Book Title: XXXXXXXXXXXXXXXXXXXXXXXXXXX

Editors

April 2, 2007

Contents

Chapter 1

Multipath Unicast and Multicast Video Communication over Wireless Ad Hoc Networks

Wei Wei, Avideh Zakhor
University of California at Berkeley,

1.1 Introduction

With the increase in the bandwidth of wireless channels and in the computational power of mobile devices, video applications are expected to become more prevalent on wireless ad hoc networks in a near future. Examples of video communication applications over wireless ad hoc networks include spontaneous video conferencing at a location without wireless infrastructure, transmitting video on the battlefield, and search and rescue operations after a disaster.

Video communication is fundamentally different from data communication, since they are delay and loss sensitive. Unlike data packets, late arriving video packets are useless to the video decoder. Thus, the retransmission techniques are not generally applicable to video communication applications with low delay requirements, especially in multicast situations.

There are additional challenges for supporting video communication over wireless ad hoc networks. Due to the mobility of wireless nodes, the topology of ad-hoc networks may change frequently. Thus the established connection routes between senders and receivers are likely to be broken during video transmission, causing interruptions, freezes, or jerkiness in the received video signal. An end-to-end connection route in wireless ad hoc networks generally consists of multiple wireless links, resulting in higher random packet loss than single hop wireless connections in wireless networks with infrastructure, such as base stations. Other challenges include lower wireless network capacity compared to wired networks, and limited battery life of mobile devices. These constraints and challenges, in combination with the delay and loss sensitive nature of interactive video applications, make video communication over wireless ad hoc networks a challenging proposition [1].

Multiple Description Coding (MDC) generates multiple compressed descriptions of the media in such a way that a reasonable reconstruction is achieved if any one of the multiple descriptions is available for decoding, and the reconstruction quality is improved as more descriptions become available [35][36]. The main advantage of MDC over layered coding is that no specific one description is needed in order to render the remaining descriptions useful. As such, unless none of the description make it to the receiver, video quality degrades gracefully with packet loss. However, there is a penalty in coding efficiency and bit rate in using MDC as compared to Single Description Coding (SDC) [35][36]. Specifically, for a given visual quality, the bit rate needed for MDC exceeds that of SDC depending on the number of descriptions. In this chapter, we will use MDC video in multipath unicast and multicast scenarios.

In this chapter, we introduce new path diversity schemes in order to provide robustness for both unicast and multicast video communication applications over wireless ad hoc networks.

1.1.1 Unicast

For the unicast case, we propose a class of techniques to find two node-disjoint paths, which achieve minimum concurrent Packet Drop Probability (PDP) of all path pairs. For MDC streaming, different descriptions are transmitted on different paths in order to fully utilize path diversity, and the worst case scenario occurs when all descriptions are missing. Streaming over the Path Pair with Minimum concurrent Packet Drop Probability, denoted by PP_MDP, minimizes the probability of concurrent loss of all the descriptions, thus optimizing the worst case video quality over all times. While most of our simulation results refer to MDC, our basic results and conclusions can be easily extended to Forward Error Corrected (FEC) video as well. For FEC streaming, concurrent packet drop over the selected PP_MDP can be shown to be less likely than that of simple node disjoint paths, resulting in lower unrecoverable probability.

In this work, we use a *conflict graph* [2][3][4] to model effects of interference between different wireless links. The conflict graph indicates which groups of links interfere with each other, and hence can not be active simultaneously. We propose a model to estimate the concurrent PDP of two node-disjoint paths, given an estimate of cross traffic flows' rates, and bit rate of the video flow. We then propose a heuristic PDP aware multipath routing protocol based on our model, whose performance is shown to be close to that of the "optimal routing", and significantly better than that of the node-disjoint multipath routing, and the shortest-widest routing.

1.1.2 Multicast

Multicast is an essential technology for many applications, such as group video conferencing and video distribution, and results in bandwidth savings as compared to multiple unicast sessions. Due to the inherent broadcast nature of wireless networks, multicast over wireless ad hoc networks can be potentially more efficient than over wired networks [5].

In this chapter, we first introduce an architecture for multiple tree video multicast communication over wireless ad hoc networks. The basic idea is to split the video into multiple parts and send each part over a different tree, which are ideally disjoint with each other so as to increase robustness to loss and other transmission degradations. We then propose a simple serial Multiple Disjoint Tree Multicast Routing protocol (Serial MDTMR), which constructs two disjoint multicast trees sequentially in a distributed way, to facilitate multiple tree video multicast. This scheme results in reasonable tree connectivity while maintaining disjointness of two trees.

However Serial MDTMR has a larger routing overhead and construction delay than conventional single tree multicast routing protocols, as it constructs the trees in a sequential manner. To alleviate these drawbacks, we further propose parallel multiple nearly-disjoint trees multicast routing (Parallel MNTMR) in which nearly disjoint trees are constructed in parallel, and in a distributed way. Using the Parallel MNTMR, each receiver is always able to connect to two trees, regardless of the node density. Simulations show that multiple tree video multicast with both Serial MDTMR and Parallel MNTMR improve video quality significantly compared to single tree video multicast; at the same time routing overhead and construction delay of Parallel MNTMR is approximately the same as that of a single tree multicast protocol.

The rest of this chapter is organized as follows. Section 1.2 discusses related work. Then in Section 1.3 we propose a method to estimate the concurrent PDP of two node-disjoint paths, formulate the optimal multipath selection problem for video streaming over wireless ad hoc networks, and develop a heuristic PDP aware multipath routing protocol. We present the testbed implementation and experimental results in Section 1.4. In Section 1.5, we propose multiple tree multicast framework. Then we present Serial MDTMR and Parallel MNTMR in Sections 1.6 and 1.7 respectively. We conclude this chapter in Section 1.8.

1.2 Related Work

1.2.1 Unicast

Several researchers have proposed to distribute MDC video flow over multiple paths for multimedia transport [6][7][8][9][10]. These efforts have successfully demonstrated that the combination of path diversity and MDC provides robustness in video communication applications. However most of these either assume that the set of paths is given, or simply select two node/link disjoint paths.

Only a few recent approaches address the problem of selecting the best path pair for MDC video streaming [11][12][13]. The path selection model used in [11][12] is more suitable for the Internet overlay networks. In [13], the authors select two paths with minimal correlation for MDC streaming over Internet overlay networks. In contrast, we consider path selection over wireless ad hoc networks when interference plays an important role.

Multipath routing for wireless ad hoc networks has been an active research area recently [14][15][16][17][18][19]. Most existing approaches focus on how to obtain multiple node/link disjoint paths. In [19], the authors propose a heuristic algorithm to select multiple paths to achieve the best reliability, assuming failure probability of different links are independent and given. In contrast, we focus on how to estimate PDP of each link in wireless ad hoc networks considering interference.

The problem of finding rate constraints on a set of flows in a wireless ad hoc network is studied in [2][3]. Both papers model the interference between links in an hoc network using conflict graphs and find capacity constraints by finding the global independent sets of the conflict graph. In [4], the authors develop a different set of rate constraints using the cliques, i.e. complete subgraphs, of the conflict graph.

Our approach differs from previous work in two significant ways. First, our proposed multipath selection model estimates the concurrent congestion probability of two paths by taking into account the interference between different links, which reflects actual constraints of a wireless ad hoc network. Second, our proposed heuristic Interference aWare Multipath (IWM) protocol provides reasonable approximation to the solution of the optimal multipath selection problem for video streaming over wireless ad hoc networks.

1.2.2 Multicast

Multicasting MDC video was first introduced in CoopNet [20] in the context of peer-to-peer networks to prevent web servers from being overwhelmed by large number of requests. CoopNet uses a centralized tree management scheme, and each tree link is only a logical link, which consists of several physical links and as such is inefficient in wireless ad hoc networks. In [21], the authors propose a genetic algorithm based solution for multiple tree multicast streaming, assuming that (a) they obtain each link's characteristics, and (b) consecutive links' packet loss rates are independent.

There has also been a great deal of prior work in the area of multicast routing in wireless ad hoc networks [22][23][26][24][28][27][25]. The On-Demand Multicast Routing (ODMRP) [22] builds multicast mesh by periodically flooding the network with control packets to create and maintain the forwarding state of each node, when the source has packets to send. It takes advantage of the broadcast nature of the wireless network by forwarding group flooding, which provides a certain amount of diversity. A mesh structure is equivalent to a tree structure with *tree flood* enabled[24]. In the remainder of this paper, we refer to ODMRP as a single tree multicast protocol. The Adaptive Demand-Driven Multicast Routing (ADMR) [24] attempts to reduce non-on-demand components within the protocol as much as possible. ADMR does not use periodic network-wide floods of control packets, periodic neighbor sensing, or periodic routing table exchanges. In ADMR, forwarding state is specific to each sender rather than being shared by the entire multicast group. This approach reduces unnecessary forwarding data redundancy. There is also a local subtree repair scheme to detect a broken link by downstream node in ADMR. The Adaptive Core Multicast Routing Protocol (ACMRP) [25] is an on-demand core-based multicast routing protocol that is based on a multicast mesh. A multicast mesh is created and maintained by the periodic flooding of the adaptive core. A core emerges on demand and changes adaptively according to the current network toplogy. This scheme outperforms ODMRP in multi-source scenarios. The Independent-Tree Ad Hoc Multicast Routing (ITAMAR) [29] creates multiple multicast trees based on different metrics in a centralized way. ITAMAR constructs multiple edge disjoint or nearly disjoint trees. The main objective of this protocol is to increase the average time between multicast tree failures. The ITAMAR algorithms are basically based on Dijkstra SPF algorithm [30], which is a centralized approach, and requires knowledge of network topology. There are two obvious advantages of our proposed techniques as compared to ITAMAR: first, our protocols are distributed, rather than centralized, and hence do not require the knowledge of network topology in advance; second, our protocols' overhead is $O(n)$, rather than $O(n^2)$ of ITAMAR, where n is the number of total nodes.

1.3 Multipath Selection for Unicast Streaming

Our approach is to minimize concurrent PDP of two node-disjoint paths in a wireless ad hoc network. As stated earlier, this is equivalent to optimizing the worst case video quality at clients. The node-disjoint constraint is useful for mobile wireless ad hoc networks, because it reduces the correlation of packet drop in different paths significantly.

1.3.1 Envisioned Network Model

A wireless ad hoc network can be modelled as a directed graph $G(V, E)$, whose vertices V correspond to wireless stations, and the edges E correspond to wireless links. Let $n_i \in V$, $1 \leq i \leq N$ denote the nodes, and d_{ij} denote the distance between nodes n_i and n_j. Each node is equipped with a radio with communication range r, and a potentially larger interference range ω. There is a link l_{ij} from vertex n_i to vertex n_j if and only if $d_{ij} < r$. If the transmission over link l_{ij} makes the transmission over link l_{kl} unsuccessful, link l_{ij} interferes with link l_{kl}. We use a model similar to the protocol interference model introduced in [3] to determine whether two links interfere with each other.

1.3.2 The Optimal Multipath Selection Problem

Let $P^1_{S,D}$ and $P^2_{S,D}$ be any two paths connecting nodes N_S and N_D, $L^1_{S,D}$ and $L^2_{S,D}$ denote the set of links on each path respectively, and $N^1_{S,D}$ and $N^2_{S,D}$ denote the set of the nodes on each path respectively. We define two indication vectors $\mathbf{x} = (\ldots, x_{ij}, \ldots)^T$ and $\mathbf{y} = (\ldots, y_{ij}, \ldots)^T$ to represent $P^1_{S,D}$ and $P^2_{S,D}$ respectively, where x_{ij} is set to be 1 if link $l_{ij} \in L^1_{S,D}$ and is set to be 0 otherwise, and y_{ij} is defined similarly for path 2.

For the optimal multipath selection, we select two node-disjoint paths with minimum concurrent PDP. This corresponds to the following optimization problem:

$$\text{Minimize } P_{\text{drop}}(P^1_{S,D}; P^2_{S,D})$$

$$\text{with respect to } x_{ij}, y_{mn} \in \{0, 1\},\ \forall (i,j), (m,n) \in E$$

subject to

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} 1 & i = N_S \\ -1 & i = N_D \\ 0 & otherwise \end{cases} \tag{1.1}$$

$$\sum_{i:(i,j)\in E} x_{ij} \leq 1 \tag{1.2}$$

$$\sum_{n:(m,n)\in E} y_{mn} - \sum_{n:(n,m)\in E} y_{nm} = \begin{cases} 1 & m = N_S \\ -1 & m = N_D \\ 0 & otherwise \end{cases} \tag{1.3}$$

$$\sum_{m:(m,n)\in E} y_{mn} \leq 1 \tag{1.4}$$

$$N^1_{S,D} \cap N^2_{S,D} = \{N_S, N_D\} \tag{1.5}$$

Equations (1.1) and (1.2) are flow constraints to guarantee the first path to connect the source N_s and the destination N_d. They represent: (a) for each node in the first path, except the source and the destination, both the number of incoming links and the number of outgoing links are 1; (b) for the source node, the number of outgoing links is 1; (c) for the destination node, the number of incoming links is 1. Similarly, Equations (1.3) and (1.2) are flow constraints for the second path. Equation (1.5) is the node-disjoint constraint to ensure that the two selected paths do not share nodes.

We can show the following claim for the optimal multipath selection problem.

*Claim 1:*The optimal multipath selection over wireless ad hoc networks as defined in Equation (1.1) through (1.5) is NP-hard.

The proof is shown in Section 1.9.

1.3.3 Concurrent PDP of two node-disjoint paths

In this section, we show how to compute the concurrent PDP of any given two node-disjoint paths connecting the same source and destination nodes, in order to solve the optimal multipath selection problem.

Since two node-disjoint paths only share the source and the destination nodes, packet drop over two node-disjoint paths also have low correlation. Thus we can approximate the concurrent PDP over two node-disjoint paths $P^1_{S,D}$ and $P^2_{S,D}$ as

$$\begin{aligned} P_{\text{drop}}(P^1_{S,D}; P^2_{S,D}) &\approx P_{\text{drop}}(P^1_{S,D}) \cdot P_{\text{drop}}(P^2_{S,D}) \\ &= [1 - \prod_{l_{ij} \in L^1_{S,D}} (1 - P_{\text{drop}}(l_{ij}))] \cdot [1 - \prod_{l_{mn} \in L^2_{S,D}} (1 - P_{\text{drop}}(l_{mn}))] \end{aligned} \tag{1.6}$$

We now argue that PDP of two node-disjoint links have low correlation. In a wireless ad hoc network, congestion, contention, time-varying wireless channel, and mobility of nodes are four main factors contributing for packet loss. We argue that PDP due to each of the above factors is little correlated, thus PDP of two node-disjoint links is little correlated. First, packet drop due to mobility of two node-disjoint links is independent of each other, assuming nodes' movement is independent of each other. Second, PDP due to contention or wireless channel error is generally small, because of the 802.11 MAC layer retransmission scheme. Thus we do not need to consider their contributions here. Third, as for congestion, even though two node-disjoint links may interfere with each other, causing their PDP to be correlated, we expect that the random backoff scheme in the 802.11 MAC layer protocol reduces the correlation significantly. We have applied NS simulations to verify our conjecture [44]. Specifically, our results show that packet drop over two node-disjoint interfering links have low correlation, as long as PDP of each link is small [44].

1.3.4 Computation of PDP over a link

In order to complete the computation of the concurrent PDP of two node-disjoint paths, we now show how to estimate PDP over one link, assuming that we have already estimated the flow rates F_i over each link l_i. As stated earlier, in a wireless ad hoc network, congestion, contention, time-varying wireless channel errors, and mobility of nodes are four main reasons for packets loss. Thus PDP over link l_{ij} can be represented as

$$P_{\text{drop}}(l_{ij}) = 1-[1-P_{\text{drop-cong}}(l_{ij})][1-P_{\text{drop-cont}}(l_{ij})][1-P_{\text{drop-chan}}(l_{ij})][1-P_{\text{drop-mob}}(l_{ij})] \tag{1.7}$$

where $P_{\text{drop-cong}}(l_{ij})$, $P_{\text{drop-cont}}(l_{ij})$, $P_{\text{drop-chan}}(l_{ij})$, and $P_{\text{drop-mob}}(l_{ij})$ are packet drop over link l_{ij} due to congestion, contention, wireless channel error, and mobility respectively. It is possible to apply the broadcast packet technique described by De Couto et al. [31] to estimate PDP due to contention and wireless channel error, and apply results on link availability [32] to estimate the PDP over a link due to mobility. In this chapter, we only focus on PDP due to congestion, since we assume (a) static scenarios, and (b) packet loss caused by channel error and contention is mostly recovered by 802.11 MAC layer retransmissions.

In the remainder of this section, we describe how to compute PDP over link l_{ij} due to congestion $P_{\text{drop-cong}}(l_{ij})$. An *interfering link set* of link l_{ij} is defined to be a set consisting of all links that interfere with it. We partition the interfering link set $I(l_{ij})$ into several disjoint subsets, such that each subset is an independent set. An *independent set* denoted by IS is defined to be a set of links, which can transmit successfully simultaneously without interfering with each other. The set of independent sets resulting from partitioning $I(l_{ij})$ is denoted by $PT(l_{ij})$. We define equivalent rate of flows over all links in the k^{th} independent set IS_k as follows:

$$CF_k = \max_{l_m \in IS_k} F_m \tag{1.8}$$

where F_m is the aggregate incoming flow rate over the m^{th} link l_m in the k^{th} independent set IS_k. Since links of the same independent set can transmit simultaneously, the equivalent rate of an independent set denotes link l_{ij}'s channel resource needed by all the links in that independent set per unit of time.

Given a partition of the set $I(l_{ij})$, we could estimate the PDP due to congestion of link l_{ij} as follows:

$$P_{\text{drop-cong}}(l_{ij}|PT(l_{ij})) \approx \max(1 - \frac{C}{\sum_{IS_k \in PT(l_{ij})} CF_k}, 0) \approx P_{\text{drop}}(l_{ij}) \tag{1.9}$$

where C is wireless channel capacity. the last equality reflects our assumption that congestion is the main reason of packet drop.

We name the partition $PT(l_{ij})^*$ that minimizes $P_{\text{drop-cong}}(l_{ij}|PT(l_{ij}))$ *the most efficient partition.* Since computing the actual PDP due to congestion is prohibitively compute intensive, we choose to use its lower bound instead, i.e. the PDP of the *most efficient partition*, as a metric in comparing PDP of two links, and subsequently two paths. We note that using the most efficient partition results in underestimating the PDP due to congestion, and the total PDP. However simulations show that it is sufficient to use the lower bound of PDP due to congestion to compare and select paths. Also with the development of more efficient MAC layer protocol in the future, our underestimation is likely to approach the actual results.

We propose a greedy algorithm to approximately find the most efficient partition. The basic idea behind the greedy partitioning algorithm is to combine as many links with large flow rates together as possible to reduce the sum of equivalent flow rates of independent sets, thus minimizing $P_{\text{drop-cong}}(l_{ij}|PT(l_{ij}))$ [44].

Combining Equations (1.6), (1.7), and (1.9), we obtain an estimate of PDP of two node disjoint paths.

One approach to solve the optimal multipath selection problem described in Section 1.3.2 is to enumerate all possible pairs of node-disjoint paths from a source N_S to a destination N_D, estimate the concurrent PDP for each path pair using the scheme proposed in this section, and choose the best one. We refer to this solution as the Optimal Multipath Routing (OMR). Unfortunately the computation complexity of the OMR grows exponentially with the size of the network; thus it can not be run in real time. However, as will be seen shortly, OMR can be used in non-realtime simulations to provide an upper bound on the performance of other lower complexity heuristic schemes.

1.3.5 A Heuristic solution to the optimum multipath selection

Since the optimal multipath selection problem is NP-hard, we propose a heuristic solution, called Interference aWare Multipath Routing (IWM), which can be implemented in real time. By assuming that the PDP of each link is small, we can approximate $P_{\text{drop}}(P^1_{S,D}; P^2_{S,D})$ in Equation (1.6) as follows:

$$P_{\text{drop}}(P^1_{S,D}; P^2_{S,D}) = \sum_{l_{ij} \in L^1_{S,D}} P_{\text{drop}}(l_{ij}) \cdot \sum_{l_{mn} \in L^2_{S,D}} P_{\text{drop}}(l_{mn}) \tag{1.10}$$

Our approach is to first determine the first path so as to minimize PDP, and then to choose to minimize the second path's PDP among paths node-disjoint from the first one. Note that this approach is similar to the one proposed in [13]. The main difference is that our metric is PDP and theirs is correlation. Specifically, we apply the techniques described in Section 1.3.4 to compute PDP for each link in wireless ad hoc networks.

The optimization problem of finding the first path can be formulated as follows.

$$\underset{\mathbf{x}}{\text{Minimize}} \sum_{l_{ij} \in E} x_{ij} P_{\text{drop}}(l_{ij})$$

such that the flow constraint in Equation (1.1) is satisfied. $P_{\text{drop}}(l_{ij})$ denotes the cost assigned to link l_{ij}, and is estimated using Equations (1.7) and (1.9) presented in Section 1.3.4. To obtain the first path, we solve this optimization problem using the Dijkstra's algorithm, whose complexity is polynomial [30].

After obtaining the first path, we first update flowrate over each link, by taking into account the allocated video flow into corresponding links. Given the first path, we compute the second path, by defining a link cost for each link as follows:

$$C_{mn} = P_{\text{drop}}(l_{mn}) + \text{nd\_cost}_{mn} \tag{1.11}$$

where

$$\text{nd\_cost}_{mn} = \begin{cases} b_1 \gg 1 & \text{destination node of link } l_{mn} \in P^1_{S,D} \\ 0 & \text{otherwise} \end{cases}$$

is a penalty factor to maintain the node-disjointness between the two paths. The optimization problem to find the second path minimizing PDP and node-disjoint from the first path can be formulated as follows:

$$\underset{\mathbf{y}}{\text{Minimize}} \sum_{l_{mn} \in E} y_{mn} C_{mn}$$

subject to the constraint, the indicator vector for the second path satisfies (1.3). We also solve the second optimization problem with the Dijkstra's algorithm.

1.3.6 Simulation results

In this Section, we compare the optimal multipath routing (OMR) as described in Sections 1.3.2 and 1.3.4, IWM as described in section 1.3.5, the node-disjoint multipath routing (NDM) [15], and the shortest widest path routing (SWP) [33]. We use a simulation model based on NS-2[34], and focus on the case of static wireless ad hoc networks. Each node's radio range is 250 meters, and its interference range is 550 meters. We consider a grid network consisting of 49 nodes, placed in a 7×7 grid with the distance between neighboring nodes being 200 meters.

We randomly choose one video sender and one video receiver. For MDC we encode one frame into two packets, and the Group Of Pictures (GOP) size is chosen to be 15. Standard MPEG QCIF sequence Foreman is coded with a matching pursuit multiple description codec[35] at 121.7 kbps. We insert 20 one-hop cross traffic flows, whose bit rates are uniformly distributed in the range [0,200.0] kbps. The bit rates of cross flows are changed every 30 seconds. We run 30 simulations for different network topologies, and select different sender and receiver pair in each scenario. Each simulation lasts 900 seconds.

We evaluate the performance using the following metrics:

a. **The ratio of bad frames:** The ratio of bad frames is the ratio of the number of non-decodable frames to the total number of video frames that should have been decoded in the receiver. A description of an I-frame is non-decodable, if the packet corresponding to the description is not received on time. A description of a P-frame is non-decodable, if at the playback deadline, either the packet corresponding to the description is not received or the same description of the previous frame is non-decodable. A frame of a MDC stream is non-decodable, if both of its two descriptions are non-decodable. This metric takes into account the dependency between consecutive frames in a predictive coding scheme, and also reflects the fact that MDC can, to some extent, conceal the undesirable effects caused by missing packets.

b. **The number of bad periods:** A bad period consists of contiguous bad frames. This metric reflects the number of times that received video is interrupted by the bad frames.

Figures 1.1(a) and 1.1(b) show the ratio of bad frames and the number of bad periods of the four schemes for 30 runs with NS2 simulations. As seen, the average performance of IWM is very close to that of OMR, and is significantly better than that of NDM and SWP, even though its computational complexity is similar to NDM and SWP. Specifically, shown in Table 1.1, IWM has the lowest ratio of bad frames among all protocols in 26 out of 30 runs. The results show that the relaxation of the optimal multipath selection problem used by IWM is very efficient.

Table 1.1: Summary: the ratio of bad frames

	OMR	IWM	NDM	SWP
Average	0.0655	0.0685	0.1864	0.1755
Num. of Best	29	26	7	8

1.4 Testbed Implementation and Evaluation

To demonstrate the feasibility of the proposed multipath selection framework and the IWM, we have built a small wireless ad hoc network testbed, consisting of 11 desktops and laptops. In this section, we summarize the key components of the testbed, and report the results obtained from the performance study conducted on it.

1.4.1 Software Architecture

We have implemented the proposed IWM protocol in the Mesh Connectivity Layer (MCL), which is an ad hoc routing framework provided by Microsoft Research [37]. MCL implements a virtual network adapter, i.e. an interposition layer between layer 2 (the link layer) and layer 3 (the network layer). The original MCL maintains a link cache in each node to store loss rate and bandwidth information of each link. Also the original MCL implements a routing protocol named Link Quality Source Routing (LQSR) to route packets. LQSR is close to SWP as described in Section 1.3.6. The LQSR supports different link-quality metrics, e.g. Weighted Cumulative Expected Transmission Time (WCETT) and Expected Transmission Count (ETX) [37]. In our experiments, LQSR uses WCETT as the link-quality metric.

It may be argued that applying LQSR with WCETT twice, could result in two node-disjoint paths with similar performacne to IWM. However, LQSR attempts to obtain the path with the largest bandwidth, rather than the one with largest *available* bandwidth. Unlike IWM, LQSR does not take into account the impact of interference from cross traffic flows and the video flow itself on the path selection. The two paths resulting from LQSR are likely to be close to each other, because the metrics for different paths are computed with the same network parameters. Rather, IWM considers from both cross traffice flows and video flow in order to compute PDP. As such, the two paths obtained by IWM adapt to availalble bandwidth resources. When there is sufficient bandwidth, the two paths obtained by IWM are likely to be close to each other, otherwise the two paths are distributed within different regions of network to minimize PDP.

We have made two major modifications to MCL. First we implement IWM inside the MCL framework such that it coexists with the LQSR in MCL. When forwarding a packet, the MCL uses one bit of information transmitted from the upper layer to decide which routing protocol to use. If the packet is high priority video packet, MCL uses IWM to route it, otherwise, it still uses LQSR. This way, we can run IWM and LQSR simultaneously in the network, and compare them under same network conditions. In our experiments, IWM is used to route MDC packets and LQSR is used to route SDC packets[1]. The second modification we have made is to enable the estimation of flow rate of each link in order to compute the PDP using the scheme described in Section 1.3.2.

We have also implemented both MDC and SDC streaming protocol in the application layer. In the streaming protocol, we have implemented timestamping, sequence numbering, feedback functions and the rate control scheme to be described in the next section. UDP sockets are used at the transport layer. The deadline of each frame is 2 seconds after the transmission time. If a packet is received after its deadline, it is discarded.

1.4.2 A Simple Rate Control Scheme

In our multipath selection framework, we assume that there exists a simple rate control scheme to determine the video application's sending rate. This way the sending rate can be adjusted according to the amount of congestion in the network.

The basic idea behind our rate control scheme is to employ an Additive Increase Multiplicative Decrease (AIMD) algorithm, which is the default congestion control mechanism used in TCP today. The receiver transmits a feedback packet to the sender periodically, in order to inform the sender whether the network is congested. Since PDP due to contention

[1]Recall that SDC rate is about 30% lower than that of MDC video due to compression inefficiency of MDC.

and wireless channel error is generally small, due to 802.11 MAC layer retransmissions, the scheme uses lost packets as a signal to detect congestion. The receiver detects lost packets using sequence numbers carried by each packet. In order to alleviate the out-of-order problem caused by multipath transmission, the receiver counts packets received from each path separately.

At the sender side, after receiving the feedback packet, if the network is not congested, the sender increases the video transmission rate by 1 fps in each time period. If the network is congested, the sender decreases the video frame rate immediately by half. If the sender has not received one feedback packet in a time interval twice of the feedback period, this triggers a timeout event, and the sender reduces the video transmission rate to the minimum transmission rate.

For simplicity, we change the transmission bit rate through changing the number of transmitted video frames per unit time without even dropping a frame. This has the effect of changing the playback duration of a given chip at the receiver. Our motivation for doing so is purely ease of implementation. This way, we do not have to implement fine grain or temporal scalability in order to compute our metrics, such as ratio of bad frames or bad periods. For a fixed GOP, this method results in the same metrics as modifying the encoding and decoding rate on-the-fly, i.e. applying temporal scalability. For example, assuming GOP of 15, if frame #4 is non-decodable, the number of bad frames for both methods is 12.

1.4.3 Testbed Setup

We deploy an 11-node wireless ad hoc network testbed on the third floor of Cory Hall, the office building of EECS, University of California at Berkeley. The nodes are placed in offices and in the aisles, which are separated from each other with floor-to-ceiling walls and solid wood doors.

Each node in the testbed is either a standard desktop or laptop running Windows XP. Each desktop is equipped with either a Linksys 802.11 a/b/g PCI card or a Netgear 802.11 a/b/g PCI card. Similarly, each laptop is equipped with either a Linksys 802.11 a/b/g

PCMCIA card or a Netgear 802.11 a/b/g PCMCIA card. All cards operate in the ad hoc mode.

All of our experiments are conducted over IPv4 using statically assigned addresses. Except for configuring ad hoc mode and fixing the frequency band and channel number, we use the default configuration for the radios. The cards all perform autorate selection.

1.4.4 802.11a wireless ad hoc network result: static nodes

We have performed a series of tests in 802.11a wireless ad hoc networks. We have carried out ten 360 second long experiments with varying cross traffic level. The maximum throughput of each link is 54 Mbps. The senders and receivers are the same as those of the previous experiments. In runs 1 through run 8, there are two one hop UDP cross traffic, whose bit rate is changed every 30 seconds based on uniform distribution. In runs 9 and 10, the cross traffic is one two-hop TCP connection.

Figures 1.2(a) and 1.2(b) show the result of the ratio of bad frames and the number of bad periods of all ten runs. The horizontal axis shows the average bit rate of combined cross traffic. As seen, IWM/MDC significantly outperforms LQSR/SDC in nine out of ten runs, and the performance gap with IWM/MDC and LQSR/SDC increases as cross traffic increases. Once again, this shows the advantage of path diversity with MDC over single path transmission of SDC video.

Figure 1.3 compares PSNR of two schemes for all ten runs. On average, IWM/MDC outperforms LQSR/SDC by 1.1 dB, and in eight out of ten runs. The reason for slightly worse performance in runs 2 and 3 is low packet loss rate for both schemes in these runs. As a result, the PSNR of received video in these runs are close to the PSNRs of original MDC and SDC videos respectively. The PSNR of encoded MDC is slightly lower than that of encoded SDC, because in practice it is very hard to make two video flows achieve the exact same PSNRs.In general, we would expect performance gain of IWM/MDC over LQSR/SDC to become wider as packet drop probability increases, which is also in agreement with the results in Figure 1.2.

We plot PSNR, loss traces and frame rate traces of run 7, i.e. the first run with cross traffic 8000 kbps, using LQSR/SDC and IWM/MDC in Figures 1.5 and 1.4 respectively. IWM/MDC outperforms LQSR/SDC by 1.1 dB in run 7. As seen in Figure 1.4(a), for IWM/MDC, PSNR drops gracefully, when there is packet loss in only one substream. As seen in Figure 1.4(b), most of the time, packet losses of two substreams do not overlap, thus reducing both the number and the amount of PSNR drops. The PSNR curve of LQSR/SDC shown in Figure 1.5(a) has more frequent and severe drops than that of IWM/MDC; this is because PSNR drops for every packet drop in SDC video, and would drop severely when there is a burst of packet loss. As seen in Figure 1.4(e), our simple rate/frame control scheme adjusts the video rate promptly, whenever there is packet drop in any path, and keeps the maximum sending rate, whenever there is no packet drop.

1.4.5 802.11a wireless ad hoc network result: moving nodes

We also carried out experiments with one moving node in 802.11a wireless ad hoc networks. In these experiments, we do not take into account PDP due to mobility even though the nodes are slowly moving. During the experiment, we randomly select one laptop, move it to a random position, and repeat the process. The senders and receivers are the same as those of previous experiments. At any time, there is always one laptop moving. Figures 1.6 and 1.7 show the results of three 600 seconds experimental run.

As seen in Figure 1.6, the ratio of bad frames and the number of bad periods are both greatly reduced for IWM/MDC in all three runs. With the continuous movement of one node, one path is broken from time to time. If the path selected by LQSR is broken during the video transmission, the SDC receiver suffers from packet loss and interruption of video playback. In contrast, even if one path selected by IWM is broken, the received video quality is still acceptable. Figure 1.7 compares PSNR of two schemes. Averaged over three runs, IWM outperforms LQSR by 2.1 dBs.

1.5 Multiple tree multicast video communication over wireless ad hoc networks

Our proposed multiple tree multicast video communication system consists of two parts: a multicast routing protocol to construct multiple trees, and a simple scheme to distribute video packets into different trees. For the latter part, we employ MDC video to form multiple video streams, and transmit different video streams through different trees. In this section, we assume that the network is lightly loaded, i.e. mobility and poor channel condition rather than congestion are major reasons for packet drop. In this case, multiple tree multicast with MDC effectively alleviates undesirable effects caused by packet drop due to mobility and poor channels.

In this section, we begin by showing the feasibility of multiple tree multicast, and then move on to describe ways to forward packets through multiple trees. We describe the proposed multiple tree protocols in detail in Sections 1.6 and 1.7. Without loss of generality, we limit our discussion to the case of two trees, with each tree carrying one description.

1.5.1 Tree Connectivity and Tree Similarity

In order to measure the tree construction capability of multicast routing protocols, we define tree connectivity level P as follows [38]:

$$P \triangleq \frac{E[N]}{M} \tag{1.12}$$

where M is the product of the total number of receivers and the number of trees, $N = \sum_{i=1}^{m} n_i$, with n_i denoting the number of trees that receiver i connects to, and m denoting the number of receivers. It can be shown that in general $0 \leq p \leq 1$. Given a random topology with n nodes, one random sender and m random receivers, N is the sum of all receivers connected to each multicast tree, and $E[N]$ is the expected value of N over all topologies. Tree connectivity is a measure of the tree construction ability of a multicast routing protocol. Obviously, it is desirable to design tree construction scheme with as high tree connectivity level, P, as possible.

To measure the level of disjointness of two trees, we define tree similarity, S, between two trees as the ratio of the number of shared nodes to the number of middle nodes of the tree with a smaller number of middle nodes. Tree similarity between two disjoint trees is zero, and between two identical trees is one. The lower tree similarity between two trees, the lower correlated packet drop across two trees, and hence, the more effective multiple tree video multicasting is in achieving high video quality.

Ideally, we desire a multicast routing protocol to achieve both a high tree connectivity level and a low tree similarity level in video applications space. Intuitively, if the node density is low, it is difficult to construct disjoint trees that connect to all m nodes, and hence either tree connectivity has to be low or tree similarity has to be high. On the other hand, for sufficiently high node density or sufficiently large radio range, we would expect a routing protocol to be able to achieve both a high tree connectivity level and a low tree similarity level.

Thus, an important issue in multiple tree video multicasting is whether or not the required node density to obtain disjoint trees with high connectivity is too high to make it feasible in practice. We have developed the following theorem to address this issue for tree connectivity of two disjoint trees. Before stating the theorem, we need to introduce the term critical density λ_c. Dousse et. al. [39] have stated that there exists one critical density λ_c for a wireless ad hoc network, such that if the density $\lambda < \lambda_c$, all the connected clusters are almost surely bounded; otherwise, almost surely there exists one unique unbounded super connected cluster.

Theorem 1: Consider an infinite wireless network, with nodes assumed to be distributed according to two-dimensional poisson process. Let D_1 denote the required node density to achieve a given tree connectivity level, P, in a single tree case. If $D_1 > \lambda_c$, there exists at least one double disjoint tree whose required node density D_2 to achieve P satisfies

$$D_2 - \frac{\ln(\pi D_2 r^2 + 1)}{\pi r^2} \leq D_1 \leq D_2 \tag{1.13}$$

where r is the radio link range.

The detailed proof is included in [38]. We can see from the above theorem that the difference between D_1 and D_2 is only a logarithm factor of D_2, which is small compared to the value of D_1 and D_2. The difference is negligible as $D_1, D_2 \to \infty$, which are requirements for keeping the network connected as the number of total nodes $n \to \infty$ [40][41]. Thus we conclude that the required density for double disjoint tree schemes is not significantly larger than that of single tree schemes, and that tree diversity is a feasible technique to improve the robustness of multicast video transmission over wireless ad hoc networks.

1.5.2 Multiple Tree Multicast Packet Forwarding

Our approach is to transmit different descriptions of MDC video flow through different trees simultaneously. If packet drop over two trees are not correlated, when some packets in one tree do not arrive at the destination on time, the receiver continues to decode and display packets corresponding to the other description on the other tree, resulting in acceptable video quality without interruption [42].

Our proposed multiple tree multicast packet forwarding works as follows. The application layer protocol sets a tree-flag in each packet's header to determine the tree to which the packet should be forwarded. The multiple tree multicast protocol forwards the packet in different trees according to the tree-flag as follows: when a node receives a data packet, it checks the node's *Forwarding Table* for the forwarding status and *Message Cache* to avoid forwarding duplicate data packet. The node forwards a non-duplicate packet forwarded in tree-y, if it is a forwarder for tree-y. Each packet flows along the corresponding tree from the sender to the receivers, but is not constrained to follow pre-set branches in the tree, as in the *tree flood* approach [24] or the *forwarding group flooding* approach [22]. Thus our packet forwarding scheme utilizes the broadcast nature of wireless ad hoc networks to obtain extra diversity gain without using extra network resources. Our packet forwarding scheme does not support packet forwarding across the trees, since nodes in one tree are unaware of the status of nodes in the other tree.

1.6 Serial Multiple Disjoint Trees Multicast Routing Protocol (Serial MDTMR)

Due to the nature of MDC, the less correlated packet drop between two trees, the more robust the video multicast. We assume that the network is lightly loaded, i.e. mobility and poor channel conditions rather than congestion are major causes of packet drop. In this case, if two trees do not share any middle nodes, packet drop over two trees are independent. Thus our main objective in the design of Serial MDTMR is to construct two node-disjoint multicast trees.

The proposed serial MDTMR constructs two node-disjoint trees in a distributed way. First we build a shortest path multicast tree. Then after requiring all the middle nodes in the first tree not to be middle nodes of the second tree, we construct another shortest path tree. Since these two trees do not share middle nodes at all, they are node disjoint. Since Serial MDTMR is a way of constructing two disjoint multicast trees, it can be easily applied on top of any suitable single tree multicast routing protocol. Without loss of generality, we design the detailed Serial MDTMR based on ODMRP [22], since ODMRP has been demonstrated to perform well and is well known [23]. By comparing Serial MDTMR and ODMRP, it is easy to quantify the performance gain obtained by the multiple tree multicast routing. We can also design detailed Serial MDTMR based on other multicast routing protocols [14-20], taking advantage of their individual strengths. For example, we could apply a *local repair* scheme similar to [24] to maintain the tree structure with less control overhead. When a middle node or receiver detects that it is disconnected from the corresponding multicast forwarding tree tree-x, where x is 0 or 1, it initiates a *local repair* process for tree-x, which searches the neighborhood of the middle node or receiver in order to find a new upstream node to reconnect the middle node or receiver to tree-x. To keep the disjointness between two trees, the middle node or receiver only selects a node, which is not a forwarding node for tree-(1-x), as its new upstream node.

Similar to ODMRP, group membership and multicast trees in Serial MDTMR are established and updated by the source on demand. When a multicast source has packets to send, it periodically triggers a two step multicast tree construction/refresh process. In the first step, the multicast source broadcasts to the entire network a JOIN REQUEST message, which includes the tree ID. When a node receives a non-duplicate JOIN REQUEST message for the first tree, it stores the upstream node ID, and rebroadcasts the packet. When the JOIN REQUEST message reaches a multicast receiver, the receiver unicasts a JOIN ACK message to the multicast source via the reverse shortest path. When a middle node in the reverse path receives a non-duplicate JOIN ACK message, it updates its corresponding forwarding state in the Forwarding Table, and forwards the message to its upstream node. Each middle node of the tree only forwards the JOIN ACK message once in one tree construction cycle.

After receiving the first JOIN ACK message, the multicast source waits for a short time period before broadcasting another round of JOIN REQUEST message for the second tree in order to ensure the disjointness of two trees. When a node receives a non-duplicate JOIN REQUEST message, it forwards the packet only if it is not a middle node of the first tree in this round. When the JOIN REQUEST message reaches a receiver, the receiver unicasts back a JOIN ACK message to the multicast source to set up the second tree.

We compare tree connectivity of single shortest path tree and Serial MDTMR as a function of node density through simulations, as shown in Figure 1.8. Note that for Serial MDTMR, tree connectivity is averaged across two trees. The total number of nodes is 1000, with 50 receivers. The nodes are randomly distributed according to a two-dimensional poisson process. The results are averaged over 5000 runs. As seen, there is only a small performance gap between the two schemes when node density is larger than 7 nodes per neighborhood. For example, when node density is 8.2 nodes per neighborhood, tree connectivity of a single tree scheme and Serial MDTMR is around 0.99 and 0.95 respectively.

1.7 Parallel Multiple Nearly-Disjoint Trees Multicast Routing Protocol (Parallel MNTMR)

Serial MDTMR achieves reasonable tree connectivity, while maintaining the disjointness of two trees. However its routing overhead and construction delay are potentially twice as much as that of a parallel scheme that would build two trees simultaneously. In this section, we propose a novel parallel double tree scheme, named Parallel MNTMR, to overcome the above disadvantages of Serial MDTMR. We have three main design goals for the Parallel MNTMR:

- *Low routing overhead and construction delay*: The routing overhead and construction delay of Parallel MNTMR should be similar to that of a typical single tree multicast protocol.

- *High tree connectivity*: if a receiver is connected to the sender, it should be able to connect to both trees.

- *Near disjointness*: The ratio of the number of shared nodes of two trees to the number of nodes of the smaller tree should be minimized.

During multicast operation, the application layer protocol sets a tree-flag in each packet's header to determine to which tree the packet should be forwarded. The multiple tree multicast protocol forwards the packet in different trees according to the tree-flag. We also apply the *tree flood* approach [24] to achieve extra diversity gain without consuming extra network resources.

1.7.1 Overview of Parallel MNTMR

In a general single-tree multicast protocol, e.g. ODMRP [22], when a multicast source has packets to send, it triggers a multicast tree construction process by flooding a Join-Query (JQ) message to the network. Upon receiving the JQ message, each receiver unicasts back a Join-Reply (JR) message to the sender to construct the multicast tree. In Parallel MNTMR, we apply similar JQ and JR processes to construct two nearly-disjoint trees simultaneously.

The basic idea behind parallel tree construction is to first classify all the nodes randomly into one of two categories: group 0 or group 1. We define a *pure JQ message* as a JQ message whose route only consists of nodes in the same group, and a *mixed JQ message* as a JQ message whose route consists of nodes in both groups. The protocol uses a technique based on delay timer to select and forward *pure JQ messages* with a priority over *mixed JQ messages.* As will be seen shortly, this improves the disjointness of the constructed trees in the JR process.

We also propose an *upstream node selection rule* so that nodes close to each other tend to select the same upstream node for the same tree, thereby avoiding nodes of the other tree. This rule improves the disjointness of two trees, and forwarding efficiency of the multicast protocol.

1.7.2 Conditions and Rules

In order to construct two trees with both high tree connectivity and low tree similarity, Parallel MNTMR applies the following conditions and rules at each node to control the flow of JQ and JR messages. Without loss of generality, we assume the current node a is in group x, where x is 0 or 1. For brevity, we call a JOIN-QUERY message with a group-x node as the last hop, a group-x JQ message.

- *JQ message storing condition*: In order to obtain two loop-free trees in the JR process, each node only stores JQ messages satisfying the *storing condition* into its *JQ Message Cache*. A JQ message satisfies the *storing condition*, either if it is the first received JQ message, or if the following two conditions are satisfied: (a) the number of hops it travelled is no larger than that of the first received JQ message at node a plus one, and (b) the JQ message has not been forwarded by node a.

- *JQ message forwarding condition*: A JQ message satisfies the *forwarding condition*, if the following two conditions hold true: (a) node a has not forwarded a JQ message in this JOIN-QUERY round, and (b) the message's last hop is the sender or a group-x node. The *forwarding condition* results in *pure group-x JQ messages* to be selected and forwarded with a priority over *mixed JQ messages*, thus helping the protocol to construct trees that are as disjoint as possible.

- *Upstream node selection rule*: The objective of the *upstream node selection rule* is to maximize the disjointness of two trees. Let JQM_a denote the set of all the messages in the *JQ Message Cache* of node a. If there exist both group-0 and group-1 JQ messages in JQM_a, node a selects last hops of the earliest received group-0 and group-1 JQ messages as upstream nodes for tree-0 and tree-1 respectively. Otherwise, we assume all the JQ messages in JQM_a are group-y JQ messages. In this case, if $|JQM_a| > 1$, node a selects last hops of the earliest and the second earliest received JQ messages as upstream nodes for tree-y and tree-$(1-y)$ respectively; otherwise if there is only one message in JQM_a, the last hop of the only JQ message is selected as upstream nodes for both tree-0 and tree-1.

1.7.3 Detailed Double Nearly-Disjoint Tree Construction

When a multicast source has packets to send, it triggers a multicast tree construction process by broadcasting a Join-Query (JQ) message to its neighbors. When a node receives a group-y JQ message, if the message satisfies the *storing condition*, the node stores it into the *JQ Message Cache* for later usage in the JR process, otherwise the message is simply discarded. If the message also satisfies the *forwarding condition*, the current node forwards the JQ message to its neighbors immediately; otherwise if the JQ message is the earliest received JQ message in the current Join Query round, the node sets a JQ-delay timer. When the JQ-delay timer expires, if the node has not forwarded a JQ message in this JQ round, it forwards the earliest received JQ message at that time. The JQ-delay scheme encourages *pure JQ messages* to be selected and forwarded with a priority over *mixed JQ messages* in the distributed tree construction process.

When a receiver receives a group-y JQ Message, if the message is a *pure JQ message*, and the node has not initiated a JOIN-REPLY (JR) message in this JQ round for tree-y, it selects the last hop of this JQ message as its upstream node for tree-y, and unicasts a JR message to the sender via the selected upstream node. All nodes, receiving and forwarding the JR message for tree-y, become middle nodes of tree-y. The receiver also sets a timer upon receiving the earliest JQ message. When the timer expires, for each tree for which it has not already initiated a JR message, the receiver selects an upstream node according to the *upstream node selection rule* and unicasts a JR message to the sender via the selected upstream node to construct that tree. In the end, we obtain one tree mainly consisting of group-0 nodes and another mainly consisting of group-1 nodes.

1.7.4 Discussion

In this section, we argue that Parallel MNTMR achieves our three design goals. Firstly, the Parallel MNTMR builds two trees simultaneously, and each node forwards the JQ message at most once in one JOIN-QUERY round. Therefore the routing overhead and the construction delay is similar to that of a typical single tree multicast routing protocol. Secondly, as long as a receiver is connected to the sender, the protocol requires it to send JR messages for both trees; therefore the tree connectivity is the same as that of a single tree protocol. Thirdly, regarding the disjointness of the two trees constructed by MNTMR, we have the following claim:

Claim 2: Given any two nodes N_a and N_b, which are middle nodes for tree-0 and tree-1 respectively, let JQ_a and JQ_b denote node sets of last hops of JQ messages stored in the JQ Message Caches of nodes N_a and N_b respectively. We sort nodes in JQ_a and JQ_b according to the arrival time of corresponding JQ messages. Let nodes N_c and N_d denote upstream nodes obtained by the Parallel MNTMR of nodes N_a and N_b respectively. We have $N_c \neq N_d$, if the first two nodes of JQ_a and JQ_b are the same.

The proof is shown in Section 1.10.

Intuitively, *claim 2* shows that if two nodes in different trees share the same first two JQ messages in their JQ message caches, they will not select the same node as their upstream nodes. Thus for many scenarios, the Parallel MNTMR is likely to maintain disjointness between two trees.

1.7.5 Simulation results

We use a simulation model based on NS-2[34]. We only consider the continuous mobility case with zero pause time, and vary the maximum speed from 2.5 m/s to 15 m/s. In each run, we simulate a 50 node wireless ad hoc network within a 1500 $\times$ 300 square meters area. Each simulation is 900 seconds long, and results are averaged over 30 runs.

We randomly choose one sender and eight receivers. For MDC we encode one frame into two packets, while for Single Description Coding (SDC) we encode one frame into one packet. We set the frame rate as 8 fps, and GOP size as 15. For fairness, we set the Peak Signal to Noise Ratio (PSNR) of MDC and SDC to be approximately the same, i.e. 33 dB. To achieve similar quality, standard MPEG QCIF sequence Foreman is coded with a Matching Prusuit Multiple Description Video Codec called MP-MDVC [35] at 64.9 kbps for MDC, and with Matching Pursuit Codec [45] at 41.2 kbps for SDC sequence. The playback deadline of each packet is set to 150 milliseconds (ms) after it is generated.

We evaluate the performance using the following metrics:

a. **The ratio of bad frames:** In multicast scenario, the ratio of bad frames is the ratio of the total number of non-decodable frames to the total number of frames that should have been decoded in all the receivers.

b. **The number of bad periods:**

c. **Normalized packet overhead:** The total number of control packets transmitted by any node in the network, divided by the total number of video frames received by all the receivers.

d. **Forwarding efficiency:** The total number of data packets transmitted by any node in the network, divided by the total number of packets received by all the receivers.

We compare the following four schemes:

- Multiple tree multicast with Parallel MNTMR and MDC;

- Multiple tree multicast with Serial MDTMR [42] and MDC;
- Single tree multicast with ODMRP [22] and MDC;
- Single tree multicast with ODMRP [22] and SDC.

For fair comparison, all of three multicast routing protocols use 3 seconds for the JOIN REQUEST flooding interval, and use 4.5 seconds as a forwarding state lifetime.

Figures 1.9(a) and 1.9(b) show the result of the ratio of bad frames and the number of bad periods of the four schemes respectively. As expected, both the number of bad frames and the number of bad periods increase with maximum speed. As seen, performance of multiple tree multicast with Parallel MNTMR is close to Serial MDTMR, and they both perform much better than the other two schemes with ODMRP. Shown in Figure 1.10, two trees obtained by Parallel MNTMR only share approximately eight percent of nodes, which means they are nearly disjoint. This explains the reason the two multiple tree protocols perform similarly. The combination of our proposed multiple tree multicast protocols, e.g. Parallel MNTMR or Serial MDTMR, and MDC reduces contiguous packet loss caused by broken links of multicast tree, since links of two nearly-disjoint trees fail nearly independent, resulting in much better received video performance than that with ODMRP and MDC. By comparing ODMRP with MDC and ODMRP with SDC respectively, we conclude MDC by itself could also reduce scattered packet loss caused by wireless channel error, or packets collision, thus reducing both the ratio of bad frames and the number of bad periods.

Figure 1.11(a) shows the normalized control packets for the four schemes. Simulation results show that the number of normalized control packets of Parallel MNTMR is very similar to that of ODMRP, and is about 50 percent lower than that of Serial MDTMR. Figure 1.11(b) shows that the number of the normalized forwarded data packets is almost the same for all four schemes with Parallel MNTMR being slightly worse. This indicates that the performance gain of Parallel MNTMR and Serial MDTMR is not at the expense of forwarding a packet more times than ODMRP, rather by the combined effect of independent trees and MDC.

We plot PSNR and loss traces of a randomly selected receiver using Parallel MNTMR with MDC and ODMRP with SDC in Figures 1.13 and 1.12 respectively. Every node moves randomly with a maximum speed 5.0 m/s. For MDC, it can be seen in Figure 1.13(a) that PSNR drops gracefully, when there is packet loss only in one substream. As seen in Figure 1.13, in Parallel MNTMR, most of the time, packet losses of two substreams do not overlap, thus reducing both the number and the amount of PSNR drops. The PSNR curve of ODMRP with SDC shown in Figure 1.12(a) has more frequent and severe drops than that of Parallel MNTMR with MDC; this is because PSNR drops for every packet drop in SDC video, and would drop severely when there is a burst of packet loss. We also visually examine the reconstructed video sequences under different schemes. For the video sequence obtained via Parallel MNTMR with MDC, we experience 6 short periods of distorted video in 900 seconds, while for the video sequence obtained via ODMRP with SDC, we experience 16 longer periods of more severely distorted video in the same time period.

1.8 Conclusion

In this chapter, we propose multipath unicast and multicast streaming in wireless ad hoc networks. In the unicast case, we have proposed a model to estimate the concurrent PDP of two paths by taking into account the interference between different links, and formulate an optimization problem in order to select two paths with minimum concurrent PDP. The solution to the optimal problem "OMR" is shown to be NP hard. Then we propose a heuristic routing protocol based on our path selection model, whose performance is shown to be close to that of the "optimal routing", and significantly better than that of existing schemes, through both NS simulations and actual experiments in a testbed.

In the multicast case, we have proposed multiple tree video multicast with MDC to provide robustness for video multicast applications. Specifically, we first propose a simple distributed protocol, Serial MDTMR, which builds two disjoint trees in a serial fashion. This scheme results in good tree connectivity while maintaining disjointness of two trees. In order to reduce the routing overhead and construction delay of Serial MDTMR, we further propose Parallel MNTMR, which constructs two nearly disjoint trees simultaneously in a distributed way. Simulation shows that video quality of multiple tree multicast video communication is significantly higher than that of single tree multicast video communication, with similar routing overhead and forwarding efficiency.

1.9 Appendix: Proof of *Claim 1*

Let A_i denote the event that $P^i_{S,D}$ does not drop packets, for $i = 1, 2$, and L_{ij} denote the event that link (i, j) does not drop packets.

$$\begin{aligned} P_{\text{drop}}(P^1_{S,D}; P^2_{S,D}) &= P(\overline{A_1 \bigcup A_2}) \\ &= P(\overline{(\bigcap_{(i,j)\in L^1_{S,D}} L_{ij}) \bigcup (\bigcap_{(m,n)\in L^2_{S,D}} L_{mn})}) \\ &= P(\bigcup_{(i,j)\in L^1_{S,D},(m,n)\in L^2_{S,D}} (\overline{L}_{ij} \bigcap \overline{L}_{mn})) \end{aligned} \tag{1.14}$$

By assuming that the PDP of each link is small, we can approximate $P_{\text{drop}}(P^1_{S,D}; P^2_{S,D})$ as follows:

$$P_{\text{drop}}(P^1_{S,D}; P^2_{S,D}) \approx \sum_{(i,j)\in L^1_{S,D}} \sum_{(m,n)\in L^2_{S,D}} P(\overline{L}_{ij}, \overline{L}_{mn}) \tag{1.15}$$

The simplified optimization problem, which uses Equation (1.15) as its metrics and keeps all the constraints of the original optimization problem in Equations (1.1-1.5), has been shown to be an Integer Quadratic Programming problem [13][46], which is a known NP-hard problem as proved in [47]. □

1.10 Appendix: Proof of *Claim 2*

We prove the claim through enumerating all possible scenarios. We list all the scenarios in Table 1.2.

Table 1.2: All scenarios of Claim 2

Scenario	Types of Nodes in JQ_a	Types of Nodes in JQ_b
1	All group 0 nodes	All group 0 nodes
2	All group 0 nodes	Both group 0 and 1 nodes
3	All group 1 nodes	All group 1 nodes
4	All group 1 nodes	Both group 0 and 1 nodes
5	Both group 0 and 1 nodes	All group 0 nodes
6	Both group 0 and 1 nodes	All group 1 nodes
7	Both group 0 and 1 nodes	Both group 0 and 1 nodes

Let message sets JQM_a and JQM_b denote JQ Message Caches of nodes N_a and N_b respectively. In scenarios 2, 6 and 7, according to the *upstream node selection rule*, N_c is the last hop of the first received group-0 JQ message in JQM_a, and N_d is the last hop of the first received group-1 JQ message in JQM_b. Therefore $N_c \neq N_d$.

In scenario 1, using the *upstream node selection rule*, N_c is the last hop of the first received group-0 JQ message, which is also the first received JQ message in JQM_a. N_d is the last hop of the second received JQ message in JQM_b. Since the first two JQ messages of JQM_a and JQM_b are the same, $N_c \neq N_d$. Similarly in scenario 3, we arrive at the same conclusion.

In scenario 4, N_c is the last hop of the second received JQ message. Since the first two JQ messages of JQM_a and JQM_b are the same, the first two JQ messages of JQM_b are group-1 JQ messages. Thus N_d is the last hop of the first group-1 JQ message, which is also the first JQ message. Therefore $N_c \neq N_d$. We could arrive at the same conclusion in scenario 5 in a similar fashion.

Therefore for all seven possible scenarios, $N_c \neq N_d$. □

References

[1] S. Mao, S. Lin, S. Panwar, and et al. "video transport over ad hoc networks: multistream coding with multipath transport", *IEEE Journal on Selected Areas in Communications*, Dec. 2003, pp. 1721 - 1737.

[2] H. Luo, S. Lu, and V. Bhargavan, "A New Model for Packet Scheduling in Multihop Wireless Networks", *ACM Mobicom*, 2000, pp. 76-86.

[3] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-hop Wireless Network Performance", *ACM MobiCom*, Sep. 2003.

[4] R. Gupta, J. Musacchio, and J. Walrand, "Sufficient Rate Constraints for QoS Flows in Ad-Hoc Networks", *under revision at Ad-Hoc Networks Journal*

[5] K. Obraczka, and G. Tsuduk. "Multicast routing issues in ad hoc networks", *IEEE International Conference on Universal Personal Communications*, 1998.

[6] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity", *Visual Communications and Image Processing (VCIP)* 2001.

[7] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming in content delivery networks", *IEEE INFOCOM*, 2002, pp. 1736-1745.

[8] S. Mao, S. Lin, S. Panwar, Y. Wang, and E. Celebi, "video transport over ad hoc networks: multistream coding with multipath transport", *IEEE Journal on Selected Areas in Communications*, Dec. 2003, pp. 1721 - 1737.

[9] W. Wei, and A. Zakhor, "Robust Multipath Source Routing Protocol (RMPSR) for Video Communication over Wireless Ad Hoc Networks", *International Conference on Multimedia and Expo (ICME) 2004*.

[10] J. Chakareski, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," *Multimedia Systems*, Springer, online journal publication, January 2005.

[11] A. Begen, Y. Altunbasak, and O. Ergun. "Multi-Path Selection for Multiple Description Encoded Video Streaming", *IEEE Internationl Conference on Communications (ICC)*, May 2003.

[12] S. Mao, Y. Hou, X. Cheng, H. Sherali, and S. Midkiff. "Multipath Routing for Multiple Description Video in Wireless Ad Hoc Networks", *IEEE Infocom*, March 2005.

[13] Z. Ma, H. Shao and C. Shen. "A New Multi-path Selection Scheme for Video Streaming on Overlay Netwoks", *IEEE Internationl Conference on Communications (ICC)*, June 2003.

[14] D B. Johnson, D A. Maltz, and Y. Hu, "The dynamic source routing protocol for mobile ad hoc network," Internet-Draft, draft-ietf-manet-dsr-09.txt, April 2003.

[15] S.J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad

hoc networks," *ICC 2001*, pp.3201-3205.

[16] K. Wu and J. Harms, "On-demand multipath routing for mobile ad hoc networks," *Proceedings of 4th European Personal Mobile Communication Conference (EPMCC 01)*, Vienna, Austria, Feb. 2001, pp.1-7.

[17] P. Pham and S. Perreau, "Performance Analysis of Reactive Shortest Path and Multipath Routing Mechanism With Load Balance", *IEEE Infocom 2003*.

[18] A. Nasipuri and S.R. Das, "On-demand multipath routing for mobile ad hoc networks," *Proceedings of the IEEE International Conference on Computer Communication and Networks (ICCCN'99)*, Boston, October, 1999, pp.64-70.

[19] P. Papadimitratos, Z.J. Haas, and E.G. Sirer, "Path Set Selection in Mobile Ad Hoc Networks," ACM Mobihoc 2002, Lausanne, Switzerland, June 9-11, 2002.

[20] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai. "Distributing streaming media content using cooperative networking," *ACM NOSSDAV*, Miami Beach, FL, USA, May 2002.

[21] S. Mao, X. Cheng, Y. Hou, H. Sherali, "Multiple description video multicast in wireless ad hoc networks," in Proc. BroadNets 2004, Oct. 25-29, San Jose, CA.

[22] S. Lee, M. Gerla, and C. Chiang. "On-Demand Multicast Routing Protocol", *Proceedings of IEEE WCNC'99*, New Orleans, LA, Sep. 1999, pp. 1298-1302.

[23] S. Lee, W. Su, et al. "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols", *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000.

[24] J. Jetcheva and D. Johnson. "Adaptive Demand-Driven Multicast Routing in Multi-hop Wireless Ad Hoc Networks", ACM Mobihoc 2001.

[25] S. Park, and D. Park. "Adaptive core multicast routing protocol", *Wireless Networks 2004*.

[26] C. Toh, G. Guichala, and S. Bunchua. "ABAM: OnDemand Associativity-Based Multicast Routing for Ad Hoc Mobile Networks", *In Proceedings of IEEE Vehicular Technology Conference, VTC 2000*, pages 987–993, September 2000.

[27] J. Xie, R. Talpade, et al. "AMRoute: ad hoc multicast routing protocol", *Mobile Networks and Applications*, v.7 n.6, p.429-439, December 2002.

[28] C. Wu and Y. Tay. "AMRIS: A multicast protocol for ad hoc wireless networks", *in Military Communications Conference Proceedings. 1999*, vol. pp. 25-29. New York: IEEE,

1999.

[29] S. Sajama, and Z. Haas. "Independent-tree ad hoc multicast routing (ITAMAR)", *ACM Mobile Networks and Applications*, Oct. 2003.

[30] E. W. Dijkstra. "A note on two problems in connexion with graphs", *Numerische Mathematik*, 1 (1959), pp. 269-271.

[31] D. De Couto, D. Aguayo, J. Bicket, and et al, "High-throughput path metric for multi-hop wireless routing". ACM Mobicom 2003.

[32] S. Jiang, D. He, and J. Rao, "A Prediction-based Link Availability Estimation for Mobile Ad Hoc Networks", IEEE Infocom 2001.

[33] Z. Jia, R. Gupta, J. Walrand, and P. Varaiya, "Bandwidth guaranteed routing for ad-hoc networks with interference consideration", ISCC 2005.

[34] ns-2: network simulator. http://www.isi.edu/nsnam/ns/

[35] X. Tang, and A. Zakhor. "Matching pursuits multiple description coding for wireless video," *IEEE Trans. on CSVT*, June 2002, pp. 566 -575.

[36] Y. Wang, and S. Lin. "Error-resilient video coding using multiple description motion compensation," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 438-452, June 2002.

[37] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks", *ACM MobiCom'04*, Philadelphia, September 2004.

[38] W. Wei and A. Zakhor, "Connectivity for Multiple Multicast Trees in Ad Hoc Networks", in International Workshop on Wireless Ad-hoc Networks (IWWAN), Oulu, Finland, June 2004.

[39] O. Dousse, P. Thiran and M. Hasler, "Connectivity in ad-hoc and hybrid networks", *IEEE Infocom 2002*

[40] T. Philips, S. Panwar, A. Tantawi. "Connectivity Properties of a Packet Radio Network Model", *IEEE Trans. On Information Theory*, Sep. 1989

[41] P. Gupta, and P. Kumar. "Critical Power for Asymptotic Connectivity in Wireless Networks". *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming* 1998

[42] W. Wei and A. Zakhor, "Multipath Unicast and Multicast Video Communication over Wireless Ad Hoc Networks", *International Conference on Broadband Networks (Broad-*

nets) 2004, San Jose, CA, USA, Oct. 2004 *(invited)*, pp. 496 - 505.

[43] W. Wei and A. Zakhor, "Path Selection For Multi-path Streaming in Wireless Ad Hoc Networks", *International Conference on Image Processing (ICIP) 2006*, Atlanta, Georgia, Oct. 2006.

[44] w. Wei, "Multipath Unicast and Multicast Video Communication over Wireless Ad Hoc Networks", PhD thesis, University of California, Berkeley, 2006.

[45] R. Neff and A. Zakhor. "Very Low Bit-Rate Video Coding based on Matching Pursuits", *IEEE Trans. on CSVT*, Feb. 1997, pp. 158-171.

[46] W. Cui, I. Stoica, and R. Katz, "Backup Path Allocation Based On a Correlated Link Failure Probability Model in Overlay Networks", *ICNP* 2002.

[47] M. Garey, and D. Johnson, "Computers and intractability: A guide to the theory of NP-Completeness", W. H. Freeman Company, 1979.

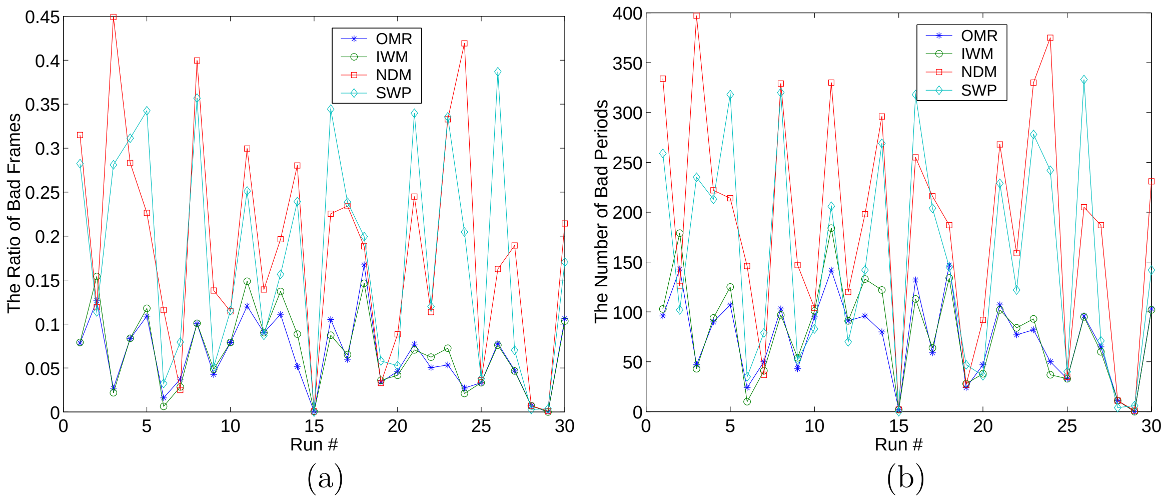


Figure 1.1: Simulation Results for the 7×7 grid network: (a) The ratio of bad frames; (b)The number of bad periods.

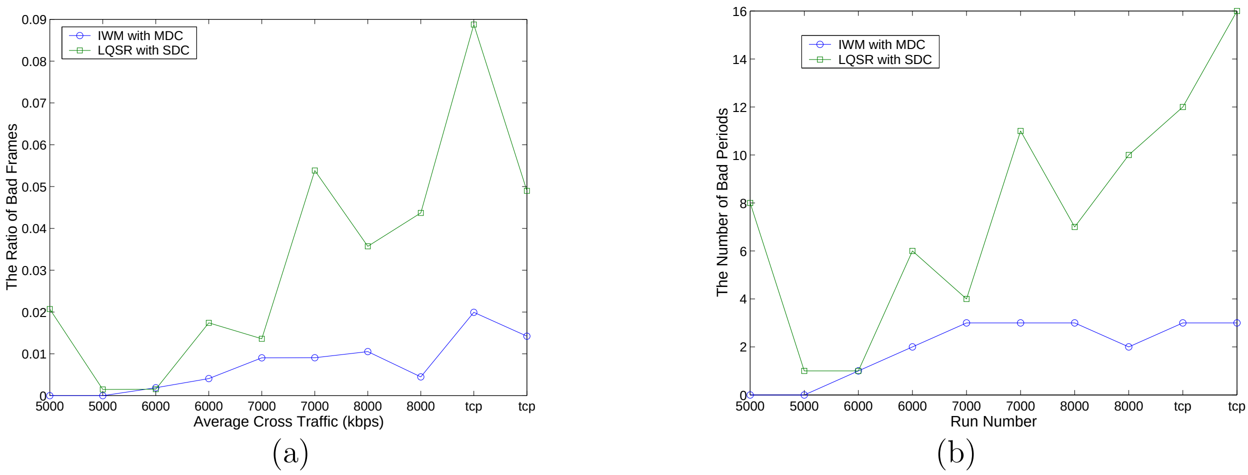


Figure 1.2: Performance Evaluation of IWM/MDC over 802.11a (a) The ratio of bad frames; (b)The number of bad periods.

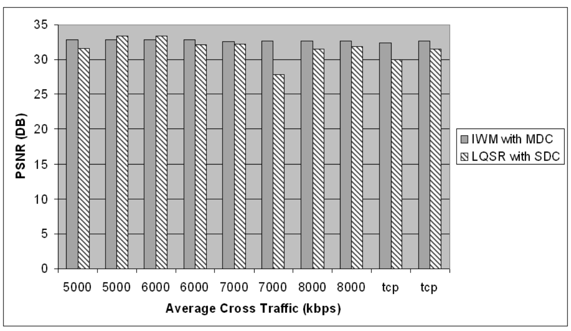


Figure 1.3: PSNR performance evaluation of IWM/MDC over 802.11a.

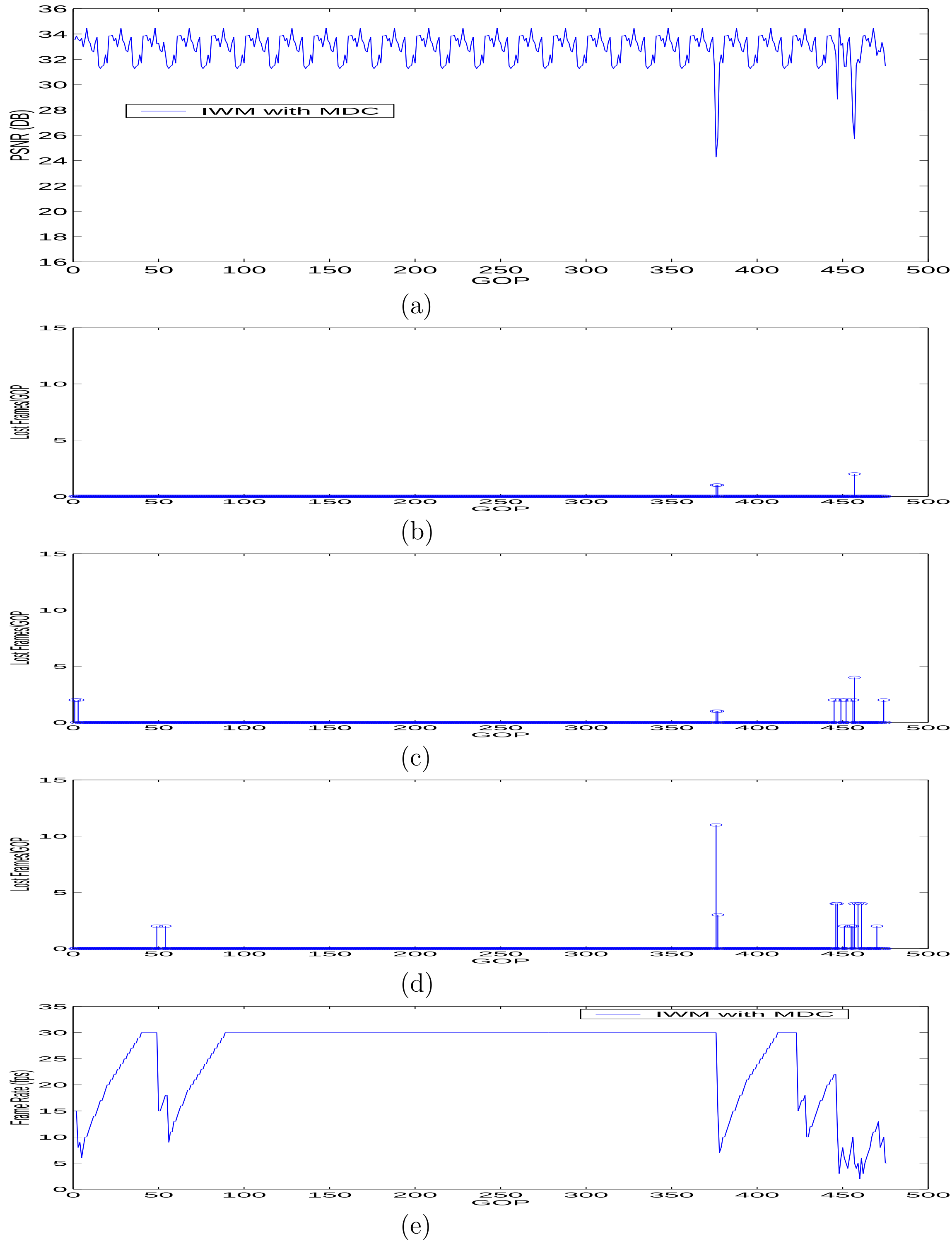
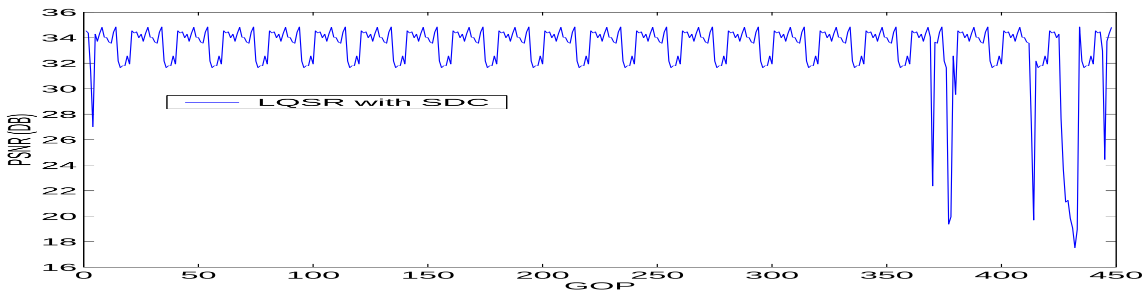
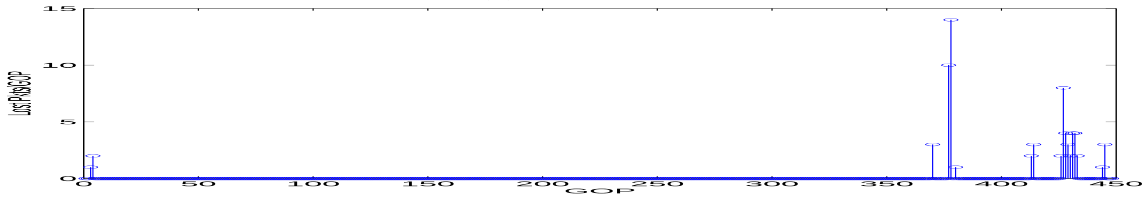


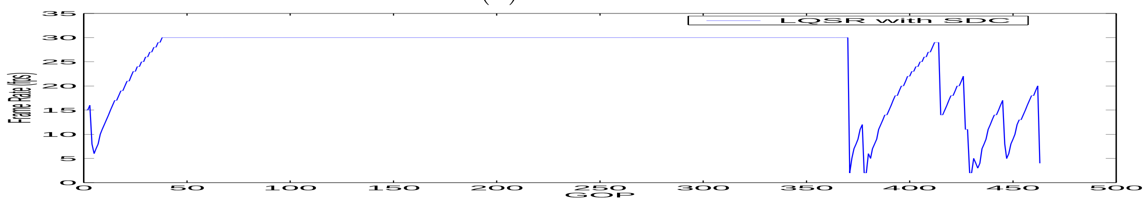
Figure 1.4: Performance Evaluation of IWM/MDC over 802.11a (a) PSNR of the received frames; (b) Number of Frames lost in both descriptions; (c) Lost frames per GOP for substream 0; (d) Lost frames per GOP for substream 1; (e) Sending frame rate.



(a)

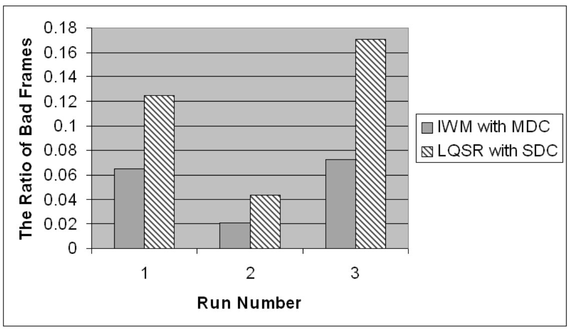


(b)

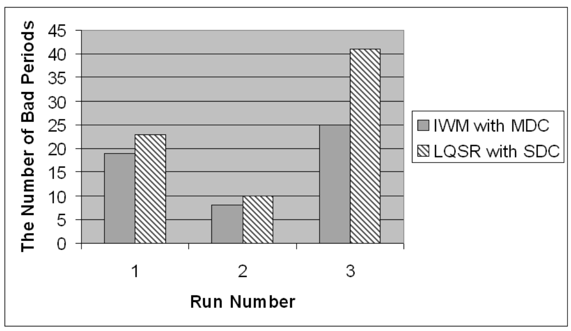


(c)

Figure 1.5: Performance Evaluation of LQSR/SDC over 802.11a (a) PSNR of the received frames; (b) Lost frames per GOP for the stream; (c) Sending frame rate.



(a)



(b)

Figure 1.6: Performance Evaluation of 802.11a with moving nodes (a) The ratio of bad frames; (b)The number of bad periods.

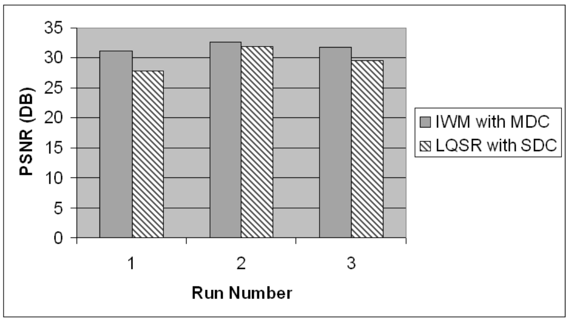


Figure 1.7: PSNR performance evaluation of IWM/MDC for 802.11a with moving nodes.

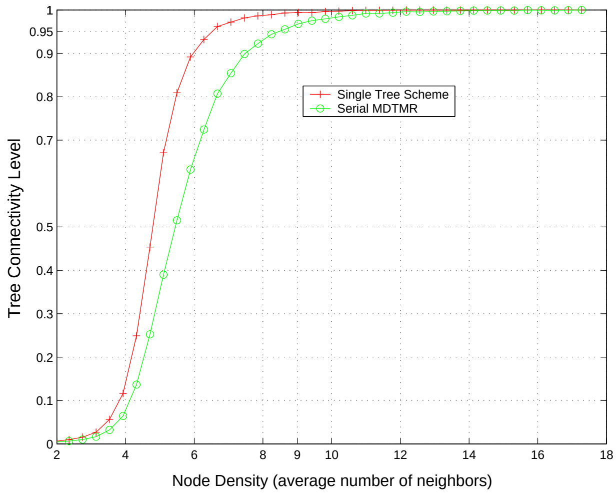
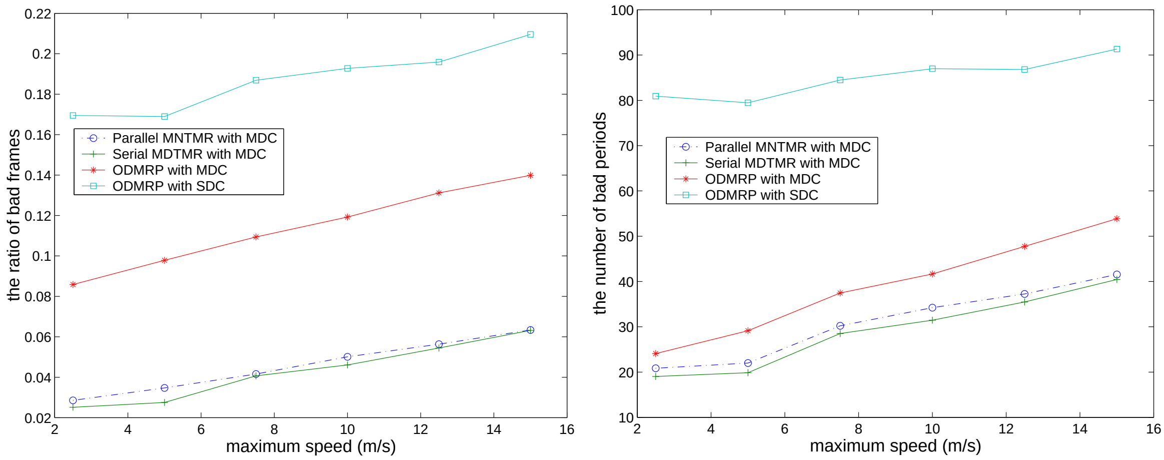


Figure 1.8: Tree connectivity of Serial MDTMR



(a) (b)

Figure 1.9: Performance evaluation for multiple tree video multicast: (a) The ratio of bad frames; (b)The number of bad periods.

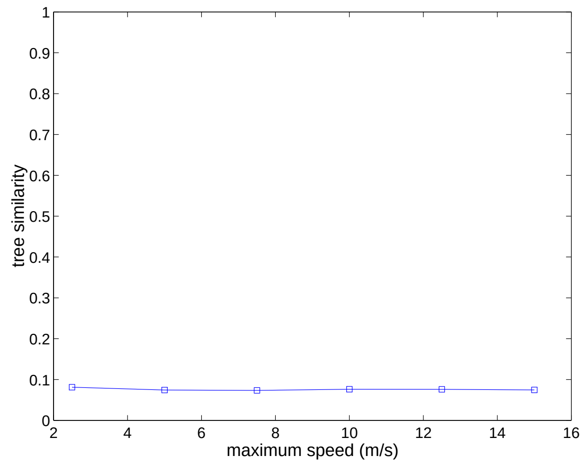


Figure 1.10: Tree Similarity of Parallel MNTMR

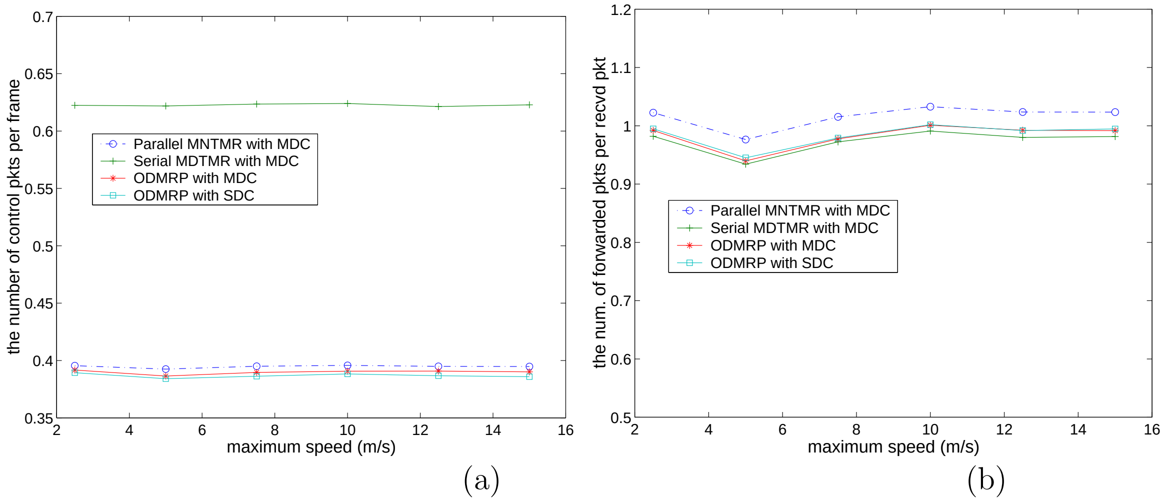


Figure 1.11: Performance evaluation for multiple tree protocols: (a) The normalized control packets; (b) The normalized forwarded data packets.

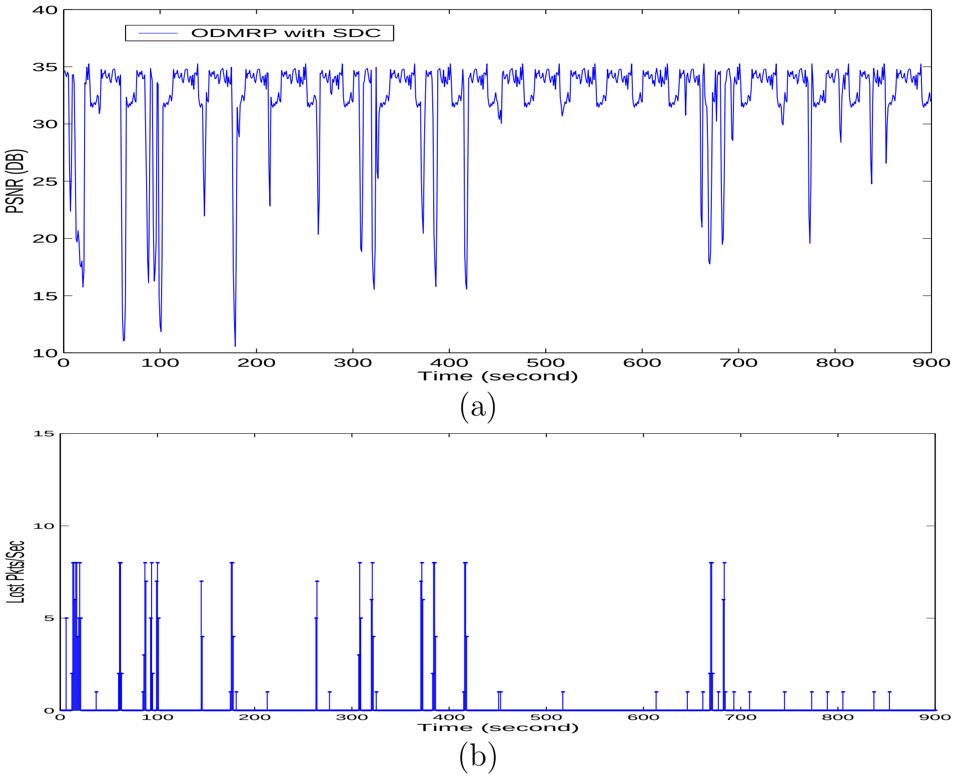


Figure 1.12: Performance evaluation of ODMRP and SDC(a) PSNR of the received frames; (b) Lost packets per second for the stream.

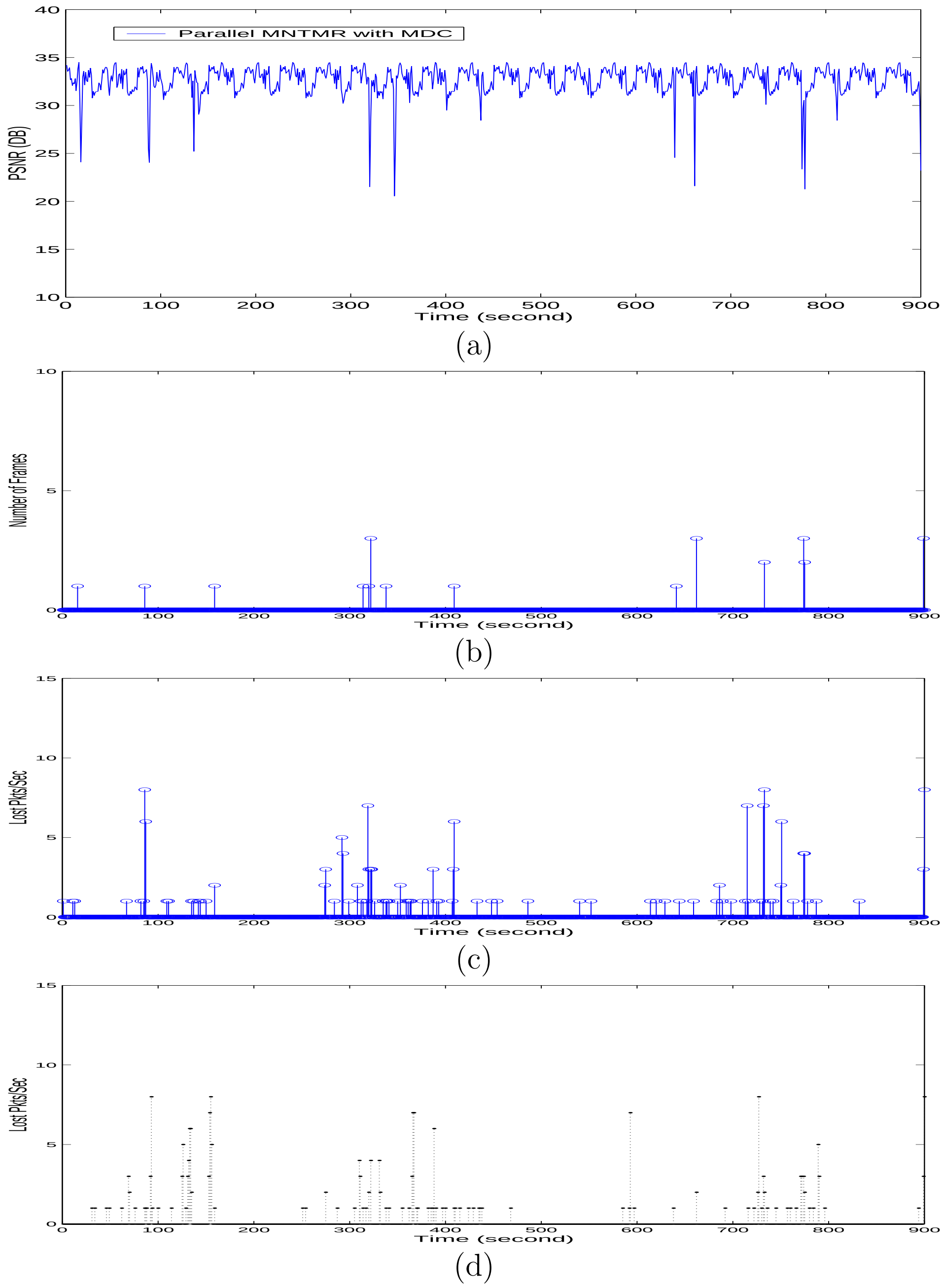


Figure 1.13: Performance evaluation of Parallel MNTMR and MDC(a) PSNR of the received frames; (b) Number of Frames that both descriptions are lost; (c) Lost packets per second for substream 0; (d) Lost packets per second for substream 1.