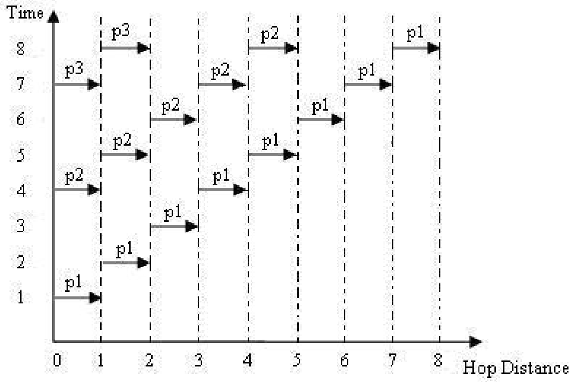


Figure 4.2: Pipelining mechanism of data transmission in sensor networks with intra-flow interference taken into account.

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 [52] for more information.

4.1.4 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 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 , $P = (src, i)$, src is the source node; , $dest(p) = i$, 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:

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

The above objective function is subject to:

$$\forall p \in P, ETX_p = \max_{i=0}^{dest(p)-3} \sum_{j=i}^{i+2} ETX_j$$

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

$$\forall j \in E, 0 <= ETX_j <= 1$$

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

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

The first constraint defines ETX_p based on the ETX value for the sub-path. The second constraint is the definition of delayp 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.

4.2 Multi-path Protocol DCHT

Single path protocols with a single metric are severely constrained by bandwidth; while multipath protocols are affected by interference that degrades throughput which must be high enough for multimedia data transmission. In designing a multipath protocol to support multimedia streaming over wireless sensor networks, the key challenge is to find an effective way to establish multiple paths which maximize throughput and minimize deadline miss ratio and interference.

We assume nodes are stationary or have little mobility, which means mobility does not contribute to the wireless loss. Each node is equipped with one 802.11 radio and they use the same channel to communicate. We assume all links are symmetric. To make it simple, we only consider one source and one sink. The goal is to find multiple disjoint paths with high throughput and low end-to-end delay. Disjoint paths with low inter-path interference are selected. The following are definitions of some terms.

Deadline (DL): The time period in which data from the source must reach the sink. Technically, it corresponds to the playout deadline in a certain video streaming application. Disjoint paths: Paths that do not share any link or node.

Bottleneck nodes: Nodes that have to be shared by multiple paths because of low density at a locality.

Cumulative SNR: Weighted SNR (Signal-to-Noise Ratio) over time used to estimate ETX. SNR_i is the cumulative SNR at time interval i and $SNR_i + 1$ at next time interval $i + 1$, i.e.

$$SNR_{i+1} = \gamma \times SNR_i + (1 - \gamma) \times SNR \tag{4.5}$$

where SNR is determined from the packet just received. γ is a positive fraction. If the recent SNR has more weight, γ should be greater than 0.5.

SNR is closely related with BER (Bit error rate) [15]. Lee et. al. [72] derived the mathematical formula to calculate BER:

$$BER = 0.5 \times erfc(\sqrt{\frac{P_r \times W}{N \times f}}) \tag{4.6}$$

P_r is the received power, W the channel bandwidth, N the noise power, f the transmission bit rate, and erfc the complementary error function. Most wireless card typically measure:

$$SNR = 10log\frac{P_r}{N} \tag{4.7}$$

In order to consider interference, we change Equation 4.7 to

$$SNR = 10log\frac{P_r}{N + I} \tag{4.8}$$

I is interference component. A method for calculating I is given in [16]. Given the packet size, packet loss rate could be calculated from BER. According to [52], we need to know the forward delivery ratio d_f and the reverse delivery ratio d_r. Since all links are symmetric in our assumption, d_f equals d_r, which is the packet delivery ratio.

Definition of path metric $Cost_p$: In order to maximize throughput and minimize delay with interference consideration, we use the path metric Costp proposed in [73]. α and β are non-negative integers.

$$Cost_p = {ETX_p}^{\alpha} \times {delay_p}^{\beta} \tag{4.9}$$

$$ETX_p = \max_{i=0}^{N-3} \sum_{j=i}^{i+2} ETX_j \tag{4.10}$$

where N is the number of hops in the path and ETX_j is the ETX value of the jth hop. The goal of our routing algorithm is to find multiple disjoint paths with minimum $Cost_p$. Paths with close proximity interfere with each other. Since links suffering severely from interference have poor SNRs (which are indirectly used to estimate ETX), they have less probability to be selected in the paths.

4.3 Interference Limiting

In this section, we first introduce the error and interference model we use to capture the lossy and interference natures of WSNs. We then discuss how to design an efficient routing algorithm to address the problem.

4.3.1 Overview of Directed Diffusion and Its Variants

Directed diffusion uses a publish/subscribe communication model in which a sink node floods interests as requests for a named data. As the interest is propagated through the network, each intermediate node sets up a gradient with its neighbors and enables data that match the interest to be pulled towards the sink. Sensor nodes with data that match the interest will forward exploratory data propagated by intermediate nodes through established gradients to the sink. The sink initiates a reinforcement message to the node that first forwarded the new data to it. Other nodes use the same rule to reinforce the upstream neighbor. The source node continues to send data through the reinforced path after it received the reinforcement.

Based on the above rule, basic diffusion generally selects route with the lowest delay. In the past few years, researchers have proposed a variety of single or hybrid metrics with the purpose of improving the performance, including throughput, delay, jitter, and deadline-hit ratio, of wireless networks. Single metrics include RTT [51], PktPair [51], ETX [52], WCETT [53], and EDR [54]. Most of them can be implemented in directed diffusion. In order to consider both throughput and delay and take into account intra-path interference, a hybrid metric was proposed in [73].

4.3.2 Route Selection Problems

Directed diffusion and its variants select paths which perform best during the exploratory data phase. However, during this phase, the network has very high traffic since the exploratory data follow all the gradients set up during the interest propagation phase, which effectively results in flooding. Different topologies and traffic types lead to different interference levels for each node in the network. As a result, the performance of a candidate path is determined by how each link performs under high interference level due to exploratory data flooding. As shown in Figure 4.3, the interference range usually differs from the transmission range, typically by a factor of 2. Node 3 can send to Nodes 2, 4, 6, and 7 but not Nodes 1, 5, 8, and 9. However, Node 3s transmission will interfere with that of Nodes 1, 5, 8, and 9.

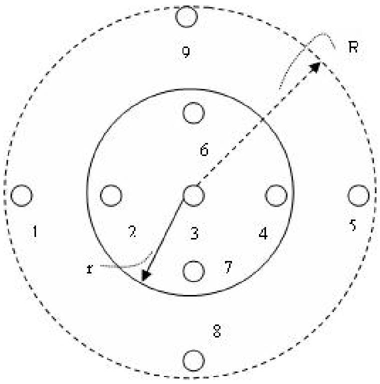


Figure 4.3: Transmission and interference ranges of Node 3. The solid circle is the transmission range and the dashed circle is the interference range. $R = 2r$. We assume homogeneous nodes.

After reinforcement, one path is selected for application data transmission and it rarely suffers from interference from other paths used in route discovery since no data will be sent

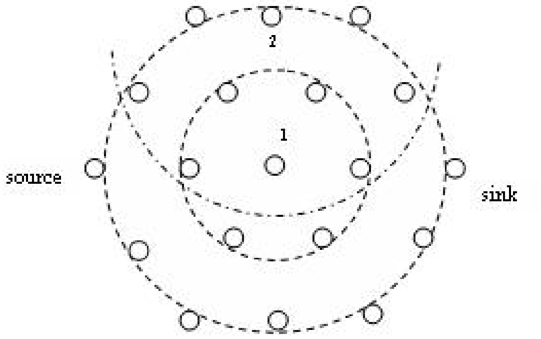


Figure 4.4: A simple topology of triangular tessellation with 19 nodes. Each node has the same distance to its nearest neighbors.

over them once a route is selected and reinforced. Different interference level in the two phases causes incorrect routes to be selected since a path that performs best under high interference environment may perform worst when transmitting data without other flooded paths. Border links are good examples (Figure 4.4). The inner circle is the transmission range and the outer circle is the interference range. Consider Nodes 1 and 2. Almost every node in the topology is in the interference range of Node 1; thus all the links starting from or ending at Node 1 should have very bad performance when exploratory data are propagated. In contrast, Node 2 is on the border and only the upper half of the nodes is in its interference range. This is also true of other border nodes. However, when there is only one data path in the network, a central path (through Node 1) will perform better than a detour path composed of border links.

Interference causes route selection problems. Unpredictable traffic pattern and unknown geographic information aggravate this problem.

Chapter 5

Protocol Design

5.1 EDGE Design

In this section, we present centralized and distributed algorithms to compute routes that more accurately estimates interference, maximizes the throughput and minimizes the delay over lossy links in multi-hop WMSNs. [62] 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, scheduling tasks may drain their energy.

Transport protocols like MRTP [74] 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) [7], then we get 533 bytes (64k/1485*99/8) as the frame size.

Table 5.1: Pseudo code of Centralized Algorithm.

```
$T2 \leftarrow Systemtime$
$MinCost \leftarrow MAXIMUM$
$Flowlabel \leftarrow 0$
for each flow do
    $ETX_p = \max_{i=0}^{N-3} \sum_{j=i}^{i+2} ETX_j$
    $Cost_p = (ETX_p)^{\alpha} \times (T2 - T1)^{\beta}$
    if $Cost_p < MinCost$
        $MinCost \leftarrow Cost_p$
        $Minlabel \leftarrow Flowlabel$
```

Suppose we do not decompose each 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 [52] 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.

5.1.1 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.

We illustrate the centralized algorithm with the example below in Figure 5.1. In this example, there are four routes with Costp metric of 42, 81, 120 and 80 ($\alpha = 1$ and $\beta = 1$).

This algorithm will choose the best route $src \rightarrow x \rightarrow y \rightarrow sink$ with the $Cost_p$ metric of 42.

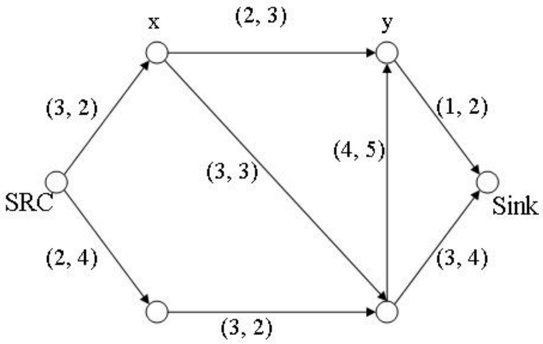


Figure 5.1: Illustration of the centralized algorithm. In each tuple, the first element is the ETX value of the link and the second element is the delay of the link which includes the packet queuing delay at the upstream node. Gradients are indicated by arrows.

5.1.2 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 [75] problem. The definition of PSPACE is x: x requires exponential memory. Figure 5.2 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.

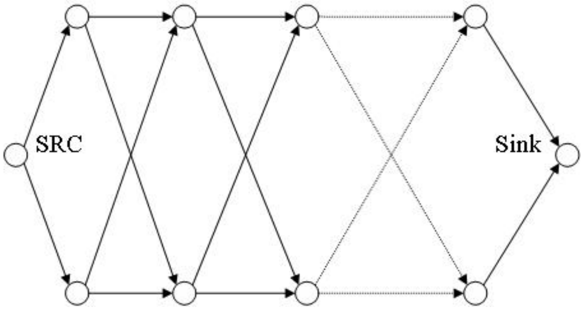


Figure 5.2: The worst case routes. Every 2 nodes except the source and sink are put in a vertical line. Every node in a line has two gradients pointing to both nodes in the next line. The number of routes is $O(2^{\frac{n}{2}})$. If we put B nodes in each vertical line, the number becomes $O(B^{\frac{n}{B}})$.

Table 5.2: Packet format. n is the hop number. $T1$ is the timestamp at which the source sent the packet.

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

Since this is a PSPACE problem, 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. Each node except the source runs the pseudo code in Table 5.4. The packet format and local table format are shown in Table 5.6 and Table 5.3 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.

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

Table 5.3: 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)

Table 5.4: 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 $FALSE$
```

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 5.3.

In Figure 5.3, 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 one 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 one 56, which means we made the wrong decision at Node A.

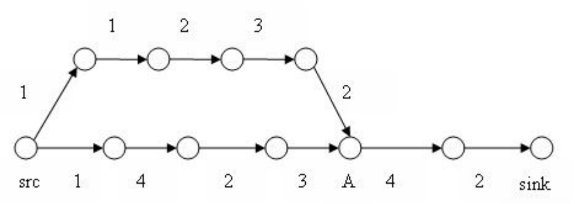


Figure 5.3: An example showing that $Cost_p$ does not have sub-solution optimality property. The numbers denote the ETX values of those links. delay for each link is 1. Suppose $\alpha = 1$ and $\beta = 1$.

5.1.3 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 5.5. ETX_i is the ETX value of the current link.

$delay_p$ 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.

Table 5.5: 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 $FALSE$
```

5.2 DCHT Design

In this section, we modify directed diffusion by 1) using $Cost_p$ as the metric instead of pure delay; 2) reinforcing multiple links at the sink to obtain disjoint paths from the source. These modifications will maximize the throughput and minimize the delay over lossy links in multi-hop WMSNs. We are not using specific multimedia QoS requirements such as bandwidth to guide the routing decision process or prioritized packet scheduling to avoid fast depletion of energy in sensor nodes. However we do consider the playout deadline since end-to-end delay constraint is one of the most important QoS requirement because data arriving later than a deadline are simply useless, making them equivalent to data dropped. We discard paths which fail to meet the playout deadline in video streaming. Throughput is

Table 5.6: Packet format. n is the hop number. $T1$ is the timestamp at which the source sent the packet.

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

influenced by ETX, which is estimated by the cumulative SNR. The use of historical SNRs offers more stable and accurate estimation of ETX.

5.2.1 Route discovery

Routing metric collection

When interests are first flooded, a timestamp t_0 is inserted in the interest packets. Exploratory data packets are flooded after the source receives interests from the sink. At each intermediate node which received an exploratory data packet, SNR is read from the packet. Cross-layer design is probably needed here since SNR is a MAC layer metric. ETX information of the previous three upstream links, calculated from SNR, are inserted in the packet header in the ETX(n%3) fields [73]. ETX_p is computed at the same time according to the previous chapter. $Cost_p$ of each subpath is kept at the intermediate nodes local table in ascending order. The format of the local table is shown in Table 5.7. Only the one with the lowest $Cost_p$ is forwarded to the next hop. Another timestamp at the sink t_1 is recorded when the first exploratory packet reaches the sink. t_i is the timestamp for the ith exploratory packet to reach the sink. We only consider packets whose timestamp t_i satisfies the constraint $t_i - t_0 <= DL$, where DL is set to be slightly less than the real playout deadline to take into account the time for finding disjoint paths. If no path can satisfy the delay constraint, the sink adjusts the metric $Cost_p$ by giving more weight to β (for delay) and piggybacks the new value in new INTEREST messages.

Table 5.7: Local table format. Local timestamp is used to get local time synchronization [73].

Last hop	ETX	Local timestamp	$Cost_p$	Cumulative SNR

Multiple path reinforcement

The sink stops putting exploratory data packets in the candidate pool when it received one that cannot meet the deadline or when the multi-path timer expires. It then sorts the candidate paths in ascending order of $Cost_p$ and selects the first ρ paths to reinforce, where $\rho > \lambda$ and λ is the number of paths needed at the source. We need to find more than the required number of paths because some candidate paths may not be reinforced i disjoint nodes cannot be found or the delay exceeds the playout deadline. If two nodes try to reinforce a link that converges to the same node, the first one to reinforce would win. In Figure 5.4, A reinforces C first. Then, B tries to reinforce C and C will drop the packet because C regards it as an old message. C will then send B a NEG RESPONSE so that B could delete the entry of C in its local candidate table and selects the next candidate, e.g. D. In addition, delay constraints must be satisfied by computing the difference between two local timestamps. We can guarantee that if we choose the node within the delay constraint in each step, the final route can also meet the delay constraint, no matter how many times we have to choose the next-best node. This technique not only guarantees disjoint nodes (paths with disjoint nodes are definitely disjoint paths), but also ensures loop-free path since loops are broken when selecting the next candidate in the local table.

In Figure 5.5, A has already reinforced B and C, D, E are reinforced sequentially. When E tries reinforcing B, B drops the packet and sends NEG RESPONSE to E. So E is able to select the next candidate F to reinforce. Standard diffusion only reinforces one

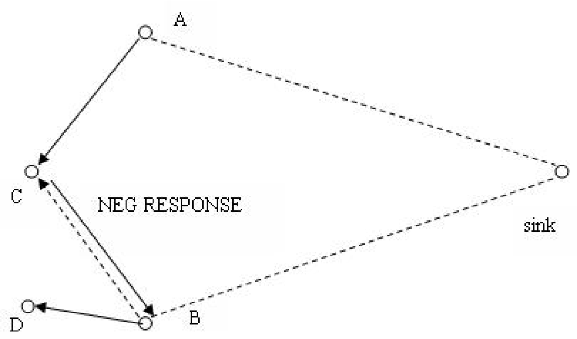


Figure 5.4: How to avoid the node already reinforced in another path.

route and when there is a loop, the reinforcement packet is simply dropped and no packet is received at the source. Our algorithm guarantees the success of reinforcement packets reaching the source provided that the delay constraint could be satisfied.

When a node discards the already-reinforced node and selects the next candidate, it must estimate the new end-to-end delay so that the deadline set earlier can still be met. In Figure 5.4, B reads the local timestamps of C and D (t_C and t_D respectively) from its local table and if $t_C - t_D <= \frac{\xi}{n}$ (ξ is the slack between the real playout deadline and the deadline used by the sink to get candidate paths; n is the estimated number of hops in the reinforced path), we reinforce D; else, we search for more candidates in the table. If it still fails, the reinforcement packet is dropped. It is possible that the throughput of the newly-selected route with disjoint nodes is degraded. Since delay is more important than throughput in multimedia data transmission, the slight difference in throughput is tolerable.

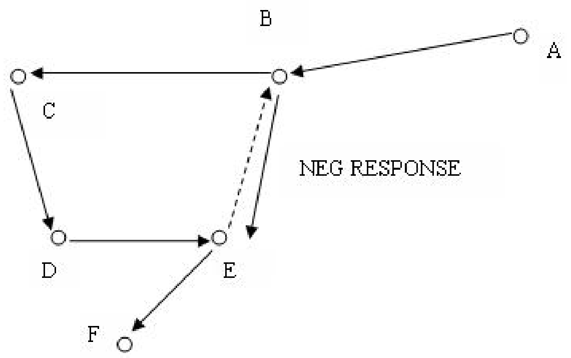


Figure 5.5: How to avoid loops.

Probabilistic traffic splitting

Multiple Description Coding (MDC) is used for multimedia traffic. We let the coder generate the same number of descriptions (streams) as that of paths found. Descriptions are sent through different paths. In order to avoid using the same paths all the time, redundant paths are built to share the traffic load. Assume the source has 2λ routes and λ descriptions to be sent. Their $Cost_p$ values are $C_1 < C_2 < \cdots < C_2$. Then, *path* 1 and *path* 2λ are allocated to transmit the first description, *path* 2 and *path* $2\lambda - 1$ the second, etc. The transmission of each description alternates between the paired paths with certain probabilities. For example, *Description* 1 has the probability of $\frac{1/C_{2\lambda}}{1/C_1+1/C_{2\lambda}}$ to be sent over *Path* 1. In a pair of paths, the one with better quality is being used more often.

5.2.2 Parameter settings

α, β adjustment

The problem of finding the suitable values for α and β can be converted into a Constraint Satisfaction Problem (CSP) [76]. We have two constraints: delay and throughput. The initial goal of our algorithm is to maximize throughput within a certain delay constraint. The playout deadline is a natural upper bound for delay constraint. We use ETX to estimate throughput and ETX also has an upper bound in different hardware or simulators. For example, since the maximum retransmission time in NS2 is 7, ETX has a maximum value of 8. Besides, we can also estimate the upper bound for ETX_p given the required bandwidth requirement.

We use a min-conflicts algorithm [77] to solve the CSP problem in this paper. A similar method is also use in [78] to solve the problem of finding a path with two additive constraints called multi-constrained path (MCP) problem. Since ETX_p is not a completely additive constraint and our formula is a weighted product instead of weighted sum, the algorithm we use is slightly different. The basic idea is to first combine the delay and ETX, i.e. $Cost_p$, and then find the corresponding shortest path. In Table 5.8, D and E are delay and ETX_p upper bounds respectively. d and e are the delay and ETX_p of a certain path. p, q, and r are paths between source and sink. q and r can be obtained using any algorithm or enumeration. p is obtained from the Dijkstras algorithm. The maximum probability that exists in a feasible path is less than 1/2 if algorithm MCP-DE cannot find a solution [78].

Table 5.8: Algorithm MCP-DE (D, E, d, e). $Cost_p$ formula is simplified to $d \times e^{\alpha}$ (α is not necessarily an integer) and it is equivalent to $d^{\beta} \times e^{\alpha}$ as the combined metric for routing.

```
$q \leftarrow$ path with lowest $ETX_p$
if ($e(q) > E$) then
  return $NULL$
else if ($d(q) <= D$) then
  return $q$
$p \leftarrow Dijk(d)$
if ($d(p) > D$) then
  return $NULL$
else if ($e(p) <= E$) then
  return $p$
while $TRUE$ do
  $\alpha \leftarrow [lg(d(q)) - lg(d(p))]/[lg(e(p)) - lg(e(q))]$
  $r \leftarrow$ path with lowest $d \times e^{\alpha}$
  if ($e(r) = e(q)$ or $e(r) = e(p)$) then
    return $NULL$
  else if ($e(r) > E$) then
    $p \leftarrow r$
  else if ($d(r) > D$) then
    $q \leftarrow r$
  else
    return $r$
```

Bottleneck nodes

One of our goals is to find disjoint paths. In real situations, some nodes have to be shared by more than one path. In Figure 5.6, Node A is a bottleneck node since nodes in $C1$ can only communicate with nodes in $C2$ through Node A and vice versa. A is a cut-vertex and the two edges incident to A are cut-edges [79]. There might be more than one cut-vertex in a certain topology. Min-cut algorithm is used to find bottleneck nodes the failure of which will cause the least partition [80]. The time to compute the min-cut value of a given graph G is $O(n^4 logn)$. We can also use this algorithm to find the bottleneck nodes which prevent us from finding disjoint routes. However, it is not scalable. When the network size increases, it consumes too much energy to find bottleneck nodes than to deploy more nodes in their neighbourhood to build disjoint paths.

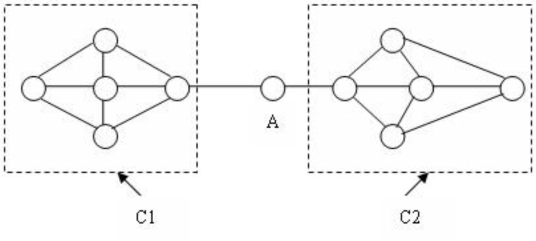


Figure 5.6: Unavoidable bottleneck nodes. Nodes connected by a line are able to communicate with each other. Or else, they are out of range.

In real situations, there is human intervention to the deployment of sensor nodes. Given the lower bound of node density, the percentage of bottleneck nodes is limited. In Figure 5.7, if C is the only node in the overlapping area, it has a high probability to become a bottleneck node (it may not be if A has other nodes in its transmission range that have paths to B without the help of C). If more than one node is put in the overlapping area, no bottleneck nodes will exist. The condition is sufficient but not necessary. Hence, we are giving a looser bound for node density.

Assume node deployment follows Poisson distribution. x is the number of nodes in the overlapping area mentioned above. Given the average density λ and overlapping area A, $x = \lambda A$.

$$P(x = k) = \frac{e^{-\lambda A}(\lambda A)^k}{k!} \tag{5.1}$$

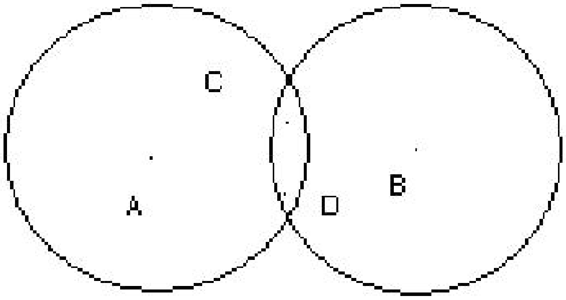


Figure 5.7: Overlapping of A's and B's transmission ranges.

Suppose the entire network is a square with the side of E and the number of nodes is N. R is the radius of the transmission range.

$$\lambda = \frac{N}{E^2} \tag{5.2}$$

If nodes are evenly distributed and we only consider the scenario in Figure 5.7, that is, $\frac{E}{\sqrt{N}} > R > \frac{1}{2} \times \frac{E}{\sqrt{N}}$. So,

$$A = 2(R^2 arccos \frac{E}{2R(\sqrt{N}-1)} - \frac{E\sqrt{4R^2(\sqrt{N}-1)^2 - E^2}}{4(\sqrt{N}-1)^2}) \tag{5.3}$$

Then, the probability that there is more than one node in the overlapping area is

$$P(x > 1) = 1 - P(x = 0) - P(x = 1) = 1 - e^{-\lambda A}(1 + \lambda A) \tag{5.4}$$

Given $E = 1200$, $N = 100$, $R = 100$, $P(x > 1)$ is 0.2492. N-P relations are shown in Figure 5.8. Likewise, given $P(x > 1)$, E and R, we can also find N. As seen from the figure, the network should have very high density to guarantee the probability of more than one node in the overlapping area above a certain threshold.

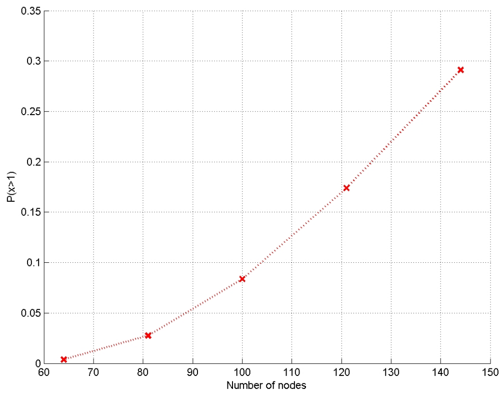


Figure 5.8: The relation of $P(x > 1)$ and N.

5.3 Epidemic Algorithm in Diffusion

In this section, we present a set of localized epidemic algorithms (without using neighbor information) for a modified directed diffusion. The main purpose is to reduce interference during the exploratory data phase, make the interference level more similar to that during the actual data transmission phase and improve route selection. The key challenge for designing such a protocol is that probability-based schemes usually makes use of some basic understanding of the network topology to assign to a node a probability p to broadcast, which means the sender knows the IDs of potential receivers. However, directed diffusion is a localized ID-free routing protocol, where no node knows its neighbors before interest flooding. (We assume this original property of directed diffusion although in practice it may be possible for the sender to determine the MAC addresses of the receivers.) So we

Table 5.9: Pseudocode segment for epidemic Interest flooding (receiver side)

```
If it is a new interest packet
    Set $r = random()$
    If $r > 1 - p$
        Update gradient table
    Else
        Drop interest
Else
    Execute original code
```

have two choices here: receivers probabilistically drop interest or senders probabilistically send exploratory data since neighbor information is known by the sender. In both ways, we are taking the risk of losing candidates. There is a trade-off between the degree of interference level match and the completeness of the candidate pool. (Our results below show the optimal probability that will maximize throughput or minimize delay for each network configuration.)

5.3.1 Epidemic Interest Flooding

Interest flooding is the first phase of directed diffusion. To adapt epidemic flooding to directed diffusion, when a node receives an interest packet, it updates the gradient table with a probability of p, i.e. with a probably of $1 - p$, the node will not update its gradient table (Table 5.9). (We assume the sender has no knowledge about its neighbors.) Strictly speaking, interests are still forwarded through the network with a reduced rate. However, exploratory data are propagated selectively, which satisfies our goal, since only parts of the gradients, with the percentage of p, are established in the earlier phase.

In Figure 5.9, interests from Nodes 2 and 4 are probabilistically dropped by Node 1; thus only the interest from Node 3 is updated in Node 1s gradient table. In the next phase, exploratory data flows from the other side to Node 1 and only Node 3 will be chosen as the downstream node. However, if Node 1 does not drop interests from Nodes 2 and 4, the exploratory data flowing from Node 1 to Node 3 will suffer from the interference coming from Node 2 and 4 since Node 1 is also sending exploratory data to them.

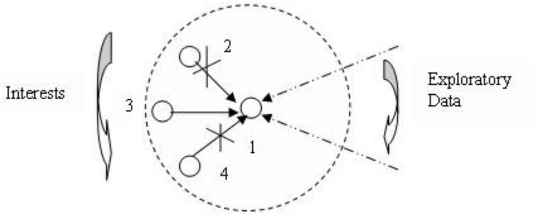


Figure 5.9: Probabilistic interest dropping reduces interference level during exploratory data phase. The dotted circle is the interference range of Node 1.

5.3.2 Epidemic Exploratory Data Propagation

Another way to limit exploratory data flooding is to drop exploratory data probabilistically during the exploratory data phase. The sender selects gradients from the table probabilistically as the downstream links. This is the direct implementation to reduce flooding during the exploratory data phase. Both GOSSIP and Fireworks algorithms can be used.

Obviously, GOSSIP algorithm eliminates a variety of candidates, where the change in interference level may be very abrupt. In Figure 5.10, Node 1 discards the exploratory data completely while Node 1 randomly selects two neighbors ($c = 2$). Although GOSSIP reduces interference in this case, it also eliminates all candidates links from Node 1. Nodes 2 and 2 send to all nodes in the gradient table. Links in the transmission range of Node 2 has the same interference range as those in the transmission range of Node 2. The transmission

Table 5.10: Pseudocode segment for epidemic exploratory data propagation (sender side)

1) Gossip version

```
If it is exploratory data
    Set $r = random()$
    If $r <= p$
        Send it to all nodes in gradient table
    Else
        Discard it
Else
    Execute original code
```

2) Fireworks version

```
If it is exploratory data
    Set $r = random()$
    If $r <= p$
        Send it to all nodes in gradient table
    Else
        Send to randomly selected $c$ nodes in gradient table
Else
    Execute original code
```

of Node 1 rarely affects that of Node 2 if Node 1 and 2 are beyond the interference range. The two transmissions in the transmission range of 1 cause limited interference with each other compared to full flooding. However, the advantage is that more candidate links are being considered.

The combination of Table 5.9 and Table 5.10 above is worth considering. Intuitively, there will be less interference and fewer candidates than either scheme. We will not discuss the details here.

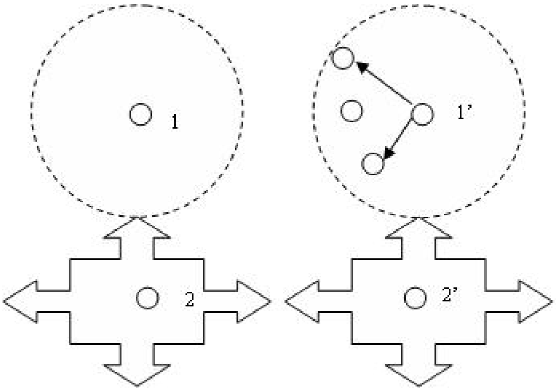


Figure 5.10: Comparison of two gossip-based algorithms. The left is GOSSIP and the right is Fireworks.

CHAPTER 6

IMPLEMENTATION

6.1 EDGE implementation

We conducted both packet level simulation study [73] 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.

6.1.1 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 802.11b protocol in 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 [81]. Even if the encoder generates VBR (variable-bit-rate) traffic, there are methods to optimize its transmission on CBR channel [82]. Usually, it is difficult for sensor nodes to implement the sophisticated video coding techniques used in the MPEGx or H.26x series [83]. Encoding/compression schemes suitable in WMSNs are divided into three categories [34]: 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 we are testing is 40 packets per second, which exceeds 15 packets per second, the afore-mentioned lowest throughput requirement of H.264.

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 ETX to be a little-varying value for a given link. 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. Suppose we have a link composed of nodes A and B ($A \rightarrow B$). T_A and R_A are the error rates of node As 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.

$$d_f = (1 - T_A) \times (1 - R_B), d_r = (1 - T_B) \times (1 - R_A) \tag{6.1}$$

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. delayp 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,

$\cdots$, T_N respectively ($T_1 < T_2 < \cdots < T_N$). *Flow* 1 (T_1) are assigned C_0 as $delay_p$ in the $Cost_p$ formula, *Flow N* (T_N) $INTERVAL$ and *Flow i* (T_i):

$$INTERVAL - \frac{(T_3 - T_i)(INTERVAL - C_0)}{T_3 - T_1} \tag{6.2}$$

or

$$C_0 + \frac{(T_i - T_1)(INTERVAL - C_0)}{T_3 - T_1} \tag{6.3}$$

6.1.2 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 6.1).

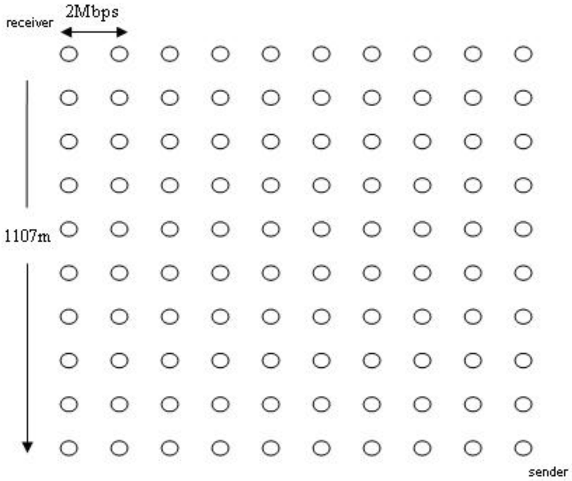


Figure 6.1: A 10-by-10 Grid Topology

Table 6.1: 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 6.1. 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 [84]. In this paper, we use the end-to-end delay of the first packet of a synchronization unit [85] as the target delay. The synchronization unit used in our simulation is 360s, the whole simulation time. We take the maximum of each packets jitter as the jitter of the current run and we take the average of each runs jitter as the jitter of current setting. The jitter we measure is formulated as follows. $jitter_{ji}$ 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 is given by:

$$(\sum_{j=1}^{r}(\max_{i=1}^{p}(jitter_{ji}))/r \tag{6.4}$$

6.2 Multi-path Protocol Implementation

We use NS2 as the simulator to demonstrate the effectiveness of our protocol for achieving much higher throughput and ensuring more packets meet the deadline than single path protocols such as EDGE [73] and basic directed diffusion. We evaluate the performance on a rectangular network with two ray ground and Wu's error model [16] combined with other mathematical formulation [86]. We use the multiple description coded foreman sequence from Video Traces Research Group at Arizona State University [87].

There is a sender and a receiver in the topology. The sizes of the frames the sender generates strictly follow the video trace file. Nodes are evenly distributed in a rectangular network with equilateral triangular tessellations where each node has the same distance to all its closest neighbors (Figure 6.2). The sender is at the bottom left corner and the receiver is at the top right corner. Both of them are one hop inside the border. We modified the codes of directed diffusion to implement our algorithm. All results are obtained from simulations using ns2.29. MAC and channel settings are the same as introduced in section 6.1.

We use the MDC video traces with two descriptions, which are sent over different paths. For EDGE and basic diffusion, the two streams are multiplexed onto a single path. Although the sizes of the frames of each stream vary much, every frame is split up into packets of equal size at the application layer. Thus what the routing layer receives is still CBR traffic. Each stream has 200 frames and they are sent in 8 seconds. Since we are using historical SNR to estimate ETX, we transmit 36 dummy packets ahead of each stream to collect relatively accurate cumulative SNR. The traffic interval of the dummy packets is 5 seconds. BER is calculated from SNR in Chapter 4. We changed the constant $erfc$ 0.5 there to 5×10^{-7} in order to show high enough throughput.

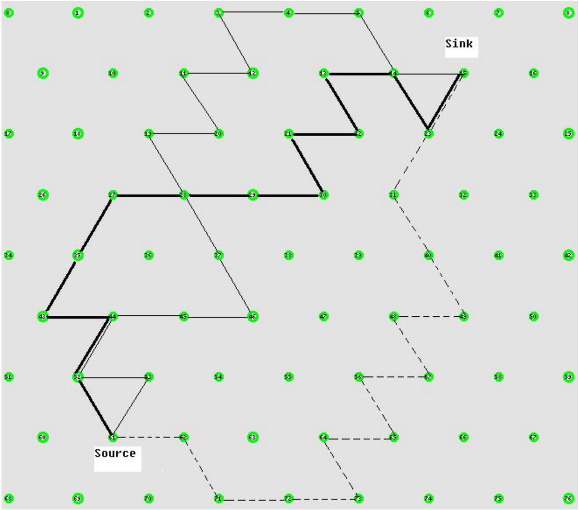


Figure 6.2: Topology with equilateral triangular tessellations. There are 77 nodes in this graph.

Intermediate nodes in our protocol start a timer on receiving the first exploratory data packet. It then buffers all incoming exploratory data packets until the timer expires. The timeout value depends on current link quality and how far the current node is from the source due to the accumulated delay difference of sub-paths. 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. As each node forwards the exploratory data, the ETX_p field in the packet header is updated at each hop. The sink only considers exploratory data packets that meet the deadline. The deadline of exploratory data is set to a different value from that of real application data because of the use of timer in our implementation. When the sink reinforces three links from which the exploratory data with least costs arrived, the reinforcement propagates all the way back to the source through the least cost links recorded by all the intermediate nodes. Using the NEG RESPONSE scheme described in the earlier section, different paths avoids using the same node. The source only chooses the first two paths to send application data.

Periodically the source marks one of its data packets as exploratory and floods it to the sink, in order to discover the two best paths (if they exist) when the network has unstable links.

Probabilistic traffic splitting is not implemented due to the low density of the topology. It is hard to find twice as many paths as number of descriptions.

6.3 Implementation of Epidemic Interest Flooding

We conducted the simulation of epidemic interest flooding with a hybrid metric $Cost_p$ [73] (we use $delay \times ETX^3$ in the simulation) for a modified directed diffusion in ns2. We measure the throughput and delay performance in different topologies with different network sizes and identify the optimal percentage of neighbors that drop interests.

There is a source and a sink in each of the four topologies: grid, triangular tessellation, hexagon tessellation (Figure 6.3) and random. For the first three topologies, the source is at the bottom left corner and the sink is at the top right corner. Both are one hop inside the border. For the random topology, we manually select source-sink pairs whose distances are almost the same. We simulate the algorithm using a modification of directed diffusion release 3.2.0 in ns2.29 with the error and interference model introduced in the earlier section. The MAC and channel settings are the same as in Section 6.1.

We fix the size of random topology to 2000m by 2000m with different numbers of nodes. The simplest CBR traffic is generated at the rate of 10 packets per second.

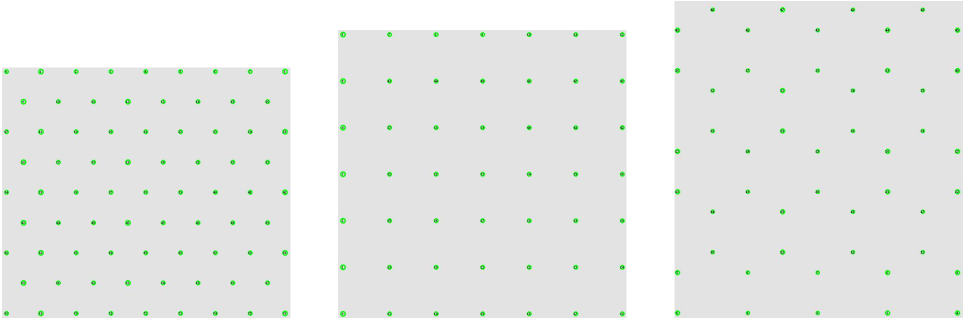


Figure 6.3: Topologies of triangular, grid and hexagon tessellation, from left to right. Every node has 6, 4, and 3 neighbors, respectively.

We plug in the BER formula (we use a much smaller constant 0.0000005 instead of 0.5 because it matches the experimental data more accurately) into the interference model and calculate FER from BER. To make it simple, we set the packet size the same as frame size so that packet loss rate is FER. Although ETX can be computed by broadcasting probe packets before interest flooding, to simplify the simulation, we put the FER into the common header of ns2 and let the protocol read FER and calculate ETX. The noise levels we use in the error model are from the specification of the Orinoco cards. Ns2 has a global scheduler to record all the traffic so that the error model is able to capture the dynamic interference.

CHAPTER 7

PERFORMANCE EVALUATION

7.1 EDGE

We compare the performance of EDGE against the traditional implementation of directed diffusion, in terms of 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 7.1(a). It also shows that as the network size increases, throughput of EDGE decreases more slowly than that of directed diffusion.

7.1(b) compares their end-to-end delay and shows that the delay performance of EDGE is also much better than standard diffusion. Besides, delay of diffusion rises faster than that of EDGE. Both differences of delay and throughput between EDGE and diffusion can be explained by the better (border) links EDGE reinforces. Lossy links not only tend to drop packets; they also bring longer delay due to retransmissions. The larger the network size, the more throughput and delay will be degraded. In [66], 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 $\frac{D(n)}{T(n)}$ increases in Figure 7.1 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 7.2(a), 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: 0.6, 0.7) because EDGE tends to select the route with lossless links. The throughput of EDGE drops from error rate 0.7 to 0.5, goes up till 0.3, and gets stable from then on. On the other hand, the trend of delay is generally reverse C it increases from error rate 0.2 and then decreases at error rate 0.5. The reason why medium error rate 0.5 scenario performs worse than 0.6 and 0.7 scenarios is that exploratory data which flow through those lossy (error rate 0.5) links has a higher probability to be 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 the average error rate of 0.5.

To test EDGEs sensitivity to traffic rate, we use CBR (Constant Bit Rate) traffic. In Figure 7.3(b), as the traffic rate increases from 40 packets per second to 58.8 packets per second, the delay of EDGE gradually becomes longer. When the traffic continues to get busier, delay jumps up drastically. We conjecture that congestion 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 till 166.7 packets per second while directed diffusion stops rising at 100 packets per second. There is a turning point, 100 packets per second, for EDGE during the increasing process, after which the increasing rate of throughput drops 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 packets per second and 166.7 packets per second), the throughput of EDGE keep rising because the packets dropped by congestion are compensated by higher traffic rate. Congestion with a higher degree reduces the throughput. Similar state transition could be found in [88]. Compared with standard diffusion, EDGE gains almost 1.5 times as much as the throughput of standard diffusion when the traffic rate is 166.7 packets per second.

We compare the network delivery ratios of EDGE and DD with three traffic rates 125, 166.7 and 200 packets per second in Figure 7.4. It further proves that the percentage of packets received drops down when the traffic gets busier. EDGE always has higher ratios than DD. When the traffic rate rises, nevertheless, the gap becomes smaller because both EDGE and DD are suffering from severe congestion.

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, which is shown by the result that jitter of diffusion, on the average, is 8 times as long as that of EDGE. EDGE achieves best performance when the links are very good (error rate: 0.2) and very bad (error rate: 0.7), which can be explained as lossless links are reinforced instead of very bad ones. For links with medium error rate (0.3, 0.4 and 0.5), error links may have a higher probability to be selected than worse links (error rate: 0.6 or 0.7), which contributes to the worst jitter.

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

$$ETX_p = \max_{i=0}^{N-3} \sum_{j=i}^{i+4} (ETX_j) \tag{7.1}$$

We compare throughput and delay between EDGE with different ETX_p formulae, in which the 5-hop one is closer to our settings, in Figure 7.6. It is observed that 5-hop EDGE has more chance to reinforce lossless links than its counterpart. In lower traffic rate scenarios, there is little difference in throughput and delay. When the traffic gets busier, the benefits of 5-hop EDGE get clearer.

We adjust the α and β values in $Cost_p$ formula trying to improve throughput or delay of our algorithm. In Figure 7.7(a), throughput increases rapidly when ETX is taken into account (α from 0 to 1). For other combinations, there is no significant improvement for throughput. It achieves the highest throughput when only ETX is considered ($\beta = 0$). Likewise, delay is lowest when $\alpha = 0$. It is different from diffusion because EDGE has an adaptive timer at each node. ETX somewhat helps decrease delay, as we can see in Figure 7.7(b) that peak point is at $\alpha = 3$ or 4 instead of 6. However, pure ETX ($\beta = 0$) worsens delay to a large extent.

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 [9], 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 less than the max jitter 32 ms in [89]. We plan to test our protocol with test-bed PC104 and observe whether EDGE is still able to reach the QoS requirements mentioned earlier in video data transmission. Our current results also demonstrate the tradeoff between throughput and delay.

7.2 DCHT

7.2.1 Performance

We compare our protocol with EDGE and basic diffusion over different network sizes of 46, 77, 116 and 163 nodes configured in the rectangular area similar to Figure 6.1. The distance of every neighbor pair is always 200m. The size of packets is 128 bytes, greater than half of the smallest video frame which is 187 bytes. We use a deadline of 200ms because the playout deadline varies from 50ms to 200ms [9]. The deadline for the exploratory packets is 2000ms. To be fair, EDGE uses similar ETX estimation from SNR.

We measure the performance of 11 simulation runs with randomly generated seed. The simulation time of each run is 1000s in ns2. To guarantee the robustness of our protocol, we changed the gradient expiration time period to 1000s, the same as the simulation time. We compare throughput (packets per second), end-to-end delay (ms), and goodput (packets per second). Goodput is defined as the number of packets that meet the deadline. We use the same seed to show the number of loss packets per frame for the 2 streams in the network of 77 nodes (Figure 6.2) with both our protocol and EDGE. In the formula,$Cost_p = {ETX_p}^{\alpha} \times {delay_p}^{\beta}$, we use $\alpha = 10$, $\beta = 1$ since delay has only slight difference for each link as observed in the experiment. $\gamma = 0.7$ in Equation 4.5. The total number of application packets sent at the source, excluding dummy packets, is 2334.

In Figure 7.8, we show that the DCHT gives twice as much throughput as EDGE. Basic diffusion can hardly receive any packet when network size goes up to 77 or above because it does not consider link quality at all. Intuitively, the throughput of DCHT should be less than that of EDGE because only the better path is as good as the path chosen in EDGE and the other path is worse. Our protocol, in fact, achieves load balancing by using more than one path, compared with EDGE which drops many packets caused by congestion in a single path. When the network size is small, such as 28, there is little difference between basic diffusion and EDGE because throughput is not affected much by the link quality when there are only a few hops. When the network size increases, both our protocol and EDGE suffer from accumulative link loss.

As we have defined, goodput only considers packets that meet the deadline. In Figure 7.9, both EDGE and basic diffusion perform much worse than our protocol because they do not take into account the playout deadline. Apparently, diffusion should have good performance because it always reinforces the link with best delay, which definitely meets the deadline. Without collecting link quality statistics, however, failure may occur because link quality changes often even within the same cycle (60s) and before the reinforcement is sent in the next cycle. By adding deadline comparison at the sink side in DCHT, paths that have poor end-to-end delay performance are filtered ahead of time. When link quality varies with time, ETX calculated by historical SNR helps in selecting relatively good paths. When network sizes are greater than 100, even DCHT cannot guarantee 1/4 of the packets are received on time.

The average end-to-end delay in Figure 7.10 does not exclude the packets that cannot meet the deadline. Basic diffusion shows low delay because it always chooses a path with lowest delay (delay is the only routing metric). We also explain the low delay of diffusion because of its routing metric delay. The end-to-end delay of DCHT stabilizes a little above 200ms, the playout deadline. EDGE rises quickly in delay and drops after the network size of 77 when a large number of packets are dropped.

Another metric important for MDC video streaming is number of loss packets per video frame. Packet loss includes those that cannot meet the deadline. We only compare our protocol with EDGE in Figures 7.11 and 7.12 with the network size of 77 nodes (Figure 6.2) since the throughput of diffusion is very low. We only show the profile with a certain seed. It makes no sense if we average the profiles of all seeds. It can be seen that EDGE has higher loss rates than DCHT. Besides, the two loss traces of EDGE are highly correlated since they are using the same path. Furthermore, the two paths found in DCHT are quite different, which leads to more successful transmission of Stream 2 than that of Stream 1. The reason why we have regular spikes in both Figure 7.11 and 7.12 is that the video trace we use for the foreman sequence has a large frame (more than 2000 bytes) every 12 frames. These large frames have a large number of loss packets per frame.

We marked the two paths in solid and dashed lines in Figure 6.2 and generated the profile in Figure 7.11. Since the solid path reinforcement reaches the source first, Stream 1 is sent over the solid path, which gives a worse performance. We analyze the performance difference between the solid and dashed paths as follows: 1) The solid path has two more hops than the dashed path and thus has more chance to be affected by cumulative link loss; 2) the solid path is relatively closer to the border, which has less interference. The single path reinforced in EDGE, which matches Figure 7.12, is marked in bold. Although there is high packet loss, the bold path is composed of links close to the border. The overhead of our protocol, i.e. packets used for routing, is almost the same as that of basic diffusion because we do not introduce any new routing packet type except $NEG\_RESPONSE$ which rarely occurs even in reinforcement.

The number of paths should match the number of descriptions of the MDC encoder. We also send 3 and 4 descriptions over 3 and 4 paths respectively and compare the throughput and goodput performance in Figure 7.13. Due to the low density of the network, it is difficult to find 3 or even 4 paths within each cycle. We multiplex the rest of the streams with those which have already got a path to send over when there is not adequate number of paths found, e.g. σ streams are to be disseminated over ρ paths ($\sigma > \rho$). Streams $\rho + 1$, $\rho + 2$, $\cdots$, σ are multiplexed with Streams 1, 2,$\cdots$, $\sigma - \rho$ respectively. The 1-description case is different from EDGE because it considers deadline when selecting the paths. Also, for the 1-path case, we do not reinforce one extra path at the sink side. As we can see from the curves, both the throughput and goodput improve much when the number of paths (descriptions) goes up. Goodput from the 2-path case to the 3-path case does not increase much because in some cycles only one or two paths can be found. 4-path case still gets high goodput because four paths can be found with a high probability according to our simulation results. In Figure 7.14, we show the number of packets and packets that meet the deadline received at the sink. The total numbers of packets sent at the source increase with more descriptions because overhead is added to each description, which only causes slight difference as seen from both Figure 7.13 and Figure 7.14. The end-to-end delay (Figure 7.15) decreases when we use more paths since packets generated at the source are split up into more paths which lightens the traffic burden. Congestion is alleviated by using more paths although finding more paths increase overhead.

We also compare the energy consumption and routing overhead in the DCHT protocol with basic diffusion and EDGE. Intuitively, our protocol may consume more energy since we are using more resources of the networks. In Figure 7.16, it is shown that the DCHT protocol does not consume additional energy, compared with basic diffusion and EDGE (two single-path protocols). A single path does not have enough bandwidth, which leads to congestion in video streaming. This results in more retransmission which requires more energy consumption and end-to-end delay. Also, larger network consumes less energy, which is a surprising fact observed in our simulation. The reason might be that it is easier to build multiple paths in a larger network which helps disseminate the traffic. The influence of different paths (descriptions) over energy consumption is a concave (Figure 7.17), which is better than linear increase. More energy consumption leads to the discovery of more paths, which transmit video data more efficiently.

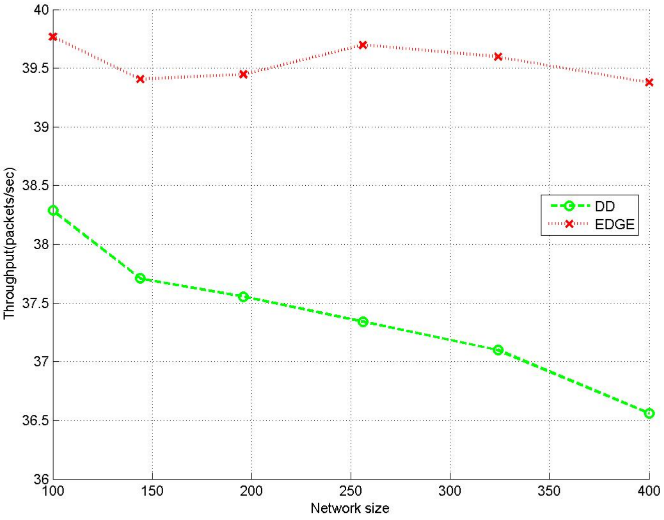
The only extra overhead in our protocol is the use of $NEG\_RESPONSE$ in order to find multiple disjoint paths (Figure 7.18). As we have estimated, it is harder to find more paths, especially when we try to avoid the same node. So it requires more $NEG\_RESPONSE$ messages to find 4 paths, considering that each node only has 6 neighbours in our topology. When we have larger network sizes, we have higher probability to get multiple paths without sharing nodes with other paths.

7.2.2 Discussion

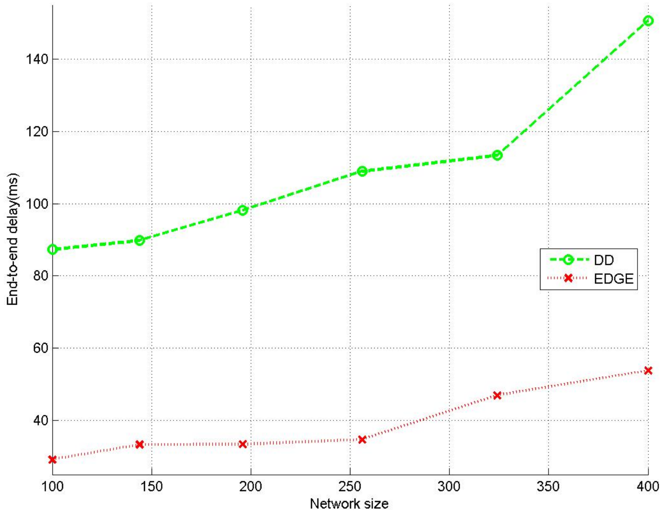
The goodput of our protocol with 4 paths in 77-node network can reach is above 180 packets per second. The packet size is 128 bytes and foreman sequence finishes transmitting in 8 seconds. The corresponding data rate is 23 kbps, which barely meets 28 kbps, the data rate of the low quality video. By sending video through more paths, the goodput could even be increased to meet the high quality video requirements. The average end-to-end delay of the 4-path case is about 120 ms, much smaller than the 200 ms deadline we set.

7.3 Epidemic flooding

We run each setting, with different topology and percentage of nodes that do not drop interests, for 50 times and take the average of the metrics. In our discussions below, let p be the fraction of nodes that do not drop interests. We compare the throughput, end-to-end delay and directed diffusion related metrics, such as, percentage of cases where no interest reaching the source, percentage of cases where no path was built (which includes the previous cases) and percentage of cases with zero throughput (which includes the previous two cases). Each run lasts for 300 seconds.

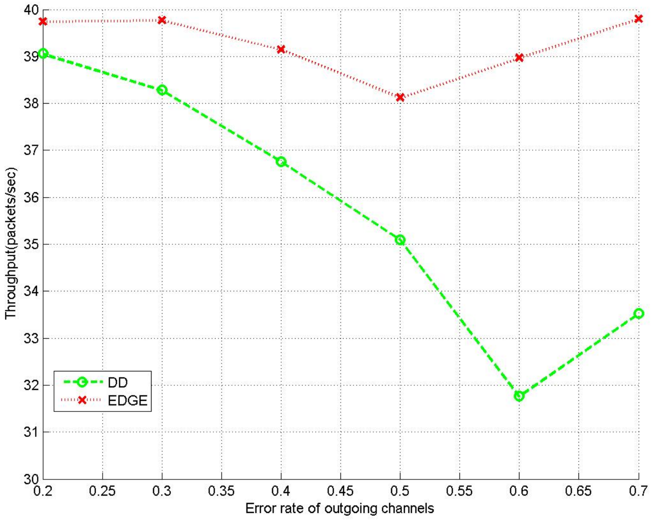


(a) Throughput with different network sizes

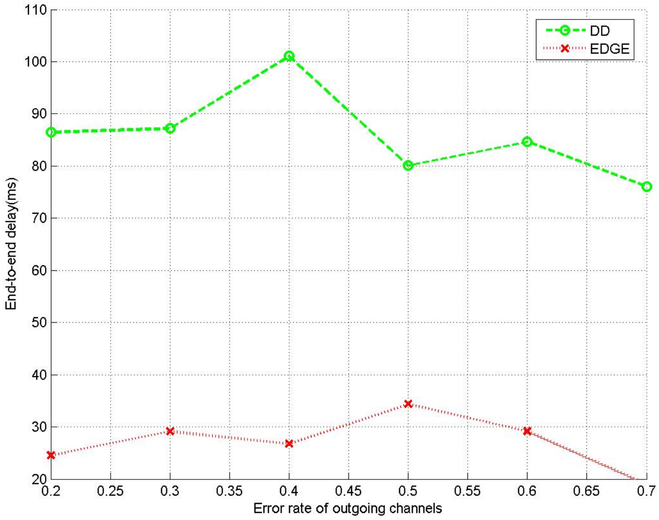


(b) End-to-end delay with different network sizes

Figure 7.1: Throughput and delay comparison between EDGE and directed diffusion in grid topologies. The error rate of outgoing channels follows uniform distribution with average error rate of 0.7. $\alpha = 1$, $\beta = 1$ in $Cost_p$. Packet size is 500 bytes.

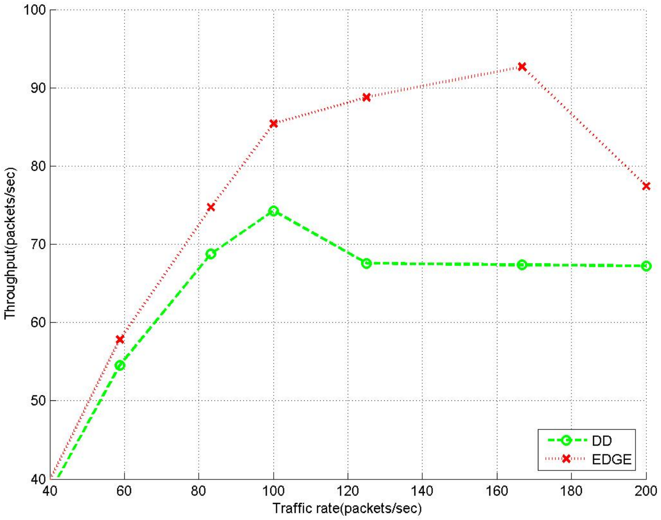


(a) Throughput with different delivery ratios of lossy links

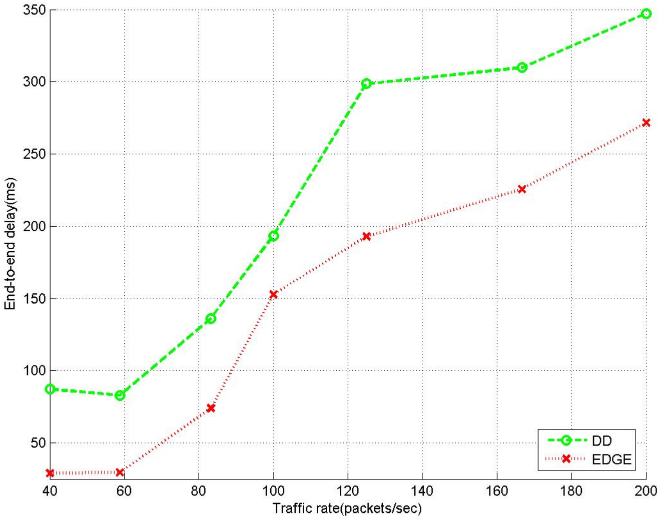


(b) End-to-end delay with different delivery ratios of lossy links

Figure 7.2: 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.



(a) Throughput with different traffic rates



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

Figure 7.3: 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 average error rate of 0.7. $\alpha = 1$, $\beta = 1$ in $Cost_p$. Packet size is 500 bytes.

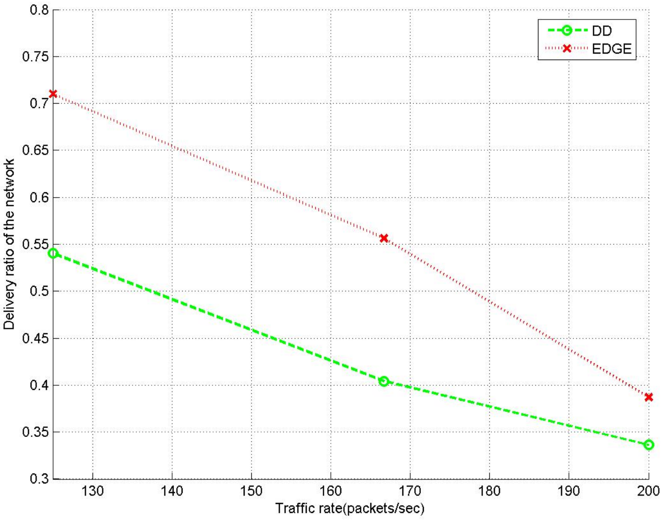


Figure 7.4: Network delivery ratio comparison between EDGE and directed diffusion in 10-by-10. The error rate of outgoing channels follows uniform distribution with average error rate of 0.7. $\alpha = 1$, $\beta = 1$ in $Cost_p$. Packet size is 500 bytes.

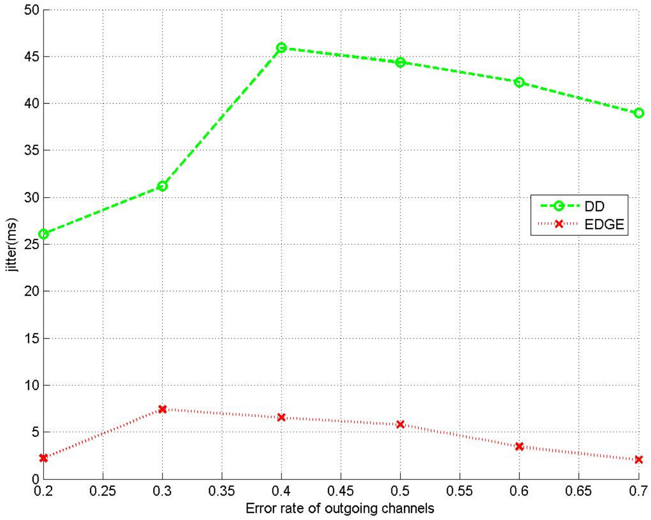
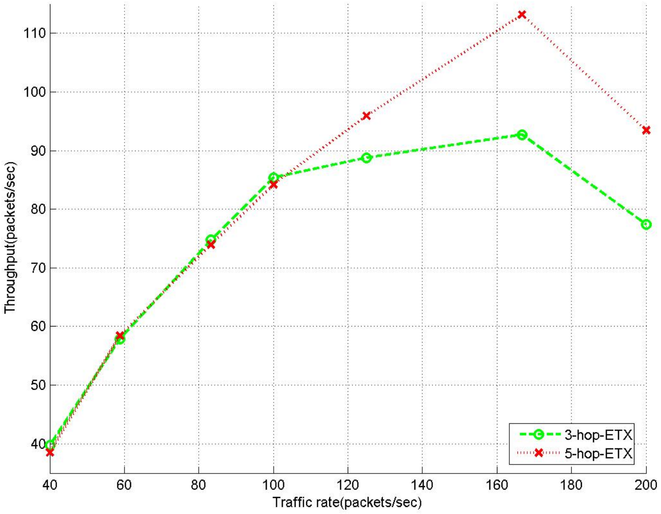
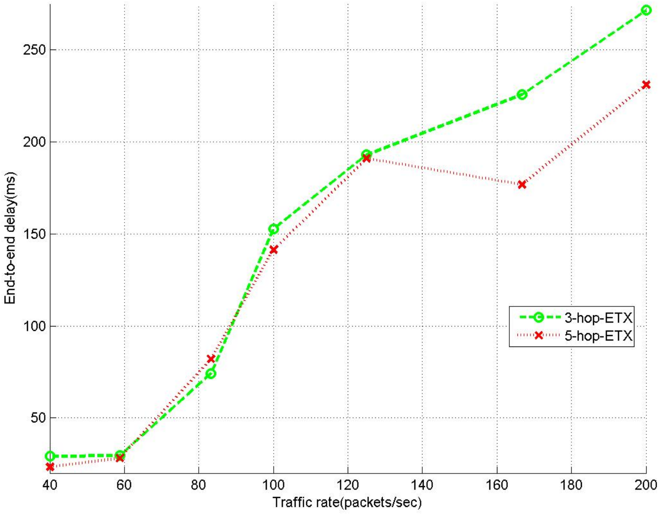


Figure 7.5: Jitter comparison between EDGE and directed diffusion in 10-by-10 grid with different delivery ratios. $\alpha = 1$, $\beta = 1$ in $Cost_p$. Traffic rate is 40 packets per second. Packet size is 500 bytes.

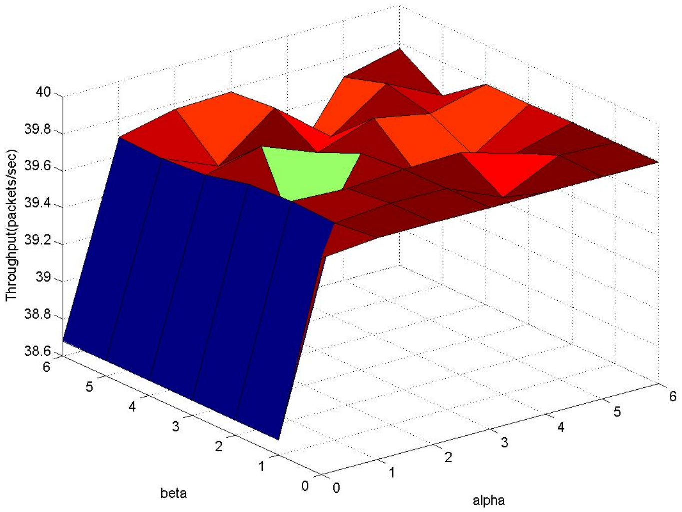


(a) Throughput with different traffic rates

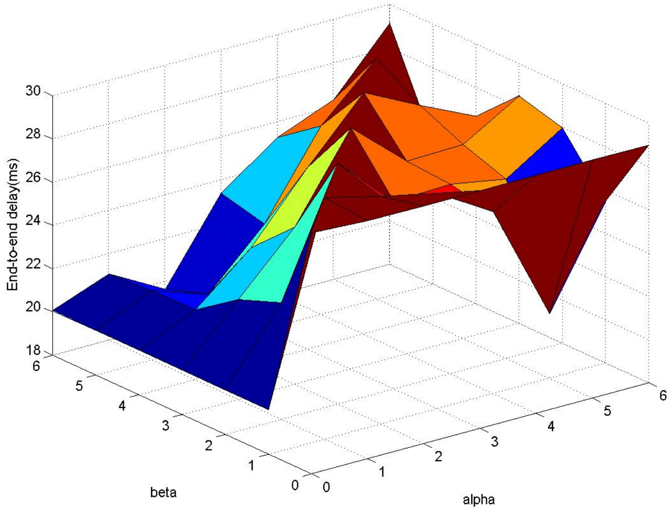


(b) End-to-end delay with different traffic rates

Figure 7.6: Throughput and delay comparison between 3-hop EDGE and 5-hop EDGE in 10-by-10 grid with different traffic rates. The error rate of outgoing channels follows uniform distribution with 0.7. $\alpha = 1$, $beta = 1$ in $Cost_p$. Packet size is 500 bytes.



(a)



(b)

Figure 7.7: Throughput and delay comparison with different α and β values in $Cost_p$ of EDGE in 10-by-10. The error rate of outgoing channels follows uniform distribution with average error rate of 0.7. $\alpha = 1$, $\beta = 1$ in $Cost_p$. Packet size is 500 bytes. Traffic rate is 40 packets per second.

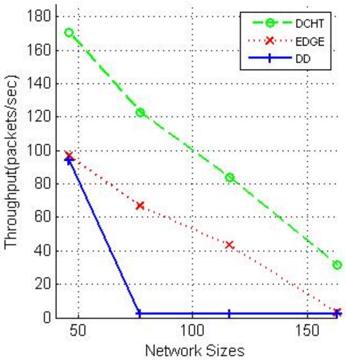


Figure 7.8: Throughput of DCHT, EDGE and basic diffusion with different network sizes.

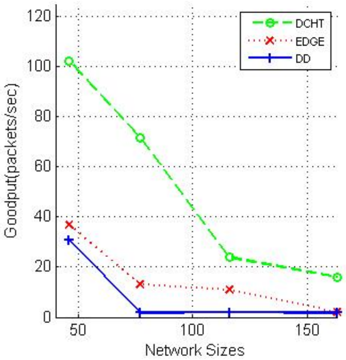


Figure 7.9: Goodput of DCHT, EDGE and basic diffusion with different network sizes.

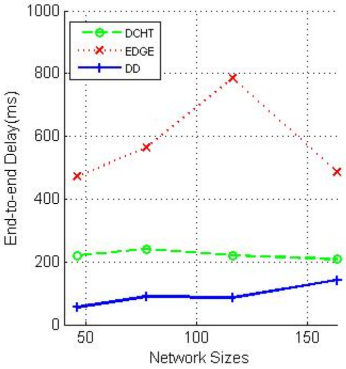


Figure 7.10: End-to-end delay of DCHT, EDGE and basic diffusion with different network sizes.

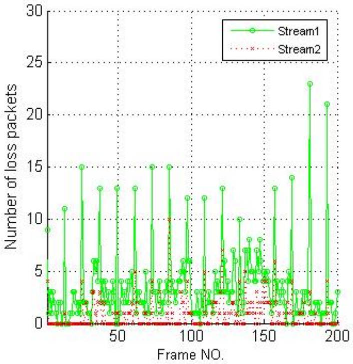


Figure 7.11: Number of loss packets per frame in DCHT. There are 29 packets in a large video frame.

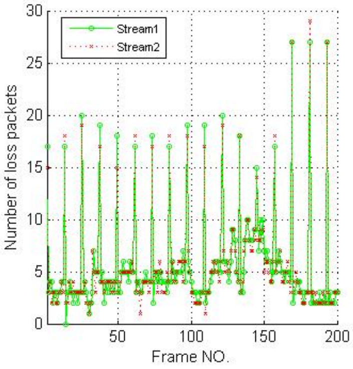


Figure 7.12: Number of loss packets per frame in EDGE. There are 29 packets in a large video frame.

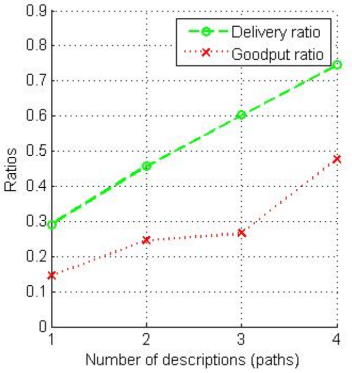


Figure 7.13: Delivery ratio and goodput ratio of DCHT with different numbers of descriptions/paths. The network size is 77 nodes.

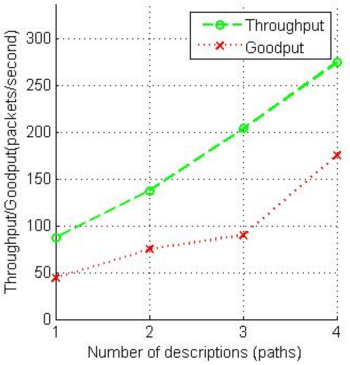


Figure 7.14: Throughput and goodput of DCHT with different numbers of descriptions/paths. The network size is 77 nodes.

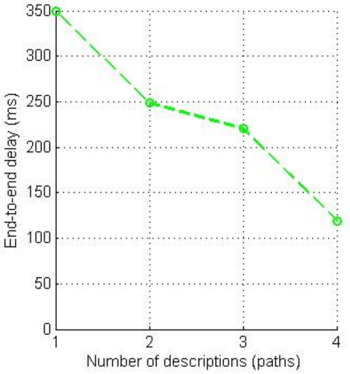


Figure 7.15: End-to-end delay of DCHT with different numbers of descriptions/paths. The network size is 77 nodes.

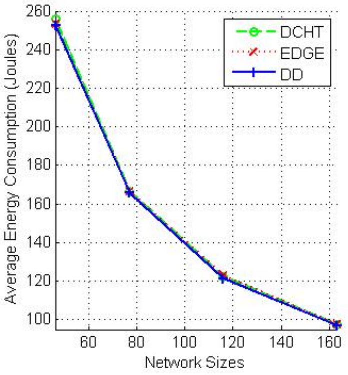


Figure 7.16: Average energy consumption of a single node in the network of DCHT with 3 paths (descriptions), EDGE and basic diffusion with different network sizes.

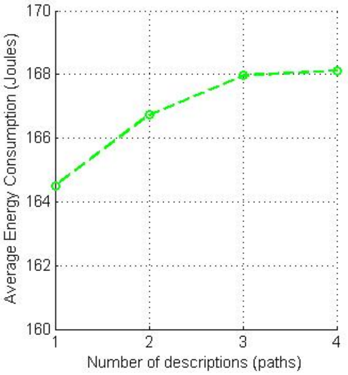


Figure 7.17: Average energy consumption of a single node in the network of DCHT with different number of paths (descriptions). The network size is 77 nodes.

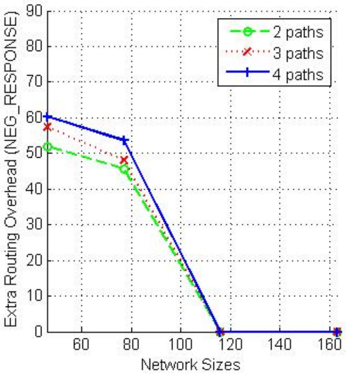


Figure 7.18: Extra routing overhead (we only consider $NEG\_RESPONSE$) of the whole network of DCHT with different number of paths (descriptions) in different network sizes.

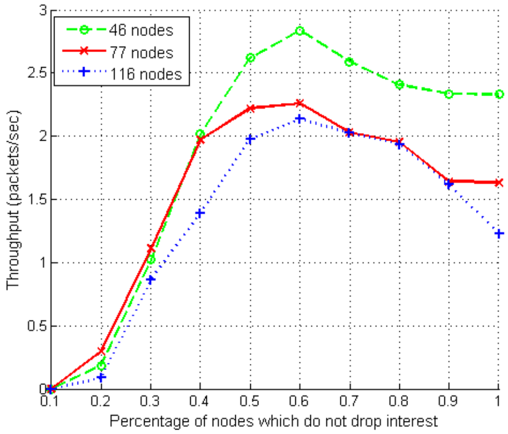


Figure 7.19: Throughput of triangular tessellation topology with different sizes.

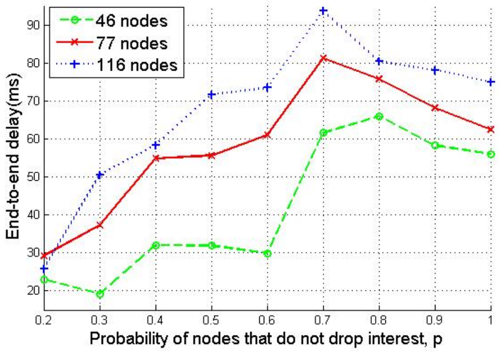


Figure 7.20: End-to-end delay of triangular tessellation topology with different sizes.

We first run the application for the triangular tessellation with different network sizes as shown in Figure 6.3 and measure the throughput (Figure 7.19) and end-to-end delay (Figure 7.20). The throughput increases rapidly when *p* increase from 0.1 to 0.5. It achieves the highest throughput when 60% of the nodes do not drop interests. Beyond 60%, the throughput drops because of the extra interference during exploratory data phase when more or all candidate gradients are considered (fewer or no node drop interests). Note that when $p = 1.0$, it becomes the original directed diffusion algorithm and the throughput is lower than the peak, when $p = 0.6$, which is 20% higher than the original directed diffusion. When many nodes drop interests, i.e. p is small, the throughput is lower because the number of candidates is very small. Network with smaller sizes achieves higher throughput since it suffers less from cumulative packet loss. Figure 7.20 shows that end-to-end delay goes up to 80 ms when $p = 0.7$ or $p = 0.8$ and drops down after that. The higher delay is due to high traffic when the throughput peaks. Larger networks have longer delay since there are more hops between the source and the sink. Trade-off between throughput and delay is also shown in these two figures. Although we do not have the highest delay when $p = 0.7$ where throughput is maximum, we believe they reflect the trend that there should be a magic percentage, between 60% and 70% (or 80%), for this topology.

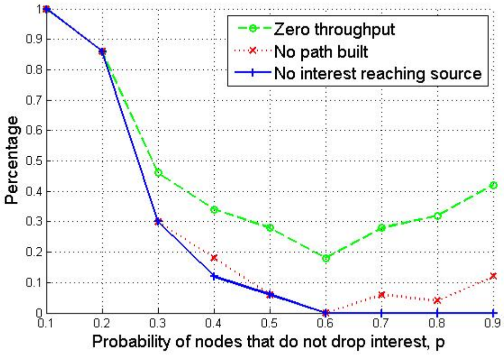


Figure 7.21: Percentage of zero throughput, no path built and no interest reaching source of triangular tessellation topology with 77 nodes.

One of the shortcomings of probabilistic forwarding is that in some cases no forwarding path for interest, exploratory data or application data, may be found. We analyze the above-mentioned probability phase by phase. In the first phase, interests may fail to reach the source due to the epidemic flooding. We measure the percentage of such instances in Figure 7.21. If more nodes are updating the gradient table, the percentage of interests reaching the source is monotonically higher. Also, we compare the percentage of cases where no path was built, which includes the cases where no interest reaches the source. Another reason for this might be that interference causes exploratory data to be dropped half way even if interests can reach the source successfully. As a result, it is no surprise to find that when $p = 0.6$, more paths are built than when $p = 1$ (original directed diffusion). Percentage of cases with zero throughput (plotted in the same figure) includes the cases where no path is built. When $p = 0.6$ we not only achieves the highest throughput, but also achieve the highest percentage of successful delivery of application data.

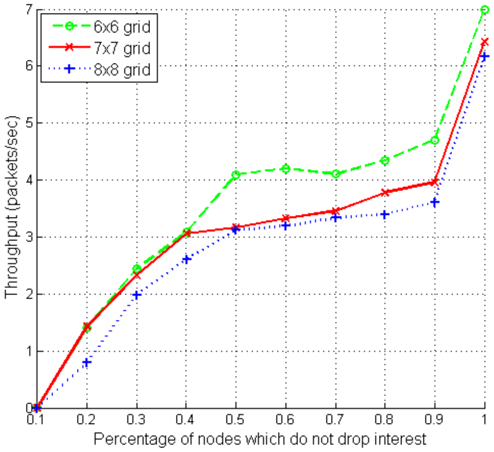


Figure 7.22: Throughput of grid topology with different sizes.

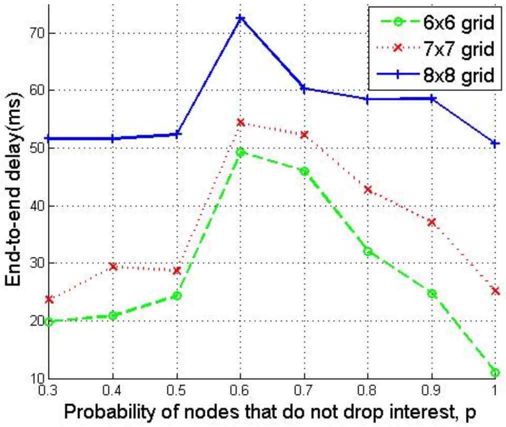


Figure 7.23: End-to-end delay of grid topology with different sizes.

Figure 7.22 and Figure 7.23 show the throughput and delay in grid topology with different network sizes. Delay follows almost the same trend as in the previous topology although the magic percentage is 0.6 here. Throughput looks quite different in grid topology and it keeps increasing till all nodes update their gradient tables. The reason is that every node has only four neighbors, which is small and hard to guarantee connectivity. Compared to the previous topology with six neighbors, the product of 6 and 0.6 is approximately 4, which is evidence that a magic number, instead of magic percentage, does exist for the number of neighbors with different topologies.

Hexagon tessellations with different sizes are compared in Figure 7.24 and Figure 7.25 in terms of throughput and end-to-end delay. Throughput keeps increasing with higher percentages of nodes that do not drop interests since the topology has only three neighbors, even smaller than grid topology. The peak of end-to-end delay is $p = 0.7$ or $p = 0.8$, which is higher than grid topology at 0.6. The highest delay is due to high traffic level.

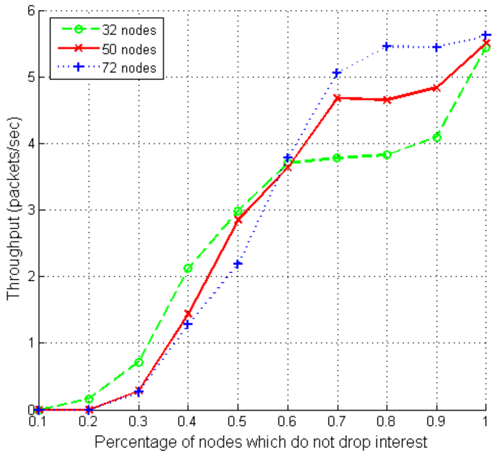


Figure 7.24: Throughput of hexagon tessellation with 32, 50, 72 nodes.

For throughput, $p = 0.6$ is a breakpoint, beyond which smaller network sizes have larger delay than larger network sizes. We explain this phenomenon as: the probability of larger networks having connectivity increases much more rapidly than that of smaller networks. Figure 7.25 shows that in all network sizes, the highest end-to-end delay occurs at $p = 0.7$ or $p = 0.8$ due to the high traffic level.

We also test random topology with different network sizes (Figure 7.26 and Figure 7.27). Throughput always follows the up and down trend and the network with 60 nodes performs the best. The magic percentage varies between 0.3 and 0.8. Network size of 60 nodes achieves the highest throughput at a higher percentage than that of 80, 90, or 100 nodes because every node has fewer neighbors. The delay graph (Figure 7.27) also indicates the magic number. The delay of 80-node topology keeping increasing till $p = 1$. Smaller network achieve lower end-to-end delay than larger ones most of the time. Since nodes are uniformly distributed in this random topology, given the transmission range, area of the

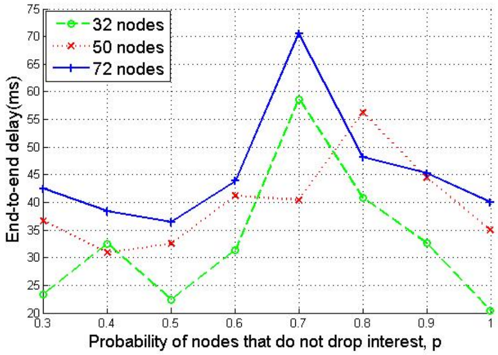


Figure 7.25: End-to-end delay of hexagon tessellation with 32, 50, 72 nodes.

network, and number of nodes, we calculate the average number of neighbors is 3.9 (almost 4) in the 80-node network and 2.9 (almost 3) in the 60-node network. We may give the same explanation as in Figure 7.22 or Figure 7.23 although that is for throughput.

To show the magic number for different topologies, we put all four of them in the same figure (Figure 7.28 and Figure 7.29). We keep the hop count between the source and the sink the same for them. In Figure 7.28, hexagon tessellation performs the best and random topology the worst. Fewer neighbors cause less interference. Although the average number of neighbors of random topology is 4, the uneven deployment aggravates the side-effects of interference. An interesting thing is that hexagon tessellation throughput exceeds that of the 7 by 7 grid at $p = 0.6$. The reason is that it is hard for the hexagonal tessellation topology to maintain connectivity with two few neighbors. Hexagon tessellation also performs best in end-to-end delay.

To further test the detrimental effect of full flooding, we put the throughput (end-to-end delay) of the three regular topologies with different sizes in the same figure (Figure 7.30 and Figure 7.31). We still make the hop count between the source and the sink the

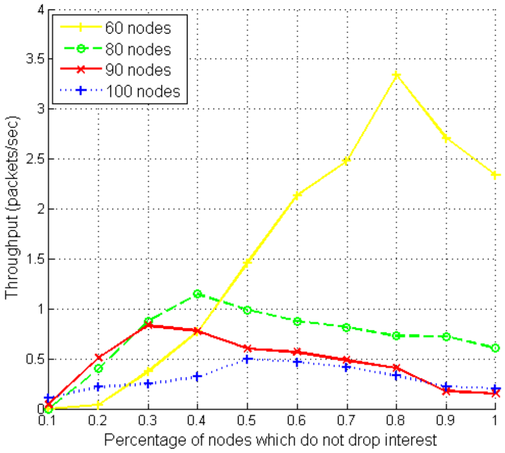


Figure 7.26: Throughput of random topology with 60, 80, 90, 100 nodes.

same for each point in the same column. Hexagon tessellation is affected less than others in both throughput and end-to-end delay since it has the smallest number of neighbors. The grid topology has the best overall throughput and delay performance since it has the magic number of neighbors.

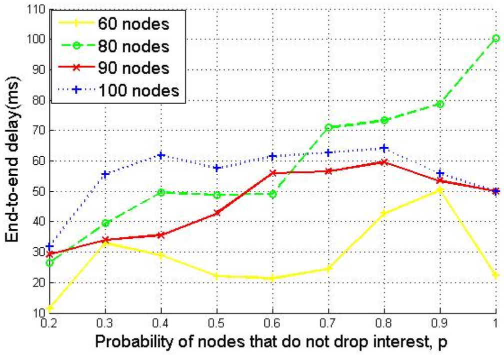


Figure 7.27: End-to-end delay of random topology with 60, 80, 90, 100 nodes.

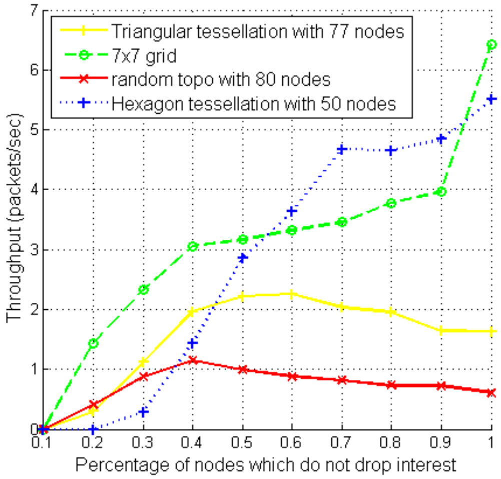


Figure 7.28: Throughput comparison of 4 topologies.

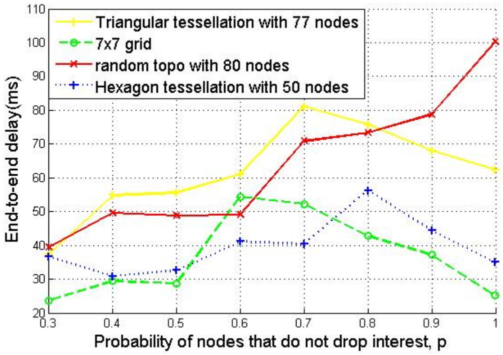


Figure 7.29: End-to-end delay comparison of 4 topologies.

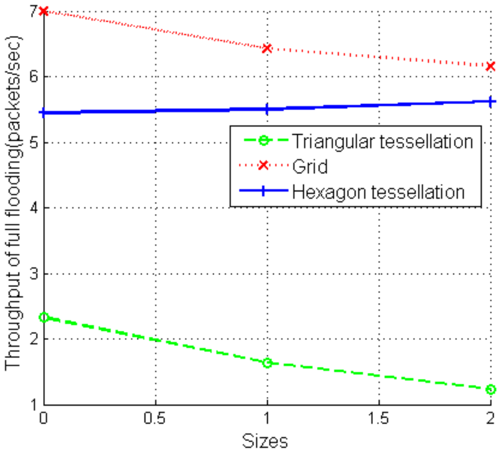


Figure 7.30: Throughput comparison of 3 topologies with full flooding.

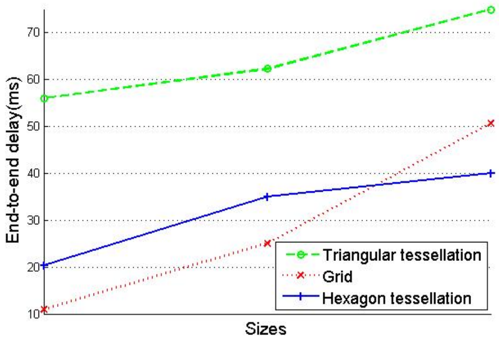


Figure 7.31: End-to-end delay comparison of 3 topologies with full flooding.

Chapter 8

Conclusions and Future Work

We have proposed, implemented, and evaluated three diffusion-based protocols for wireless multimedia sensor networks. They improve throughput and maintain reasonably low delay. They select paths with high throughput, mitigate the effects of interference, and support real-time traffic transmission.

The main contributions of our work are listed below.

- Incorporation of a hybrid metric into directed diffusion to increase throughput and maintain low day.
- Implementation of a multi-path routing protocol in sensor networks to achieve load balancing and error recovery with more accurate error model.
- Improvement on routing decision by limiting flooding and reducing interference.

The protocols we implemented meet the overall goals of video sensor network. The throughput performance of EDGE is on the average 15% better, jitter is 8 times shorter and delay is 20% less than directed diffusion for the scenarios we investigated. The main reason is EDGE has more chance 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.

In DCHT, multiple disjoint paths can achieve high throughput and desirable delay and meet the QoS requirement of multimedia streaming. The use of Wus error model and related mathematical formulas allow us to simulate real-world wireless link loss properly. Our protocol gives twice as much throughput as EDGE and its goodput is even better than that of EDGE. The end-to-end delay of our protocol is 1/4 of that of EDGE in the best case. Basic diffusion is not comparable at all since it can hardly receive any packet on time, especially in large networks.

For the epidemic flooding algorithm, we observe that different topologies have different magic percentages of neighbors to achieve optimal routing decision (which is different from the connectivity problem). We also identify the magic number of neighbors that can be applied to different topologies for optimizing throughput and delay. For all the topologies we have tested, the optimal number of neighbors for achieving the best throughput is four. When the number of neighbors is greater than four, maximum throughput is achieved when some nodes drop interests, i.e. the percentage of nodes not dropping interest is less than 100%. In these cases, when more nodes drop interests, the level of interference is reduced and higher throughput is achieved in the resulting routes. As the number of neighbors decreases, the optimal percentage of nodes that do not drop interests for achieving maximum throughput moves toward 100%. In most cases, delay is highest when the percentage of nodes not dropping interest is optimal. Random topologies perform worse than topologies with regular tessellation no matter what density we choose because of the uneven distribution of nodes resulting in hotspots with high interference level and difficulty for the routing algorithm to estimate interference characteristics.

There are many possibilities for the future research work in the area of routing protocols for video sensor networks. Inter-flow interference is hard to formulate and locally detect due to the irregular topology and unpredictable traffic patterns. In this way, interference caused by inter-flow interference is not accurately estimated. Another possibility is to test adapt our EDGE and DCHT protocols to random topologies to achieve equivalently desirable performance in regular topologies such as grid and tessellations. Besides, the incorporation of the epidemic flooding protocol into DCHT is worth investigating.

BIBLIOGRAPHY

[1] V.K. Goyal. Multiple description coding: compression meets the network. *Signal Processing Magazine, IEEE*, 18(5):74–93, Sep 2001.

[2] 21 ideas for the 21st century. Business Week, pp. 78-167, August 1999.

[3] Chee-Yee Chong and S.P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug. 2003.

[4] V. K Varadan J. W. Gardner. *MEMS and Smart Devices.* New York: Wiley, 2001.

[5] S. Kumar and D. Shepherd. Sensit: Sensor information technology for the warfighter. In *Proc. 4th Int. Conf. on Information Fusion*, 2001.

[6] T.; Chowdury K.R. Akyildiz, I.F.; Melodia. Wireless multimedia sensor networks: A survey. In *Wireless Communications, IEEE [see also IEEE Personal Communications] , vol.14, no.6, pp.32-39*, 2007.

[7] H.264/mpeg-4 avc. http://en.wikipedia.org/wiki/H.264.

[8] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, 2003.

[9] A.C. 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*, 1:I–517–20 vol.1, 6-9 July 2003.

[10] S. Mao, S.S. Panwar, and Y.T. Hou. On optimal partitioning of realtime traffic over multiple paths. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 4:2325–2336 vol. 4, 13-17 March 2005.

[11] S.-J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. *Communications, 2001. ICC 2001. IEEE International Conference on*, 10:3201–3205 vol.10, 2001.

[12] Marc R. Pearlman, Zygmunt J. Haas, Peter Sholander, and Siamak S. Tabrizi. On the impact of alternate path routing for load balancing in mobile ad hoc networks. In *MobiHoc '00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 3–10, Piscataway, NJ, USA, 2000. IEEE Press.

[13] Deepinder Sidhu, Raj Nair, and Shukri Abdallah. Finding disjoint paths in networks. In *SIGCOMM '91: Proceedings of the conference on Communications architecture & protocols*, pages 43–51, New York, NY, USA, 1991. ACM.

[14] Shiwen Mao, Shunan Lin, S.S. Panwar, Yao Wang, and E. Celebi. Video transport over ad hoc networks: multistream coding with multipath transport. *Selected Areas in Communications, IEEE Journal on*, 21(10):1721–1737, Dec. 2003.

[15] Yang Wang and Shunan Lin. Error resilient video coding using multiple description motion compensation. *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, pages 441–446, 2001.

[16] Wu Xiuchao. Simulate 802.11b channel within ns2. Technical report, SOC, NUS.

[17] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, 2000.

[18] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM.

[19] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward. A distance routing effect algorithm for mobility (dream). In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 76–84, New York, NY, USA, 1998. ACM.

[20] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353, 1996.

[21] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, 25-26 Feb 1999.

[22] Zygmunt J. Haas and Marc R. Pearlman. The performance of query control schemes for the zone routing protocol. In *SIGCOMM '98: Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 167–177, New York, NY, USA, 1998. ACM.

[23] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, page 1405, Washington, DC, USA, 1997. IEEE Computer Society.

[24] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated

database maintenance. In *PODC '87: Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 1–12, New York, NY, USA, 1987. ACM.

[25] Zygmunt Haas, Joseph Y. Halpern, and Li Li. Gossip-based ad hoc routing. Technical report, Ithaca, NY, USA, 2001.

[26] L. Orecchia, A. Panconesi, C. Petrioli, and A. Vitaletti. Localized techniques for broadcasting in wireless sensor networks. In *DIALM-POMC '04: Proceedings of the 2004 joint workshop on Foundations of mobile computing*, pages 41–51, New York, NY, USA, 2004. ACM.

[27] Ranveer Chandra, Venugopalan Ramasubramanian, and Kenneth Birman. Anonymous gossip: Improving multicast reliability in mobile ad-hoc networks. Technical report, Ithaca, NY, USA, 2001.

[28] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks, 2002.

[29] K. Kar P. Chaporkar and S. Sarkar. Throughput guarantees through maximal scheduling in wireless networks. In *Proc. 43rd Allerton Conf. on Commun., Control, and Comp.*, 2005.

[30] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 66–80, New York, NY, USA, 2003. ACM.

[31] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: enabling ultra-low power wireless research. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 48, Piscataway, NJ, USA, 2005. IEEE Press.

[32] Crossbow technology. http://www.xbow.com/. Accessed July 24, 2007.

[33] Pc104.com. http://www.32.com/. Accessed July 24, 2007.

[34] Martin Reisslein Satyajayant Mishra and Guoliang Xue. A survey of multimedia streaming in wireless sensor networks. under submission.

[35] J. L. Wang 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.

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

[37] Video sensor network laboratory at ucr receives federal funding. http://www.newsroom.ucr.edu/cgi-bin/display.cgi?id=1298. Accessed July 24, 2007.

[38] Wu-chang Feng Calton Pu 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.

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

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

[41] Wireless multimedia sensor networks. http://www.ece.osu.edu/ ekici/. Accessed July 24, 2007.

[42] J. Konrad 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.

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

[44] S. Kumar 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.

[45] L. Savidge, Huang Lee, H. Aghajan, and A. Goldsmith. Qos-based geographic routing for event-driven image sensor networks. *Broadband Networks, 2005 2nd International Conference on*, pages 991–1000 Vol. 2, 3-7 Oct. 2005.

[46] Chenyang Lu, Brian M. Blum, Tarek F. Abdelzaher, John A. Stankovic, and Tian He. Rap: A real-time communication architecture for large-scale wireless sensor networks. Technical report, Charlottesville, VA, USA, 2002.

[47] A spatiotemporal communication protocol for wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 16(10):995–1006, 2005. Member-Tian He and Fellow-John A. Stankovic and Member-Chenyang Lu and Member-Tarek F. Abdelzaher.

[48] Mmspeed: Multipath multi-speed protocol for qos guarantee of reliability and timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 5(6):738–754, 2006. Student Member-Emad Felemban and Member-Chang-Gun Lee and Member-Eylem Ekici.

[49] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, and Robert Morris. Performance of multihop wireless networks: shortest path is not enough. *SIGCOMM Comput. Commun. Rev.*, 33(1):83–88, 2003.

[50] Charles E. Perkins and Pravin 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*, pages 234–244, New York, NY, USA, 1994. ACM.

[51] Richard Draves, Jitendra Padhye, and Brian 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*, pages 133–144, New York, NY, USA, 2004. ACM.

[52] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Wirel. Netw.*, 11(4):419–434, 2005.

[53] Richard Draves, Jitendra Padhye, and Brian 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*, pages 114–128, New York, NY, USA, 2004. ACM.

[54] 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.

[55] C. Chandler J. Tang, G. Xue. Interference-aware routing in multihop wireless networks using directional antennas. In *Proceedings of IEEE INFOCOM*, 2005.

[56] Jian Tang, Guoliang Xue, and Christopher Chandler. Interference-aware routing and bandwidth allocation for qos provisioning in multihop wireless networks: Research articles. *Wirel. Commun. Mob. Comput.*, 5(8):933–943, 2005.

[57] Csaba D. Tth Yunhong Zhou Chiranjeeb Buragohain, Subhash Suri. Improved throughput bounds for interference-aware routing in wireless networks. In *Proc. 13th Computing and Combinatorics Conference*, 2007.

[58] G. Hiertz B. Walke E. Weiss, O. Klein. Capacity and interference aware ad hoc routing in multi-hop networks. In *Proceedings of 12th European Wireless conference*, 2006.

[59] Fengguang An Liran Ma, Qian Zhang and Xiuzhen 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.

[60] H. Mahmood and C. Cornaniciu. Interference aware routing for cdma wireless ad hoc networks. *Military Communications Conference, 2005. MILCOM 2005. IEEE*, pages 1059–1063 Vol. 2, 17-20 Oct. 2005.

[61] Tamer ElBatt and Timothy 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*, pages 153–158, New York, NY, USA, 2006. ACM.

[62] Dang-Quan Nguyen and Pascale 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*, pages 428–435, Washington, DC, USA, 2005. IEEE Computer Society.

[63] Ahmed Bader and Eylem 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*, pages 274–285, New York, NY, USA, 2006. ACM.

[64] 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.

[65] P. Gupta and P.R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, Mar 2000.

[66] A.El. 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*, 1:–475, 7-11 March 2004.

[67] Shunan Lin, Shiwen Mao, Yao Wang, and S. Panwar. A reference picture selection scheme for video transmission over ad-hoc networks using multiple paths. *Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on*, pages 96–99, 22-25 Aug. 2001.

[68] S. Panwar S. S. Wang Y. Mao, S. Lin. Reliable transmission of video over ad hoc networks using automatic repeat request and multipath transport. In *EEE VEHICULAR TECHNOLOGY CONFERENCE 2001, CONF 54; VOL 2, pages 615-619*, 2001.

[69] Rita Cucchiara. Multimedia surveillance systems. In *VSSN '05: Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, pages 3–10, New York, NY, USA, 2005. ACM.

[70] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA, 2002. ACM.

[71] Jinyang Li, Charles Blake, Douglas S.J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 61–69, New York, NY, USA, 2001. ACM.

[72] J. J. Lemmon. Wireless link statistical bit error model. Technical report, NTIA Report 02-394, 2002.

[73] Shuang Li, Alvin Lim, Santosh Kulkarni, and Cong Liu. Edge: A routing algorithm for maximizing throughput and minimizing delay in wireless sensor networks. *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–7, 29-31 Oct. 2007.

[74] Shiwen Mao, D. Bushmitch, S. Narayanan, and S.S. Panwar. Mrtp: a multiflow real-time transport protocol for ad hoc networks. *Multimedia, IEEE Transactions on*, 8(2):356–369, April 2006.

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

[76] http://aima.cs.berkeley.edu/newchap05.pdf.

[77] P. Norvig S. Russell. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[78] Gang Feng, K. Makki, N. Pissinou, and C. Douligeris. An efficient approximate algorithm for delay-cost-constrained qos routing. *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on*, pages 395–400, 2001.

[79] D.B. West. *Introduction to Graph Theory*. Prentice-Hall, 1996.

[80] Z. Ye, S.V. Krishnamurthy, and S.K. Tripathi. A framework for reliable routing in mobile ad hoc networks. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 1:270–280 vol.1, 30 March-3 April 2003.

[81] Jun Sun, Wen Gao, and Qingming 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*, pages 373–376, Washington, DC, USA, 2004. IEEE Computer Society.

[82] M.T. Ko R.I Chang, M. Chen and J.M. Ho. Optimizations of stored vbr video transmission on cbr channel. In *SPIE VVDC*, 1997.

[83] 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*, 5:2357–2361 vol.5, 15-18 Sept. 2002.

[84] D. Ferrari. Client requirements for real-time coomunication services. *IEEE Communications*, 1990.

[85] Zheng Wang and Jon Crowcroft. Analysis of burstiness and jitter in real-time communications. In *SIGCOMM '93: Conference proceedings on Communications architectures, protocols and applications*, pages 13–19, New York, NY, USA, 1993. ACM.

[86] Seungjoon Lee, Bobby Bhattacharjee, and Suman Banerjee. Efficient geographic routing in multihop wireless networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 230–241, New York, NY, USA, 2005. ACM.

[87] http://trace.eas.asu.edu/.

[88] Yogesh Sankarasubramaniam, Özgür B. Akan, and Ian F. Akyildiz. Esrt: event-to-sink reliable transport in wireless sensor networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 177–188, New York, NY, USA, 2003. ACM.

[89] 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*, 2:790–794 vol.2, 14-16 Nov 1995.