

# Tele-Immersion for Wireless Networks

Rahul Malik  
Department of Computer  
Science,  
UIUC, USA  
rmalik4@uiuc.edu

Klara Nahrstedt  
Department of Computer  
Science,  
UIUC, USA  
klara@cs.uiuc.edu

Jennifer C. Hou  
Department of Computer  
Science,  
UIUC, USA  
jhou@cs.uiuc.edu

## ABSTRACT

Tele-immersive 3D multi camera environments are beginning to emerge and they bring with them lots of applications as well as challenging research problems. They enable cooperative interaction between geographically distributed sites. One of the important questions that arises in the design and implementation of such systems is that how to transmit huge amounts of 3D data consisting of multiple streams, which is generated by multiple interdependent sources in one tele-immersive environment. Due to data requirements, in the past a tele-immersive session occurred only over wired networks. In this paper, we propose an algorithm to transmit this enormous amount of data using multiple streams over wireless mesh networks. One of the significant novelties of our work is to find multiple transmission paths from a given source to a destination and to transmit data concurrently over them. This, according to us, would comprise the first step towards porting a tele-immersive system from the wired to the wireless domain. We have implemented the algorithm over ns-2 simulator. Through our extensive simulation studies, we have compared the performance of the proposed algorithm with existing wireless multi-path and single-path protocols and show that our protocol achieves markedly better performance.

## 1. INTRODUCTION

Tele-immersion is an emerging technology that enables cooperative interaction between users located at different geographical locations. The main strength of tele-immersion lies in its use of resources such as a shared virtual space and the free viewpoint stereo videos, which greatly enhance the immersive experience of each participant. Tele-immersive systems have great potential applications, especially in tele-medicine, remote education, virtual offices, etc. There is active ongoing research in the diverse fields of computer vision, graphics, data compression and high-speed networking to deliver a realistic 3D immersive experience in real time.

The Tele-immersive Environments for **EVERY**body (TEEVE)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Immerscom 2009, May 27-29, 2009, Berkley, USA.  
Copyright C 2009 ICST ISBN # 978-963-9799-39-4

[18] project currently underway at some universities aims to integrate, deploy, and experiment with existing tele-immersive components in order to explore the possibilities and challenges of deploying this cutting edge technology in a more cost effective manner and across a broader audience. The TEEVE system features a distributed multi-tier application model in which each user is surrounded by cameras and audio devices to capture his or her body movements. The implementation of the capturing level of the TEEVE system captures user input and transmits it across the transmission tier to the receiving user, while the implementation of the displaying level allows the receiving user to observe the transmitting user's physical activity from arbitrary angles.

As discussed in [17], one of the most important challenges of tele-immersive systems lies in transmitting multiple 3D video streams from different fields of view over the Internet. As opposed to 2D video streams, the 3D streams consume much more bandwidth and are deployed for a much wider field of view. Even for moderate data rates, the bandwidth requirements are very high.

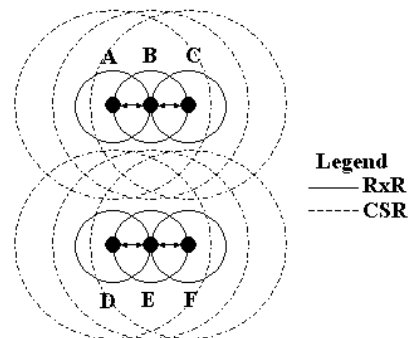


Figure 1: Intra-flow and inter-flow interference. Here  $RxR$  is the reception range of a node and  $CSR$  is the carrier-sensing range.

In this paper, we explore ways of transmitting 3D video streams over wireless networks. The advantage of this system would be that it would lead to much more flexibility in comparison to the existing model, because on the displaying level, the user need not be constrained to sit in a fixed room to watch the 3D video. There is already some research going on regarding displaying the 3D video using PDAs and mobile devices[15].

For the purpose of clarity, let us assume that the sender

of 3D streams is a fixed wireless access point and the receiver is a wireless node connected via mesh network. All the intermediate nodes are also wireless nodes. In order to improve the throughput from a given source to a destination, we use multiple non-interfering paths from a source to a destination, and transmit data concurrently over them. This would require coordinated scheduling from the source node. In addition, we increase the throughput of each path by alleviating bottleneck. To explain the concept in greater detail, in a wireless network, there are two major kinds of interferences. The first one is intra flow interference. This is the interference between consecutive wireless links in the same flow. Figure 1 shows an example of interference in a single flow. Here, the flow between nodes  $A-B-C$  and  $D-E-F$  is experiencing intra-flow interference within themselves. The intra-flow interference is considered as a self capacity limiting mechanism. In order to avoid this interference, schemes such as TDMA [2] have been described. In this paper, our major focus is inter-flow interference. So, we assume that a suitable scheme such as TDMA has already been incorporated in order to avoid intra-flow interference. The other kind of interference is inter-flow interference. This is interference between different flows. In Figure 1, the two flows i.e.  $A-B-C$  and  $D-E-F$  are an example of inter-flow interference among them. As already described, our major focus in this paper is inter-flow interference.

Although, previously also some work [12],[14] has been done in relation to finding node-independent multiple paths from a source to a destination, but their main intention has been to use alternate paths only when the existing path currently being used for transmission gets broken. Unlike our proposed scheme, at any given time, only one path was used for transmission in their scheme. Main contributions of this paper are:

1. Finding multiple non-interfering paths from the source to the destination and concurrently transmitting data over them.
2. Alleviating bottlenecks in each of these paths found by looking at the least bandwidth link in the path.

This paper is organized as follows. Section 2 presents an overview of a tele-immersive system, Section 3 presents the details of the proposed protocol, Section 4 provides the performance evaluation results of the protocol, Section 5 gives an overview of related work and finally Section 6 gives the conclusion.

## 2. OVERVIEW OF A TELE-IMMERSIVE SYSTEM

### 2.1 Tele-Immersion Architecture

The tele-immersive system is modeled as a distributed multi-level application [18] (Figure 2). The first level consists of the *capturing level*. This consists of a set of 3D cameras organized in a half-circle within a room. Each 3D camera cluster consists of four 2D cameras, two of which are used for 3D reconstruction, one for verification of reconstruction and one, which is a color camera, for adding color. Each camera cluster is connected to a *camera PC* that reconstructs one 3D video stream from these camera outputs. The *control PC* is responsible for controlling all the camera PCs in synchronizing the time of taking their images. The

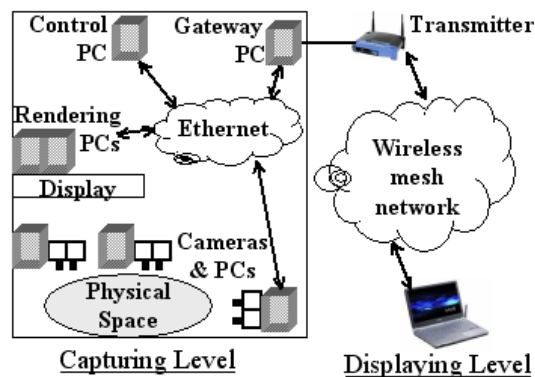


Figure 2: Tele-immersion Application Model

3D camera environment at the capturing level is responsible for 3D scene acquisition and reconstruction. The 3D video image generated for each of the cameras is further processed by *gateway PC* of the middleware layer to perform tasks including traffic shaping, multi-stream coordination and synchronization, and skew control so that remote clients can generate the realistic 3D model of the subject in real time. Each of individual 3D streams is compressed before sending. The gateway is connected to the wireless transmitter which is source for transmission.

The second level consists of the *transmission level*. In the current architecture, this level receives the compressed 3D video streams, shapes them and sends them over the Internet to the receiving nodes within a remote room. This is a mesh network. Finally, the third level consists of *displaying level*. It takes the received 3D video streams, decompresses them, renders them into an immersive video and forwards the rendered video over LAN to one or more displays.

### 2.2 Format of 3D Data

The capturing level consists of  $N$  identical 3D cameras organized around the subject and synchronized. At time  $T_j$ , each camera captures one frame of depth image from its viewpoint. So, at time  $T_j$ , the remote renderer uses  $N$  3D reconstructed frames to show the same scene from different viewing angles.

One major challenge is to accommodate the huge data rate requirement from the application layer. This is especially true for higher frame rates and resolutions. In current experiments, the image size is chosen as  $320 \times 240$  pixels, although the maximum resolution supported by underlying hardware is  $640 \times 480$ . Each pixel consists of 5 bytes, with 2 bytes of depth and 3 bytes of color information, yielding an uncompressed frame size of  $320 \times 240 \times 5 = 384,000$  bytes or 2.9 Mbits. Furthermore, 12 such 3D video cameras are used, where each 3D camera produces 10 frames per second. The whole setup indicates a bandwidth requirement of 348 Mbps. Using compression techniques, a compression ratio of 25:1 is achieved, reducing the end-to-end bandwidth requirement to 14 Mbps. However, we would ideally like to support higher frame rates and resolutions.

### 2.3 End-to-End Streaming Protocol

The end-to-end streaming protocols [18] address the chal-

allenges of bandwidth management and rate adaptation, especially for streaming of 3D video data where the resource demand is much higher than for any 2D streaming. Here the focus is mainly on capturing level since it is the main determinant factor for bandwidth consumption requirement.

Under the bandwidth constraints (when the available bandwidth is not sufficient to support all the 3D video streams at the minimum frame rate), the gateway controller PC allocates the available bandwidth for each stream. A view-based allocation scheme is adopted. This is characterized by pre-defined priorities and dynamic view changes. As an example, the 3D cameras of the first view are generally more important than those on the side view and are given higher priority. During runtime, the user may select different views, which affects the bandwidth allocation as well.

It is also important to realize that although one macro-frame consists of  $N$  3D reconstructed frames (which is same as the number of cameras) at time  $T_j$ , they cannot be transmitted at the same time since their simultaneous transmissions would cause a strong congestion in the network. Hence, the individual frames of the macro frame need to be shaped and spaced out within a certain completion interval.

### 3. EXPLOITING MULTIPLE PATHS FOR SIMULTANEOUS DATA TRANSMISSION

In this section, we look at how multiple paths can be found from the source<sup>1</sup> to the destination for the concurrent transmission of data. In the present application of tele-immersion, most of the data transmission is of down-link traffic. This is because all the 3D streams are generated at capturing level and displayed at the displaying level. The capturing level consists of an access point for sending the data to the transmission level which then transmits the data to displaying level through multiple paths that simultaneously transmit the data.

The rationale behind choosing multiple paths for the simultaneous data transmission for current application is logical, considering the model of 3D data that we have described earlier. At time  $T_j$ , each camera  $i (i \in \{1, 2, \dots, N\})$  captures one frame from its viewpoint, denoted by  $f_{i,j}$ . This means that at time  $T_j$ , all the frames  $f_{i,j}$ , where  $f_{i,j} \in \{f_{1,j}, f_{2,j}, \dots, f_{N,j}\}$  need to be transmitted from capturing level to displaying level. The capturing level consists of  $N$  TCP connections, one for each of the frames coming from each of the cameras. Therefore, it is logical to use multiple paths for simultaneously transmitting data as there exist multiple connections for different streams.

#### 3.1 Background and Terminology

We consider a wireless mesh network with  $n$  nodes located in a plane. This work assumes statically located nodes; however, nodes can be placed irregularly. In this model, the transmission power is equal at all nodes. The received power is modeled by path loss with exponent  $\alpha$ .

In our protocol, each wireless node has a single communication radio which has a communication range  $R_i$  and a larger interference range of  $R_i'$ . The wireless transceiver supports the CSMA/CA medium access control protocol. In our protocol, we assume a single physical channel. A trans-

<sup>1</sup>Here, the *source* for our protocol is the gateway/wireless base-station as shown in Figure 2 and the *destination* is the end wireless node that renders all the received streams.

mission from node  $i$  to node  $j$  is successful if the following conditions are satisfied:

$$d_{ij} < R_j \quad \text{and} \quad d_{kj} > R_j' \quad \forall k \neq i, j$$

where  $d_{ij}$  represents the distance between node  $i$  and  $j$ .

The *connectivity graph*  $G$  is a directed graph whose vertices correspond to the wireless nodes ( $N_G$ ) and the edges correspond to the wireless links ( $L_G$ ) between the nodes. There is a directed link from  $i$  to  $j$  if  $d_{ij} < R_i$  and  $i \neq j$ .

A *conflict graph*  $F$  is a graph whose vertices correspond to the links in the connectivity graph  $G$ . There is an edge between the vertices  $l_{ab}$  and  $l_{cd}$  in  $F$  if the links  $l_{ab}$  and  $l_{cd}$  may not be active simultaneously. This is similar to the model of conflict graph as presented in [9],[6]. We use the technique as used by [5] to compute the conflict graph. We assume that each node knows its position (e.g. using GPS) and disseminates its position information to other nodes in the neighborhood. Each station then geometrically computes which stations are within an interference radius. The *conflicting set* of a link  $l_{ij}$ ,  $C(l_{ij})$  is defined as the set of all the links that cannot be active simultaneously with  $l_{ij}$ .

#### 3.2 Finding Multiple Paths from the Transmission Level to the Displaying Level

Now, we discuss the algorithm for finding multiple paths from the access point located at the transmission level to the displaying node, for simultaneous transmission of data. As shown by Jain et al. [6], computing the maximum throughput that can be supported by an underlying network is a NP-hard problem. Therefore, in this paper, we present a method of computing such paths heuristically.

As shown by Ye et al. [19], the number of disjoint paths that can be found from the source to the destination depends on the node density. Furthermore, as the distance between the source and destination increases, one can find no more than a very limited number of paths between them, even at moderate node densities. So, this leads us to take a look at paths which are not totally node-independent between the source and the destination, but finds a way to increase the throughput by intelligently taking care of the bottlenecks that occur in these node disjoint paths.

Thus, our algorithm runs in three steps. In the first step, the algorithm tries to find a path from the source to the destination and tries to optimize it in terms of throughput and transmission power. Once, this is achieved, it tries to find another node disjoint path between the source and the destination, containing nodes different from the first one and so on. The final step involves scheduling the packets at the source among these different routes.

First of all, we describe the procedure of finding one path from the source to the destination and maximizing its performance in terms of overall throughput and transmission power. Previously, issues for routing the traffic over wired networks have been discussed so that Quality of Service (QoS) is maintained [3], [8]. The QoS Routing Problem consists of finding an optimal cost path from a source to a destination subject to one or more constraints (e.g., total delay and loss probability) on the path. It is well known that this problem is NP-complete [4] and therefore many heuristics have been provided. From the connectivity graph, the source node tries to find the shortest path to the destination. This can be easily determined using shortest path algorithm. Now, in order to improve the throughput of this path, the

link with the lowest capacity is chosen, which is the bottleneck in this path. If we are able to improve its throughput, then the overall throughput of the whole path can be improved. If this link does not contain source or destination nodes, i.e. it is not the first or last link in the path, then it is optimized, otherwise not. This algorithm for finding the paths is a source initiated link-state based reactive algorithm. Any node that is a source and wants to transmit the tele-immersion video streams, sends route-request packets to all the nodes in the network and in reply, each node sends its local information to the source.

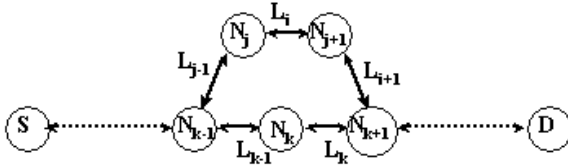


Figure 3: Example scenario

After choosing the bottleneck in the path, we try to intelligently increase the throughput of whole path by choosing another link for transmission that can transmit simultaneously with the bottleneck link, thus reducing the burden on the bottleneck link. As in Figure 3, the bottleneck link is  $l_k$ , which is between nodes  $N_k$  and  $N_{k+1}$ . We find the conflicting set of link  $l_{k-1}$ , which is previous to link  $l_k$ , and divide it into independent sets. An independent set is a set of conflict graph nodes which do not have a link between them. Intuitively, this corresponds to any set of links whose transmission does not interfere with each other. One possible such independent set in the example is  $\{l_k, l_{j-1}\}$ . We choose a set that contains link  $l_k$  and we partition the data between these two so as to improve the throughput. In order to find the next completely node disjoint path, the set of links that have been used in previous step and the links which interfere with the links in path are removed and the previous step is run again.

The final step is to sequence the paths temporally among the various routes found previously. Here, note that the paths found are non-interfering from the second hop onwards. However, at the first hop at the source and at the last hop at the destination, there still is congestion. In order to avoid that, we describe the Priority Based Scheme (PBS).

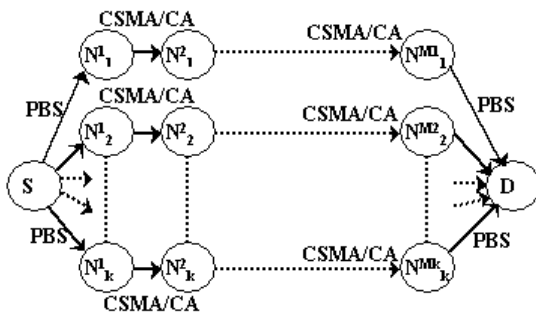


Figure 4: Temporal Sequencing of Packets

The Priority Based Scheme (PBS) is based on the Black Burst (BB) protocol that was first proposed in [16] to achieve fair channel sharing in wireless LAN. Here, we are only interested in the BB channel contention scheme and we adapt it to cope with the interference at the source and the destination. In the BB scheme, a node with a packet to deliver accesses the channel using a medium interframe spacing of length  $t_{med}$ . Upon hearing an idle channel for  $t_{med}$ , the node starts transmitting a jamming signal, called a black burst, with length  $t_{BB}^p$  proportional to the packet priority  $p$ . After sending a BB, a node reverts to channel sensing for an interval  $t_{short}$  shorter than the medium interframe spacing  $t_{med}$ . If the channel remains idle for such an interval, the node begins transmitting its packet, otherwise it reverts to channel sensing until the channel is sensed idle for at least  $t_{med}$  before transmitting another BB. Note that the BB scheme can be used in conjunction with the CSMA/CA scheme of 802.11 and it always has higher priority than CSMA/CA. This is because in CSMA/CA, instead of transmitting a packet immediately when the backoff timer expires, each node must first perform channel sensing for a long interframe spacing  $t_{long} > t_{med}$ ; if the channel is perceived busy, the node enters backoff again.

In the priority based scheme, the BB protocol is used at the source and the destination node and rest of the nodes use the CSMA/CA protocol. At the source, a value of priority  $p$  is assigned to different streams based on the priority determined from the views, as described in section 2. The destination uses the same priorities as the source for the different streams. As the protocol is a link-state based protocol, the priorities of the various links are conveyed to the destination and this can be achieved by piggybacking the information in the beginning. Thus, there is no collision at the source as well as at the destination because a stream with a higher priority will be allowed to be transmitted before a lower priority one.

## 4. PERFORMANCE EVALUATION

### 4.1 Existing Protocols Used for Comparison

#### 4.1.1 SMR (Split Multi-path Routing)

SMR[7] is based on DSR (Dynamic Source Routing) as it is a source based on-demand routing protocol. This protocol attempts to discover maximally disjoint paths. The routes are discovered on demand in the same way as it is done with DSR. That is, the sender floods a Route REQuest (RREQ) message in the entire network. However, the main difference is that intermediate nodes do not reply even if they know a route to the destination. From the received RREQs, the destination then identifies multiple disjoint paths and sends a Route REPLY (RREP) packet back to the source for each individual route.

#### 4.1.2 AOMDV (Ad hoc On demand Multi-path Distance Vector routing)

AOMDV[11] extends AODV [13] to provide multiple paths. In AOMDV each RREQ and respectively RREP defines an alternative path to the source or the destination. Multiple paths are maintained in routing entries in each node. The routing entries contain a list of next-hops along with corresponding hop counts for each destination. To ensure loop-free paths AOMDV introduces the *advertised\_hop\_count* value

at node  $i$  for destination  $d$ . This value represents the maximum hop-count for destination  $d$  available at node  $i$ . Consequently, alternate paths at node  $i$  for destination  $d$  are accepted only with a lower hop-count than the advertised hop count value. Node-disjointness is achieved by suppressing duplicate RREQ at intermediate nodes.

### 4.1.3 AODV (Ad hoc On demand Distance Vector routing)

AODV [13] is used as a reference on demand single path protocol. AODV is used as a benchmark for comparison with single path protocols.

## 4.2 Simulation Environment

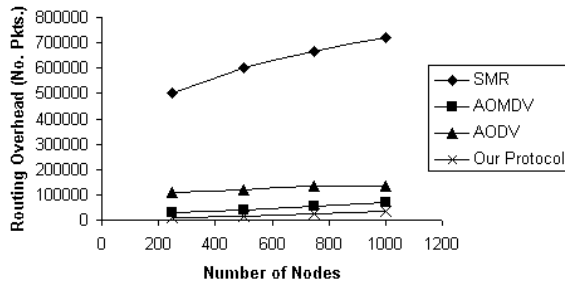


Figure 5: Routing Overhead

We use the *ns-2* simulator [1] for simulation purposes. The distributed coordination function (DCF) of IEEE 802.11 for wireless nodes is used at the MAC layer. The radio model uses characteristics similar to Lucent's WaveLAN radio interface. WaveLAN is a shared-media radio with a nominal bit-rate of 2 Mb/sec and a nominal radio range of 250 meters. An  $n$  node network in a field with dimensions  $1200m \times 800m$  is used. The number of nodes was varied to see the effect of node density and the number of paths found. The number of nodes corresponds the number of nodes in the mesh network. The sender and receiver were randomly chosen. We use the Constant Bit Rate (CBR) model in *ns-2* for video traffic. Cross traffic is specified randomly, up to 25% of the capacity on each link, with exponentially distributed packet sizes.

### 4.3 Simulation Results

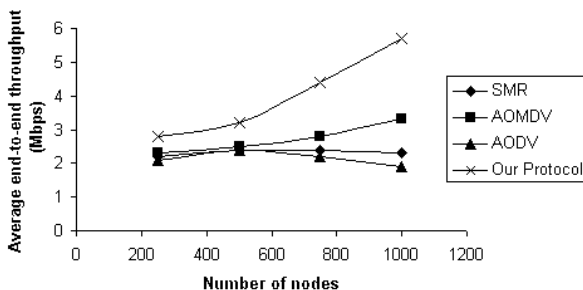


Figure 6: Average end-to-end throughput

We use the following metric for comparing the performance of our protocol with other protocols: routing overhead, average end-to-end throughput, average end-to-end delay of data packets and the average number of paths discovered.

In terms of routing overhead, the results are displayed in Figure 5. As can be seen from the results, SMR has the highest overhead in terms of number of packets. Also, as the node density increases, the overhead increases. This is obvious as more information needs to be collected for a larger number of nodes. Note here that the number of nodes here refers to the number of nodes in the mesh network.

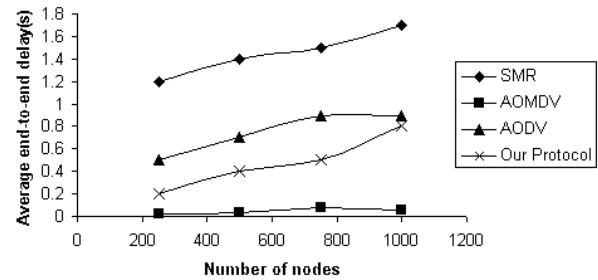


Figure 7: Average end-to-end delay

The results of average end-to-end throughput and delay are displayed in Figures 6 and 7 respectively. Clearly, in terms of end-to-end throughput, our protocol outperforms all the protocols. This is because the paths that are discovered by our protocol are non-interfering paths. Thus, these paths are able to deliver maximum throughput as compared to the paths of other multi-path protocols. Also, in terms of throughput and delay, SMR performs the worst. This is because high routing overheads of the SMR penalize data packets, therefore high buffering delays contribute to high end-to-end delay and low throughput.

Finally, in terms of number of paths found from source to destination, as the node density increases, the number of paths that can be discovered in any protocol increases. The results are shown in Figure 8. Again, SMR was not able to find a large number of simultaneous paths. However, our protocol performs well here. This shows that our protocol is better than existing ones.

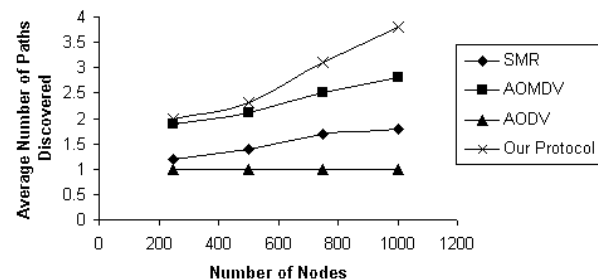


Figure 8: Average number of paths discovered

## 5. RELATED WORK

Most of the work in this area has either been done in relation to the sending of one video stream over multiple paths or finding multiple paths and then using one of the multiple paths for improving the performance. In relation to transmitting one video stream over multiple paths, recently some work has been carried out. In [10], the authors propose to transmit layered video coding over multiple paths in wireless ad hoc networks. In their scheme, base layer and enhancement layer packets are transmitted over different paths. Base layer packets are protected by an automatic repeat request (ARQ). A lost base layer packet is retransmitted through the path through which enhancement layer packets are transmitted. In [20], Zhu, et al. propose a rate allocation scheme to optimize the expected received video quality based on models of the encoder's rate-distortion performance and the network's rate-congestion tradeoffs. However the traffic model used in the above papers assumes that all nodes transmit simultaneously, and the capacity of each link is decided by the Signal Interference Noise Ratio (SINR).

In [12], the effect of the number of multiple paths on routing performance has been studied using an analytical model. Most of the multi-path routing protocols are implemented as extensions or modifications of existing single path routing protocols like the proactive DSDV (Destination-Sequenced Distance Vector) and OLSR (Optimized Link State Routing), or the reactive on demand protocols: AODV or DSR. Analysis and comparison of single path and multi-path routing protocols in ad hoc networks has also been conducted in [14]. There, protocol performance is examined with regard to protocol overhead, traffic distribution, and throughput.

## 6. CONCLUSION

In this paper, we have presented tele-immersive data transmission over wireless mesh networks. We find multiple simultaneous paths which are non-interfering with each other and use those paths to concurrently transmit data between one source and destination pair. As can be seen from the results, we were able to achieve a maximum average end-to-end throughput of 6Mbps, whereas the desired throughput was 14Mbps. Thus, the end-to-end streaming protocol needs to remove certain streams that are not in the field of view.

In future, we plan to experiment with real testbeds to evaluate the protocol in real scenarios. In addition, we will also consider using schemes such as layered video encoding [10] for important streams over multiple paths.

## Acknowledgment

This work was supported in part by the National Science Foundation grant NSF CNS 05-20182.

## 7. REFERENCES

- [1] [www.isi.edu/nsnam/ns/](http://www.isi.edu/nsnam/ns/).
- [2] B. Buy, R. Pellizzoni, M. Caccamo, C. Cheah, and A. Tzakis. Soft real-time chains for multi-hop wireless ad-hoc networks. *IEEE Real-Time and Embedded Technology and Applications Symposium*, April 2007.
- [3] S. Chen, K. Nahrstedt, and Y. Shavitt. A qos-aware multicast routing protocol. *IEEE Journal on Special Areas in Communication*, December 2000.
- [4] M. Garey and D. Johnson. *Computer and Intractability: A guide to the theory of NP-completeness*. Freeman, 1978.
- [5] R. Gupta, J. Musacchio, and J. Walrand. Sufficient rate constraints for qos flows in ad-hoc networks. *Ad Hoc Netw.*, pages 429–443, May 2007.
- [6] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. *ACM Mobicom*, September 2003.
- [7] S. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. *IEEE ICC*, pages 3201–3205, 2001.
- [8] K. S. Lui, J. Wang, L. Xiao, and K. Nahrstedt. Qos multicast routing with heterogeneous receivers. *IEEE Globecom*, December 2003.
- [9] H. Luo, S. Lu, and V. Bhargavan. A new model for packet scheduling in multihop wireless networks. *ACM Mobicom*, pages 76–86, 2000.
- [10] S. Mao, S. Lin, S. Panwar, and Y. Wang. Reliable transmission of video over ad-hoc networks using automatic repeat request and multi-path transport. *IEEE Vehicular Technology Conference*, pages 615–619, 2001.
- [11] M. Marina and S. Das. On-demand multipath distance vector routing in ad hoc networks. *International Conference for Network Protocols*, 2001.
- [12] A. Nasipuri, R. Castaneda, and S. Das. Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *Mob. Netw. Appl.*, 6(4):339–349, 2001.
- [13] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. *RFC 3561*, 2003.
- [14] P. Pham and S. Perreau. Performance analysis of reactive shortest path and multipath routing mechanism with load balance. *IEEE Infocom*, 2003.
- [15] S. Shi, K. Nahrstedt, and R. Campbell. View-dependent real-time 3d video compression for mobile devices. *Proceeding of the 16th ACM international conference on Multimedia*, pages 781–784, 2008.
- [16] J. Sobrinho and A. Krishnakumar. Real-time traffic over the ieee 802.11 medium access control layer. *Bell Labs Technical Journal*, 1(2):172–187, 1996.
- [17] H. Towles, S. Kum, T. Sparks, S. Sinha, S. Larsen, and N. Beddes. Transport and rendering challenges of multi-stream, 3d tele-immersion data. *NSF Lake Tahoe Workshop on Collaborative Virtual Reality and Visualization*, October 2003.
- [18] Z. Yang, Y. Cui, B. Yu, J. Liang, K. Nahrstedt, S. H. Jung, and R. Bajscy. Teeve: The next generation architecture for tele-immersive environments. *7th IEEE International Symposium on Multimedia*, 2005.
- [19] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi. A framework for reliable routing in mobile ad hoc networks. *IEEE Infocom*, 2003.
- [20] X. Zhu, S. Han, and B. Girod. Congestion-aware rate allocation for multipath video streaming over ad hoc wireless networks. *IEEE International Conference on Image Processing*, pages 2547–2550, October 2004.