Interference Aware Multi-Path Selection for Video Streaming in Wireless Ad Hoc Networks

Wei Wei and Avideh Zakhor, Fellow, IEEE
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley, CA 94720, USA

***Abstract*— In this paper, we propose a novel multi-path selection framework for video streaming over wireless ad hoc networks. We propose a heuristic interference-aware multipath routing protocol based on the estimation of concurrent packet drop probability of two paths, taking into account interference between links. Through both simulations and actual experiments, we show that the performance of the proposed protocol is close to that of the optimal solution, and is better than that of other heuristic protocols.**

I. Introduction

A wireless ad hoc network is a collection of wireless mobile nodes that dynamically form a temporary network without an infrastructure. With the increase in the bandwidth of wireless channels, and in the computing power of mobile devices, video applications are expected to become widespread in wireless ad hoc networks in a near future.

There are many challenges to support video communication over wireless ad hoc networks. An end-to-end connection route in wireless ad hoc networks generally consists of multiple wireless links, which have much smaller throughput and higher random packet loss than single hop wireless connections in a wireless network with an infrastructure. Due to the mobility of wireless nodes, 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. These constraints and challenges, in combination with the delay and loss sensitive nature of video applications, make video communication over wireless ad hoc networks a challenging proposition.

Recent efforts on multipath routing of Multiple Description Coded (MDC) video have successfully demonstrated improved robustness in video communication applications[4-8]; this is done either by assuming that the set of paths is given, or by simply selecting two node/link disjoint paths. A number of recent papers have addressed the difficult problem of selecting optimal paths for MDC video streaming[9-11]. Begen et al. have studied how to select multiple paths that maximize the *average* video quality at clients on Internet overlay networks [9]. Mao et al. have further proposed a meta-heuristic approach based on Genetic Algorithms to solve the path selection problem [10]. In [11], the authors propose to select two paths with minimal correlation for MDC streaming over Internet overlay networks. In [17], the authors address the problem of minimizing the joint failure probability of two overlay paths, assuming that joint failure probability of every two links are given. These approaches however are too complex to be performed in real-time. Also these models consider neither the interference of flows on neighboring links, nor the influence of the incoming video flow on the characteristics of links; this is needed in current wireless ad hoc networks, because a new video flow generally consumes a large percentage of wireless resource, thus changing characteristics of wireless links significantly.

In [23], layered video coding combined with a selective ARQ scheme is proposed, in which base layer and enhancement layer packets are transmitted over different paths, and only base layer packets are retransmitted. In [24], a framework for simultaneous streaming of video from multiple mirror sites to a single receiver in the Internet is proposed. In [25], A rate allocation scheme with FEC is proposed for path diversity based video streaming on the Internet. In [26], a video client determines how to allocate rate between several throughput-limited forwarders to maximize received video quality.

Multipath routing for wireless ad hoc networks has been an active research area recently [27-36]. Most existing approaches focus on how to obtain multiple node/link disjoint paths. In [36], the authors propose a heuristic algorithm to select multiple paths to achieve the best reliability, assuming the independence of failure probability of different links.

The problem of finding rate constraints on a set of flows in a wireless ad hoc network is studied in [12][13]. 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 [14], the authors develop a different set of rate constraints using the cliques, i.e. complete subgraphs, of the conflict graph.

In this paper, we propose a technique for choosing two node-disjoint paths, which achieve minimum concurrent Packet Drop Probability (PDP) of all path pairs. Our motivation is to increase robustness of video applications over wireless ad hoc networks. 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 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

[1]W. Wei and A. Zakhor are with department of EECS, UC Berkeley. Email: {weiwei, avz}@eecs.berkeley.edu.

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. 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 paper, we use a *conflict graph* [12][13][14] 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 show that the above optimization is an NP-hard problem. 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 better than that of the node-disjoint multipath routing, and the shortest-widest routing.

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.

The rest of this paper is organized as follows. In Section II, we propose a method to estimate the concurrent PDP of two node-disjoint paths, and formulate the optimal multipath selection problem for video streaming over wireless ad hoc networks. Section III presents a heuristic PDP aware multipath routing protocol. We show our simulation results in Section IV, and introduce the testbed implementation and experimental results in Section V. Section VI concludes the paper.

II. Optimum Multipath Selection

Our goal is to minimize concurrent PDP of two node-disjoint paths in a wireless ad hoc network which 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 can significantly decorrelate packet drop between different paths.

A. Envisioned Network Model

We consider a wireless ad-hoc network with N nodes arbitrarily distributed in a plane. Let n_i, $1 \leq i \leq N$ 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. Our interference model is similar to that of the protocol model introduced in [13][14], with changes to reflect the implementation of 802.11 MAC protocol in NS-2 [16].

The 802.11 MAC protocol defines two access methods, Distributed Coordination Function (DCF) and Point Coordination Function (PCF). We focus on DCF in our research, as it is more popular than PCF. DCF applies a Carrier Sense Multiple Access with Collision Avoidance mechanism (CSMA/CA). When a station wants to transmit a packet, it senses the medium first. If the medium is sensed as free for a specified time period, the station is allowed to send. Each correctly received unicast packet is followed by an Acknowledgement (ACK) packet to the sender. The sender retransmits the packets for a limited number of times until it receives the ACK packet. In order to avoid hidden terminal problem, the standard also defines an optional Request-to-Send/Clear-to-Send (RTS/CTS) scheme, which reserves the wireless channel for transmission of a data packet.

Protocol Interference Model: Suppose node n_i wishes to transmit to node n_j. We use SS_{ij} to denote the signal strength of n_i's transmission as received at node n_j. The transmission between nodes n_i and n_j is successful if all of the following conditions are satisfied:

- $d_{ij} \leq r$; intuitively this is equivalent to nodes n_i and n_j being within each other's communication range.
- Any node n_k, such that $d_{ki} \leq \omega$ is not transmitting. This is motivated by the CSMA/CA scheme in the 802.11 MAC protocol, which states that node n_i can not transmit if any node in its interference range is transmitting.
- Any node n_k, such that $\frac{SS_{ij}}{SS_{kj}} \leq CPThresh$, is not transmitting, where $CPThresh$ denotes the capture threshold, with default value of 10 in NS-2. This implies that no node with sufficiently large signal strength interfering with link n_i to n_j is transmitting.

The main difference between the Protocol Interference Model applied in this paper and the model used in [13][14] is that the latter requires only the receiver to be free of interference, whereas our model requires both the receiver and the sender be free of interference, reflecting the 802.11 MAC better.

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. There is a link from vertex n_i to vertex n_j if and only if $d_{ij} < r$. As in [12][13][14], we make use of a "conflict graph" to model the interference relationship between different links of a network. Every directed link in the graph $G(V, E)$ is represented by a node in the directed conflict graph $CG(V^C, E^C)$. Without confusion, we use l_{ij} to denote both a link in the original graph, and a node in the corresponding conflict graph. If the transmission over link l_{ij} makes the transmission over link l_{kl} unsuccessful, link l_{ij} interferes with link l_{kl}, resulting in a directed link from node l_{ij} to node l_{kl} in the conflict graph. To avoid confusion in the paper, we use the terms "node" and "link" in reference to the connectivity graph $G(V, E)$, while using "CG-node" and "CG-link" to refer the conflict graph $CG(V^C, E^C)$. Figure 1(a) shows an example of a conflict graph. CG-nodes 1 through 5 correspond to five wireless links in the original wireless network. The wireless link represented by CG-node 1 interferes with wireless links represented by CG-nodes 2, 3, 4 and 5, while none of the other CG-nodes interfere with each other.

For simplicity, in the above protocol interference model, we assume that communication and interference ranges are the same for all nodes. For a more general case, where different

nodes have different communication and interference ranges, we can replace r and ω with specific values. In this case, we can still obtain a conflict graph, which describes interference relationship between different links. As such, this assumption does not affect the results in this paper.

B. The Optimal Multipath Selection Problem

The main objective of the optimal multipath selection problem for MDC video streaming over wireless ad hoc networks is to select two node-disjoint paths with minimum concurrent PDP.

Definition 1: A path $P_{S,D}$ connecting nodes N_S and N_D in a graph $G(V, E)$, is a sequence of nodes $v_1, \ldots v_n$, which satisfy the following two conditions. (a) $\forall i, 1 \leq i < n$, we have $(v_i, v_{i+1}) \in E$; (b) no node appears more than once. The set of the nodes on this path is represented by $N_{S,D} \subseteq V$, and the set of the links on this path is denoted by $L_{S,D} \subseteq E$.

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 1 if link $(i,j) \in L^1_{S,D}$ and is set to 0 otherwise. Similarly y_{ij} is set to 1 if link $(i,j) \in L^2_{S,D}$ and is set to 0 otherwise. The dimension of vectors $\mathbf{x}$ and $\mathbf{y}$ is the number of links in the graph.

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

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

$$\sum_{i:(i,j)\in E} x_{ij} \leq 1 \quad (2)$$

$$\sum_{n:(m,n)\in E} y_{mn} - \sum_{n:(n,m)\in E} y_{nm} = \begin{cases} 1 & m = N_S \\ -1 & m = N_D \\ 0 & otherwise \end{cases} \quad (3)$$

$$\sum_{m:(m,n)\in E} y_{mn} \leq 1 \quad (4)$$

$$N^1_{S,D} \cap N^2_{S,D} = \{N_S, N_D\} \quad (5)$$

Equations (1) and (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 for 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 (3) and (4) are flow constraints for the second path. Equation (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) through (5) is NP-hard.

The proof is shown in [43].

Since the optimal multipath selection problem is NP-hard, the required computation needed by the optimal solution will be very high. One approach 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 to be discussed in Section II.C, and choose the best one. We refer to this solution as the Optimal Multipath Routing (OMR). Unfortunately as computation complexity of the OMR grows exponentially with the size of the network, it can not be run in real time. For instance, it takes a Matlab implementation of OMR approximately 8.2 seconds to select the best path pair in a network of 49 nodes, and 237.6 seconds in a network of 100 nodes. However, as will be seen shortly, OMR can be used to provide an upper bound on the performance of other low complexity heuristic schemes that can be run in real time. In Section III, we propose one such heuristic solution, and compare its performance with OMR. Before doing so, we will first develop a technique for estimating concurrent PDP in the next section.

C. 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.

We assume that we have already estimated the flow rates F_i over each link l_i. Before computing the PDP, we hypothetically include the new arriving video flow into the network by increasing the flow rate over each link in $L^1_{S,D} \cup L^2_{S,D}$ by the amount of video flow rate that will be transmitted over that link.

We define random variables

$$X_{ij} = \begin{cases} 1 & \text{packet drop in link } l_{ij} \\ 0 & \text{otherwise} \end{cases}$$

and

$$Y_{mn} = \begin{cases} 1 & \text{packet drop in link } l_{mn} \\ 0 & \text{otherwise} \end{cases}$$

We refer to the correlation of two random variables X_{ij} and Y_{mn} as follows.

$$\rho_{xy} = \frac{Cov(X_{ij}, Y_{mn})}{\sqrt{Var(X_{ij})}\sqrt{Var(Y_{mn})}} \quad (6)$$

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. According to the standard, large packets are retransmitted up to four times and small packets are retransmitted up to seven times. PDP due to channel error is 0.4% after four retransmissions, even if PDP due to channel error without retransmission is as high as 25%. 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 the above conjecture. We deploy 12 nodes in a 100 by 100 square meters area, with all links interfering with each other. We transmit two UDP flows of 500 kbps each over two node-disjoint links, and vary the number of cross traffic flows, whose bitrates are uniformly distributed in the range of [200, 300] kbps over other links. The cross traffic flows do not share nodes with each other. We have used trace files in NS to verify that using these parameters, the main cause of packet drop is congestion. We apply sequence number to packets of these two UDP flows. Concurrent packet drop is increased by one if two packets of two flows with the same sequence number are both dropped. The correlation between packet drop of two UDP flows as computed by Equation (6) are shown in Table I. The results show that if packet drop rate over each link is small, the correlation between packet drop over two node-disjoint links is also small. We carried out 30 groups of simulations, and arrived at similar conclusions. Our conclusions may be extended to other scenarios with hidden terminal nodes, in which two links do not interfere with each other directly, because in these cases two links are less correlated than those in our simulations.

TABLE I
CORRELATION OF PACKET DROP OVER TWO NODE-DISJOINT LINKS

Number of Cross Flows	2	3	4
Packet Drop Rate of Flow 1	0.0421	0.1610	0.3211
Packet Drop Rate of Flow 2	0.0364	0.1515	0.3131
Concurrent Packet Drop Rate of Flow 1 & 2	0.0002	0.0073	0.0647
Correlation	-0.0374	-0.1336	-0.1715

Since correlation between packet drop over two node-disjoint links is small, so is the correlation between packet drop over two node-disjoint paths. This is because two node disjoint paths only share two nodes, i.e. source and destination. Thus we can compute the concurrent PDP over two node-disjoint paths $P^1_{S,D}$ and $P^2_{S,D}$ as follows

$$\begin{aligned} P_{\text{drop}}(P^1_{S,D}; P^2_{S,D}) &\approx P_{\text{drop}}(P^1_{S,D}) \cdot P_{\text{drop}}(P^2_{S,D}) \\ &= [1 - \prod_{l_{ij} \in L^1_{S,D}} (1 - P_{\text{drop}}(l_{ij}))] \\ &\cdot [1 - \prod_{l_{mn} \in L^2_{S,D}} (1 - P_{\text{drop}}(l_{mn}))] \end{aligned} \quad (7)$$

In the next section, we will show how to estimate PDP over one link in order to compute the concurrent PDP of two node-disjoint paths.

D. Computing 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 packet loss. Thus PDP over link l_{ij} can be represented as

$$\begin{aligned} P_{\text{drop}}(l_{ij}) = 1 - [1 - P_{\text{drop-cong}}(l_{ij})] \\ [1 - P_{\text{drop-cont}}(l_{ij})][1 - P_{\text{drop-chan}}(l_{ij})] \\ [1 - P_{\text{drop-mob}}(l_{ij})] \end{aligned} \quad (8)$$

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. [40] to estimate PDP due to contention and wireless channel error, and apply results on link availability [42] 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 $P_{\text{drop-cong}}(l_{ij})$. For brevity, we refer to PDP due to congestion as PDP-congestion in the rest of this section. One possibility is to measure packet drop due to interface queue (ifq) overflow at each node, and use the ifq packet drop rate at node n_i to approximate $P_{\text{drop-cong}}(l_{ij})$. The disadvantage of this method is that it does not consider the influence of the incoming video flow on the packet drop rate of the link. In current wireless ad hoc networks, a new video flow generally consumes a large percentage of wireless resource, which can change PDP-congestion significantly. Thus we propose a new scheme to estimate a link's PDP-congestion based on the estimation of equivalent bandwidth used in the link's neighborhood.

We define the *interfering link set*, consisting of all the links that interfere with link l_{ij} as follows:

$$I(l_{ij}) = \{l \in E, \text{and } l \text{ interferes with } l_{ij}\} \bigcup \{l_{ij}\}$$

where E is the link set consisting of all the links in the graph. Assuming all flows are long lived, and the buffer size of each node is small, a naive way to compute the PDP-congestion of link l_{ij} is as follows:

$$P_{\text{drop-cong}}(l_{ij}) \approx \max(1 - \frac{C}{\sum_{l_k \in I(l_{ij})} F_k}, 0) \quad (9)$$

where F_k corresponds to the aggregate incoming flow rate into the k^{th} link l_k of the set $I(l_{ij})$, and C is the channel capacity. However Equation (9) unnecessarily overestimates the PDP-congestion, since simply summing up all flow rates overestimates the bandwidth needed for transmitting all flows

[14], and as such can not effectively differentiate between congested and uncongested links. Later in the paper, we will show that we can break a link's interfering links set into several disjoint independent sets, reducing the required bandwidth for transmitting all flows.

Figure 1 shows two conflict graphs that illustrate an example of the ineffectiveness of the naive estimation. Remember that for a conflict graph, each CG-node is a link in the original connectivity graph, and if link l_i interferes with link l_j, there is a directed CG-link connecting CG-node i and CG-node j. To simplify our explanation, we assume bi-directional CG-links in the example. Also assume that flow rates $F_j, j = 1 \ldots 5$ are equal. Intuitively, the PDP-congestion of link l_1 in Fig. 1(a) is smaller than that of Fig. 1(b), since there are extra interferences between other four links in conflict graph 1(b). However, Equation (9) would incorrectly imply PDP-congestion of link l_1 in both conflict graphs to be the same.

We now propose a method to estimate the PDP-congestion more accurately. Note that set $I(l_{ij})$ represents both an interfering link set in the connectivity graph, which consists of all the links that interfere with link l_{ij}, and a CG-node set in the corresponding conflict graph. We partition the interfering link set $I(l_{ij})$ into several disjoint subsets, such that each subset is an independent set. In a conflict graph, an *independent set* denoted by IS, is defined to be a set of CG-nodes that have no edges between them. Intuitively, this corresponds to any set of links whose transmissions do not interfere with each other. Note that for each link l_{ij} there are multiple possible partitions. The set of independent sets resulting from a particular partition of $I(l_{ij})$ is denoted by $PT(l_{ij})$, and can be represented as follows:

$$PT(l_{ij}) = \{IS_1, IS_2, \ldots IS_{q_i}\}$$

where

$$\bigcup_{k=1,\ldots q_i} IS_k = I(l_{ij}) \text{ and } IS_k \bigcap IS_m = \emptyset,\ 1 \leq k, m \leq q_i$$

IS_k is the k^{th} independent set, and q_i is the number of subsets in this particular partition. Physically, each independent set is selected by the MAC layer protocol with some probability at each time.

Assume that we can compute all the possible partitions of set $I(l_{ij})$, calling them $PT(l_{ij})_1, PT(l_{ij})_2, \ldots PT(l_{ij})_{N_i}$, where N_i is the number of partitions. We model the selection of an independent set at time t as a two level process. First, the MAC layer selects partition $PT(l_{ij})_k$ with probability p_k; second, it selects one independent set in the partition $PT(l_{ij})_k$ using the corresponding schedule for that partition, to be described shortly. The estimation of PDP-congestion of link l_{ij} can be written as

$$P_{\text{drop-cong}}(l_{ij}) \approx \sum_{k=1}^{N_i} p_k \times P_{\text{drop-cong}}(l_{ij}|PT(l_{ij})_k) \quad (10)$$

We define a link is an *active link*, if either of its two nodes have available packets to transmit through it. At a given time t, we define an independent set to be an *active independent set*, if at least one of the links in the independent set is active. From the definition of the independent set and partition, we know that all links in the same independent set can transmit simultaneously, but not those links in different independent sets. For a given partition of the set $I(l_{ij})$ with q_i independent sets, we define the corresponding schedule as follows. Let $m_i(t)$ denote the number of active independent sets in the partition at time t. The corresponding schedule allows the k^{th} active independent set at time t to access the shared wireless medium with probability $p(k, m_i(t))$. In essence the corresponding schedule for a partition allows every active independent set to access the shared wireless medium according to some predefined schedule. All *active links* in one independent set can transmit at time t, if the independent set is assigned by the MAC layer to access the wireless medium at that time.

We can define equivalent rate of flows over all links in the independent set IS_k as follows:

$$CF_k = \max_{l_m \in IS_k} F_m \quad (11)$$

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

Given a partition of the set $I(l_{ij})$, assuming that the corresponding schedule is applied by the MAC layer, and all flows are long lived, we can estimate the PDP-congestion of link l_{ij} as follows:

$$P_{\text{drop-cong}}(l_{ij}|PT(l_{ij})) \approx \max(1 - \frac{C}{\sum_{IS_k \in PT(l_{ij})} CF_k}, 0) \quad (12)$$

Note that we treat an independent set as a super-flow; since all independent sets share the resources, we add the equivalent rates for different independent sets in order to compute PDP-congestion. In deriving Equation (12), we assume the durations of the long lived flows to be large enough and the buffer sizes at the nodes to be small enough so that we can ignore the time it takes to fill the buffer to compute PDP-congestion. In practice, if the buffer sizes are large, Equation (12) would overestimate $P_{\text{drop-cong}}$.

Combining Equations (10) and (12), it is possible to obtain PDP-congestion of link l_{ij}. Unfortunately, computing all of the independent sets in a graph grows exponentially in the number of nodes [13]; also it is difficult to estimate the probability p_k that the 802.11 MAC layer selects partition $PT(l_{ij})_k$, $k = 1, 2, \ldots N_i$; as such, the computational overhead of the above method is too high and the implementation is impractical. To circumvent this, we name the partition $PT(l_{ij})^*$ that minimizes $P_{\text{drop-cong}}(l_{ij}|PT(l_{ij}))$ as *the most efficient*

partition. Since

$$\begin{aligned} P_{\text{drop-cong}}(l_{ij}) &\approx \sum_{k=1}^{N_i} p_k \times P_{\text{drop-cong}}(l_{ij}|PT(l_{ij})_k) \\ &\geq \sum_{k=1}^{N_i} p_k \times P_{\text{drop-cong}}(l_{ij}|PT(l_{ij})^*) \\ &= [\sum_{k=1}^{N_i} p_k] \times P_{\text{drop-cong}}(l_{ij}|PT(l_{ij})^*) \\ &= P_{\text{drop-cong}}(l_{ij}|PT(l_{ij})^*) \end{aligned} \quad (13)$$

Therefore $P_{\text{drop-cong}}(l_{ij}|PT(l_{ij})^*)$ is the lower bound of link l_{ij}'s PDP-congestion. Since computing the actual PDP-congestion is prohibitively computing-intensive, our approach is to use its lower bound instead, i.e. the PDP-congestion of the *most efficient partition*, as a metric in comparing PDP of two links, and subsequently two paths. Specifically, combining Equations (12) and (13), we get

$$P_{\text{drop-cong}}(l_{ij}) \geq \max(1 - \frac{C}{\sum_{IS_k \in PT(l_{ij})^*} CF_k}, 0) \quad (14)$$

where $PT(l_{ij})^*$ denotes the most efficient partition.

As such, we will shortly propose a greedy algorithm to approximately find the most efficient partition. We note that using the most efficient partition results in underestimating the interference around link l_{ij}, and the PDP. Nevertheless, we have verified through simulations that it is sufficient to use the lower bound of each link's PDP as an approximation to our metric in order to compare and select paths. Also, it can be argued that with the development of more efficient MAC layer protocol in the future, our estimation approaches the optimal results.

For a network shown in Figure 1(b), the most efficient partition consists of three independent sets, $\{l_1\}$, $\{l_2, l_4\}$, $\{l_3, l_5\}$. Thus link l_1's PDP-congestion is approximated as follows:

$$\begin{aligned} P_{\text{drop-cong}}(l_1) &\approx P_{\text{drop-cong}}(l_1|PT(l_1)^*) \\ &\approx \max(1 - \frac{C}{F_1 + \max(F_2, F_4) + \max(F_3, F_5)}, 0) \end{aligned} \quad (15)$$

E. Estimating the most efficient partition

We now propose a greedy partition algorithm to estimate the most efficient partition. The basic idea of the greedy partitioning algorithm is to combine links with large flow rates together in order to reduce the sum of equivalent flow rates of independent sets, thus minimizing the probability $P_{\text{drop-cong}}(l_{ij}|PT(l_{ij}))$. The algorithm first selects the link with the largest flow rate into the first independent set, then selects other qualified links into the same independent set in the order of flow rate. After obtaining one independent set, the algorithm repeats the above process to obtain other independent sets, until every link in the interfering link set $I(l_{ij})$ is in one independent set. Algorithm 1 shows the pseudocode for the proposed greedy algorithm.

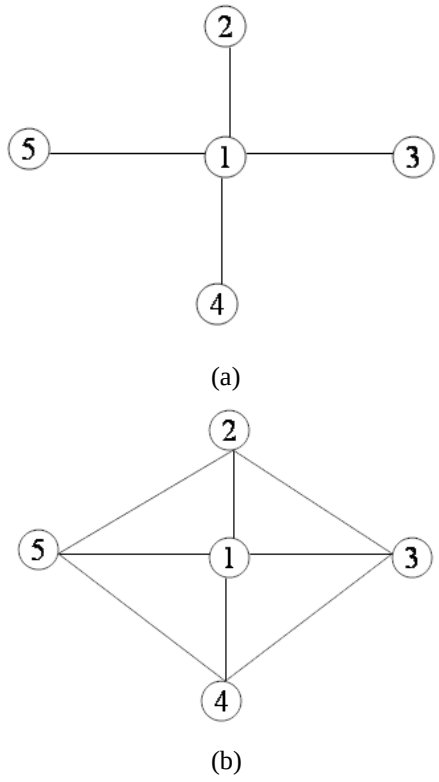


Fig. 1. An example to show the ineffectiveness of the naive estimation; (a) conflict graph 1(a); (b) conflict graph 1(b).

F. Summary and Implementation Issues

In our current implementation, we apply a centralized approach in order to estimate PDP. Each node measures flow rate and packet loss probability of broadcast packets of links to which it is connected, and broadcasts this information to the network periodically. In the end, each node receives flow rate and packet loss probability of broadcast packets of all other links.

In order to estimate flow rate of each link, the routing agent of each node parses all the outgoing packets, obtaining next hop and the packet's size. The number of bytes transmitted over a link in a time window $bytes\_sent\_win$ can be computed through summing up the size of all packets transmitted through this link during the time window. The flow rate can be computed using a moving window average scheme, and updated as follows:

$$curr\_flow\_rate = \alpha \times prev\_flow\_rate + (1-\alpha) \times \frac{bytes\_sent\_win}{win\_size} \quad (16)$$

where $curr\_flow\_rate$, $prev\_flow\_rate$, win_size represent current flow rate, previously estimated flow rate, and length of time window respectively. α is a parameter that can be used to trade off importance of past measurements versus new ones.

The procedure of estimating concurrent PDP of two paths can be summarized as follows:

- Given flow rate of links interfering with link l_{ij}, PDP over link l_{ij} due to congestion is estimated by first estimating the most efficient partition and then applying Equation (14). Our current implementation in this paper does not

Algorithm 1 Partitioning set $I(l_{ij})$

```
Sort CG-nodes in the set I(l_ij) based on the the flow rate,
in the order from the largest to the smallest
Set k = 0
while ( I(l_ij) not empty ) do
  Start with an empty independent set IS_k
  Add the first CG-node I_0 into IS_k
  Update I(l_ij) = I(l_ij)\I_0
  for (Haven't finished searching I(l_ij)) do
    if (CG-node I_m does not have an edge to any of the
    CG-nodes in IS_k) then
      Add I_m into IS_k
      Update I(l_ij) = I(l_ij)\I_m
    end if
  end for
  Increase k by one
end while
```

take into account PDP over a link due to mobility.

- Using Equation (8), we compute PDP over link l_{ij} from PDP over link l_{ij} due to channel error, contention and congestion.
- Using Equation (7), we compute concurrent PDP of two paths from PDP over a link.

To summarize, in Section II, we have developed a technique to estimate PDP of two node disjoint paths. We can use this in the next section to arrive at a practical path selection algorithm.

III. A Heuristic solution to the optimum multipath selection

Since the optimal multipath selection problem is NP-hard and the OMR algorithm described in Section II is prohibitively compute intensive, in this section we propose a heuristic solution, called Interference aWare Multipath Routing (IWM), for choosing two paths with minimum concurrent PDP. IWM applies the technique summarized in Section II-F to estimate PDP of each path. The basic idea behind IWM is to obtain a path with approximately minimum PDP as the first path. After updating all the link metrics of the network graph, such as flow rate, IWM finds the second path with approximately minimum PDP based on the new graph.

A. Centralized Implementation

We begin by proposing a centralized protocol. We assume that flow rate and packet loss probability of broadcast packets of each link, are distributed over the entire network periodically. Thus the sender knows both the topology of the network and characteristics of each link. In this case, the sender is able to compute the PDP given any two paths in the network.

By assuming that the PDP of each link is small, we approximate $P_{\text{drop}}(P^1_{S,D}; P^2_{S,D})$ represented in Equation (7) as follows:

$$P_{\text{drop}}(P^1_{S,D}; P^2_{S,D}) = \sum_{l_{ij} \in L^1_{S,D}} P_{\text{drop}}(l_{ij}) \cdot \sum_{l_{mn} \in L^2_{S,D}} P_{\text{drop}}(l_{mn}) \tag{17}$$

Therefore, we relax the optimal multipath selection problem by allowing the first path to minimize PDP and the second path to minimize PDP among all node disjoint paths with the first one. Note that this approach is similar to the one proposed in [11]. The main difference is that our metric is PDP while theirs is correlation. Specifically, we apply the techniques described in Section II-D to compute PDP for each link in wireless ad hoc networks.

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

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

such that the constraint in Equation (1) is satisfied. $P_{\text{drop}}(l_{ij})$ as defined by Equation (8) denotes the PDP over link l_{ij}, which can be viewed as the cost assigned to link l_{ij}. $P_{\text{drop}}(l_{ij})$ is estimated through the procedure described in Section II-F. To obtain the first path, we solve this optimization problem using the Dijkstra's algorithm, whose complexity is polynomial [39].

After obtaining the first path, we first update flow rate over each link, by taking into account the incoming video flow rate into corresponding links. Given the first path, for computing the second path, we define a link cost as follows:

$$C_{mn} = P_{\text{drop}}(l_{mn}) + \text{nd\_cost} \tag{18}$$

where

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

is a penalty factor to maintain the node-disjointness between the two paths. b_1 is chosen to be an arbitrarily large constant to trade off between disjointness and minimizing PDP.

The optimization problem to find the second path is formulated as follows:

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

such that the constraints in Equation (3) are satisfied. We also apply the Dijkstra's algorithm to solve this optimization problem. Thus the complexity of IWM is polynomial and is comparable to other Link State routing algorithms [20].

The advantage of the proposed centralized approach is that it is very easy to implement. Also when a node needs to transmit video applications, it can compute two paths from the link state cache immediately, i.e. there is no startup delay with this approach. However there are several disadvantages of the centralized approach.

- Each node has to collect link state information of all the links in the network and store them to the link state cache. In order to build and maintain the link state cache, each node needs to periodically broadcast characteristics of its links to the entire network. Collecting link state information requires a large amount of control overhead. Two techniques can be applied to reduce the amount of control overhead. The first one is Multipoint Relay (MPR) used by Optimized Link State Routing (OLSR) [37]. The second one is partial topology information

report technique applied by Topology Broadcast based on Reverse-Path Forwarding (TBRPF) [38].
- The period for updating the link state information can not be too short, otherwise the amount of control overhead becomes prohibitively large. So this approach is only suitable for static networks or networks with slow moving nodes. For the kind of network, whose condition changes very fast, the link state information in the cache is not accurate, affecting the selection of the best path pair. In our simulations and experiments, we set the period of updating the link state information to 1 second. The video sender recalculates the best two paths after it obtains updated link state information, and changes the two current paths to the new ones if the performance gain is above a certain threshold. Thus our algorithm can perform well when cross traffic flows are more dynamic.

B. Distributed Implementation

In order to reduce the amount of control overhead, we further propose a distributed protocol for IWM. The basic idea behind the distributed implementation of the IWM is that the protocol builds two paths in two steps. In the first step, the sender initiates a route discovery process by sending out a Route Query (RREQ) message. The RREQ message carries a value representing cost of the path traversed by the message. When a node receives a non-duplicate RREQ message, before forwarding it, it updates the path cost as follows:

$$\text{new\_path\_cost} = \text{old\_path\_cost} + \text{link\_cost} \tag{19}$$

where the link_cost is PDP of the link connecting the previous hop and the current node, which is computed based on the link's two hop neighbors' information and Equation (8). The receiver collects paths carried in arrived RREQ messages within a short time period $[t_0, t_0 + d]$, where t_0 is the time that the first RREQ message arrives. Then the receiver selects the path with the smallest path cost and sends a Route Reply (RRER) message carrying the path back to the sender. After receiving the RRER message carrying the first path, the sender sends out another RREQ message with a different sequence number. We use odd sequence numbers representing RREQ messages for the first path, and even ones for the second path. This time, the RREQ message carries both the path cost and the nodes' IDs of the first path. The middle nodes update a path cost carried by a RREQ message as shown in Equation (19), except the link_cost is represented by Equation (18). The sender will select the second path in a similar way.

In order to increase the probability of selecting the best path pair without increasing too much routing overhead, we further propose two enhancement techniques.

- When receiving a duplicate RREQ message, instead of simply discarding it, the middle node first compares the $path\_cost$ value carried in the message with the minimum path cost value stored in the node for the same route discovery process. If the $path\_cost$ value carried in the current message is smaller, the middle node updates the minimum stored path cost value, and forwards the newly received RREQ message.
- When a node forwards a RREQ message, the node applies an extra forwarding delay, which is proportional to the path cost carried in the message. Thus the path with smaller cost has a larger chance to arrive at the receiver. This way, the receiver has a better chance of selecting the ideal best path.

In order to learn two hop neighbors' information, each node sends beacon messages to its neighbors periodically. The beacon message carries both characteristics of links connected to the current node, and information of links connected to the current node's neighbors. Thus each node can learn its two hop neighbors' links' characteristics through exchanging beacon messages.

IV. Numerical Results

In this section, we present simulation results to demonstrate the efficacy of the proposed multi-path selection scheme for a streaming application.

A. Methodology

We use a simulation model based on NS-2 [16]. The distributed coordination function (DCF) of IEEE 802.11 for wireless LANs is used as the MAC layer. The radio model is a shared-media radio with a transmission rate of 2 Mbps, a radio range of 250 meters, and an interference range of 550 meters. A detailed description of the simulation environment and the models is available in [21].

Note that the wireless channel capacity can not be fully utilized due to the inefficiencies in the distributed nature of the 802.11 MAC protocol [22]. Based on our observation, we choose the empirical capacity, C, to be 1.0 Mbps, in our PDP estimation model. We assume that each node knows the flow rate of other links. In our simulations, we mostly study the case of static wireless ad hoc networks with stationary nodes, and assume PDP due to contention and channel error is very small after retransmissions. Thus in this section, the only contribution to PDP is assumed to be congestion.

We randomly choose one video sender and one video receiver. Standard MPEG QCIF sequence Foreman is MDC coded with MP-MDVC [2]. We encode each frame into two descriptions, and the group-of-pictures (GOP) size is chosen to be 15. Intra-frame encoding is identical for both descriptions. For each description, an I-frame is packetized into two packets, and two consecutive P-frames are packetized into one packet, in order to make each packet smaller than Maximum Transmission Unit (MTU) of 802.11's link layer, and to reduce the number of total packets. For the same visual quality, as measured by Peak Signal to Noise Ratio (PSNR), the bit rate needed for MDC is around 30% larger than that for Single Description Coding (SDC). This is due to inevitable compression inefficiency of MDC as compared to SDC [2][3].

To describe our performance metrics, we first define a "bad" frame.

Definition 2: A description of an I-frame is decodable at the playback deadline, if both packets corresponding to the description are received. A description of a P-frame is decodable, if at the playback deadline, both the packet corresponding

to the description is received, and the same description of the previous frame is decodable. A frame of a MDC stream is called a bad frame, if none of its two descriptions is decodable.

We evaluate the performance of video streaming applications using the following metrics:

a. **The ratio of bad frames:** The ratio of bad frames is the ratio of the number of bad frames to the total number of video frames that should have been decoded in the receiver. Note that the ratio of bad frames is different from packet delivery ratio or the number of packet loss bursts. This metric is related to the quality of received video due to the following two reasons. First it considers the dependency between different frames. Second this metric 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 received video is interrupted by the bad frames.

B. Verification of the Proposed Multipath Selection Model for Centralized IWM

In this section, we verify that concurrent PDP could be a reasonable indicator for streaming applications' performance. We do this by comparing the results of NS simulations for ratio of bad frames and that of the estimation model based on concurrent PDP. Intuitively, we would expect these two terms to be related, and that the lower concurrent PDP of two paths, the lower the ratio of bad frames observed at the receiver side.

We consider a grid network consisting of 49 nodes, placed in a 7×7 grid. The distance between neighboring nodes is 200 meters, slightly shorter than the communication range. We randomly choose a video sender and receiver. The shortest path between the sender and the receiver is five hops. The bitrate of the MDC video flow is 121.7 kbps. We insert 20 one-hop cross traffic flows, whose bit rates are uniformly selected in the range of [0,200.0] kbps, and packet size is 512 bytes.

We manually select six paths connecting the sender and the receiver, and consider 21 transmission scenarios as follows: 15 path pairs with all possible combinations of every two paths, plus 6 single paths. We transmit a different description of the video flow over each path in a path pair case, and both descriptions over one path in a single path case.

We obtain ratio of bad frames for different transmission scenarios through packet level NS simulations. Each simulation lasts 3000 seconds in order to obtain statistically reliable results. We also compute concurrent PDP for each transmission scenario through the estimation model summarized in Section II-F. We then order each transmission scenario based on bad frame ratio for NS simulation results and concurrent PDP for the estimation results, and show the rank of each transmission scenario in Figure 2. As seen, the results of the estimation model match those obtained by the NS simulation quite well. Based on the estimation model, the 3 best transmission scenarios are $1, 2$, and 3, which also happen to be best performing transmission scenarios according to NS simulations. This means that if we select the optimal transmission scenario based on the concurrent PDP estimation model of Section II-F, we are likely to have chosen the best performing transmission scenario in terms of the ratio of bad frames. We have also tested our PDP estimation model with other networks, whose nodes are placed randomly, and have reached similar conclusions [43].

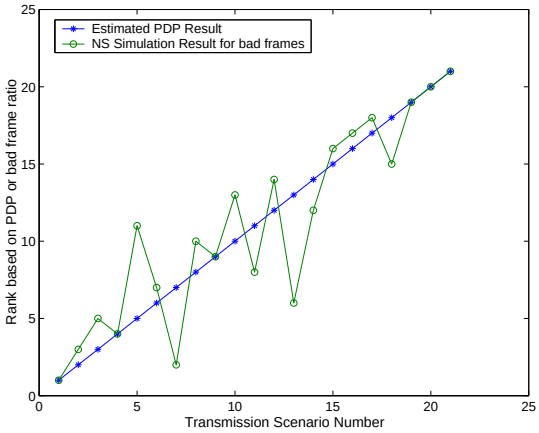


Fig. 2. Verification of the PDP model

C. Performance of the Centralized IWM

In this section, we use NS simulations to compare OMR, Centralized IWM, the node-disjoint multipath routing (NDM), and the shortest widest path routing (SWP) [19]. For OMR and IWM we use the PDP estimation method summarized in Section II-F, where flow rates are assumed to be perfectly known at every node.

We have tested these four protocols in the 7×7 grid network described in the previous section. The bit rates of cross flows are changed every 30 seconds. All other settings are identical to those of simulations in Section IV-B.

We run 30 simulations for different network topologies and select different senders and receivers in each scenario. Each simulation lasts 900 seconds. Figures 3(a) and 3(b) show the ratio of bad frames and the number of bad periods of all 30 runs of the four schemes respectively. We sort the results based on the performance of SWP. As seen, the average performance of IWM is very close to that of OMR, and is better than that of NDM and SWP, even though its computational complexity is as low as NDM and SWP. Note that although OMR should outperform IWM in theory, due to the approximations we make in computing the concurrent PDP, in a few runs IWM outperforms OMR.

We count a protocol as the best under a scenario, if its ratio of bad frames is no more than 5 percent higher than the lowest among all the other protocols. Specifically, as shown in Table II,IWM performs close to the best among all protocols in 26 out of 30 runs. Figure 4 shows the length of the achievable shortest path between the sender and the receiver in all 30 scenarios. IWM is particularly effective when the distance between the sender and the receiver is large, e.g. runs #18, #23, #26 . In this case, IWM distributes the video traffic between two paths which are far from each other. This has two advantages. First, packet drop over two paths far from each

other are independent. Second, the aggregate bandwidth of two paths far from each other is larger. Thus IWM outperforms NDM and SWP in this case. On the other hand, when the sender and the receiver are close to each other, e.g. runs #1, #2, #3, the gain brought by longer detoured paths are offset by the extra resource consumed. In this case, IWM selects two paths close to each other or even a single path, resulting in similar performance to NDM and SWP. Both the simulation results and the analysis show that the relaxation of the optimal multipath selection problem used by IWM is very efficient.

We have also run simulations over a random wireless network consisting of 100 nodes, distributed in a 1250 by 1250 meters square area. Similar conclusions are reached in this scenario [43].

We have also compared performance of NDM and Centralized IWM as a function of frequency of change in the bit rate of cross traffic flows. We assume that there are 20 one-hop random cross traffic flows, whose bit rates are randomly selected in the range of [0, 200] kbps. The bit rates of cross traffic flows are changed from once every second to once every thirty seconds. Each simulation lasts 1500 seconds. Tables IV and V show the ratio of bad frames and the number of bad periods of two schemes under different dynamic levels of cross traffic flows. As seen, Centralized IWM performs better than NDM under different dynamic levels of cross traffic flows. The proposed centralized IWM performs a little worse as the cross traffic flows change faster.

TABLE II

SUMMARY FOR THE GRID NETWORK: THE RATIO OF BAD FRAMES

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

TABLE III

SUMMARY FOR THE GRID NETWORK: THE NUMBER OF BAD PERIODS

	OMR	Centralized IWM	NDM	SWP
Average	74.7	79.1	186.0	153.3
Num. of Best	20	17	2	7

TABLE IV

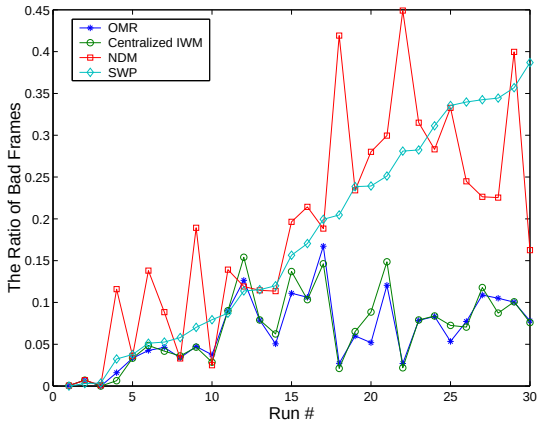
SUMMARY FOR THE GRID NETWORK WITH DYNAMIC CROSS TRAFFIC: THE RATIO OF BAD FRAMES

Traffic Change Interval (second)	1	2	5	10	30
NDM	0.1311	0.1223	0.1063	0.1064	0.1074
Centralized IWM	0.0666	0.0589	0.0553	0.0564	0.0545

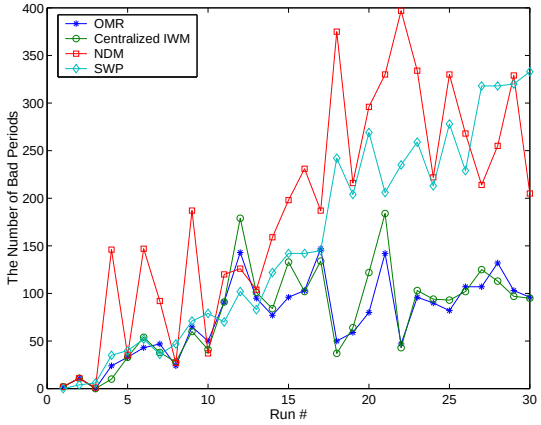
TABLE V

SUMMARY FOR THE GRID NETWORK WITH DYNAMIC CROSS TRAFFIC: THE NUMBER OF BAD PERIODS

Traffic Change Interval (second)	1	2	5	10	30
NDM	318	270	223	220	220
Centralized IWM	83	84	80	75	70



(a)



(b)

Fig. 3. Simulation Results comparing OMR, IWM, NDM, SWP on the 7×7 Grid Network for 30 runs: (a) The ratio of bad frames; (b)The number of bad periods.

D. Performance of the Distributed IWM

In this section, we compare Centralized IWM, Distributed IWM, NDM and SWP. We test these four protocols in the 7×7 grid network described in Section IV-C. The bit rates of cross flows are changed every 100 seconds. All other settings are identical to those of simulations in Section IV-C. We run 30 simulations for different network topologies and select different senders and receivers in each scenario. Each simulation lasts 1500 seconds. There are 20 cross traffic flows in the network, whose bitrates are selected uniformly between 0 and 180 kbps.

Figures 5(a) and 5(b) show the ratio of bad frames and the number of bad periods of all 30 runs of the four schemes respectively. We sort the results based on the performance of distributed IWM. Simulation results show that the average performance of Distributed IWM is better that of NDM and SWP, but is worse than Centralized IWM. As shown in Table VI, the average ratio of bad frames of Centralized IWM, Distributed IWM, NDM and SWP are 0.0334, 0.0684, 0.1041, 0.1009 respectively. Distributed IWM has the lowest ratio of bad frames among all protocols in 18 out of 30 runs, and has the lowest number of bad periods in 13 out of 30 runs. In summary, the performance of Distributed IWM lies between Centralized IWM and NDM or SWP. There are mainly two

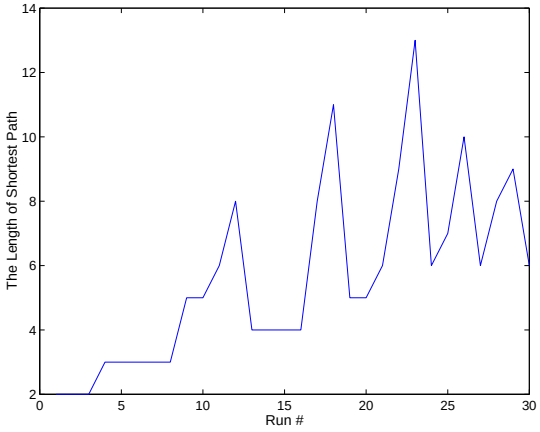


Fig. 4. Length of the achievable Shortest Path for the 7×7 Grid Network

reasons for the performance gap between Distributed IWM and Centralized IWM. First, since Distributed IWM only uses local knowledge of each node, the path cost computed by the protocol is not as accurate as that by Centralized IWM. A node does not learn the accurate topology in its two hop neighborhood, and thus does not model the interference around it as accurately as Centralized IWM does. Second, during the process of flooding RREQ messages to the network, and sending RRER messages back to the sender, some useful RREQ messages might be dropped or lost in the middle of the network, thus the sender might not obtain the best path all the time.

TABLE VI
PERFORMANCE OF THE DISTRIBUTED IWM IN A GRID NETWORK: THE RATIO OF BAD FRAMES

	Centralized IWM	Distributed IWM	NDM	SWP
Average	0.0334	0.0684	0.1041	0.1009
Num. of Best	28	18	13	10

TABLE VII
PERFORMANCE OF THE DISTRIBUTED IWM IN A GRID NETWORK: THE NUMBER OF BAD PERIODS

	Centralized IWM	Distributed IWM	NDM	SWP
Average	74.8	109.5	189.9	157.0
Num. of Best	23	13	10	9

V. TESTBED IMPLEMENTATION AND EVALUATION

To demonstrate the feasibility of the proposed multipath selection framework and the Centralized IWM, we have built a small wireless ad hoc network testbed, consisting of 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.

A. Software Architecture

We have implemented the proposed IWM protocol in the Mesh Connectivity Layer (MCL), which is an ad hoc routing

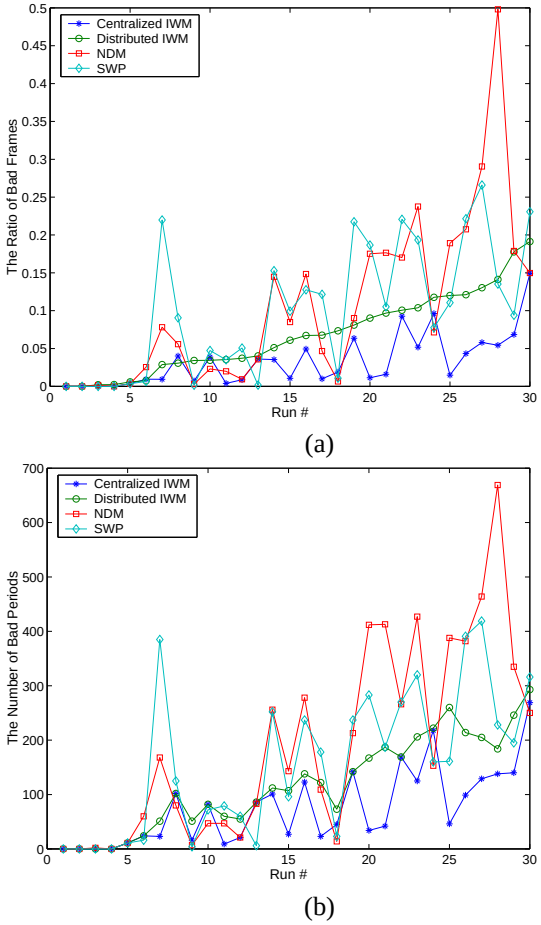


Fig. 5. Performance of the Centralized IWM, Distributed IWM, NDM, SWP in a Grid Network for 30 runs: (a) The ratio of bad frames; (b)The number of bad periods.

framework provided by Microsoft Research [30]. 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. The LQSR supports different link-quality metrics, e.g. Weighted Cumulative Expected Transmission Time (WCETT) and Expected Transmission Count (ETX) [30]. 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 performance to IWM. However, LQSR attempts to obtain the path with the largest bandwidth, rather than the one with largest available bandwidth. Specifically, 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 both cross traffic flows and video flow in order to compute PDP. As such, the two paths obtained by IWM adapt to available 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 II-F. We set α in Equation (16) to be 0.1.

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.

In our multipath selection framework, we need a 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. For simplicity, we opt to employ an Additive Increase Multiplicative Decrease (AIMD) algorithm, which is the default congestion control mechanism used in TCP today [43].

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.

B. 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 were 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.

C. 802.11g wireless ad hoc network result

We first performed a series of tests to show the performance of our proposed IWM in 802.11g wireless ad hoc network. We carried out eight 300 second long experiments. Only nodes 1 to 9 are activated in this scenario. Nodes 1 and 2 are MDC and SDC video senders respectively, and nodes 8 and 9 are MDC and SDC video receivers respectively. In ad hoc mode, both Netgear and Linksys cards' maximum throughput is only 11 Mbps.

We compare our proposed IWM and MDC with LQSR using metric WCETT and SDC. Metric WCETT has been shown to be more effective than other path selection metrics, e.g. ETX and shortest path, for single path routing [30].

Figure 6 shows the result of PSNR of the received video for all eight runs. In seven out of eight runs, IWM outperforms LQSR by several dBs, and on average, IWM outperforms LQSR by 2.8 dB. Since MDC's bit rate is 30% higher than SDC's, it is possible that in some scenario, i.e. run 6, LQSR with SDC outperforms IWM with MDC.

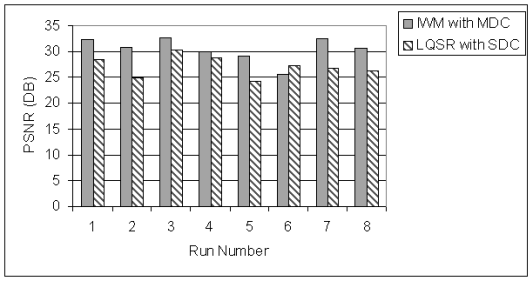


Fig. 6. PSNR performance evaluation for IWM/MDC in 802.11g.

D. 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 7(a) and 7(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

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

again, this shows the advantage of path diversity with MDC over single path transmission of SDC video.

Figure 8 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 7.

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 10 and 9 respectively. IWM/MDC outperforms LQSR/SDC by 1.1 dB in run 7. As seen in Figure 9(a), for IWM/MDC, PSNR drops gracefully, when there is packet loss in only one substream. As seen in Figure 9(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 10(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 9(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.

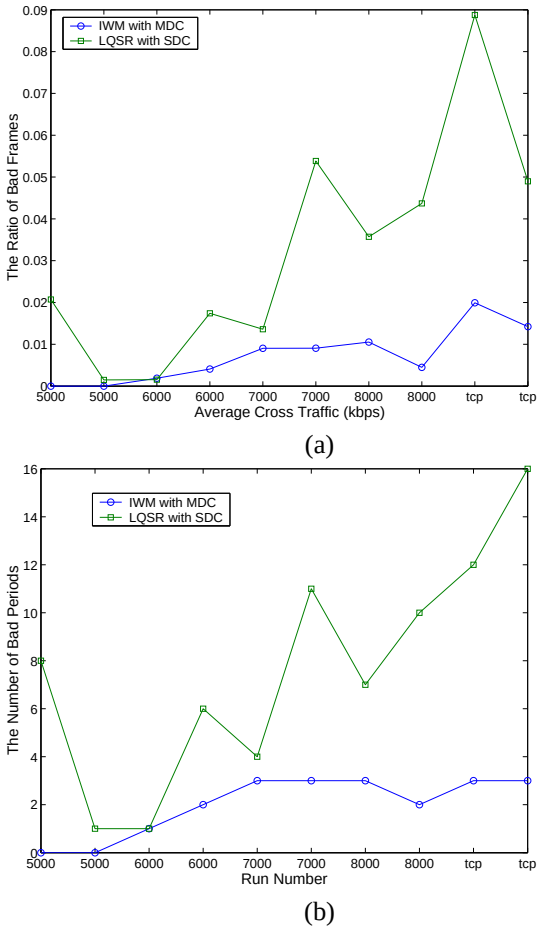


Fig. 7. Performance Evaluation in 802.11a (a) The ratio of bad frames; (b)The number of bad periods.

E. 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 11 and 12 show the results of three 600 seconds experimental run.

As seen in Figure 11, 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 12 compares PSNR of two schemes. Averaged over three runs, IWM outperforms LQSR by 2.1 dBs.

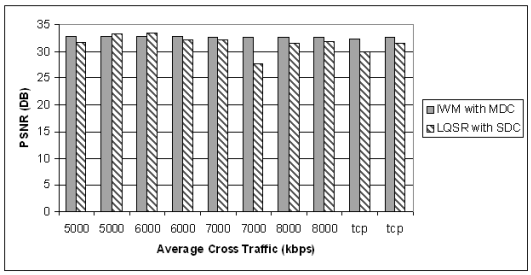


Fig. 8. PSNR performance evaluation for IWM/MDC in 802.11a.

VI. Conclusions and Future Work

In this paper, we propose a novel multipath streaming framework in order to provide robustness in video communication applications over wireless ad hoc networks. 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, which optimizes the worst case MDC video quality over all times. Then we propose a heuristic IWM routing protocol based on our path selection model, whose performance is shown to be close to that of the ”optimal routing”, and better than that of existing schemes, through both NS simulations and actual experiments in a testbed.

In this paper, we show the performance of our proposed protocol when there is only one video session in the network. It would be interesting to see how multiple video sessions perform with our protocol. It is possible to prevent multiple

video sessions from selecting paths simultaneously by applying random jitter to path update interval of each video session. Using our protocol, the second video session would not select links used by the first video session, if these links do not have sufficient bandwidth for two streams. Another area of future work is to investigate how the performance of our proposed protocol scales with network size. We would also like to study the tradeoff between protocol overhead and performance in the future.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Jean Walrand, Rarjarshi Gupta in EECS of UC Berkeley for the helpful discussions. The authors also thank Cindy Liu, Allan Gu, Minghua Chen, Gap N Thirathon, Steven Jian, and John Lo for their help running the testbed experiments.

REFERENCES

[1] V. Goyal, "Multiple description coding: compression meets the network", *IEEE Signal Processing Magazine*, Vol. 18, No. 5, Sept. 2001, pp. 74 - 93.

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

[3] 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.

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

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

[6] 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.

[7] 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*.

[8] 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.

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

[10] 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.

[11] 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.

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

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

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

[15] P. Gupta, and P. R. Kumar, "The Capacity of Wireless Networks", *IEEE Trans. on Information Theory*, Vol. 34, No. 5, pp. 910-917, 2000.

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

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

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

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

[20] J. Moy, "OSPF Version 2", *RFC 1583*, March 1994.

[21] J. Broch, D.A. Maltz, D.B. Johnson, et al. "A performance comparison of multi-hop wireless ad hoc network routing protocols", *ACM MobiCom 1998*, pp.85-97.

[22] G.Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", *IEEE Journal on Selected Areas in Communicatioins"*, vol. 18, no. 3, March 2000.

[23] 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", *Proceedings of the 2001 Fall IEEE Vehicular Technology Conference*, Atlantic City, NJ, October 2001, pp.615-619.

[24] T. Nguyen and A. Zakhor, "distributed video streaming over internet", *in Multimedia Computing and Networking 2002, Proceedings of SPIE*, San Jose, California, January 2002, Vol. 4673, pp. 186-195.

[25] T. Nguyen, and A. Zakhor, "distributed video streaming with forward error correction", *Packet Video 2002*, Pittsburgh, April 2002.

[26] D. Jurca, and P. Frossard, "Media-Specific Rate Allocation in Multi-path Networks", *Signal Processing Institute Technical Report - TR-ITS-2005.032*, Mar. 2006.

[27] C. Kuo and C. Liang, "A Meshed Multipath Routing Protocol in Mobile Ad Hoc Networks *", Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06)*, pp. 306-310.

[28] L. Reddy and S. Raghavan, "SMORT: Scalable multipath on-demand routing for mobile ad hoc networks", *Ad Hoc Networks*, Vol. 5, Issus 2, March 2007, pp. 162-188.

[29] Y. Ping, B. Yu, and W. Hao, "A Multipath Energy-Efficient Routing Protocol for Ad hoc Networks", *2006 International Conference on Communications, Circuits and Systems Proceedings*, June 2006, pp. 1462-1466.

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

[31] 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.

[32] S.J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," *ICC 2001*, pp.3201-3205.

[33] 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.

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

[35] 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.

[36] 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.

[37] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", RFC 3626, Oct. 2003.

[38] R. Ogier, M. Lewis, and F. Templin. "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). *IETF Internet RFC 3684*, http://www.ietf.org/rfc/rfc3684.txt.

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

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

[41] A. Bruce McDonald and T.F. Znabi, "A path availability model for wireless ad hoc networks", Proc. IEEE WCNC, New Orlean, USA, (Sept. 1999), pp.35-40.

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

[43] W. Wei, "Multipath Unicast and Multicast Video Communication over Wireless Ad Hoc Networks", PhD thesis, University of California, Berkeley, 2006, http://www-video.eecs.berkeley.edu/papers/weiwei/WeiThesis.pdf.

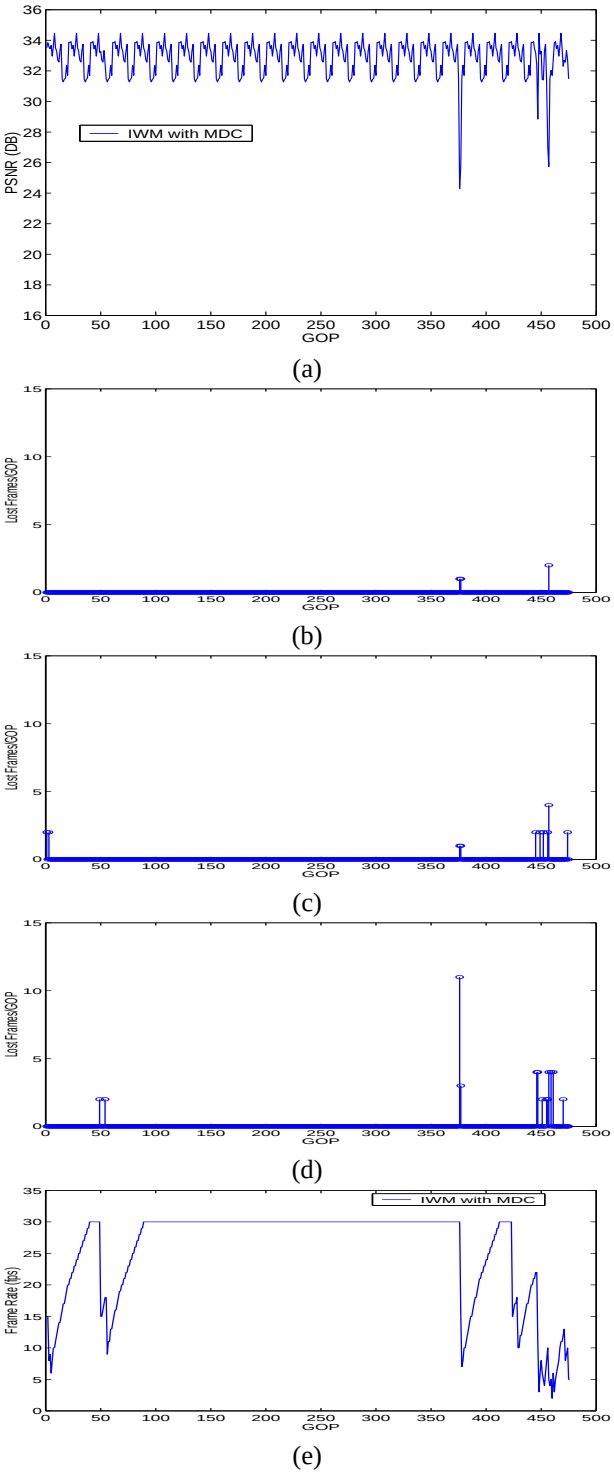


Fig. 9. Performance Evaluation in 802.11a (a) PSNR of the received frames using IWM/MDC; (b) Number of Frames lost in both descriptions; (c) Lost frames per GOP for substream 0; (d) Lost frames per GOP for substream 1; (e) Sending frame rate.

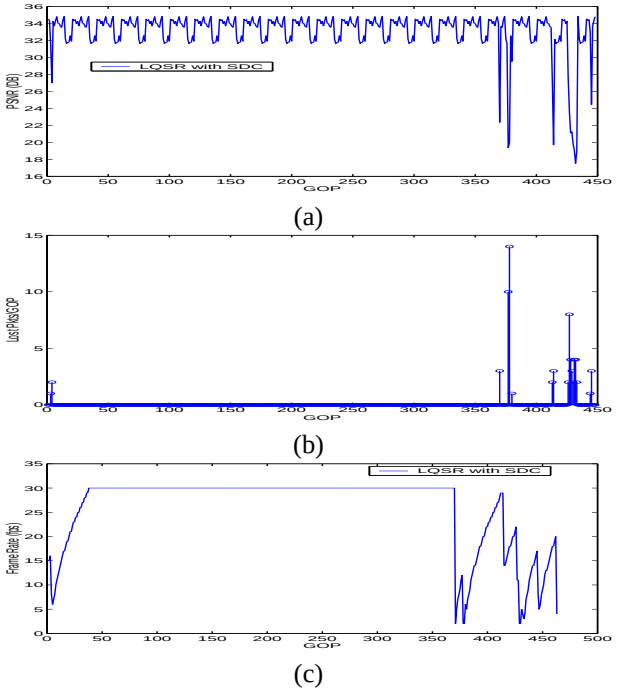


Fig. 10. Performance Evaluation in 802.11a (a) PSNR of the received frames using LQSR/SDC; (b) Lost frames per GOP for the stream; (c) Sending frame rate.

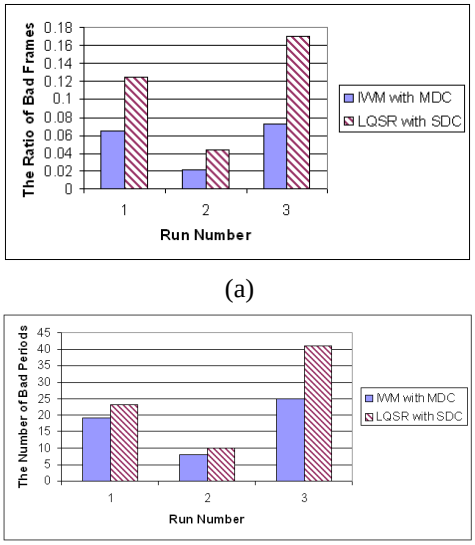


Fig. 11. Performance Evaluation in 802.11a with moving node (a) The ratio of bad frames; (b)The number of bad periods.

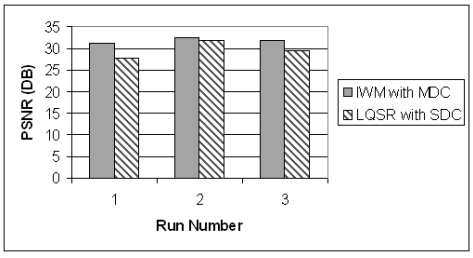


Fig. 12. Performance evaluation for IWM/MDC (802.11a with moving node): PSNR.