

# Experimental Evaluation of Large Scale WiFi Multicast Rate Control

Varun Gupta<sup>†</sup>, Craig Gutterman<sup>†</sup>, Yigal Bejerano<sup>\*</sup>, Gil Zussman<sup>†</sup>

<sup>†</sup> Electrical Engineering, Columbia University, New York, NY, USA.

<sup>\*</sup> Bell Labs, Nokia, Murray Hill, NJ, USA.

Email: {varun@ee, clg2186}.columbia.edu, bej@research.bell-labs.com, gil@ee.columbia.edu

**Abstract**—WiFi multicast to *very large groups* has gained attention as a solution for multimedia delivery in crowded areas. Yet, most recently proposed schemes do not provide performance guarantees and *none have been tested at scale*. To address the issue of providing high multicast throughput with performance guarantees, we present the design and experimental evaluation of the Multicast Dynamic Rate Adaptation (*MuDRA*) algorithm. *MuDRA* balances fast adaptation to channel conditions and stability, which is essential for multimedia applications. *MuDRA* relies on feedback from some nodes collected via a light-weight protocol and dynamically adjusts the rate adaptation response time. Our experimental evaluation of *MuDRA* on the ORBIT testbed with over 150 nodes shows that *MuDRA* outperforms other schemes and supports high throughput multicast flows to hundreds of receivers while meeting quality requirements. *MuDRA* can support multiple high quality video streams, where 90% of the nodes report excellent or very good video quality.

## I. INTRODUCTION

Multimedia (e.g., video) delivery is an essential service for wireless networks and several solutions were proposed for crowded venues [1], [2]. Most of them are based on dense deployments of Access Points (APs) and require considerable capital and operational expenditure, may suffer from interference between APs, and may exacerbate hidden node problems [3]. *Multicast* offers another approach for video delivery to large groups of users interested in venue specific content (e.g., sports arenas, entertainment centers, and lecture halls). However, WiFi networks provide *limited multicast support at a low rate* (e.g., 6Mbps for 802.11a/g) *without a feedback mechanism that guarantees service quality*. To improve performance, there is a need for a multicast system that *dynamically adapts the transmission rate* [4]. Yet, designing such a system poses several challenges, as outlined below.

**Multicast Rate Adaptation (RA) - Challenges:** A key challenge in designing multicast RA schemes for large groups is to obtain accurate quality reports with low overhead. Some systems [5], [6] experimentally demonstrated impressive ability to deliver video to a few dozen nodes by utilizing Forward Error Correction (FEC) codes and retransmissions. However, most approaches do not scale to very large groups with hundreds of nodes, due to the following:

(i) Most schemes tune the rate to satisfy the receiver with the worst channel condition. As shown in [7], [8] in crowded venues, a few unpredictable outliers, referred to as *abnormal nodes*, may suffer from low SNR and Packet Delivery Ratio (PDR) even at the lowest rate and without interference.

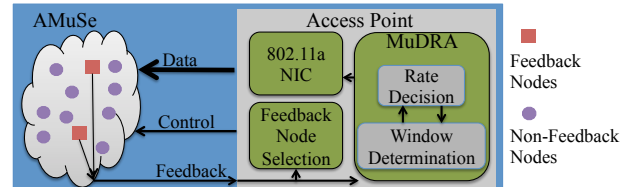


Fig. 1. The Adaptive Multicast Services (*AMuSe*) system consisting of the Multicast Dynamic Rate Adaptation (*MuDRA*) algorithm and a multicast feedback mechanism.

This results from effects such as multipath and fast fading. Therefore, *a multicast scheme cannot provide high rate while ensuring reliable delivery to all users*.

(ii) It is impractical to continuously collect status reports from all or most users without hindering performance. Even if feedback is not collected continuously, a swarm of retransmission requests may be sent following an interference event, thereby causing additional interruptions.

To overcome these challenges, a multicast system should *conduct efficient RA based on only limited reports from the nodes*. We have been developing the Adaptive Multicast Services (*AMuSe*) system for content delivery over WiFi multicast. In our recent papers [7], we focused on efficient feedback collection mechanisms for WiFi multicast as part of the *AMuSe* system. In this paper, we present the Multicast Dynamic Rate Adaptation (*MuDRA*) algorithm. *MuDRA* leverages the efficient multicast feedback collection of *AMuSe* and *dynamically adapts the multicast transmission rate to maximize channel utilization while meeting performance requirements*. Fig. 1 shows the overall *AMuSe* system composed of (i) *MuDRA* algorithm, and (ii) a feedback mechanism. Before describing *MuDRA*, the overall *AMuSe* system design, and our contributions in detail, we now first briefly outline the related work relevant to the system design.

## A. Related Work

Unicast RA, multicast feedback schemes, and multicast RA have received considerable attention (see survey in [9]).

**Unicast RA:** We discuss unicast RA schemes, since they can provide insight into the design of multicast RA. In *Sampling-based algorithms*, both ACKs after successful transmissions and the relation between the rate and the success probability are used for RA after several consecutive successful or failed transmissions [10]–[12]. The schemes in [13], [14] distinguish between losses due to poor channel conditions and collisions,

and update the rate based on former. However, such schemes cannot support multicast, since multicast packets are not acknowledged. In *Measurement-based schemes* the receiver reports the channel quality to the sender which determines the rate [15]–[19]. Most measurement-based schemes modify the wireless driver on the receiver end and some require changes to the standard, which we avoid.

**Multicast Feedback Schemes:** Some schemes rely on *individual feedback* from all users for each packet [20], [21]. On the other hand, *Leader-Based Schemes* [6], [22] collect feedback from a few receivers with the weakest channel quality. Pseudo-Multicast schemes [5] convert the multicast feed to a unicast flow and send it to one leader that acknowledges the reception while the other nodes receive packets in promiscuous mode. *Cluster-Based Feedback Schemes* [7] balance accurate reporting with minimization of control overhead by selecting nodes with the weakest channel condition in each cluster as *Feedback (FB) nodes*.

**Multicast RA:** In [5], [22], [23] the sender uses feedback from leaders (nodes with worst channel conditions) for RA. In [24] when the channel conditions are stable, RA is conducted based on reports of a single leader. When the channel conditions are dynamic, feedback is collected from all nodes. Medusa [6] combines Pseudo-Multicast with infrequent application layer feedback reports from all nodes. The MAC layer feedback sets backoff parameters while application layer feedback is used for RA and retransmissions of video packets. Recently, in [7] we considered multicast to a large set of nodes and provided a rudimentary RA scheme which is not designed to achieve optimal rate, maintain stability, or respond to interference.

## B. Our Contributions

We present a multicast rate adaptation algorithm *MuDRA* which is designed to support WiFi multicast to hundreds of users in crowded venues. *MuDRA* can provide high throughput while ensuring high *Quality of Experience (QoE)*. *MuDRA* benefits from a large user population, which allows selecting a small yet sufficient number of Feedback (FB) nodes with marginal channel conditions for monitoring the quality. We address several design challenges related to appropriate configuration of the feedback level. We note that using *MuDRA* does not require any modifications to the IEEE 802.11 standard or the mobile devices. We implemented *MuDRA* with the *AMuSe* system on the ORBIT testbed [25], evaluated its performance with all the operational IEEE 802.11 nodes (between 150–200), and compared it to other multicast schemes. Our key contributions are:

**(i) The need for RA:** We empirically demonstrate the importance of RA. Our experiments on ORBIT show that when the multicast rate exceeds an optimal rate, termed as *target-rate*, numerous receivers suffer from low PDR and their losses cannot be recovered. We also observed that even a controlled environment, such as ORBIT, can suffer from significant interference. These observations constitute the need for a stable and interference agnostic RA algorithm that *does not exceed the target-rate*.

**(ii) Practical method to detect the target-rate:** Pseudo-multicast schemes that rely on unicast RA [5] may occasionally sample higher rates and retreat to a lower rate after a few failures. Based on the observation above about the target rate, schemes with such sampling mechanisms will provide low QoE to many users. To overcome this, we developed a method to detect when the system operates at the target-rate, termed the **target condition**. Although the target condition is sufficient but not necessary, our experiments show that it is almost always satisfied when transmitting at the target-rate. *MuDRA* makes RA decisions based on the target condition and employs a dynamic window based mechanism to avoid rate changes due to small interference bursts.

**(iii) Extensive experiments with hundreds of receivers:** Our experiments demonstrate that *MuDRA* swiftly converges to the target-rate, while meeting the Service Level Agreement (SLA) requirements (e.g., ensuring PDR above 85% to at least 95% of the nodes). Losses can be recovered by using appropriate application-level FEC methods [26], [27].

*MuDRA* is experimentally compared to (i) pseudo-multicast with a unicast RA [28], (ii) fixed rate, and (iii) a rate adaptation mechanism proposed in [7] which we refer to as Simple Rate Adaptation (SRA) algorithm. *MuDRA* achieves 2x higher throughput than pseudo-multicast while sacrificing PDR only at a few poorly performing nodes. While the fixed rate and SRA schemes can obtain similar throughput as *MuDRA*, they do not meet the SLA requirements. Unlike other schemes, *MuDRA* preserves high throughput even in the presence of interference. Additionally, *MuDRA* can handle significant node mobility. Finally, we devise a live multicast video delivery approach for *MuDRA*. We show that in our experimental settings with target rate of 24–36Mbps, *MuDRA* can deliver 3 or 4 high definition H.264 videos (each one of 4Mbps) where over 90% of the nodes receive video quality that is classified as excellent or good based on user perception.

To summarize, to the best of our knowledge, *MuDRA* is the first multicast RA algorithm designed to satisfy the specific needs of multimedia/video distribution in crowded venues. Moreover, *AMuSe* in conjunction with *MuDRA* is the first multicast content delivery system that has been evaluated *at scale*. Due to space constraints, several design details and results are omitted and can found in a technical report [29].

## II. TESTBED AND KEY OBSERVATIONS

We evaluate *MuDRA* on the ORBIT testbed [25], which is a dynamically configurable grid of  $20 \times 20$  (400) 802.11 nodes where the separation between nodes is 1m. It is a good environment to evaluate *MuDRA*, since it provides a very large and dense population of wireless nodes, similar to the anticipated crowded venues.

**Experiments:** To avoid performance variability due to a mismatch of WiFi hardware and software, only nodes equipped with *Atheros 5212/5213* cards with *ath5k* driver were selected. For each experiment we activated all the operational nodes that meet these specifications (between 150 and 250 nodes). In all the experiments, one corner node served as a single

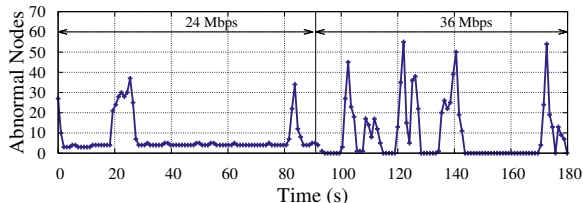


Fig. 2. Experimental measurement of the number of abnormal nodes in time, for fixed rates of 24 and 36Mbps.

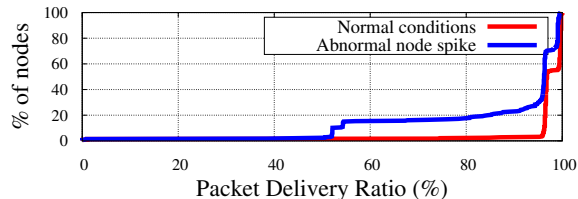


Fig. 3. The CDF of the PDR values of 170 nodes during normal operation and during a spike at rate of 36Mbps.

multicast AP. The other nodes were multicast receivers. The AP used 802.11a to send a multicast UDP flow, where each packet was 1400 bytes. *The AP used the lowest supported transmission power of  $1mW = 0dBm$  to ensure that the channel conditions of some nodes are marginal.*

**Interference and Stability:** We study the time variability of the channel conditions on the ORBIT testbed by measuring the number of nodes with low PDR (below a threshold of 85%). We call these nodes *abnormal nodes* (the term will be formally defined in Section III). The number of abnormal nodes out of 170 nodes for rates of 24 and 36Mbps is shown in Fig. 2. We repeated these experiments several times and observed that even at a low rate, the channel may suffer from sporadic interference events, which cause a sharp increase in the number of abnormal nodes. These interference spikes caused by non-WiFi devices are beyond our control and their duration varies in time.

Fig. 3 provides the Cumulative Distribution Function (CDF) of the PDR values with and without sporadic interference. The figure shows that during a spike, over 15% of the nodes suffer from PDR around 50%. Further, the location of the nodes affected by the spikes varies with time and does not follow a known pattern. These experiments show that even in a seemingly controlled environment, *nodes may suffer from sporadic continuous interference, which may cause multicast rate fluctuations.* Users are very sensitive to changes in video quality [30], and therefore, to keep a high QoE we would like to avoid rate changes due to sporadic interference.

### III. NETWORK MODEL AND OBJECTIVE

We consider a WiFi LAN with multiple APs and frequency planning such that the transmissions of adjacent APs do not interfere with each other. Thus, *for RA we consider a single AP with  $n$  associated users.* We assume low mobility (e.g., users watching a sports event). Although we consider a controlled environment, the network may still suffer from sporadic interference, as shown in Section II. The main notation used

TABLE I  
NOTATION AND PARAMETER VALUES USED IN EXPERIMENTS.

Symbol	Semantics	Exp. Val.
$n$	Number of nodes associated with the AP.	$> 150$
$X$	<i>Population threshold</i> - Minimal fraction of nodes that should experience high PDR.	95%
$A_{max}$	Maximal number of allowed abnormal nodes.	8
$L$	<i>PDR threshold</i> - Threshold between acceptable (normal) and low (abnormal) PDR.	85%
$H$	Threshold between high PDR and mid-PDR.	97%
$K$	Expected number of FB nodes.	30
$A_t$	Number of abnormal nodes at time $t$ .	
$M_t$	Number of mid-PDR FB nodes at time $t$ .	
$W_{min}$	Minimal RA window size (multiples of reporting intervals).	8
$W_{max}$	Maximal RA window size.	32

in the paper is summarized in Table I. Specifically, a *PDR-Threshold  $L$* , is defined such that a node has high QoE if its PDR is above  $L$ . Such a node is called a *normal node*. Otherwise, it is considered an *abnormal node*.

Our objective is to develop a practical and efficient rate control system which satisfies the following requirements:

**(R1) High throughput** – Operate at the highest possible rate, i.e., the *target rate*, while preserving SLAs.

**(R2) Service Level Agreements (SLAs)** – Given  $L$  (e.g.,  $L = 85\%$ ), and a *Population-Threshold  $X$*  (e.g.,  $X = 95\%$ ), the selected rate should guarantee that at least  $X\%$  of the nodes experience PDR above  $L$  (i.e., are normal nodes). Except for short transition periods, this provides an upper bound of  $A_{max} = \lceil n \cdot (1 - X) \rceil$  on the number of permitted abnormal nodes.

**(R3) Scalability** – Support hundreds of nodes.

**(R4) Stability** – Avoid rate changes due to sporadic channel condition changes.

**(R5) Fast Convergence** – Converge fast to the target rate after long-lasting changes (e.g., user mobility or network changes).

**(R6) Standard and Technology Compliance** – No change to the IEEE 802.11 standard or operating system of the nodes.

### IV. MULTICAST RATE ADAPTATION

The overall multicast rate adaptation process of *MuDRA* as a part of the *AMuSe* system relies on three main components, as illustrated in Fig. 1 and discussed below. We first provide a high level description of each component and then discuss the details in the following subsections.

**(i) Feedback (FB) Node Selection:** Selects a small set of *FB nodes* that provide reports for making RA decisions. We describe the FB node selection process in Section IV-A and calculate the reporting interval duration in Section V.<sup>1</sup>

The following two components compose the *MuDRA* Algorithm (Algorithm 1). It collects the PDR values from the FB nodes, updates their status (normal or abnormal), invokes the GETRATE procedure, which calculates the desired rate, and invokes the GETWINDOWSIZE procedure, which determines the window size of rate updates (to maintain stability).

<sup>1</sup>Unlike in unicast where each packet is acknowledged, *MuDRA*'s reporting intervals are long (in the experiments we consider 2 reports per second).

---

**Algorithm 1** *MuDRA* Algorithm

---

```
1:  $rate \leftarrow lowestRate, window \leftarrow W_{min}, changeTime \leftarrow t,$   
    $refTime \leftarrow t, t := current\ time$   
2: while ( $true$ ) do  
3:   Get PDR reports from all FB nodes  
4:   Get Status of each FB node  $i$   
5:   Calc  $\hat{A}_t$  and  $\hat{M}_t$   
6:    $rate, action, changeTime \leftarrow GetRate(...)$   
7:    $window, refTime \leftarrow GetWinSize(...)$   
8:   set multicast rate to  $rate$   
9:   sleep one reporting interval
```

---

---

**Procedure 1** Rate Decision

---

```
1: procedure GETRATE( $rate, window, changeTime, t$ )  
2:    $action \leftarrow Hold$   
3:   if ( $t - changeTime$ ) >  $window$  then  
4:      $canDecrease \leftarrow true, canIncrease \leftarrow true$   
5:     for  $\tau \leftarrow 0$  to  $window$  do  
6:       if  $\hat{A}_{t-\tau} < A_{max}$  then  
7:          $canDecrease \leftarrow false$   
8:       else if  $\hat{A}_{t-\tau} + \hat{M}_{t-\tau} > A_{max}$  then  
9:          $canIncrease \leftarrow false$   
10:    if  $canDecrease$  and  $rate > rate_{min}$  then  
11:       $rate \leftarrow NextLowerRate$   
12:       $action \leftarrow Decrease, changeTime \leftarrow t$   
13:    if  $canIncrease$  and  $rate < rate_{max}$  then  
14:       $rate \leftarrow NextHigherRate$   
15:       $action \leftarrow Increase, changeTime \leftarrow t$   
16: return  $rate, action, changeTime$ 
```

---

(ii) **Rate Decision** (Procedure 1): Utilizes the limited and infrequent FB reports to determine the highest possible rate, termed the *target-rate*, while meeting the requirements in Section III. The rate decisions (lines 5–15) rely on rate decision rules that are described in Section IV-B. To maintain rate stability, rate change operations are permitted, only if the conditions for rate change are satisfied for time equal to a window size (determined by the *Stability Preserving Method*).

(iii) **Stability Preserving Method** (Procedure 2): A window based method that maintains rate stability in the event of sporadic interference and after an RA decision. It follows the classical Additive Increase Multiplicative Decrease (AIMD) approach. The duration of the time window varies according to the network and channel characteristics (e.g., the typical duration of interference). More details appear in Section IV-C.

#### A. Feedback Node Selection

*MuDRA* uses a simple and efficient mechanism based on a quasi-distributed FB node selection process, termed *K-Worst* [7], where the AP sets the number of FB nodes and their reporting rates.  $K$  nodes with the worst channel conditions are selected as FB nodes (the node's channel condition is determined by its PDR). Hence, the selection process ensures an upper bound on the number of FB messages, regardless of the multicast group size. This upper bound is required for limiting the interference from FB reports, as explained in Section V. More details about the FB protocol are in [29].

#### B. Rate Decision Rules and Procedure

In this subsection, we describe the *target condition* which is an essential component of the rate selection rules. Then, we describe the rules and the corresponding Procedure 1.

---

**Procedure 2** Window Size Determination

---

```
1: procedure GETWINSIZE( $Action, window, refTime, t$ )  
2:   if  $Action = Decrease$  then  
3:      $window \leftarrow \min(W_{max}, 2 \cdot window), refTime \leftarrow t$   
4:   else if  $Action = Increase$  then  
5:      $refTime \leftarrow t$   
6:   else if ( $t - refTime$ ) >  $thresholdTime$   
7:     and  $Action = Hold$  then  
8:        $window \leftarrow \max(W_{min}, window - 1)$   
9:        $refTime \leftarrow t$   
10: return  $window, refTime$ 
```

---

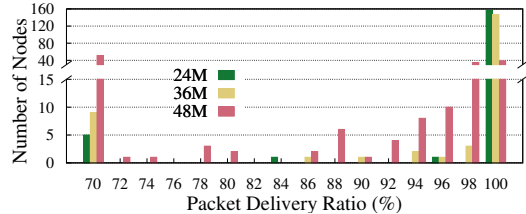


Fig. 4. The PDR distribution of one set of experiments with  $TX_{AP}$  rates of 24, 36, and 48Mbps.

**The Target Condition:** At a given time, the FB reports are available only for the current rate. To detect the target-rate, most RA schemes occasionally sample higher rates. However, the following experiment shows that this approach may cause undesired disruption to many receivers. We evaluated the PDR distribution of 160 – 170 nodes for different multicast transmission rates, denoted as  $TX_{AP}$  for 3 different experiment runs on different days. Fig. 4 shows the number of nodes in different PDR ranges for  $TX_{AP}$  values of 24, 36, and 48Mbps for one experiment with 168 nodes. When  $TX_{AP}$  is at most 36Mbps, the number of abnormal nodes is very small (at most 5). However, when  $TX_{AP}$  exceeds 36Mbps, the PDR of many nodes drops significantly. In this experiment 47 nodes became abnormal nodes which is more than  $A_{max} = 8$  (for  $X = 95\%$ ). We observed similar results in other experiments. Thus, in this case, the *target rate* is 36Mbps which is the highest rate above which the SLA requirements will be violated. We observed similar results for other experiments as well.

A key challenge is to *determine if the AP operates at the target-rate, without FB reports from higher rates*. We refer to this assessment as the *target condition*. Unfortunately, the target-rate cannot be detected from RF measurements, such as SNR. As shown in [31], [32] different nodes may have different receiver sensitivities, which may result in substantial PDR gaps between nodes with similar RF measurements. However, large scale multicast environments enable us to efficiently predict the target condition as described next.

From Fig. 4, we obtain the following important observation. *Observation 1:* When operating below the target-rate, almost all the nodes have PDR close to 100%. However, when operating at the target-rate, noticeable number of receivers experience PDR below 97%. At 36Mbps, 17 nodes had PDR below 97%, which is substantially more than  $A_{max} = 8$ .

Fig. 5 shows the average percentage of nodes that remain normal vs. their initial PDR when increasing  $TX_{AP}$  from 36Mbps to 48Mbps averaged for 3 different sets of experi-



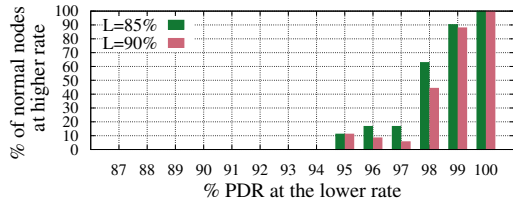


Fig. 5. The percentage of nodes that remain normal after increasing the  $TX_{AP}$  from 36Mbps to 48Mbps vs. their PDR values at the 36Mbps for different PDR-thresholds ( $L$ ).

ments. The total number of nodes in these experiments was 168. We derive the following observation from Fig. 5.

**Observation II:** There is a PDR threshold,  $H = 97\%$ , such that every node with PDR between  $L$  and  $H$  becomes abnormal after the rate increase with very high probability. Note that 97% is the highest threshold for which this observation holds. We refer to these nodes as *mid-PDR nodes*.

Observation II is not surprising. As reported in [31], [33], each receiver has an SNR band of 2 – 5dB, in which its PDR drops from almost 100% to almost 0%. The SNR of mid-PDR nodes lies in this band. Increasing the rate requires 2 – 3dB higher SNR at the nodes. Hence, mid-PDR nodes with SNR in the transition band before the rate increase will be below or at the lower end of the transition band after the increase, and therefore, become abnormal nodes.

In summary, Observations I and II imply that it is possible to assess the target condition by monitoring the nodes close to transitioning from normal to abnormal. Let  $A_t$  and  $M_t$  denote the number of abnormal and mid-PDR nodes at time  $t$ , respectively. We obtain the following empirical property.

**Property 1 (Target Condition):** Assume that at a given time  $t$ , the following condition holds,

$$A_t \leq A_{max} \quad \text{and} \quad A_t + M_t > A_{max} \quad (1)$$

then almost surely, the AP transmits on the target-rate at time  $t$ . This is sufficient but not a necessary condition.

It is challenging to analytically predict when the target condition is satisfied with the available FB information and without a model of the receiver sensitivity of all nodes. However, our experiments show that the target condition is typically valid when operating at the target-rate.

**Adjusting the Multicast Rate:** The SLA requirement (R2) and target condition (1) give us a clear criteria for changing the rate. The FB scheme only gives us estimates of  $A_t$  and  $M_t$ , denoted by  $\hat{A}_t$  and  $\hat{M}_t$  respectively. For the  $K$ -Worst scheme, if  $K > A_{max} + \delta$  ( $\delta$  is a small constant), then  $\hat{A}_t$  and  $\hat{M}_t$  are sufficient to verify if (1) is satisfied because of the following property:

**Property 2:** If  $K \geq A_{max} + \epsilon$ , then,  $\hat{A}_t = \min(A_t, A_{max} + \epsilon)$  and  $\hat{A}_t + \hat{M}_t = \min(A_t + M_t, A_{max} + \epsilon)$ , where  $\hat{A}_t$  and  $\hat{M}_t$  are the known number of abnormal and mid-PDR known to the AP, and  $\epsilon$  is a small constant. In other words, given that  $K$  is large enough, the  $K$ -worst scheme provides accurate estimates of abnormal and mid-PDR nodes.

The proof of the property is in [29]. The objective is to choose minimum  $K$  (for minimum FB overhead) that is

sufficient to verify (1). In our experiments, we found that for  $A_{max} = 8$ ,  $K > 10$  works well (Section VI-A). We now derive the following *rate changing rules*:

**Rule I**  $\hat{A}_t > A_{max}$ : The system violates the SLA requirement (R2) and the rate is reduced.

**Rule II**  $\hat{A}_t + \hat{M}_t \geq A_{max} - \epsilon$ : The system satisfies the target condition.

**Rule III**  $\hat{A}_t + \hat{M}_t < A_{max} - \epsilon$ : The target condition does not hold and the rate can be increased, under the stability constraints provided in Section IV-C.

In our experiments we use  $\epsilon = 2$  to prevent rate oscillations.

The rate change actions in Procedure 1 are based on the these rules. The flags *canIncrease* and *canDecrease* indicate whether the multicast rate should be increased or decreased. Rate change operations are permitted only if the time elapsed since the last rate change is larger than the window size determined by the *Stability Preserving Method* (line 3). The for-loop checks whether the rate should be decreased according to Rule I (line 6) or increased according to Rule II (line 9) for the window duration. Finally, based on the value of the flags and the current rate, the algorithm determines the rate change operation and updates the parameters *rate* and *action*, accordingly (lines 10–15).

### C. The Stability Preserving Method

It is desirable to change the rate as soon as Rules I or III are satisfied to minimize QoE disruption (see (R5) in Section III). We observed that such a strategy can cause severe rate and throughput fluctuations. These result from two main reasons: (i) the reporting mechanism not stabilizing after the last rate change, and (ii) interference causing numerous low PDR reports.

To address this, we introduce in Procedure 2 a *window based RA technique* which considers the two situations and balances fast convergence with stability. In Procedure 1, rate is changed only if the rate change conditions are satisfied over a given *time window*, after the last rate change operation (lines 5-9). To prevent oscillations due to short-term wireless channel degradation, when the rate is reduced, the window is doubled in Procedure 2 (line 3). The window size is decreased by 1 when a duration *thresholdTime* elapses from the last rate or window size change (line 8). This allows recalibrating the window after an atypical long interference episode. The window duration varies between  $W_{min}$  and  $W_{max}$  FB reporting periods. In the experiments,  $W_{min} = 8$  and  $W_{max} = 32$ . We defer the discussion of tuning  $W_{min}$  and  $W_{max}$  to [29].

### D. Handling Losses

*MuDRA* can handle mild losses (below 15%) by adding application level FEC [27] to the multicast streams. The PDR-Threshold in our experiments ( $L = 85\%$ ) was selected to allow nodes to handle losses in the event of short simultaneous transmission of another node. In such a situation, the collision probability is below  $2/CW_{min}$ , where  $CW_{min}$  is the minimal 802.11 contention window. For 802.11a/g/n  $CW_{min} = 16$ , which implies collision probability is below 12.5%. Therefore,

TABLE II  
THE PERCENTAGE OF  $\Delta PDR(T)$

T (ms)	100	200	300	400	500	700	1000
$\Delta PDR\%$	4.69	1.56	0.94	0.67	0.52	0.36	0.25

nodes with high PDR (near 100%) should be able to compensate for the lost packets. If there is strong interference, other means should be used. For instance, the multicast content can be divided into high and low priority flows, augmenting the high priority flow with stronger FEC during the interference period, while postponing low priority flows.

## V. REPORTING INTERVAL DURATION

*MuDRA* relies on status reports from the FB nodes. For immediate response to changes in service quality, the status reports should be sent as frequently as possible, (i.e., minimal *reporting interval*). However, this significantly impairs the system performance as described below.

**Impact of Aggressive Reporting:** Figs. 6(a)-6(c) show the impact of different reporting intervals on *MuDRA*. In these experiments, the number of FB nodes ( $K$ ) is 50 and the total number of nodes is 158. To focus on RA aspects, we set both  $W_{min}$  and  $W_{max}$  to 5 reporting intervals. Fig. 6(a) shows that when the reporting interval is too short, *MuDRA* does not converge to the target rate of 24Mbps. Fig. 6(b) shows that in the case of reporting interval of 100ms, more than 50% of the packets are transmitted at the lowest rate of 6 Mbps. Fig. 6(c) shows that the control overhead is significantly larger for short reporting intervals (shorter than 200ms). The control overhead comprises of unicast FB data sent by nodes and multicast data sent by AP to manage  $K$  FB nodes.

These phenomena result from collisions between feedback reports and multicast messages. In the event of a collision, FB reports, which are unicast messages, are retransmitted, while multicast messages are lost. Frequent reporting increases the collision probability, resulting in PDR reduction and causes the classification of many nodes as mid-PDR nodes, i.e.,  $PDR < H_{high} = 97\%$ . Thus, due to Rule II from Section IV-B, the rate is kept close to the minimal rate.

**Appropriate Reporting Interval Duration:** Assume a greedy AP which continuously transmits multicast messages. We now estimate the PDR reduction, denoted as  $\Delta PDR$ , for a given reporting interval  $T$  and upper bound  $K$  on the number of FB nodes (both normal and abnormal), when the system operates at the low rate of 6 Mbps. In this rate, the transmission duration of multicast and FB messages are  $D \approx 3.0\text{ms}$  and  $d \approx 1\text{ms}$ . With proper calculation we get,

$$\Delta PDR = \left[ \frac{2}{CW_{min}} \right]^2 \cdot \frac{K \cdot D}{T - d \cdot K} \quad (2)$$

Equation (2) confirms that  $\Delta PDR$  is reduced by increasing the reporting interval or by using a higher rate, which reduces  $D$ . Table II provides the  $\Delta PDR$  values for  $K = 50$  when  $T$  varies between 0.1 to 1s. We wanted  $\Delta PDR \leq 0.5\%$ , which implies using a reporting interval  $T \geq 500\text{ms}$ . The detailed derivations appear in [29].

## VI. EXPERIMENTAL EVALUATION

For evaluating the performance of *MuDRA* on the ORBIT testbed, we use the parameter values listed in Table I. The performance metrics are described below:

- (i) *Multicast rate and throughput*: The time instants when the target condition is satisfied are marked separately.
- (ii) *PDR at nodes*: Measured at each node.
- (iii) *Number of abnormal and mid-PDR nodes*: We monitored all the abnormal and mid-PDR nodes (not just the FB nodes).
- (iv) *Control traffic*: The feedback overhead (this overhead is very low and is measured in Kbps).

We compared *MuDRA* to the following schemes:

- (i) *Fixed rate scheme*: Transmit at a fixed rate of 36Mbps, since it is expected to be the target rate.
- (ii) *Pseudo-multicast*: Unicast transmissions to the node with the lowest SNR/RSS. The unicast RA is the driver specific RA algorithm *Minstrel* [28]. The remaining nodes are configured in promiscuous mode.
- (iii) *Simple Rate Adaptation (SRA) algorithm* [7]: This scheme also relies on measuring the number of abnormal nodes for making RA decisions. Yet, it is not designed to achieve the target rate, maintain stability, or respond to interference.

### A. Performance Comparison

We evaluated the performance of *MuDRA* in several experiments on different day with 160 – 170 nodes. Fig. 7 shows one instance of such an experiment over 300s with 162 nodes. Fig. 7(a) shows the mid-PDR and abnormal nodes for the duration of one experiment run. Fig. 7(b) shows the rate determined by *MuDRA*. The AP converges to the target rate after the initial interference spike in abnormal nodes at 15s. The AP successfully ignored the interference spikes at time instants of 210, 240, and 280s to maintain a stable rate. The target-condition is satisfied except during the spikes. The overall control overhead as seen in Fig. 7(c) is approximately 40Kbps. The population of abnormal nodes stays around 2 – 3 for most of the time which implies that more than 160 nodes (> 98%) have a PDR > 85%. The actual throughput is stable at around 20Mbps which after accounting for 15% FEC correction implies a goodput of 17Mbps.

Fig. 8(a) shows a sample of the throughput and rate performance of the pseudo-multicast scheme. The throughput achieved is close to 9Mbps. We observe that pseudo-multicast frequently samples higher rates (up to 54Mbps) leading to packet losses. The average throughput for different schemes over 3 experiments of 300s each (conducted on different days) with 162 nodes is shown in Table III. *MuDRA* achieves 2x throughput than pseudo-multicast scheme. The fixed rate scheme yields approximately 10% higher throughput than *MuDRA*. SRA has similar throughput as *MuDRA*.

Fig. 8(b) shows the distribution of average PDR of 162 nodes for the same 3 experiments. In the pseudo-multicast scheme, more than 95% of nodes obtain a PDR close to 100% (we did not consider any retransmissions to nodes listening in promiscuous mode). *MuDRA* meets the QoS requirements of 95% nodes with at least 85% PDR. On the other hand, in SRA

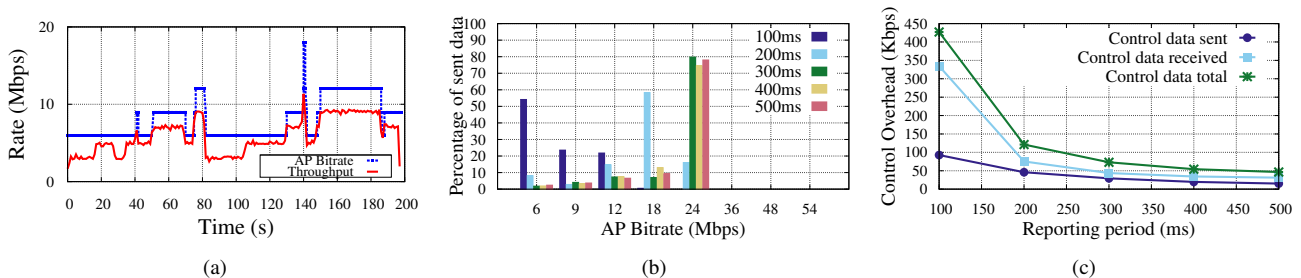


Fig. 6. (a) Rate adaptation performance for reporting intervals of 100ms, (b) Fraction of data sent at various rates with *MuDRA* for different reporting intervals, and (c) Control overhead for various reporting intervals.

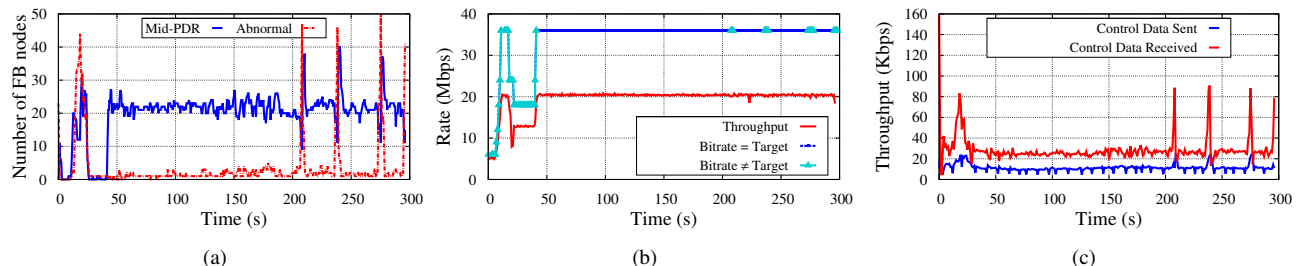


Fig. 7. A typical sample of *MuDRA*'s operation over 300s with 162 nodes: (a) Mid-PDR and abnormal nodes, (b) Multicast rate and throughput measured at the AP, and (c) Control data sent and received.

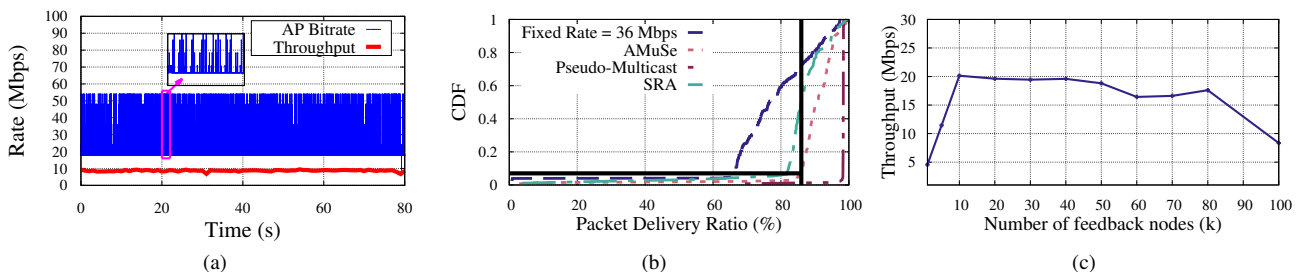


Fig. 8. (a) Rate and throughput for the pseudo-multicast scheme, (b) CDF of PDR distributions of 162 nodes for fixed rate, *MuDRA*, Pseudo-Multicast, and SRA schemes, and (c) Multicast throughput vs. the number of feedback nodes ( $K$ ).

TABLE III  
AVERAGE THROUGHPUT (MBPS) OF PSEUDO-MULTICAST, *MuDRA*, AND SRA SCHEMES WITH AND WITHOUT BACKGROUND TRAFFIC.

	No Background traffic	Background traffic
Fixed rate = 36Mbps	20.42	13.38
Pseudo-Multicast	9.13	5.36
<i>MuDRA</i>	18.75	11.67
SRA	19.30	4.55

and the fixed rate schemes 45% and 70% of the nodes have PDR less than 85%, respectively.

In pseudo-multicast, more reliable transmissions take place at the cost of reduced throughput, since the AP communicates with the node with the poorest channel quality in unicast. The significant difference in QoS performance of the fixed rate and SRA schemes is because the target rate can change due to interference, etc. In such a situation, *MuDRA* can achieve the new target rate while the fixed rate and SRA schemes lead to significant losses (we observed that exceeding the target rate even 10% of time may cause up to 20% losses and less than 5% throughput gain).

**Changing number of FB nodes:** We varied the number of FB nodes ( $K$ ) between 1 – 100 for *MuDRA*. Fig. 8(c) shows the throughput as  $K$  changes. For  $K = 1$ , *MuDRA* tunes to

the node with the worst channel quality, and consequently, the throughput is very low. On the other hand, increasing  $K$  from 30 to 90 adds similar amount of FB overhead as decreasing the report interval from 500ms to 200ms in Section V. Thus, the throughput decreases for a large number of FB nodes. The throughput for  $K$  between 10 – 50 does not vary significantly which is aligned with our discussion in Section IV that *MuDRA* needs only  $K > A_{max} + \delta$  for small  $\delta$  to evaluate the target rate conditions.

### B. Impact of Mobility

We evaluate *MuDRA* performance when emulating severe mobility conditions. In the experiments, each node leaves or joins the network with probability  $p$  after every 6s. Thus,  $p = 0.1$  implies that a node changes its state with probability of approximately 50% at least once in a minute. Initially, 50% of the nodes are randomly selected to be in the network.

We conducted 3 experiments consisting of 155 nodes (initially, 77 nodes in on state). Fig. 9(a) shows the impact of  $p$  on the distribution of time duration that the nodes remain as FB nodes. Higher values of  $p$  imply higher mobility and lead to shorter periods for which nodes serve as FB nodes.

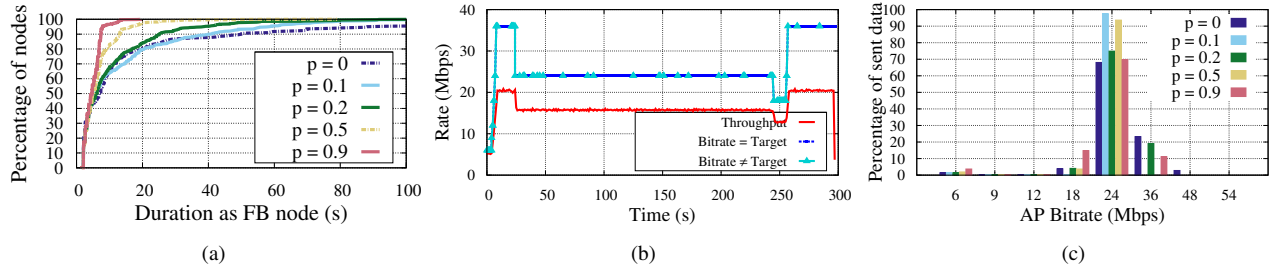


Fig. 9. Performance of *MuDRA* with emulated node mobility: (a) Distribution of time durations for which a node is a feedback node for different values of probability  $p$  of node switching its state on/off every 6s, (b) RA with  $p = 0.2$ , (c) Percentage of data sent at various rates for different values of  $p$ .

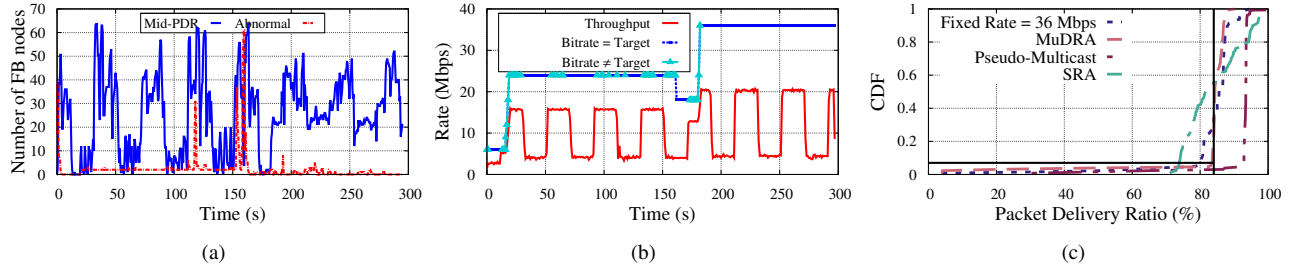


Fig. 10. Performance of *MuDRA* with 155 nodes where an interfering AP transmits on/off traffic: (a) Mid-PDR and abnormal feedback nodes, (b) Multicast rate and throughput, (c) CDF for PDR distribution with interference for fixed rate, *MuDRA*, pseudo-multicast, SRA.

The average number of changes in FB nodes per second is 2, 5, and 10 for  $p$  equal to 0, 0.2, and 0.9, respectively. Even with these changes, the average control overhead is very low (35Kbps) and is not affected by the degree of mobility. Fig. 9(b) shows one instance of the RA process with  $p = 0.2$ . We see that *MuDRA* can adapt to the changing target rate at times 10, 30, and 255s. Fig. 9(c) shows the percentage of data sent at different rates for several values of  $p$  averaged over 3 different experiment runs. *MuDRA* achieves a similar rate distribution for all values of  $p$ . Our experiments show that *MuDRA* can achieve the target rate, maintain stability, and adds low overhead, even under severe mobility.

### C. Impact of External Interference

We envision that *MuDRA* will be deployed in environments where the wireless infrastructure is centrally controlled. However, in-channel interference can arise from mobile nodes and other wireless transmissions. In addition to the uncontrolled interference spikes on ORBIT, we evaluate the impact of interference from a nearby node which transmits at the same channel as the multicast AP. We consider a scenario with two nodes near the center of the grid that exchange unicast traffic at a fixed rate of 6Mbps in a periodic on/off pattern with on and off periods 20s each. The transmission power of the interfering nodes is also 0dBm. This helps us evaluate the performance in the worst case scenario of continuous interference and study the dynamics of changing interference.

Fig. 10(a) shows the mid-PDR and abnormal nodes and Fig. 10(b) shows the rate and throughput for one experiment with 155 nodes. The number of mid-PDR nodes increases during the interference periods, due to losses from collisions. *MuDRA* converges to the target rate of 24Mbps. Notice during interference periods, *MuDRA* satisfied the target-condition and that using the stability preserving method, *MuDRA* manages

to preserve a stable rate. The average throughput of different schemes with on/off background traffic for 3 experiments of 300s each is in Table III. Pseudo-multicast achieves half while SRA has a third of the throughput of *MuDRA*. The fixed rate scheme achieves similar throughput as *MuDRA*.

The PDR distribution of nodes is in Fig. 10(c). *MuDRA* satisfies QoS requirements while maintaining high throughput. Pseudo-multicast scheme has 90% nodes with PDR more than 90% since it makes backoff decisions from unicast ACKs. SRA yields 55% nodes with PDR less than 85% as it transmits at low rates. The fixed rate scheme yields 30% nodes with PDR less than 85%. The fixed rate scheme performs better than SRA since it maintains a higher rate.

### D. Video multicast

We demonstrate the feasibility of using *MuDRA* for streaming video. The video is segmented with segment durations equal to the period of rate changes (1s) and each segment is encoded at several rates in H.264 format. For each time period, the key (I) frames are transmitted reliably at the lowest rate 6Mbps (note that transmitting the key frames can be achieved with 100% reliability even at 12Mbps on the testbed). The non-key (B and P) frames are transmitted at the rate set by *MuDRA*. At each instant, we know the expected throughput  $\hat{D}_R$  for every rate  $R$ , the fraction of key frame data  $f_k$ , and the fraction of non-key frame data  $f_{nk}$ . Denote the expected throughput at 6Mbps by  $\hat{D}_{min}$ . The video rate can be calculated by solving linear equations  $\hat{V}_R = \frac{\hat{D}_{min} \cdot \hat{D}_R}{\hat{D}_{min} \cdot f_{nk} + \hat{D}_R \cdot f_k}$ . **Experimental Results:** We use raw videos from an online dataset [34] and encode the videos with H.264 standard. In our data sets,  $f_k$  is 15 – 20%. For *MuDRA* with throughput 19Mbps and FEC correction of 15%, we can support a video rate of 13 – 15 Mbps, which is sufficient for 3 or 4 HD streams



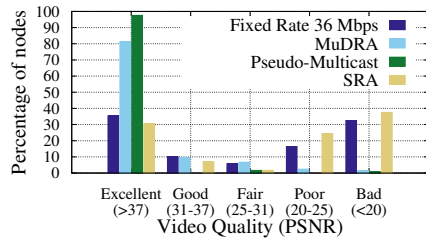


Fig. 11. Distribution of video quality and PSNR (in brackets) measured at 160 nodes for different multicast schemes.

(each 4Mbps) on mobile devices. For each node, we generated the video streams offline by mapping the video frames to the detailed packet traces collected on ORBIT from an RA experiment. In our experiments, we only considered a single video stream of rate  $V_R$ . For a fair comparison, the I frames were transmitted at 6Mbps for all schemes. We measured the PSNR of the video at each node and classified the PSNR in 5 categories based on visual perception.

Fig. 11 shows the video quality and PSNR ranges at the nodes for 3 experiments each of 300s and with 150 – 160 nodes. With *MuDRA*, more than 90% of the nodes achieve excellent or good quality, 5% achieve fair quality, and less than 5% get poor or bad quality. While the pseudo-multicast scheme results in almost all nodes obtaining excellent quality, the video throughput for this scheme is significantly lower (8Mbps). SRA and the fixed rate schemes have more than 50% nodes with poor or bad video quality.

## VII. CONCLUSION AND FUTURE WORK

We designed a novel multicast rate adaptation algorithm (*MuDRA*) that provides high throughput while satisfying SLA requirements. *MuDRA*'s performance on the ORBIT testbed with hundreds of nodes shows that it can reliably support applications such as large scale multimedia content delivery. In future work, we will refine *MuDRA* by distinguishing between losses due to channel conditions and collisions.

## VIII. ACKNOWLEDGEMENTS

The authors express their gratitude to Ivan Seskar from WINLAB (Rutgers University) for his support in conducting experiments on the ORBIT testbed and several useful technical discussions. This work was supported in part by NSF grant CNS-10-54856, CIAN NSF ERC under grant EEC-0812072, and the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/20072013) under REA grant agreement no. [PIIF-GA-2013-629740].11.

## REFERENCES

- [1] Y. Tanigawa, K. Yasukawa, and K. Yamaoka, "Transparent unicast translation to improve quality of multicast over wireless LAN," in *IEEE CCNC'10*, 2010.
- [2] "Cisco, white-paper, Cisco connected stadium Wi-Fi solution," 2011. [Online]. Available: [http://www.cisco.com/web/strategy/docs/sports/c78-675063\\_dSheet.pdf](http://www.cisco.com/web/strategy/docs/sports/c78-675063_dSheet.pdf)
- [3] K. Pelechrinis, T. Salonidis, H. Lundgren, and N. Vaidya, "Experimental characterization of 802.11n link quality at high rates," in *ACM WiN-TECH'10*, 2010.
- [4] C. Perkins and M. McBride, "Multicast WiFi problem statement," Working Draft, IETF Internet-Draft, 2015, <http://www.ietf.org/internet-drafts/draft-mcbride-mboned-wifi-mcast-problem-statement-00.txt>.

- [5] R. Chandra, S. Karanth, T. Moscibroda, V. Navda, J. Padhye, R. Ramjee, and L. Ravindranath, "DirCast: a practical and efficient Wi-Fi multicast system," in *IEEE ICNP'09*, 2009.
- [6] S. Sen, N. K. Madabhushi, and S. Banerjee, "Scalable WiFi media delivery through adaptive broadcasts," in *USENIX NSDI'10*, 2010.
- [7] Y. Bejerano, J. Ferragut, K. Guo, V. Gupta, C. Gutterman, T. Nandagopal, and G. Zussman, "Scalable WiFi multicast services for very large groups," in *IEEE ICNP'13*, 2013.
- [8] K. Papagiannaki, M. Yarvis, and W. S. Conner, "Experimental characterization of home wireless networks and design implications," in *IEEE INFOCOM'06*, 2006.
- [9] J. Vella and S. Zammitt, "A survey of multicasting over wireless access networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 718–753, 2013.
- [10] A. Kameron and L. Montebani, "WaveLAN-ii: a high-performance wireless LAN for the unlicensed band," *Bell Labs technical journal*, vol. 2, no. 3, p. 118133, 1997.
- [11] M. Lacage, M. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *ACM MSWiM'04*, 2004.
- [12] J. Bicket, "Bit-rate selection in wireless networks," *PhD thesis, MIT*, 2005.
- [13] S. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *ACM MOBICOM'06*, 2006.
- [14] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: collision-aware rate adaptation for IEEE 802.11 WLANs," in *IEEE INFOCOM'06*, 2006.
- [15] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," in *ACM MOBICOM'01*, 2001.
- [16] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee, "Diagnosing wireless packet losses in 802.11: Separating collision from weak signal," in *IEEE INFOCOM'08*, 2008.
- [17] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *ACM MobiSys'08*, 2008.
- [18] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *ACM SIGCOMM'09*, 2009.
- [19] H. Rahul, F. Edalat, D. Katabi, and C. Sodinii, "Frequency-aware rate adaptation and MAC protocols," in *ACM MOBICOM'09*, 2009.
- [20] X. Wang, L. Wang, and D. Wang, Y. and Gu, "Reliable multicast mechanism in WLAN with extended implicit MAC acknowledgment," in *IEEE VTC'08*, 2008.
- [21] Z. Feng, G. Wen, C. Yin, and H. Liu, "Video stream groupcast optimization in WLAN," in *IEEE ITA'10*, 2010.
- [22] J. Villalon, P. Cuenca, L. Orozco-Barbosa, Y. Seok, and T. Turletti, "Cross-layer architecture for adaptive video multicast streaming over multirate wireless LANs," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 699–711, 2007.
- [23] A. Basalamah, H. Sugimoto, and T. Sato, "Rate adaptive reliable multicast MAC protocol for WLANs," in *IEEE VTC'06*, 2006.
- [24] W.-S. Lim, D.-W. Kim, and Y.-J. Suh, "Design of efficient multicast protocol for IEEE 802.11n WLANs and cross-layer optimization for scalable video streaming," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 780–792, 2012.
- [25] "ORBIT testbed," <http://orbit-lab.org/>.
- [26] H.-T. Chiao, S.-Y. Chang, K.-M. Li, Y.-T. Kuo, and M.-C. Tseng, "WiFi multicast streaming using AL-FEC inside the trains of high-speed rails," in *IEEE BMSB'12*, 2012.
- [27] K. N. D. Vukobratovic, "A survey on application layer forward error correction codes for IP datacasting in DVB-H," in *3rd COST 2100 MCM*, 2007.
- [28] "Minstrel," <https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel>.
- [29] V. Gupta, C. Gutterman, Y. Bejerano, and G. Zussman, "Experimental evaluation of large scale WiFi multicast rate control," in *arXiv:1601.06425 [cs.NI]*, 2016.
- [30] N. Cranley, P. Perry, and L. Murphy, "User perception of adapting video quality," *Int. J. Hum. Comput. St.*, vol. 64, no. 8, pp. 637–647, 2006.
- [31] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," in *ACM SIGCOMM'06*, 2006.
- [32] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *ACM SIGCOMM'10*, 2010.
- [33] M. R. Souryal, L. Klein-Berndt, L. E. Miller, and N. Moayeri, "Link assessment in an indoor 802.11 network," in *IEEE WCNC'06*, 2006.
- [34] "Video dataset," <https://media.xiph.org/video/derf/>.