EFFICIENT MULTI-PATH PROTOCOL FOR WIRELESS SENSOR NETWORKS

Shuang Li[1], Raghu Kisore Neelisetti[2], Cong Liu[3], and Alvin Lim[2]

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

ABSTRACT

Wireless sensor networks are useful for streaming multimedia in infrastructure-free and hazardous environments. However, these networks are quite different from their wired counterpart and are composed of nodes with constrained bandwidth and energy. Multiple-path transmission is one of the methods for ensuring QoS routing in both wired and wireless environment. Directed diffusion, a well known wireless sensor network protocol, only routes packets through a single path, which barely meets the throughput requirement of multimedia data. Instead, we propose a multipath algorithm based on directed diffusion that reinforces multiple routes with high link quality and low latency. This algorithm retains the merits of the original directed diffusion algorithms, including its energy efficiency and scalability. A hybrid metric of link quality and latency is used as the criterion for path selection. In order to select disjoint paths, we propose a scheme for reinforced nodes to respond negatively to multiple reinforcement messages. We use the NS-2 simulation tool with video trace generated by Multiple Description Coding (MDC) to evaluate the performance. The results show that our algorithm gives better throughput and delay performance, i.e higher video quality, than standard directed diffusion that transmits over a single path, with low overheads and energy consumption.

KEYWORDS

Wireless Multimedia Sensor networks, QoS, real-time, throughput, delay.

1. INTRODUCTION

Multimedia surveillance, storage of potentially relevant activities from networked cameras, city traffic monitoring and vehicular collision avoidance are recent multimedia streaming applications over sensor networks. These applications have stringent QoS requirements of throughput and delay. For example, the data rate of H.264 varies between 64 kbps and 240 Mbps depending on the level [1]. However, wireless sensor networks (WSNs) have restrictions in supporting these video/audio streaming applications because of the lack of raw bandwidth, poor link characteristics and limited power supply. Recent advances of multimedia source coding techniques such as Multiple Description Coding and inexpensive hardware, such as CMOS cameras and microphones, have made multimedia transmission over WSNs possible. In many applications, not all video data need to be transmitted to the end-users. For example, in a surveillance sensor network, cluster heads receive the raw video data from member nodes and determine whether to forward the clip to the end-user depending on whether it contains suspicious persons or behavior, as analyzed by some image object recognition techniques. At the sink node, the end-user will make the final decision of any threat by more thorough analysis of the suspicious video clips.

Real-time multimedia data have strict QoS requirements such as bandwidth, delay, jitter and loss ratio. Meeting these QoS requirements requires more efficient sensor network routing protocols. 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 [2], [3]. For ad hoc networks, DSR and AODV are modified to support multiple paths [4][5][6] by sending back multiple REPLYs from the destination. Disjoint paths are preferred [4][5][6] 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 [7], 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 [8]. 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 resilient to errors in media streams because 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.

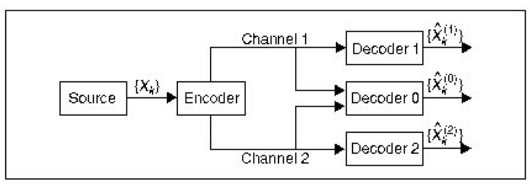

Figure 1. Scenario for MD source coding with two channels and three receivers. [9]

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 [10], 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 [11] 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.

Our results show that our new algorithm can find routes in which both throughput and video quality at the sink are much higher than our single path counterpart EDGE [12] and standard directed diffusion with retransmission. It also has fewer frames that miss the decoding deadline.

2. RELATED WORK

The network layer of WMSN needs to address QoS issues of multimedia streams. Papers such as RAP [13], SPEED [14] and its extension MMSPEED [15] are concerned with real time streaming issues. They prioritize packets based on their delivery speed, which is computed from geographic information and time interval. Prioritization is computed either at the source, hop-by-hop or every few hops. MMSPEED on the other hand selects routes based on reliability measured in terms of packet delivery ratio.

Multipath routing has been extensively researched in wired networks. One of the earliest papers [6] 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) [5] 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) [4] 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. Lin et. al. [16] proposed three variants of multiple path, c-GEDIR, c-DIR and c-MFR methods, i.e. original c-path method, alternate c-path method and disjoint c-path method, of which the last provides high success rate, and small hop counts for small values of c. Our protocol is similar to their last method where intermediate nodes will forward the message to its best neighbor among those who never received the message. However, there are two significant differences. First, we use a negative response scheme to reduce local communication. Second, we use end-to-end delay and link quality as the performance metrics for selecting disjoint paths with minimal interference and latency.

In a framework for supporting multipath video transport over ad hoc networks, multistream video coding is one of the essential components. Mao et. al. [8] 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) [17], layered coding (LC) with selective ARQ [18] and multiple description motion compensation (MDMC) [10].

Interference-aware protocols have recently been explored in multi-hop wireless networks because wireless interference causes packets to be dropped. [19] presents routing algorithms to compute interference-optimal paths with bounded cost. It uses an optimal bandwidth allocation algorithm to allocate timeslots. [19] and [20] analyze the throughput bounds and capacity for interference-aware routing in wireless networks respectively. Nguyen et. al. [21] considers radio interference and modifies the OLSR routing protocol for bandwidth reservation and interferences. Our approach is to estimate ETX, the link layer indicator, from SINR (Signal-to-Interference-and-Noise) and select multiple paths that can both maximize throughput and meet the playout deadline.

Routing metrics are designed to achieve different goals. Existing wireless ad hoc routing protocols typically select routes using minimum hop count, e.g. DSR [22] and DSDV [23]. Directed diffusion [7] uses a publish/subscribe communication model whereby a sink node requests data by sending interests for a named data. As the interest is flooded through the network, each intermediate node establishes a gradient with its neighbors and enables data that match the interest to be drawn towards the sink. Sensor nodes with data that matches the interest will forward an exploratory data that is propagated by intermediate nodes through established gradients to the sink. The sink sends a reinforcement message to the node that first forwarded the new data to it. Intermediate nodes use the same rule to reinforce their upstream neighbor. After the reinforcement stage, the source node continues to send data through the reinforced path.

Recently, many new link quality metrics have been proposed [24] such as Round Trip Time (RTT), Per-hop Packet Pair Delay (PktPair), and Expected Transmission Count (ETX). Li et. al. [12] proposed a hybrid metric which takes into account throughput, delay and interference.

3. PROBLEM STATEMENT

Designing routing protocols in WMSNs is a big challenge because of the stringent QoS requirement of throughput and delay. H.264 has variable data rates between 64 kbps and 240 Mbps for different levels [25]. 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 IEEE 802.11b 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. To be more specific,we are trying to find multiple disjoint paths with minimum *PATH_COST*, which is a path metric defined in the following paragraphs. Paths with close proximity interfere with each other. Since links suffering severely from interference have poor SNRs, they have less probability to be selected in the paths. 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.

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

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) [26]. Lee et. al. [27] derived the mathematical Equation 2 to calculate BER.

$$BER = 0.5 \times erfc(\sqrt{\frac{P_r \times W}{N \times f}}) \quad (2)$$

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:

$$SNR = 10\log\frac{P_r}{N} \tag{3}$$

In order to consider interference, we change Equation 3 to

$$SNR = 10\log\frac{P_r}{N+I} \tag{4}$$

I is the interference component. A method for calculating I is given in [28]. Given the packet size, packet loss rate could be calculated from BER. According to $ETX = \frac{1}{d_f \times d_r}$ [29], 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 PATH COST: In order to maximize throughput and minimize delay with interference consideration, we use the path metric *PATH_COST* proposed in [12]. α and β are non-negative integers.

$$PATH\_COST = PATH\_ETX^{\alpha} \times PATH\_DELAY^{\beta} \tag{5}$$

$$PATH\_ETX = \mathop{Max}_{i=0}^{N-3}\left(\sum_{j=i}^{i+2} ETX_j\right) \tag{6}$$

where N is the number of hops in the path and ETX_j is the ETX value of the jth hop.

4. ALGORITHM DESIGN

In this section, we modify directed diffusion by 1) using *PATH_COST* as the metric instead of pure delay, and 2) selecting 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 dropped data. We discard paths which fail to meet the playout deadline in video streaming. Throughput is influenced by ETX, which is estimated by the cumulative SNR. The use of historical SNRs offers more stable and accurate estimation of ETX.

When the exploratory data are being flooded through intermediate nodes based on the established gradients, each intermediate node will compute the *PATH_COST* and stores it in its local table. Since exploratory data are being transmitted simultaneously at multiple paths during this phase and the *PATH_COST* metric computes the link quality which takes into account the effects of interference from multi-path transmission, the *PATH_COST* metric gives a better estimate of the actual link quality when data is transmitted through the selected disjoint paths. Other multiple path schemes [16] do not take into account interferences from simultaneous transmission through multiple paths. Based on the *PATH_COST* metric stored locally at each intermediate node, the path selection phase will select multiple paths with the lowest delay and interference (highest link quality). Since each intermediate node can determine which of its upstream neighbor has the lowest *PATH_COST* (up to that node), it just need one message to select (reinforce) the best upstream node in each of the multiple paths (unless it has been previously selected, then another message is needed and so forth).

During the exploratory data phase, further improvements on the estimate of the interference from transmissions at multiple paths are possible by using different variants of flooding schemes that will better estimate the actual interference by limiting the transmission to only paths that are similar to the multiple paths that are eventually selected. These techniques are discussed in our future companion paper.

4.1. Route Discovery

4.1.1. 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 [12]. *PATH_ ETX* is computed at the same time according to Equation 6. *PATH_COST* of each subpath is kept at the intermediate nodes' local table in ascending order. The format of the local table is shown in Table 1. Only the one with the lowest *PATH_COST* 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 *i*th 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 *PATH_COST* by giving more weight to β (for delay) and piggybacks the new value in new *INTEREST* messages.

Table 1. Local table format. Local timestamp is used for local time synchronization [15].

Last hop	ETX	Local timestamp	$Cost_p$	Cumulative SNR

4.1.2. Mutiple path selection

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 *PATH_ COST* 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 selected if disjoint nodes cannot be found or the delay exceeds the playout deadline. If two nodes try to select a link whose upstream nodes will converge, the first one to reinforce would win. In Figure 2, *A* selects *C* first. Then, *B* tries to select *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 is a heuristic method, which guarantees that the paths are always disjoint and loop-free if we do find them.

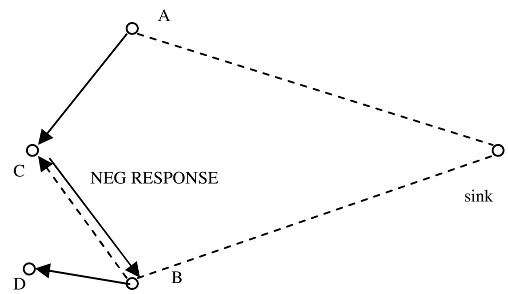

Figure 2. Avoiding a node that is already reinforced in another path.

In Figure 3, after *A* has already selected *B* then *C*, *D* and *E* are reinforced sequentially. When *E* tries to select *B*, *B* drops the packet and sends *NEG RESPONSE* to *E*. So *E* is able to select the next candidate *F* to select. Standard diffusion only selects one route and when there is a loop, the reinforcement packet is simply dropped and no packet is received at the source. Our algorithm guarantees that reinforcement packets successfully reach the source provided that the delay constraint can be satisfied.

When a node discards an 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 2, *B* reads the local timestamps of C and D (t_c and t_D respectively) from its local table and if $t_D - t_C <= \frac{\varepsilon}{n}$ (ε 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.

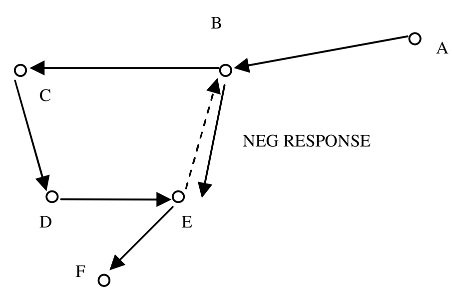

Figure 3. How to avoid loops.

4.1.3. Probabilistic traffic splitting

Multiple Description Coding (MDC) has been 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 *PATH_COST* values are $C_1 < C_2 < \ldots < C_{2\lambda}$. Then, path 1 and path 2λ are allocated to transmit the first description, path 2 and path $2\lambda-1$ the second, and so on. 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.

4.2. Parameter Settings

4.2.1. α, β adjustment

The problem of finding the suitable values for α and β can be converted into a Constraint Satisfaction Problem (CSP) [30].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. Since we use ETX to estimate throughput, the latter is limited by ETX that 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 PATH ETX given the required bandwidth requirement.

We use a min-conflicts algorithm [25] to solve the CSP problem in this paper. A similar method is also use in [31] to solve the problem of finding a path with two additive constraints called multi-constrained path (MCP) problem. Since *PATH_ETX* 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 delay and ETX, i.e. *PATH_COST*, and then find the corresponding shortest path. In Table 2, *D* and *E* are delay and ETX_p upper bounds respectively. d and e are the delay and ETXp 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 Dijkstra's algorithm. The maximum probability that exists in a feasible path is less than ½ if algorithm MCP-DE cannot find a solution [31].

Table 2: Algorithm MCP-DE. $Cost_p$ formula is simplified to $d \cdot e\ \alpha$ (α is not necessarily an integer).

Algorithm MCP-DE (D, E, d, e)

```
q ← path with lowest ETX_p
if (e(q) > E) then
    return NULL
else if (d(q) ≤ D) then
    return q
p ← Dijk(d)
if (d(p) > D) then
    return NULL
else if (e(p) ≤ E) then
    return p
while TRUE do
    α ← [lg(d(q)) – lg(d(p))]/[lg(e(p)) – lg(e(q))]
    r ← path with lowest d · e^α
    if (e(r) = e(q) or e(r) = e(p) then
        return NULL
    else if (e(r) > E) then
        p ← r
    else if (d(r) > D) then
        q ← r
    else
        return r
```

4.2.2. 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 4, 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 [32]. 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 [28]. The time to compute the min-cut value of a given graph G is $O(n^4 \log n)$. 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.

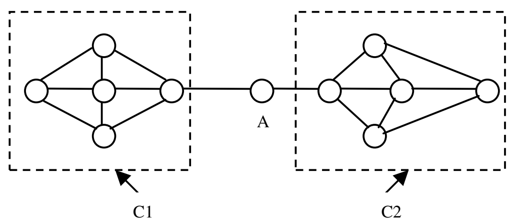

Figure 4. Unavoidable bottleneck nodes. Nodes connected by a line are able to communicate with each other. Otherwise, they are out of range.

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, 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 more conservative lower bound for node density.

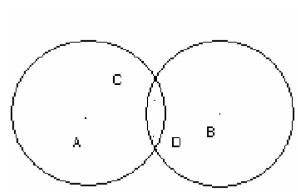

Figure 5. Overlapping of A's and B's transmission ranges

We assume that node deployment follows a Poisson distribution. *x* is the number of nodes in the overlapping area mentioned above. Given the average density λ and overlapping area *A*, x = λA.

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

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.

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

If nodes are evenly distributed and we only consider the scenario in Figure 5, that is, $\frac{E}{\sqrt{N}} > R > \frac{1}{2} \cdot \frac{E}{\sqrt{N}}$. So,

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

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

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

Given E=1200, N=100, R=100, $P(x>1)$ is 0.2492. N-P relations are shown in Figure 6. 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.

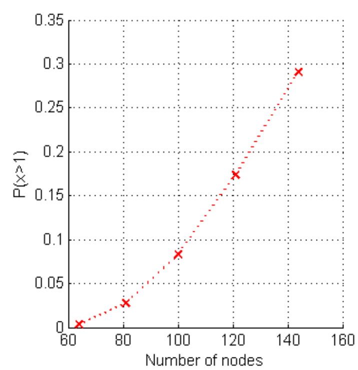

Figure 6. The relation of *P(x>1)* and *N*.}

5. SIMULATION RESULTS

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 [12] and basic directed diffusion. We evaluate the performance on a rectangular network with two ray ground and Wu's error model [11] combined with other mathematical formulation [27]. We use the multiple description coded foreman sequence from Video Traces Research Group at Arizona State University [33].

5.1. Simulation methodology

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 neighbours (Figure 7). 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.

We use the IEEE 802.11b protocol for the MAC layer. The channel has a bandwidth of 2 Mbps. The transmission range is 250m and the interference range is 550m. The distance between each pair of nodes is 200m, a little smaller than the transmission range. For a certain node, only the closest neighbours can receive packets from it. The maximum number of link layer retransmissions is seven, after which the packet is dropped.

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 may 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 by using Equation 2. We changed the constant 0.5 there to $5*10^{-7}$ in order to show high enough throughput.

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 *PATH_ETX* 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 3 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 avoid 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 two better paths (if they exist) when the network has unstable links with variable link quality.

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

5.2. Performance comparison

We compare our protocol, DCHT, with DCHT-sumETX, EDGE, c-MFR and basic diffusion. In DCHT-sumETX, we replace the *PATH_ETX* part in *PATH_COST* metric with the sum of ETXs in a path. EDGE [12] is a single-path protocol which adds *PATH_COST* metric into directed diffusion (basic diffusion).

The different protocols are compared over different network sizes of 46, 77, 116 and 163 nodes configured in the rectangular area as shown in Figure 7. The distance of every neighbor pair is always 200m. The size of packets is 128 bytes, greater than half of the smallest video frame with 187 bytes. We use a deadline of 200ms because the playout deadline varies from 50ms to 200ms [3]. The deadline for the exploratory packets is 2000ms. To be fair, EDGE uses similar ETX estimation from SNR.

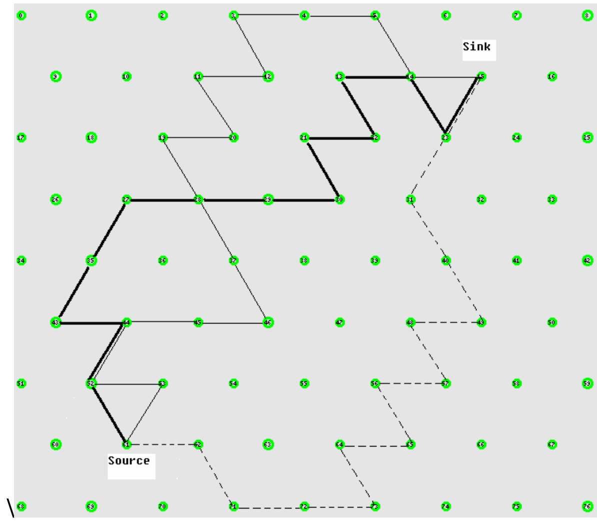

Figure 7. Topology with equilateral triangular tessellations. There are 77 nodes in this graph.

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 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 transmitted per second that meet the deadline. We use the same seed to show the number of loss packets per frame for the two streams in the network of 77 nodes (Figure 7) with both our protocol and EDGE. In the formula, $PATH\_COST = PATH\_ETX^{\alpha} \times PATH\_DELAY^{\beta}$, 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 1. The total number of application packets sent at the source, excluding dummy packets, is 2334.

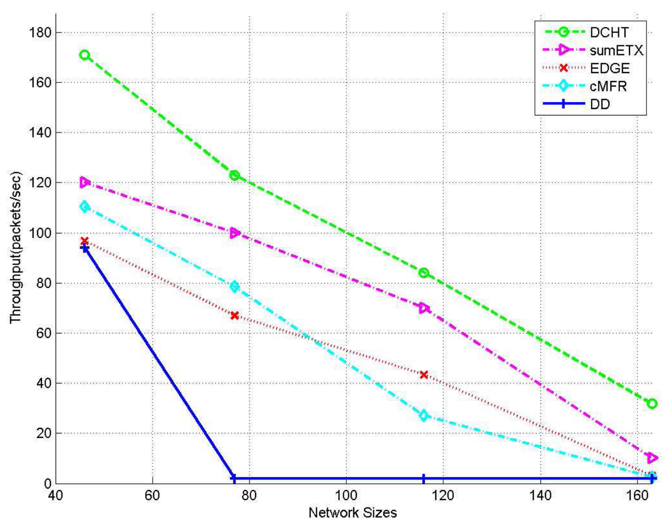

Figure 8. Throughput of DCHT, DCHT-sumETX, EDGE and basic diffusion with different network sizes.

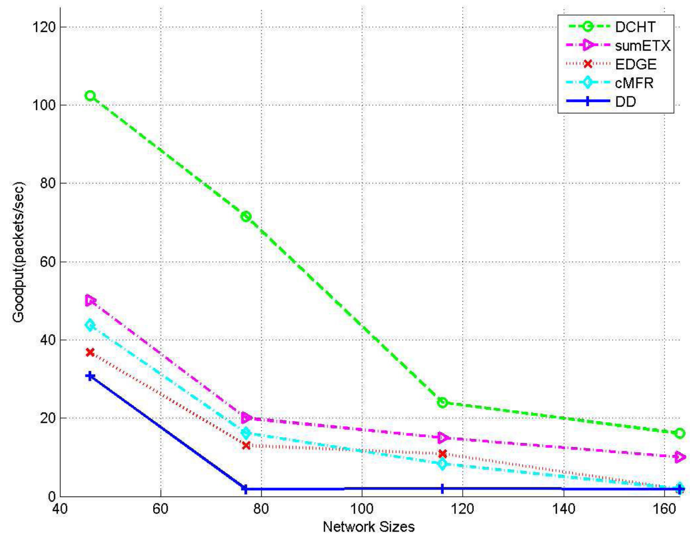

Figure 9. Goodput of DCHT, DCHT-sumETX, EDGE and basic diffusion with different network sizes.

In Figure 8, we show that our protocol gives twice as much throughput as EDGE. The performance of DCHT-sumETX is between that of two-path DCHT and that of single-path EDGE. c-MFR has similar performance compared to EDGE. However, it can be seen that the performance of c-MFR protocol decreases gradually as the network size increases. This is because as the network size increases, the average path length increases and this leads to increase in packet losses. Basic diffusion can hardly receive any packet when network size increases to 77 and higher 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, unlike 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. Still, EDGE performs better than basic diffusion because EDGE takes interference into account. 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 9, both EDGE and basic diffusion perform much worse than our protocol because they do not take into account the playout deadline. Although DCHT-sumETX considers deadline as DCHT does, it selects routes with a different metric, part of which is the sum of all ETXs on a path. DCHT-sumETX is over-estimating ETX of a path, which gives preference to short paths. c-MFR also performs similarly to EDGE but degrades more as the network size increases for the same reason as described above. 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.

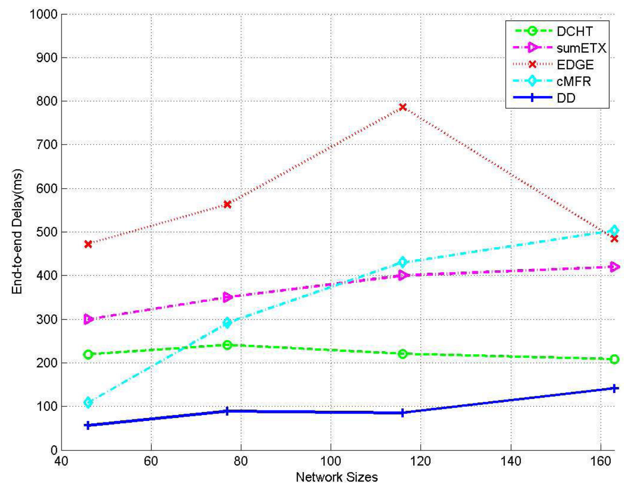

Figure 10. End-to-end delay of DCHT, DCHT-sumETX, EDGE and basic diffusion with different network sizes.

The average end-to-end delay in Figure 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. The end-to-end delay of DCHT stabilizes a little above 200ms, the playout deadline. EDGE rises quickly in delay and drops after size 77 when a large number of packets are dropped. We also evaluated the end-to-end delay of c-MFR technique. Its delay increases almost linearly as the network size increases. This indicates that c-MFR technique is not a scalable solution. Also, as the network size increases it has to evaluate much larger number of neighbors before it can determine an ideal path between source and destination.

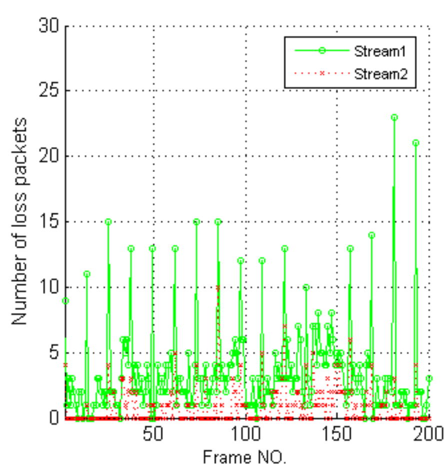

Figure 11. Number of loss packets per frame in DCHT. There are 29 packets in a large video frame.

Another metric important for MDC video streaming is the 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 11 and 12 with the network size of 77 nodes (Figure 7) 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 packet 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 11 and 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 7 and generated the profile in Figure 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 dashed path is relatively closer to the border, which has less interference. The single path reinforced in EDGE, which matches Figure 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.

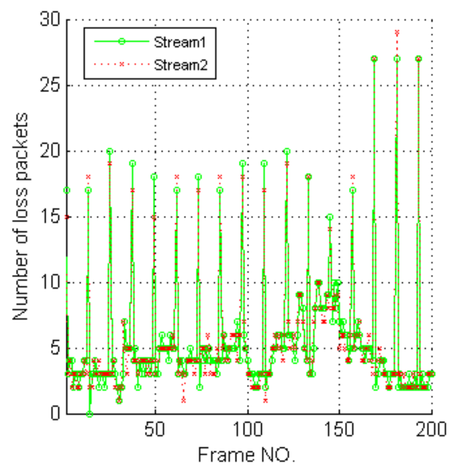

Figure 12. Number of loss packets per frame in EDGE. There are 29 packets in a large video frame

The number of paths should match the number of descriptions of the MDC encoder. We also send three and four descriptions over three and four paths respectively and compare the throughput and goodput performance in Figure 13. Due to the low density of the network, it is difficult to find three or even four 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 ρ+1, ρ+2… σ are multiplexed with Streams 1, 2… σ-ρ 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 significantly when the number of paths (descriptions) increases. Goodput from the 2-path case to the 3-path case does not increase much because in some cycles only 1 or 2 paths can be found. 4-path case still gets high goodput because 4 paths can be found with a high probability according to our simulation results. In Figure 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 13 and Figure 14. The end-to-end delay (Figure 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 increases overhead.

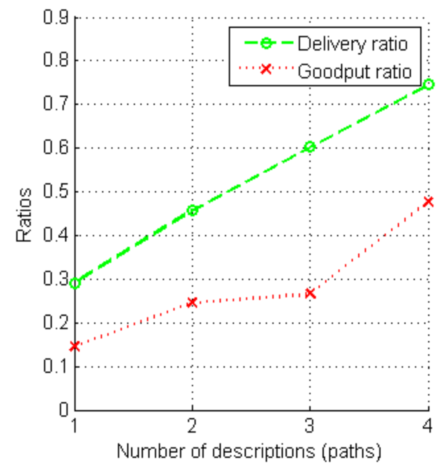

Figure 13. Delivery ratio and goodput ratio of DCHT with different numbers of descriptions/paths. The network size is 77 nodes

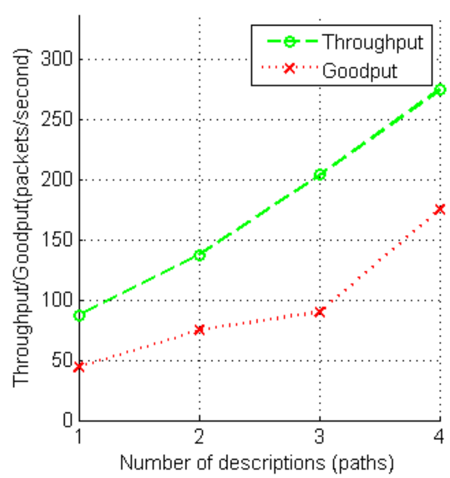

Figure 14. Throughput and goodput of DCHT with different numbers of descriptions/paths. The network size is 77 nodes

We also compare the energy consumption and routing overhead in our protocol with basic diffusion and EDGE. Intuitively, our protocol may consume more energy since we are using more resources of the networks. In Figure 16, it is shown that our protocol does not consume additional energy, compared with basic diffusion and EDGE – two single-path protocols.

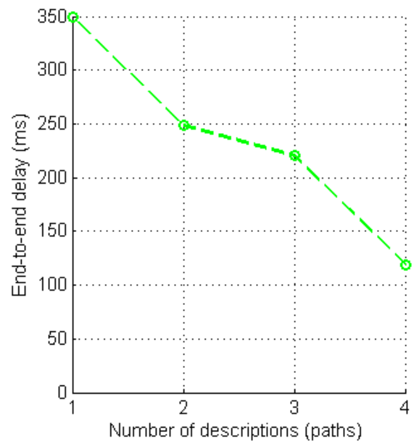

Figure 15. End-to-end delay of DCHT with different numbers of descriptions/paths. The network size is 77 nodes

On the other hand c-MFR takes more energy as it has contact each of its neighbor to determine the ideal source destination path and this step consumes lot of energy and hence consumes slightly more energy when compared to our proposed techniques. Its energy consumption increases with increase in network size. 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 17), which is better than linear increase. More energy consumption leads to the discovery of more paths, which transmit video data more efficiently.

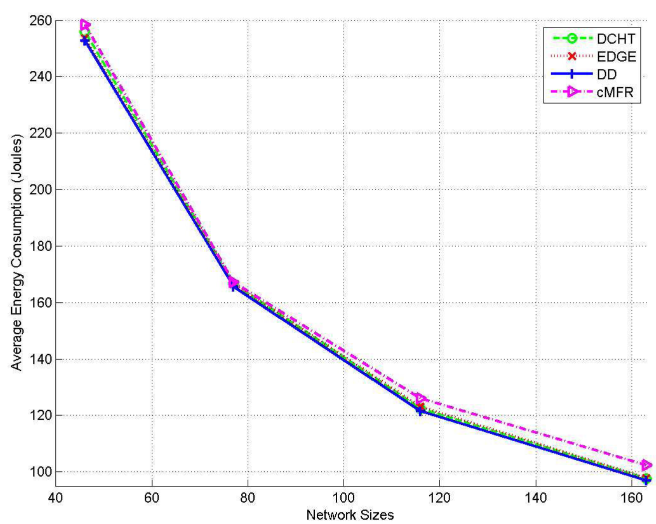

Figure 16. Average energy consumption of a single node in the network of DCHT with three paths (descriptions), EDGE and basic diffusion with different network sizes.

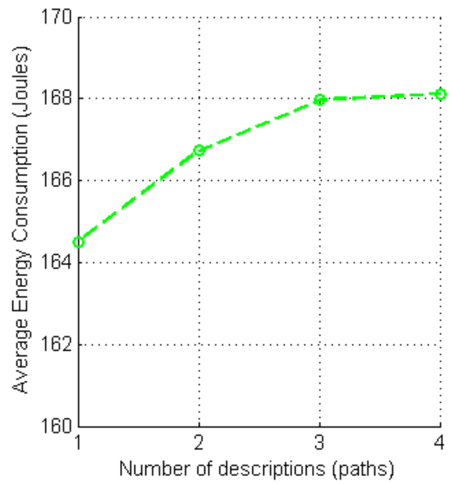

Figure 17. Average energy consumption of a single node in the network of DCHT with different number of paths (descriptions). The network size is 77 nodes.

The only extra overhead in our protocol is the use of *NEG RESPONSE* in order to find multiple disjoint paths (Figure 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 four paths, considering that each node only has six neighbours in our topology. When we have larger network sizes, we have higher probability to get multiple paths without sharing nodes with other paths.

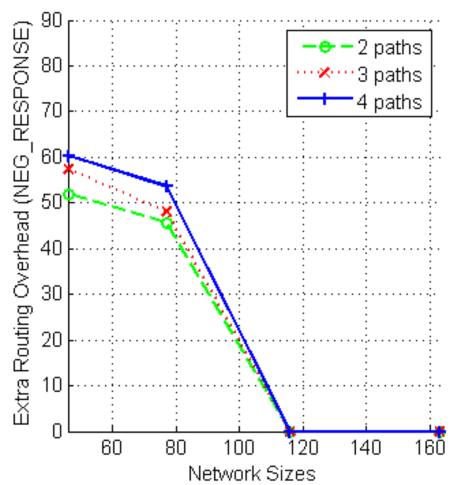

Figure 18. Extra routing overhead (we only consider *NEG RESPONSE*) of the whole network of DCHT with different number of paths (descriptions) in different network sizes.

5.3. Discussion

The goodput of our protocol with four 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 of 4 paths is 23 kbps, which barely meets the data rate of low quality video of 28 kbps. By sending video through more paths, such as more than four paths, the goodput could be increased to meet even 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, the deadline we set.

6. CONCLUSION

In this paper, we propose a protocol for multipath video streaming over wireless sensor networks, and compare its performance with EDGE and basic diffusion. Multiple disjoint paths can achieve high throughput and desirable delay and meet the QoS requirement of multimedia streaming. The use of Wu's 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. Also, our protocol does not consume much energy or routing overhead. If more paths are found, both the throughput and delay performance can be improved even further.

In addition to wireless loss, congestions control is another important issue. Our future work will focus on developing wireless multi-media protocols that will perform well in networks with congestion. We will also experiment with random topologies and a better cumulative SNR estimation formula that will increase throughput further.

REFERENCES

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

[2] A. Begen, Y. Altunbasak, and O. Ergun, Fast heuristics for multi-path selection for multiple description encoded video streaming, *ICME '03. International Conference on Multimedia and Expo*, vol. 1, pp. I-517-520, 6-9 July 2003.

[3] S. Mao, S. Panwar, and Y. 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, vol. 4, pp. 2325-2336, 13-17 March 2005.

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

[5] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. 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*. Piscataway, NJ, USA: IEEE Press, 2000, pp. 3-10.

[6] D. Sidhu, R. Nair, and S. Abdallah, Finding disjoint paths in networks, in *SIGCOMM '91: Proceedings of the conference on Communications architecture & protocols*. New York, NY, USA: ACM, 1991, pp. 43-51.

[7] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, Directed diffusion for wireless sensor networking, *IEEE/ACM Transactions on Networks*, vol. 11, no. 1, pp. 2-16, 2003.

[8] S. Mao, S. Lin, S. Panwar, Y. Wang, and E. Celebi, Video transport over ad hoc networks: multistream coding with multipath transport*, Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 10, pp. 1721-1737, Dec. 2003.

[9] V. Goyal, Multiple description coding: compression meets the network, *Signal Processing Magazine, IEEE*, vol. 18, no. 5, pp. 74-93, Sep 2001.

[10] Y. Wang and S. Lin, Error resilient video coding using multiple description motion compensation, *IEEE Fourth Workshop on Multimedia Signal Processing,*, pp. 441 - 446, 2001.

[11] W. Xiuchao, Simulate 802.11b channel within ns2, SOC, NUS, Tech. Rep.

[12] S. Li, A. Lim, S. Kulkarni, and C. Liu, Edge: A routing algorithm for maximizing throughput and minimizing delay in wireless sensor networks, *Military Communications Conference, MILCOM 2007, IEEE*, pp. 1-7, 29-31 Oct. 2007.

[13] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, Rap: A real-time communication architecture for large-scale wireless sensor networks, Charlottesville, VA, USA, Tech. Rep., 2002.

[14] Tian He, John A. Stankovic, Chenyang Lu and Tarek F. Abdelzaher A spatiotemporal communication protocol for wireless sensor networks,. *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 10, pp. 995-1006, 2005.

[15] Emad Felemban, Chang-Gun Lee and Eylem Ekici, Mmspeed: Multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks, *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738-754, 2006.

[16] X. Lin and I. Stojmenovic, Location-based localized alternate, disjoint and multi-path routing algorithms for wireless networks, *J. Parallel Distrib. Comput.*, 2002.

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

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

[19] Liran Ma, Qian Zhang, Fengguang An, 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)*, LNCS, 2005.

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

[21] D.-Q. Nguyen and P. Minet, Interference-aware QoS OLSR for mobile ad-hoc network routing, in *SNPD-SAWN '05: Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/ Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 428-435.

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

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

[24] R. Draves, J. Padhye, and B. Zill, Comparison of routing metrics for static multi-hop wireless networks, in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2004, pp. 133 - 144.

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

[26] J. J. Lemmon, Wireless link statistical bit error model, NTIA Report 02-394, Tech. Rep., 2002.

[27] S. Lee, B. Bhattacharjee, and S. 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*. New York, NY, USA: ACM, 2005, pp. 230-241.

[28] Z. Ye, S. Krishnamurthy and S. 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, vol. 1, pp. 270-280, 30 March-3 April 2003.

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

[30] Constraint Satisfaction Problems. http://aima.cs.berkeley.edu/newchap05.pdf.

[31] G. Feng, K. Makki, N. Pissinou, and C. Douligeris, An efficient approximate algorithm for delay-cost-constrained QoS routing, *Tenth International Conference on Computer Communications and Networks*, pp. 395 - 400, 2001.

[32] D. West, Introduction to Graph Theory. Prentice-Hall, 1996.

[33] Video Trace Library. http://trace.eas.asu.edu/

Authors

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

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

Raghu Kisore Neelisetti is doctoral student at Auburn University. His is currently working on various issues related to wireless sensor networks, such as reliability and application of sensor networks to target tracking. His work is presently funded by Vodafone.

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