Delay-Constrained High Throughput Protocol for Multi-Path Transmission over Wireless Multimedia Sensor Networks

Shuang Li, Raghu Neelisetti, Cong Liu and Alvin Lim
Department of Computer Science and Software Engineering, Auburn University
{lishuan, neelira, liucong, limalvi}@auburn.edu

Abstract

Real-time multimedia transport has stringent QoS requirements, such as bandwidth, delay, jitter, and loss ratio. Wireless sensor networks are useful for streaming multimedia data in infrastructure-free and hazardous environments. However, these networks are composed of nodes with constrained bandwidth and energy. In QoS routing for wired networks, multipath routing is widely used. Some existing ad hoc routing algorithms also provide multipath 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 multipath algorithm based on directed diffusion that reinforces multiple routes with high link quality and low latency. 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 than standard directed diffusion.

1. Introduction

There has been a recent emergence of multimedia streaming applications over sensor networks, such as multimedia surveillance, storage of potentially relevant activities from networked cameras, traffic conditions and collision avoidance. 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.

Real-time multimedia data have strict QoS requirements such as bandwidth, delay, jitter and loss ratio. 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 [3], [18]. For ad hoc networks, DSR and AODV are modified to support multiple paths [11][20][21] by sending back multiple REPLYs from the destination. Disjoint paths are preferred since 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 [10], 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, which cannot be guaranteed by a single reinforced path in basic directed diffusion. Thus enhancement to directed diffusion is needed.

Multiple Description Coding is a coding technique which generates multiple equally important descriptions [19]. 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. 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. Besides, MDC matches multipath routing very well.

Standard NS-2 is using primitive propagation models 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 [25] added SNR and BER and modeled interference accurately. Thus other frames received by a receiver simultaneously are also modeled.

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 [14] 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. RAP [16], SPEED [8] and its extension MMSPEED [6] are concerned with real time streaming issues. They prioritize packets based on their delivery speed. MMSPEED on the other hand selects routes based on reliability.

Multipath routing has long been researched in wired networks. One of the earliest papers [21] proposed an extension to the distance-vector algorithm for finding multiple disjoint paths. Alternate path routing (APR) [20] provides load balancing and failure prevention by distributing traffic among a set of diverse paths in mobile ad hoc networks. 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.

In a framework for supporting multipath video transport over ad hoc networks, multistream video coding is one of the essential components. Mao *et. al.* [19] combines multipath transport with multiple description coding in ad hoc networks. Other efficient coding schemes that were proposed are feedback-based reference picture selection [15], layered coding with selective ARQ [18] and multiple description motion compensation [24].

Routing metrics are designed to achieve different goals. Existing wireless ad hoc routing protocols typically select routes using minimum hop count. Directed diffusion [10] selects routes in sensor networks with the least delay. Recently, many new link quality metrics have been proposed [5] such as Round Trip Time (RTT), Per-hop Packet Pair Delay (PktPair), and Expected Transmission Count (ETX). Li *et. al.* [14] 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 [22]. 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.

3.1. Assumptions, goals and definitions

We assume nodes are stationary or have little mobility. 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.

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.

SNR is closely related with BER (Bit error rate) [13]. Lee *et. al.* [12] derived the mathematical formula (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:

$$SNR = 10 \log \frac{P_r}{N} \quad (3)$$

In order to consider interference, we change (3) to

$$SNR = 10 \log \frac{P_r}{N+I} \quad (4)$$

I is interference component. A method for calculating I is given in [25]. Given the packet size, packet loss rate could be calculated from BER. Since all links are symmetric in our assumption, d_f and d_r in [4] are both 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 $Cost_p$ proposed in [14].

$$Cost_p = ETX_p^{\alpha} \times delay_p^{\beta} \tag{5}$$

$$ETX_p = \underset{i=0}{\overset{N-3}{Max}}\left(\sum_{j=i}^{i+2} ETX_j\right) \tag{6}$$

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

3.2. Problem formulation

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. Algorithm 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 because data arriving later than a deadline are simply useless. 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.

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 [14]. ETX_p is computed at the same time according to Formula (6). $Cost_p$ of each subpath is kept at the intermediate node's local table in ascending order. The format of the local table is shown in Figure 1. 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 $\boldsymbol{t_i - t_0 <= DL}$, 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.

Last hop	ETX	Local timestamp	$Cost_p$	Cumulative SNR

Figure 1: Local table format. Local timestamp is used to get local time synchronization [14].

4.1.2. 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 if 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 2, 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. This technique not only guarantees disjoint nodes, but also ensures loop-free path since loops are broken when selecting the next candidate in the local table. Standard diffusion only reinforces 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 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 2, B reads the local timestamps of C and D (t_C and t_D respectively) from its local table and if $t_D - t_C <= \varepsilon / n$ (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.

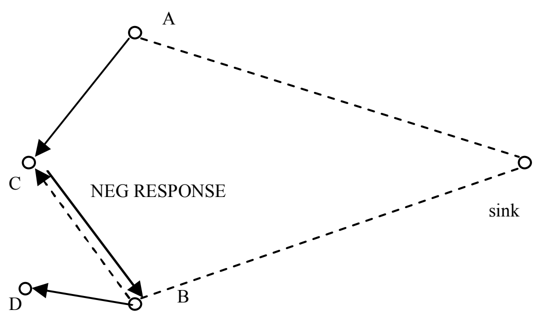

Figure 2: How to avoid the node already reinforced in another path.

4.1.3. Probabilistic traffic splitting

Multiple Description Coding (MDC) is used for multimedia traffic. The coder generates 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 $C_1 < C_2 < \ldots < C_{2\lambda}$. Then, path 1 and path 2λ are allocated to transmit the first description, path 2 and path 2λ-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 $(1/C_{2\lambda})/(1/C_1+1/C_{2\lambda})$ 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) [1]. 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. Besides, we can also estimate the upper bound for ETX_p given the required bandwidth requirement.

We use a min-conflicts algorithm [24] to solve the CSP problem in this paper. A similar method is also use in [7] 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 Figure 3, 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 Dijkstra's algorithm. The maximum probability that exists in a feasible path is less than ½ if algorithm MCP-DE cannot find a solution [7].

```
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
```

Figure 3: Algorithm MCP-DE. $Cost_p$ formula is simplified to $d \cdot e^{\alpha}$ (α is not necessarily an integer).

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. Min-cut algorithm is used to find bottleneck nodes the failure of which will cause the least partition [26]. The time to compute the min-cut value of a given graph *G* is $O(n^4 logn)$. However, when the network size increases, it consumes too much energy. We may deploy more nodes in their neighbourhood to build disjoint paths.

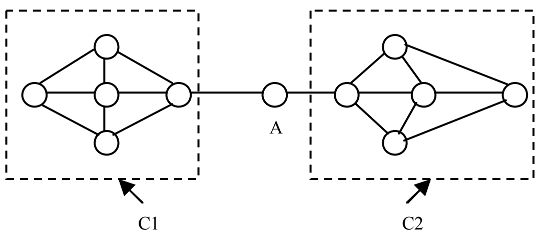

Figure 4: Unavoidable bottleneck nodes. Nodes connected by an edge are able to communicate with each other directly. Otherwise, they are out of range.

In real situations, we can manually control the deployment of sensor nodes. Given the lower bound on node density, the percentage of bottleneck nodes is limited. In Figure 5, 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.

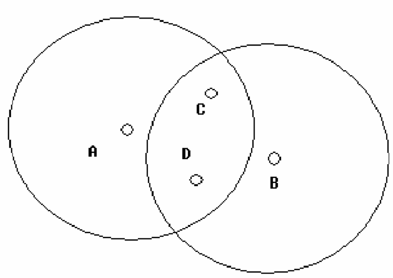

Figure 5: Overlapping of A's and B's transmission ranges.

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

$$P(x=k)=\frac{e^{-\lambda A}\cdot(\lambda A)^k}{k!} \tag{7}$$

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

$$\lambda=\frac{N}{E^2} \tag{8}$$

Nodes are evenly distributed and we only consider the scenario in Figure 5. So,

$$A=2(R^2\arccos\frac{E}{2R(\sqrt{N}-1)}-\frac{E\sqrt{4R^2(\sqrt{N}-1)^2-E^2}}{4(\sqrt{N}-1)^2}) \tag{9}$$

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

$$P(x>1)=1-P(x=0)-P(x=1)=1-e^{-\lambda A}(1+\lambda A) \tag{10}$$

N-P relations are shown in Figure 6. Given *P*(*x*>1), *E* and *R*, we can find *N*.

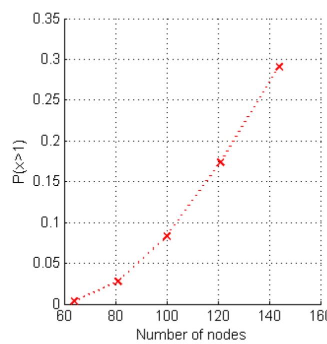

Figure 6: The relation of P(x>1) and N.

5. 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 [14] and basic directed diffusion. We evaluate the performance on a rectangular network with two ray ground and Wu's error model [25] combined with other mathematical formulation [12]. We use the multiple description coded foreman sequence from Video Traces Research Group at Arizona State University [2].

5.1. Simulation methodology

There is a sender and a receiver in the topology. 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 802.11 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. For a certain node, only the closest neighbours can receive packets from it. The maximum number of link layer retransmissions is seven.

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. 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 8s. We transmit 36 dummy packets ahead of each stream to collect relatively accurate cumulative SNR. The traffic interval of the dummy packets is 5s. BER is calculated from SNR by using Formula (2). We changed the constant 0.5 there to 5*10e-7 in order to show high enough throughput and where the BER matches more closely with experimental results [9].

5.2. Performance comparison

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 (Figure 7). The size of packets is 128B, greater than half of the smallest video frame 187B. 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.

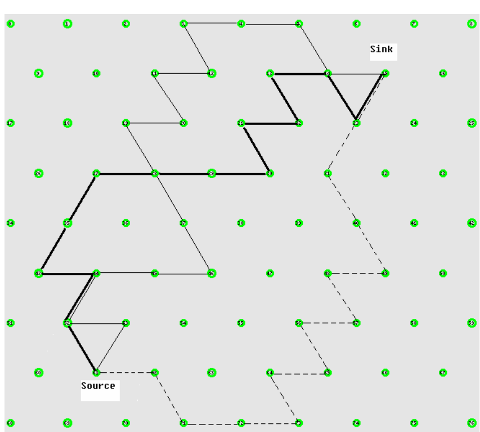

Figure 7: Topology with equilateral triangular tessellations. There are 77 nodes in this graph.

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 that meet the deadline. We use the same seed to show the number of lost packets per frame for the 2 streams in the network of 77 nodes (Figure 9) with both our protocol and EDGE. In the formula, $Cost_p = ETX_p^{\alpha} \times delay_p^{\beta}$, we use α = 10, β =1 since delay has only a slight difference in delay for each link as observed in the experiment. In Formula (1) γ = 0.7. The total number of application packets sent at the source, excluding dummy packets, is 2334.

In Figure 8, we show that our protocol gives twice as much throughput as EDGE. Basic diffusion can hardly receive any packet when network size increases to 77 and above because it does not consider link quality at all. Our protocol 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 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 9, both EDGE and basic diffusion perform much worse than our protocol because they do not take into account the playout deadline. Without collecting link quality statistics 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 ¼ of the packets are received on time.

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 since 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 the network size of 116 when a large number of packets are dropped.

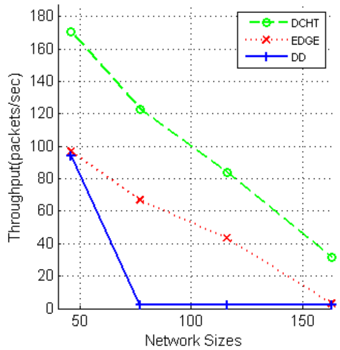

Figure 8: Throughput of DCHT, EDGE and basic diffusion with different network sizes.

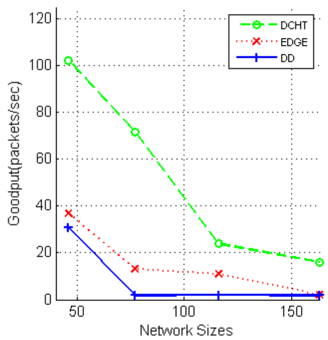

Figure 9: Goodput of DCHT, EDGE and basic diffusion with different network sizes.

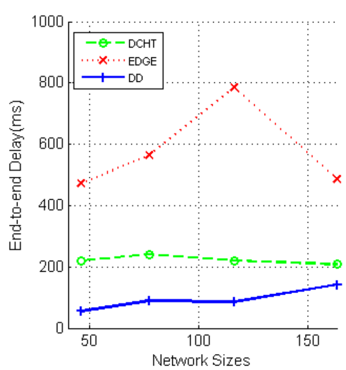

Figure 10: End-to-end delay of DCHT, EDGE and basic diffusion with different network sizes.

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 11 and 12 with the network size of 77 nodes since the throughput of diffusion is very low. We only show the profile with a certain seed. 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 figures is that the video trace we use for the foreman sequence has a large frame (more than 2000 bytes) every 12 frames.

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 solid 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.

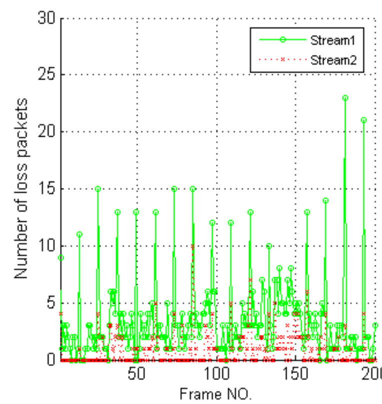

Figure 11: Number of loss packets per frame in DCHT. There are 29 packets in a large video frame.

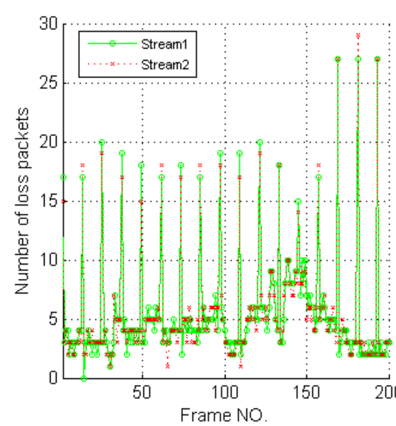

Figure 12: Number of loss packets per frame in EDGE. There are 29 packets in a large video frame.

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 accurately. Our protocol provides 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 ¼ 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.

We plan to test our protocol with more than two streams with MDC in the future. In addition to wireless loss, congestions control is another issue. We will also experiment with random topologies and a better cumulative SNR estimation formula that will increase throughput further.

7. References

[1] http://aima.cs.berkeley.edu/newchap05.pdf
[2] http://trace.eas.asu.edu/
[3] A. Begen, Y. Altunbasak, and O. Ergun, "Fast Heuristics for Multi-path Selection for Multiple Description Encoded Video Streaming," in *IEEE ICME*, July 2003, pp. 517-520.
[4] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris. *A high-throughput path metric for multi-hop wireless routing*. Proc. of MobiCom, 2003.
[5] R. Draves, J. Padhye, and B. Zill, *Comparison of Routing Metrics for Static Multi-Hop Wireless Networks*. Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), pages 133–144, ACM Press, 2004.
[6] E. Felemban, C.-G. Lee, E. Ekici, MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks, IEEE Trans. Mobile Comput. 5 (6) (2006) 738–754.
[7] G. Feng, K. Makki, N. Pissinou and C. Doulgeris, "An efficient approximate algorithm for delay-cost-constrained QoS routing," *ICCCN'2001*, pp. 395-400, Phoenix, Arizona, Oct 15-17, 2001.
[8] T. He, J.A. Stankovic, C. Lu, T.F. Abdelzaher, A spatiotemporal communication protocol for wireless sensor networks, IEEE Trans. Parallel Distr. Syst. 16 (10) (2005) 995–1006.
[9] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. *Mobile Computing and Networking*, 2001.
[10] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, Proceedings of the ACM MobiCom'00, Boston, MA, 2000, pp. 56–67.
[11] S.J. Lee and M. Gerla, Split multipath routing with maximally disjoint paths in ad hoc networks, *Proceedings of the IEEE ICC*, pp. 3201–3205, 2001.
[12] S. Lee, B. Bhattacharjee, and S. Banerjee. Efficient Geographic Routing in Multihop Wireless Networks. In ACM Mobihoc'05, May 2005.
[13] J. J. Lemmon, "Wireless Link Statistical Bit Error Model," *NTIA Report 02-394*, June 2002.
[14] S. Li, A. Lim, S. Kulkarni, C. Liu, "EDGE: A Routing Algorithm for Maximizing Throughput and Minimizing Delay in Wireless Sensor Networks", *Proceedings of the 26th Military Communications Conference (MILCOM'07), October 2007.*
[15] 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*, Aug 22-25 2001.
[16] 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. In *Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'02)*, page 55, Washington, DC, USA, 2002. IEEE Computer Society.
[17] S. Mao, S. Lin, S. S. Panwar, and Y. Wang, "Reliable transmission of video over ad hoc networks using automatic repeat request and multipath transport," in Proc. IEEE Fall VTC Atlantic City, NJ, Oct. 2001, pp. 615-619.
[18] S. Mao, S. Panwar, and Y. Hou, "On Optimal Partitioning of Realtime Traffic over Multiple Paths," in *INFOCOM 2005*, March 2005.
[19] S. Mao, S. Lin, S. Panwar, Y. Wang and E. Celebi, "Multipath video transport over ad hoc networks: Multistream Coding With Multipath Transport", in *IEEE Journal of Selected Areas In Communications, VOL. 21, NO. 10, December 2003.*
[20] Marc R. Pearlman, Zygmunt J. Haas, Peter Sholander, Siamak S. Tabrizi, On the impact of alternate path routing for load balancing in mobile ad hoc networks, Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing, November 20, 2000, Boston, Massachusetts
[21] D. Sidhu, R. Nair, and *S* . Abdallah, "Finding Disjoint Paths in Networks," *Proceedings of ACM SIGCOMM'91,* Zurich, Switzerland, Sep. 1991. pp. 43-5 I.
[22] I. Richardson, "H.264 and MPEG-4 video compression," Wiley.
[23] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, Englewood Cliffs, NJ, 1995.
[24] Y. Wang and S. Lin, "Error resilient video coding using multiple description motion compensation," IEEE Trans. Circuit and System for Video Technology, vol. 12, no. 6, pp. 438-453, Jun., 2002.
[25] Simulate 802.11b Channel within NS2. Wu Xiuchao, SOC, NUS.
[26] Z. Ye, S.V. Krishnamurthy, and S.K. Tripathi. A framework for reliable routing in mobile ad hoc networks. In Proc. of IEEE INFOCOM, 2003.