Experimental Evaluation of Large Scale WiFi Multicast Rate Control

Varun Gupta, Craig Gutterman, Yigal Bejerano, and Gil Zussman

***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.**

***Index Terms*—WiFi Multicast, Rate Adaptation, Large-Scale Evaluation**

I. Introduction

Multimedia (e.g., video) delivery is an essential service for wireless networks and several solutions were proposed for crowded venues [2]–[4]. 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 [5], [6]. *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* [7]. 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 [8]–[10] 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 [11], [12] 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. This results from effects such as multipath and fast fading [13]. 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.

V. Gupta, C. Gutterman, Y. Bejerano, and G. Zussman are with the Department of Electrical Engineering, Columbia University, NY, USA (email: varun@ee.columbia.edu, clg2168@columbia.edu, yb2406@columbia.edu, gil@ee.columbia.edu).

A preliminary version of this paper appeared in IEEE INFOCOM 2016 [1].

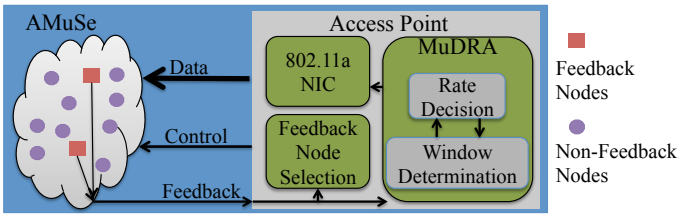


Fig. 1. The Adaptive Multicast Services (*AMuSe*) system consisting of the Multicast Dynamic Rate Adaptation (MuDRA) algorithm and a multicast feedback mechanism.

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 [11], [14], 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 surveys in [15], [16]).

Unicast RA: We discuss unicast RA schemes, since they can provide insight into the design of multicast RA. In *Sampling-based algorithms*, ACKs after successful transmissions as well as the relation between the rate and the success probability

are used for RA after several consecutive successful or failed transmissions [17]–[19]. The schemes in [20]–[22] distinguish between losses due to poor channel conditions and collisions, and update the rate based on former. Recently, [23], [24] proposed multi-arm bandit based RA schemes with a statistical analysis of performance. 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 [25]–[30]. Most measurement-based schemes modify the wireless driver on the receiver end and some require changes to the standard, which we avoid.

Multicast Feedback Mechanisms: Solutions for improving multicast service quality are based on collecting feedback from the receivers and adapting the sender rate. They integrate Automatic Repeat Request (ARQ) mechanisms into the protocol [9], [31]–[37], add Forward Error Correction (FEC) packets [8], [38]–[40], and utilize RA methods [33], [41]–[43]. The feedback mechanisms can be classified into five categories:

(i) Collecting *Individual Feedback* from all users for each received packet [8], [32], [37], [44]–[47]. Although this offers reliability, it does not scale for large groups. The other approaches provide scalability by compromising on the feedback accuracy.

(ii) The *Leader-Based Protocol with acknowledgements (LBP-ACK)* method [10], [33], [34], [39], [46] selects a few receivers to provide feedback, typically the receivers with the lowest channel quality.

(iii) *Pseudo-Broadcast* [9], [10], [48] converts the multicast feed to a unicast flow and sends it to one leader. The leader acknowledges the reception of the unicast flow while the other receivers receive packets by listening to the channel in promiscuous mode.

(iv) The *Leader-Based Protocol with negative acknowledgements* (LBP-NACK) [31], [43], [49] method improves Pseudo-Broadcast by allowing the other receivers to send NACKs for lost packets.

The leader based approaches (ii)-(iv) cannot provide guarantees on the feedback accuracy [11], [37]. Moreover, most LBP-ACK and LBP-NACK methods require changes to the standard.

(v) *Cluster-Based Feedback Mechanisms* [11], [32], [37], [50] handle the scalability issue by using the fact that adjacent receivers experience similar service quality. They partition the receivers into clusters and select the receiver with the weakest channel condition at each cluster as a *feedback (FB) node* that sends status reports to the sender. These methods, however, do not guarantee reliable delivery to all receivers.

Additionally, [38], [40], [51] propose to use strong FEC for overcoming losses without specifying any feedback mechanism. Others [9], [10], [37], [43] balance between the accuracy requirements and low overhead by using a combination of methods (e.g., Pseudo-Broadcast with infrequent reports from the other receivers). The recently proposed IEEE 802.11aa [52] amendment specifies protocols for multicast feedback. However, it is not designed to scale to hundreds of receivers and it does not specify a rate adaptation mechanism. We believe that the Block Acknowledgement based feedback mechanism proposed in the 802.11aa standard could be especially attractive and compatible for rate adaptation approach in *MuDRA* .

Multicast RA and Coding: In [9], [33], [41], [42], [53] the sender uses feedback from leaders (nodes with worst channel conditions) for RA. In [43] RA is conducted based on reports of a single leader or all nodes depending on channel conditions. Network coding schemes [54], [55] have been extensively used for ensuring reliable multicast to large number of receivers. Further, rateless coding schemes [56], [57] have been extensively studied for wireless multicast.

Medusa [10] 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. An anonymous query scheme to maximize multicast throughput by inferring the maximum transmission rate that satisfies all receivers was proposed in [58]. Recently, in [11] 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. *MuDRA* requires application layer measurements from mobile devices for multicast rate adaptation decisions. The multicast rate changes can be supported by most APs through changes in the driver-level code or through API calls (e.g., Asus APs provide simple API calls [59]). During our experiments, we utilized API calls for the ath5k driver on the ORBIT nodes.[1]

We implemented *MuDRA* with the *AMuSe* system on the ORBIT testbed [60], evaluated its performance with all the operational IEEE 802.11 nodes (between 150 and 200), and compared it to other multicast schemes. We use 802.11a to maximize the number of WiFi devices available[2]. To the best of our knowledge, this is the largest set of wireless devices available to the research community. 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*,

[1]The API calls provide a flexible way to implement the rate adaptation algorithms on the application layer. However, the rate change operations may suffer from small latency (approximately 20-50ms in our measurements which is much faster than the scale of rate changes). Another alternative is implementing the rate change algorithm in the driver, which may require AP specific changes but result in faster rate changes.

[2]The ORBIT testbed supports only about 30 802.11n enabled devices

numerous receivers suffer from low PDR and their losses cannot be easily recovered by retransmissions due to large number of retransmisisons and associated control data required. Error corrrection schemes such as rateless codes [61], [62] or network coding schemes [54], [55] may recover intended information in the presence of a large number of random packet erasures. Nevertheless, they assume that the underlying rate in the physical layer has been given. Therefore, the actual throughput or delay performance of rateless or network codes depends critically on the rate adaptation in the physical layer. 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 [9] 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 [38], [40], [51], [63], [64].

MuDRA is experimentally compared to (i) pseudo-multicast with a unicast RA [65], (ii) fixed rate, and (iii) a rate adaptation mechanism proposed in [11] which we refer to as the 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 churn. Finally, we devise a live multicast video delivery approach for *MuDRA*. We show that in our experimental settings with target rate of $24-36$Mbps, *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*. The rest of the paper is organized as follows. Section II describes the ORBIT testbed and important observations. Section III presents the model and objectives. *MuDRA*'s design is described in Sections IV and V. The experimental evaluation is presented in Section VI before concluding in Section VII.

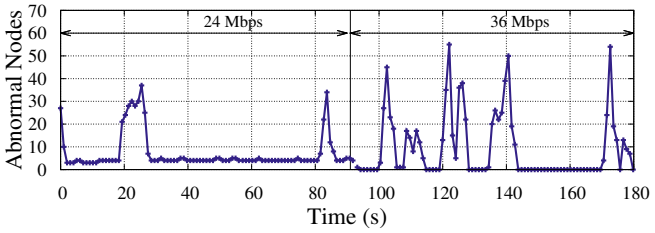


Fig. 2. Experimental measurement of the number of abnormal nodes in time, for fixed rates of 24 and 36Mbps.

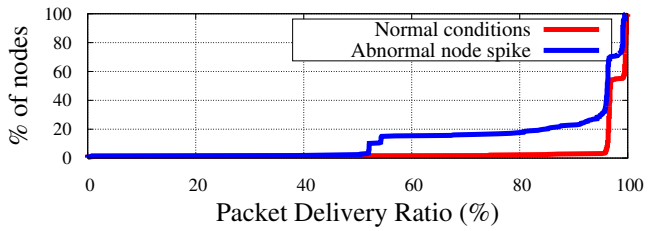


Fig. 3. The CDF of the PDR values of 170 nodes during normal operation and during a spike at rate of 36Mbps.

II. Testbed and Key Observations

We evaluate *MuDRA* on the ORBIT testbed [60], which is a dynamically configurable grid of 20×20 (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*[3] 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 each experiment, the number of active nodes varied slightly due to testbed hardware issues and is known at the beginning of each experiment. We account for these variations in the number of nodes and normalize our experimental results wherever necessary. In all the experiments, one corner node served as a single 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. Most practical applications such as video streaming include a sequence number to keep track of packet delivery at the clients. We embed an artificial sequence number for each packet in the UDP payload for measurement purposes. *The AP used the lowest supported transmission power of* $1mW = 0dBm$ *to ensure that the channel conditions of some nodes are marginal.*

Technical challenges: While analyzing the performance, we noticed that clients disconnect from the AP at high bit-rates, thereby causing performance degradation. We noticed that in

[3]We do not expect variability in hardware or software to have a major impact on the performance, since *MuDRA* does not use transceiver specific metrics such as RSSI or SNR. Instead, *MuDRA* relies on measuring Packet Delivery Ratio (PDR) performance for making rate change decisions.

TABLE I
NOTATION AND PARAMETER VALUES USED IN EXPERIMENTS.

Symbol	Semantics	Exp. Val.
n	Number of nodes associated with the AP.	> 150
X	*Population threshold* - Minimal fraction of nodes that should experience high PDR.	95%
A_{max}	The maximal number of allowed abnormal nodes.	8
L	*PDR threshold* - Threshold between acceptable (normal) and low (abnormal) PDR.	85%
H	Threshold between high PDR and mid-PDR.	97%
K	Expected number of FB nodes, $K = \alpha \cdot A_{max}$.	30
R	Reporting PDR threshold.	
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

several WiFi driver implementations, the beacon rate is set as the multicast rate. Increasing the bit-rate also increases the WiFi beacon bit-rate which may not be decoded at some nodes. A sustained loss of beacons leads to node disconnection. To counter this, we modified the ath5k driver to send beacons at a fixed minimum bit-rate.

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 [66], [67], 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 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*. Typically, the value of L will not be too small or too large. The value of L is independent of the *MuDRA* algorithm and depends solely on the desired QoE requirements and the amount of error recovery provisioned. In our experiments, we use $L = 85\%$. Choosing a large L may lead to lower transmission rate.

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.[4]

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 GETWINSIZE procedure, which determines the window size of rate updates (to maintain stability). A summary of notation used to describe the algorithm is shown in Table II.

(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.

The core of the rate adaptation algorithm relies on predicting the number of nodes that will suffer from low PDR values before performing a rate increase operation. A rate increase is performed, only if such nodes are few in number. On the other hand, if a large number of nodes violate the SLA requirements, the rate is immediately decreased. Further, 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*).

[4]Unlike in unicast where each packet is acknowledged, *MuDRA*'s reporting intervals are long (in the experiments we consider 2 reports per second).

TABLE II
NOTATION USED IN *MuDRA* ALGORITHM.

Symbol	Semantics
$\hat{A}_t$	Estimated number of abnormal nodes.
$\hat{M}_t$	Estimated number of mid-PDR nodes.
$changeTime$	Time slot of last rate increase or decrease.
$rate_{min}$	Minimum multicast rate supported by hardware.
$rate_{max}$	Maximum multicast rate supported by hardware.
ϵ	A fixed small constant.

Algorithm 1 *MuDRA* Algorithm

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

(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* [11], 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. The process works as follows: At the beginning of each reporting interval the AP sends a message with a list of K or less FB nodes as well as a *reporting PDR threshold* R. R is used for adjusting the set of FB nodes to changes due to mobility or variation of the channel condition, i.e., interference[5]. Upon receiving this message, each FB node waits a short random time for avoiding collisions and then reports its measured PDR to the AP. Every other node checks if its PDR value is below R and in such situation it volunteers to serve as an FB node. To avoid a swarm of volunteering messages in the case of sporadic interference, a non FB node verifies that its PDR values are below R for three consecutive reporting intervals before volunteering. At the end of a reporting interval, the AP checks the PDR values of all the FB and volunteering nodes, it selects the K with lowest PDR values as FB nodes and updates R. If the number of selected FB nodes is K then for keeping the stability of the FB list, R is set slightly below the highest PDR value of the FB nodes (e.g., 1% point below). Otherwise, R is set slightly above the highest PDR value of the FB nodes (e.g., 0.5% point above). The AP sends a new message and the process repeats. We note that in a quasi static scenario, the values of R do not have a significant impact on the feedback or the overhead of feedback. Tuning R is a challenge only in the rare scenario when a large number of nodes with significantly different PDR values rapidly enter or leave the multicast system.

[5]when the system is activated the FB list is empty and $R = L$.

Procedure 1 Rate Decision

procedure GETRATE($rate, window, changeTime, t$)
  $action \leftarrow Hold$
  if $(t - changeTime) > window$ **then**
    $canDecrease \leftarrow true$, $canIncrease \leftarrow true$
    for $\tau \leftarrow 0$ **to** $window$ **do**
      if $\hat{A}_{t-\tau} < A_{max}$ **then**
        $canDecrease \leftarrow false$
      else if $\hat{A}_{t-\tau} + \hat{M}_{t-\tau} > A_{max} - \epsilon$ **then**
        $canIncrease \leftarrow false$
    if $canDecrease$ and $rate > rate_{min}$ **then**
      $rate \leftarrow NextLowerRate$
      $action \leftarrow Decrease$, $changeTime \leftarrow t$
    if $canIncrease$ and $rate < rate_{max}$ **then**
      $rate \leftarrow NextHigherRate$
      $action \leftarrow Increase$, $changeTime \leftarrow t$
return $rate$, $action$, $changeTime$

Procedure 2 Window Size Determination

procedure GETWINSIZE($Action, window, refTime, t$)
  if $Action = Decrease$ **then**
    $window \leftarrow \min(W_{max}, 2 \cdot window)$, $refTime \leftarrow t$
  else if $Action = Increase$ **then**
    $refTime \leftarrow t$
  else if $(t - refTime) > thresholdTime$ and $Action = Hold$ **then**
    $window \leftarrow \max(W_{min}, window - 1)$
    $refTime \leftarrow t$
return $window, refTime$

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.

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.

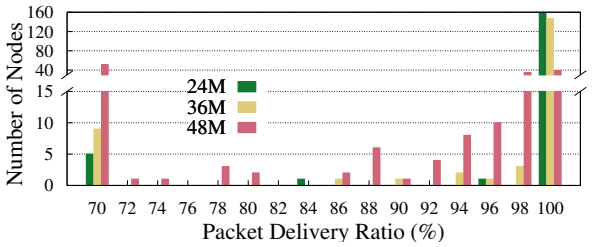


Fig. 4. The PDR distribution of one set of experiments with TX_{AP} rates of 24, 36, and 48Mbps.

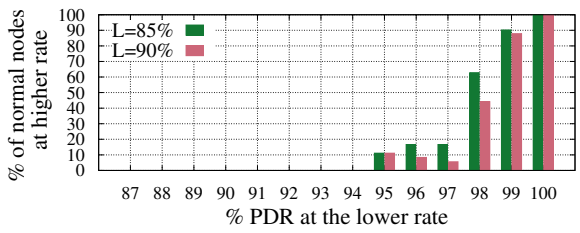


Fig. 5. The percentage of nodes that remain normal after increasing the TX_{AP} from 36Mbps to 48Mbps vs. their PDR values at the 36Mbps for different PDR-thresholds (L).

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 easily detected from coarse RF measurements, such as SNR/RSSI. As shown in [68], [69] differing receiver sensitivities as well as frequency selective fading may result in substantial PDR gaps between nodes with similar SNR/RSSI measurements. Receiver sensitivities differ between different 802.11 Network Interface Cards (NICs) and may vary based on the thermal noise, manufacturing process, etc. Typically the noise floor in off-the-shelf commodity WiFi hardware is not calibrated to account for these differing sensitivities [69]. On the other hand, predicting frequency selective fading is not possible without detailed subcarrier information which is not widely supported in current hardware and may lead to excessive feedback overhead [70]. However, large scale multicast environments enable us to efficiently predict the target condition as described next.

The above observations regarding the PDR and target rate from Fig. 4 can be summarized as the following important observation.

Observation I: 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 experiments. The total number of nodes in these experiments was 168. By calculating the percentage of abnormal nodes on the y-axis at 48Mbps vs. the PDR values on x-axis at 36Mbps in Fig. 5, we derive the following observation.

Observation II: There is a PDR threshold, H, such that every node with PDR between L and H becomes abnormal after the rate increase with very high probability. Typically, H can be a value slightly below 100%. In our experiments on ORBIT, we use $H = 97\%$ since 97% is the highest threshold for which this observation holds. With a very high or very low value of H, we observed that the algorithm may lead to more oscillations.[6] We refer to these nodes as *mid-PDR nodes*.

Observation II is not surprising. As reported in [68], [71], each receiver has an SNR band of $2-5$dB, 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 $2-3$dB 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,*

$$A_t \leq A_{max} \quad and \quad A_t + M_t > A_{max} \tag{1}$$

then with high likelihood, 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 the 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 $K > A_{max} + \epsilon$ (ϵ 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 ϵ 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.*

Proof. *First consider the claim $\hat{A}_t = \min(A_t, A_{max} + \epsilon)$. Consider the case $A_t \leq A_{max} + \epsilon$, we know that the number of estimated abnormal nodes $\hat{A}_t = A_t$ since $K \geq A_{max} + \epsilon$ and all abnormal nodes must belong in the K FB nodes set. Next, if $A_t > A_{max} + \epsilon$ then all the FB nodes chosen are abnormal by the definition of the K-worst feedback scheme which implies $\hat{A}_t = A_{max} + \epsilon$.*

A similar argument can be made for the claim $\hat{A}_t + \hat{M}_t = min(A_t + M_t, A_{max} + \epsilon)$. If $A_t + M_t \leq A_{max} + \epsilon$, then $\hat{A}_t + \hat{M}_t = A_t + M_t$ since the K feedback nodes will necessarily

[6]Further, evaluating the impact of H in different experimental conditions is an interesting future direction.

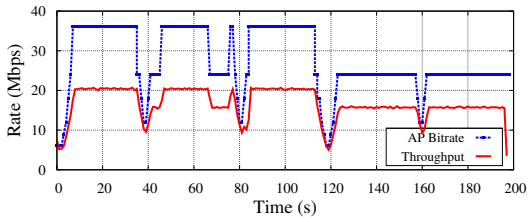


Fig. 6. Evolution of the multicast rate over time when the delay between rate changes = 1s (2 reporting intervals).

include the A_t abnormal and M_t mid-PDR nodes. If $A_t + M_t > A_{max} + \epsilon$, then $\hat{A}_t + \hat{M}_t$ which is upper bounded by $A_{max} + \epsilon$. □

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 III (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). However, as we show in Fig. 6 such a strategy may cause severe fluctuations of the transmission rate. 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, the 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 $W_{max} = 32$.

TABLE III
THE PERCENTAGE OF PDR LOSS AT NODES (ΔPDR) AS A FUNCTION OF THE REPORTING INTERVAL 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

D. Handling Losses

MuDRA can handle mild losses (below 15%) by adding application level FEC [63] 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, 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

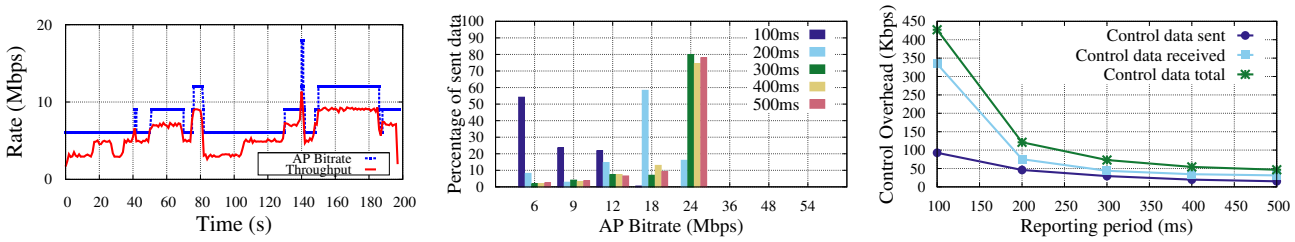
MuDRA relies on status reports from the FB nodes. For immediate response to changes in service quality, the status reports should be sent as frequently as possible, (i.e., minimal *reporting interval*). However, this significantly impairs the system performance as described below.

Impact of Aggressive Reporting: Figs. 7(a)-7(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. 7(a) shows that when the reporting interval is too short, *MuDRA* does not converge to the target rate of 24Mbps. Fig. 7(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. 7(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



(a) (b) (c)

Fig. 7. (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.

as ΔPDR, 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 6Mbps.

Packet Transmission Duration: We denote with D and d the transmission duration of multicast and feedback report message at the rate of 6Mbps, respectively. Since the length of each multicast packet is $12Kbits$, its transmission duration is $\frac{12\text{Kbits}}{6\text{Mbps}} = 2.0$ms. Given WiFi overhead of about 30%, we assume $D = 3$ms. The feedback messages are much shorter and we assume that their transmission duration is $d = 1$ms.

Number of feedback reports and multicast messages: Consider a time interval U, say a minute. The number of feedback reports, denoted as F, is

$$F = \frac{U}{T} \cdot K$$

The number of multicast message B is given by,

$$B = \frac{U - d \cdot F}{D} = \frac{U}{D} \cdot \left(1 - \frac{d \cdot K}{T}\right)$$

Collision probably of a multicast packet (ΔPDR): Let us first calculate the number of contention window slots, denoted by S, in which packet may be transmitted from the view point of the AP during the time interval U. Recall that between any two multicast transmissions, the AP waits an average of half of the contention window size $CW_{min}/2 = 8$. This leads to

$$S = \frac{CW_{min}}{2} \cdot B$$

ΔPDR is the fraction of contention window slots in which both the AP and a FB node send a message. To simplify our estimation, we ignore collisions and retransmission of FB messages, and assume that in any contention window slot only one FB node may transmit.[7] . Therefore,

$$\Delta PDR = \frac{F}{S} \cdot \frac{B}{S} = \left[\frac{2}{CW_{min}}\right]^2 \cdot \frac{F}{B}$$

With proper assignment 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 ΔPDR is reduced by increasing the reporting interval or by using a higher bit-rate, which reduces D. Table III provides the ΔPDR values for $K = 50$ when T varies between 0.1 to 1 second. In our experiments we wanted $\Delta PDR \leq 0.5\%$, which implies using reporting interval $T \geq 500$ms.

[7]These are second order effects of already low collision probabilities. For instance, the probability of a multicast message colliding with two feedback messages can be shown equal to $\left[\frac{2}{CW_{min}}\right]^3 \cdot \frac{K \cdot D}{T - d \cdot K}^2$ following an analysis similar to shown in this section. This translates to a collision probability of 2×10^{-4} for $K = 50$ and $T = 500ms$.

VI. Experimental Evaluation

For evaluating the performance of *MuDRA* on the ORBIT testbed, we use the parameter values listed in Table I. In all our evaluations, we consider backlogged multicast traffic. 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:

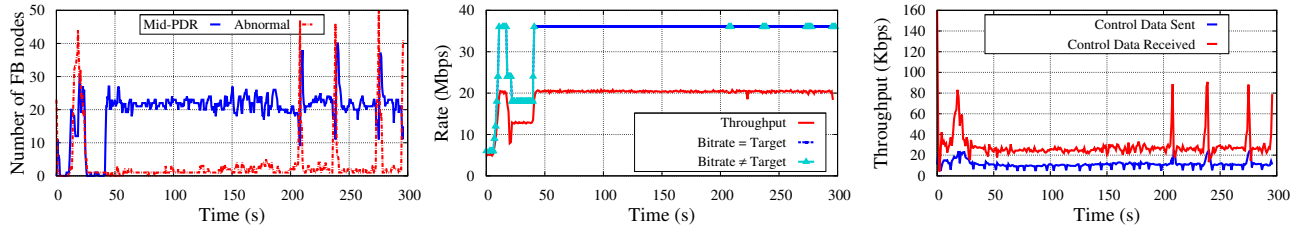
(i) *Fixed rate scheme*: Transmit at a fixed rate of 36Mbps, since it is expected to be the target rate.

(ii) *Pseudo-multicast*: Unicast transmissions to the node with the lowest SNR/RSS. The unicast RA is the driver specific RA algorithm *Minstrel* [65]. The remaining nodes are configured in promiscuous mode.

(iii) *Simple Rate Adaptation (SRA) algorithm [11]*: 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.

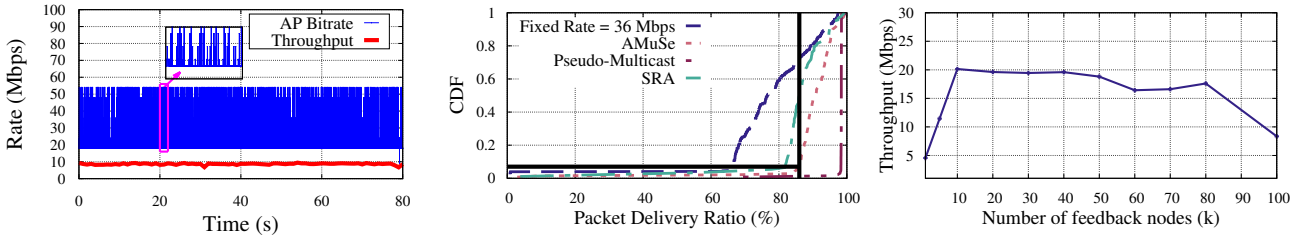
A. Performance Comparison

We evaluated the performance of *MuDRA* in several experiments on different days with $160 - 170$ nodes with the total number of feedback nodes, $K = 50$ and reporting interval duration of 500ms. To demonstrate the convergence and stability of *MuDRA* algorithm, one instance of such an experiment over 300s with 162 nodes is shown in Fig. 8. Fig. 8(a) shows the mid-PDR and abnormal nodes for the duration of one experiment run. Fig. 8(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. 8(c) is approximately 40Kbps. The population of abnormal nodes stays around $2-3$



(a) (b) (c)

Fig. 8. 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.



(a) (b) (c)

Fig. 9. (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 IV
AVERAGE THROUGHPUT (MBPS) OF PSEUDO-MULTICAST, *MuDRA*, AND SRA SCHEMES WITH AND WITHOUT ON-AND-OFF (ON AND OFF PERIODS 20S EACH) UNICAST BACKGROUND TRAFFIC BETWEEN TWO RANDOMLY CHOSEN NODES ON THE GRID.

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

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. 9(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 5 experiments of 300s each (conducted on different days) with 162 nodes is shown in Table IV. In this section, we only focus on the left-hand side of the able with no additional background traffic. *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. 9(b) shows the distribution of average PDR of 162 nodes for the same 5 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 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. 9(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 $K > A_{max}+\epsilon$ for small ϵ to evaluate the target rate conditions.

Impact of topology changes: To demonstrate that changes in the network may lead *MuDRA* to converge to a different rate, we devised a strategy to emulate network topology changes on the grid. During an experiment, a number of FB nodes are turned off at a given time. Since FB nodes have the lowest PDRs, it may lead to changes in the target rate as a large number of nodes with low PDR disappear from the network. Fig. 10 shows the scenario when 30 FB nodes are turned off after 150s during the experiment. The rate converges quickly

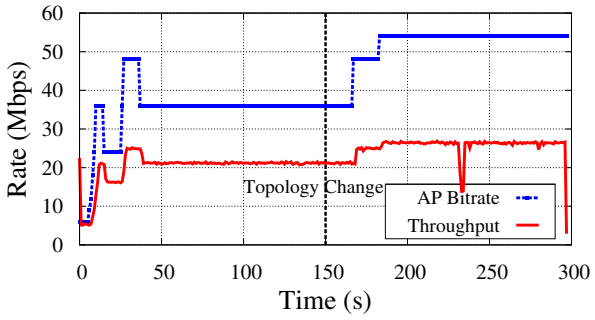


Fig. 10. Emulating topology change by turning off FB nodes after 150s results in changing optimal rate for *MuDRA*.

and without oscillations to a new target rate of 54Mbps.

B. Impact of High Node Churn

We evaluate the performance of *MuDRA* when emulating severe network changing conditions. In the experiments, each node leaves or joins the network with probability p 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 are in on state). Fig. 11(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 churn and lead to shorter periods for which nodes serve as FB nodes. 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.[8] Even with these changes, the average control overhead is very low (35Kbps) and is not affected by the degree of churn. Fig. 11(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. 11(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 network changing conditions.* It should be noted that the control overhead for the pseudo-multicast and fixed rate schemes is fixed and independent of the node churn.

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[9]. The transmission power of the interfering nodes is 0dBm which is equal to the transmission power of the multicast AP. The low transmission bitrate ensures maximum interference with multicast packets. Further, the on and off patterns help us evaluate the performance in the worst case scenario of continuous interference and study the dynamics of changing interference.

Fig. 12(a) shows the mid-PDR and abnormal nodes and Fig. 12(b) shows the rate and throughput for a single experiment with 155 nodes with background traffic. 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 unicast background traffic for 3 experiments of 300s each is in the right-most column of Table IV. Pseudo-multicast achieves half while SRA has a third of the throughput of *MuDRA* in presence of background traffic. The fixed rate scheme achieves similar throughput as *MuDRA*.

The PDR distribution of nodes is in Fig. 12(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. We also investigate the combined impact of *both interference and node churn*, where every 6s, the probability of a node switching on/off is $p = 0.2$. Fig. 13 shows the rate and throughput for this case. Similar to results in Section VI-B, *the performance of the system is not affected by node churn.*

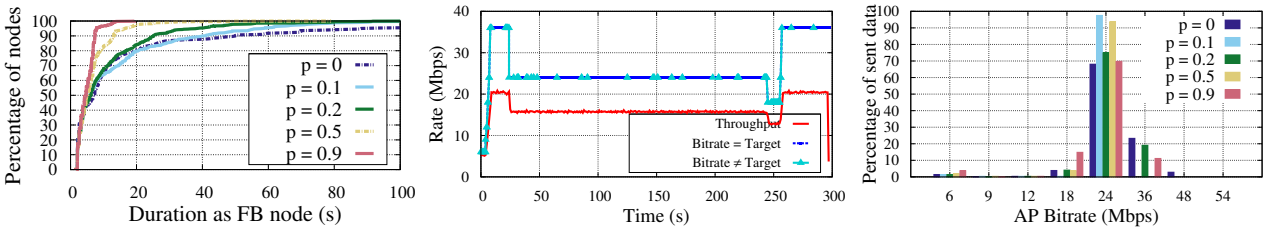
D. Video multicast

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*.

Let the multicast rate for current time period be R, the expected data throughput at this rate be $\hat{D}_R$, and the estimated throughput at the minimum rate be $\hat{D}_{min}$. Let f_k be the fraction of key frame data and f_{nk} be the fraction of non-key frame data. The video server has to determine the video rate V_R at each time t. Let the fraction of transmission time

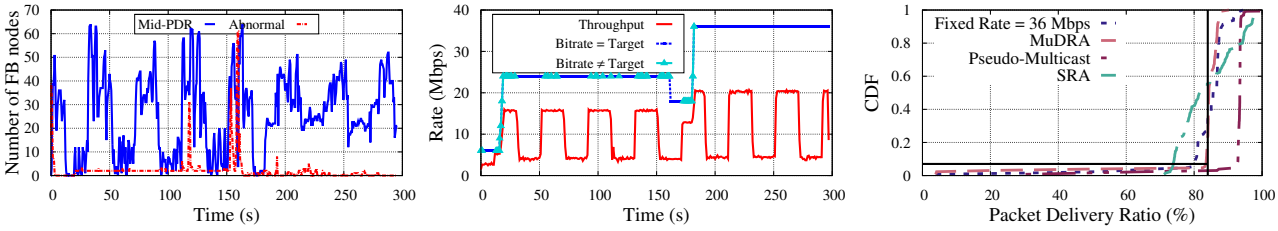
[8]The number of these changes were be calculated by comparing the identity of FB nodes in consecutive slots and is close to the expected values that can be derived analytically.

[9]The placement of the sender and receiver did not lead to any significant differences in performance of multiast traffic.



(a) (b) (c)

Fig. 11. Performance of *MuDRA* with high node churn: (a) Distribution of time durations for which a node is a FB node for different values of probability p of node switching its state on/off every 6s, (b) Multicast rate and throughput measured at the AP with $p = 0.2$, (c) Percentage of data sent at various rates for different values of p.



(a) (b) (c)

Fig. 12. Performance of *MuDRA* with 155 nodes where an interfering AP transmits on/off traffic: (a) Mid-PDR and abnormal FB nodes, (b) Multicast rate and throughput, (c) CDF for PDR distribution with interference for fixed rate, *MuDRA*, pseudo-multicast, SRA.

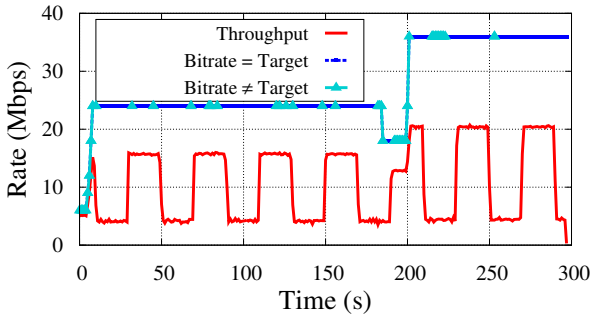


Fig. 13. Multicast throughput with node 1-8 transmitting interfering on/off packet stream with node churn.

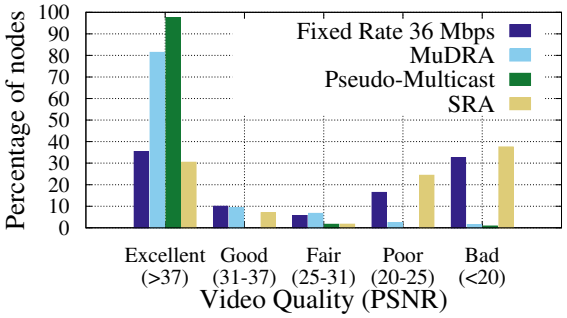


Fig. 14. Distribution of video quality and PSNR (in brackets) measured at 160 nodes for different multicast schemes.

for key frames $T_k = \frac{V_R \cdot f_k}{\hat{D}_{min}}$ and fraction of transmission time for non-key frames $T_{nk} = \frac{V_R \cdot f_{nk}}{\hat{D}_R}$. We know that

$$t_k + t_{nk} = 1$$

The video rate can be calculated by solving linear equations $V_R = \frac{\hat{D}_{min} \cdot \hat{D}_R}{\hat{D}_{min} \cdot f_{nk} + \hat{D}_R \cdot f_k}$. In environments where estimates of throughput are inaccurate due to interference, techniques such as in [72] can be utilized.

Experimental Results: We use raw videos from an online dataset [73] 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 (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. *For a fair comparison, the I frames were transmitted at 6Mbps for all schemes even though MuDRA can dynamically adjust the transmission rate to be much higher even for reliable transmissions.* In our experiments, we only considered a single video stream of rate V_R. We measured the PSNR of the video at each node and classified the PSNR in 5 categories based on visual perception[10].

Fig. 14 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. The higher thorughput from *MuDRA* can allow streaming of several concurrent video streams or streams encoded at higher rates while ensuring QoS requirements.

[10]PSNR quantifies the distortion of the received as compared to the original transmitted video.

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. Moreoever, we will also consider rate adaptation experiments in wild using a testbed of diverse set of Android mobile devices as well as evaluating performance with 802.11n/ac/aa standards. Moreover, we will develop methods that not only adapt the multicast rate but also the video coding rates based on the feedback from the nodes. We will evaluate the methods not only in the ORBIT testbed but also in a testbed of mobile devices where video streams will be sent to the nodes. Finally, we will consider the design of similar feedback and rate adaptation schemes that are tailored for the special characteristics of LTE eMBMS.

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 grants CNS-1423105 and CNS-1650685 and NSF GRFP DGE 16-44869.

References

[1] V. Gupta, C. Gutterman, Y. Bejerano, and G. Zussman, "Experimental evaluation of large scale WiFi multicast rate control," in *Proc. IEEE INFOCOM'16*, 2016.

[2] Y. Tanigawa, K. Yasukawa, and K. Yamaoka, "Transparent unicast translation to improve quality of multicast over wireless LAN," in *IEEE CCNC'10*, 2010.

[3] "Cisco, white-paper, Cisco connected stadium Wi-Fi solution," 2011. [Online]. Available: http://www.cisco.com/web/strategy/docs/sports/c78-675063_dSheet.pdf

[4] "Yinzcam," http://www.yinzcam.com/.

[5] K. Pelechrinis, T. Salonidis, H. Lundgren, and N. Vaidya, "Experimental characterization of 802.11n link quality at high rates," in *ACM WiNTECH'10*, 2010.

[6] N. Hajlaoui and I. Jabri, "On the performance of IEEE 802.11n protocol," in *ACM WiNTECH'12*, 2012.

[7] M. McBride and C. Perkins, "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.

[8] M. Wu, S. Makharia, H. Liu, D. Li, and S. Mathur, "IPTV multicast over wireless LAN using merged hybrid ARQ with staggered adaptive FEC," *IEEE Trans. Broadcast.*, vol. 55, no. 2, pp. 363 –374, 2009.

[9] 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.

[10] S. Sen, N. K. Madabhushi, and S. Banerjee, "Scalable WiFi media delivery through adaptive broadcasts," in *USENIX NSDI'10*, 2010.

[11] 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.

[12] K. Papagiannaki, M. Yarvis, and W. S. Conner, "Experimental characterization of home wireless networks and design implications," in *IEEE INFOCOM'06*, 2006.

[13] T. S. Rappaport, *Wireless Communication Principle and Practice, 2nd edition*. Prentice Hall, 2002.

[14] V. Gupta, Y. Bejerano, C. Gutterman, J. Ferragut, K. Guo, T. Nandagopal, and G. Zussman, "Light-weight feedback mechanism for WiFi multicast to very large groups–experimental evaluation," *IEEE Trans. Netw.*, vol. 24, no. 6, pp. 3826–3840, 2016.

[15] D. Nguyen and J. Garcia-Luna-Aceves, "A practical approach to rate adaptation for multi-antenna systems," in *IEEE ICNP'11*, 2011.

[16] J. Vella and S. Zammit, "A survey of multicasting over wireless access networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 718–753, 2013.

[17] A. Kamerman 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.

[18] M. Lacage, M. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *ACM MSWiM'04*, 2004.

[19] J. Bicket, "Bit-rate selection in wireless networks," *PhD thesis, MIT*, 2005.

[20] Q. Pang, V. Leung, and S. Liew, "A rate adaptation algorithm for IEEE 802.11 WLANs based on MAC-layer loss differentiation," in *IEEE BroadNets'06*, 2006.

[21] S. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *ACM MOBICOM'06*, 2006.

[22] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: collision-aware rate adaptation for IEEE 802.11 WLANs," in *IEEE INFOCOM'06*, 2006.

[23] B. Radunovic, A. Proutiere, D. Gunawardena, and P. Key, "Dynamic channel, rate selection and scheduling for white spaces," in *ACM CONEXT'11*, 2011.

[24] R. Combes, A. Proutiere, D. Yun, J. Ok, and Y. Yi, "Optimal rate sampling in 802.11 systems," in *IEEE INFOCOM'14*, 2014.

[25] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *ACM MOBICOM'01*, 2001.

[26] 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.

[27] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *ACM MobiSys'08*, 2008.

[28] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *ACM SIGCOMM'09*, 2009.

[29] H. Rahul, F. Edalat, D. Katabi, and C. Sodinii, "Frequency-aware rate adaptation and MAC protocols," in *ACM MOBICOM'09*, 2009.

[30] R. Crepaldi, J. Lee, R. Etkin, S.-J. Lee, and R. Kravets, "CSI-SF: Estimating wireless channel state using CSI sampling and fusion," in *IEEE INFOCOM'12*, 2012.

[31] J. K. Kuri and S. Kumar, "Reliable multicast in multi-access wireless LANs," *ACM/Kluwer Wirel. Netw.*, vol. 7, pp. 359–369, 2001.

[32] M.-T. Sun, L. Huang, A. Arora, and T.-H. Lai, "Reliable MAC layer multicast in IEEE 802.11 wireless networks," in *IEEE ICPP'02*, 2002.

[33] 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.

[34] N. Choi, Y. Seok, T. Kwon, and Y. Choi, "Leader-based multicast service in IEEE 802.11v networks," in *IEEE CCNC'10*, 2010.

[35] Z. Li and T. Herfet, "BLBP: a beacon-driven leader based protocol for MAC layer multicast error control in wireless LANs," in *IEEE WiCOM'08*, 2008.

[36] V. Srinivas and L. Ruan, "An efficient reliable multicast protocol for 802.11-based wireless LANs," in *IEEE WoWMoM'09*, 2009.

[37] X. Wang, L. Wang, Y. Wang, Y. Zhang, and A. Yamada, "Supporting MAC layer multicast in IEEE 802.11n: Issues and solutions," in *IEEE WCNC'09*, 2009.

[38] E. Park, S. Han, H. Kim, K. Son, and L. Jing, "Efficient multicast video streaming for IPTV service over WLAN using CC-FEC," in *IEEE ICICSE'08*, 2008.

[39] O. Alay, T. Korakis, Y. Wang, and S. Panwar, "Dynamic rate and FEC adaptation for video multicast in multi-rate wireless networks," *ACM Mobile Netw. and Appl.*, vol. 15, no. 3, pp. 425–434, 2010.

[40] 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.

[41] Y. Seok and Y. Choi, "Efficient multicast supporting in multi-rate wireless local area networks," in *IEEE ICOIN'03*, 2003.

[42] A. Basalamah, H. Sugimoto, and T. Sato, "Rate adaptive reliable multicast MAC protocol for WLANs," in *IEEE VTC'06*, 2006.

[43] 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.

[44] X. Wang, L. Wang, Y. Wang, and D. Gu, "Reliable multicast mechanism in WLAN with extended implicit MAC acknowledgment," in *IEEE VTC'08*, 2008.

[45] V. Srinivas and L. Ruan, "An efficient reliable multicast protocol for 802.11-based wireless LANs," in *IEEE WoWMoM'09*, 2009.

[46] Z. Feng, G. Wen, C. Yin, and H. Liu, "Video stream groupcast optimization in WLAN," in *IEEE ITA'10*, 2010.

[47] "IEEE draft standard for information technology telecommunications and information exchange between systems local and metropolitan area networks - specific requirements, part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications - amendment: MAC enhancements for robust audio video streaming," IEEE, July 2011.

[48] Y. Park, C. Jo, S. Yun, and H. Kim, "Multi-room IPTV delivery through pseudo-broadcast over IEEE 802.11 links," in *IEEE VTC'10*, 2010.

[49] Z. Li and T. Herfet, "HLBP: a hybrid leader based protocol for MAC layer multicast error control in wireless LANs," in *IEEE GLOBECOM'08*, 2008.

[50] Y. Bejerano, J. Ferragut, K. Guo, V. Gupta, C. Gutterman, T. Nandagopal, and G. Zussman, "Experimental evaluation of a scalable WiFi multicast scheme in the ORBIT testbed," in *3rd GENI Research and Educational Experiment Workshop (GREE)*, 2014.

[51] V. Sgardoni, M. Sarafianou, P. Ferre, A. Nix, and D. Bull, "Robust video broadcasting over 802.11a/g in time-correlated fading channels," *IEEE Trans. Consum. Electron.*, vol. 55, no. 1, pp. 69–76, 2009.

[52] P. Salvador, L. Cominardi, F. Gringoli, and P. Serrano, "A first implementation and evaluation of the IEEE 802.11 aa group addressed transmission service," *ACM SIGCOMM Comp. Comm. Rev.*, vol. 44, no. 1, pp. 35–41, 2013.

[53] K. Lin, W. Shen, C. Hsu, and C. Chou, "Quality-differentiated video multicast in multi-rate wireless networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 21–34, January 2013.

[54] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. on Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[55] F. Wu, Y. Sun, Y. Yang, K. Srinivasan, and N. B. Shroff, "Constant-delay and constant-feedback moving window network coding for wireless multicast: Design and asymptotic analysis," *IEEE J. Sel. Areas in Commun.*, vol. 33, no. 2, pp. 127–140, 2015.

[56] W. Xiao, S. Agarwal, D. Starobinski, and A. Trachtenberg, "Reliable rateless wireless broadcasting with near-zero feedback," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1924–1937, 2012.

[57] U. C. Kozat, "On the throughput capacity of opportunistic multicasting with erasure codes," in *Proc. IEEE INFOCOM'08*, 2008.

[58] F. Wu, Y. Yang, O. Zhang, K. Srinivasan, and N. B. Shroff, "Anonymous-query based rate control for wireless multicast: Approaching optimality with constant feedback," in *ACM Mobihoc'16*, 2016.

[59] V. Gupta, L. Xu, B. Wu, C. Gutterman, Y. Bejerano, and G. Zussman, "Demo: Evaluating video delivery over wireless multicast," in *Proc. IEEE INFOCOM'17*, 2017.

[60] "ORBIT testbed," http://orbit-lab.org/.

[61] M. Luby, "Lt codes," in *Proc. IEEE FOCS'02*, 2002.

[62] A. Shokrollahi, "Raptor codes," *IEEE Trans. on Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[63] 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.

[64] J. Ababneh and O. Almomani, "Survey of error correction mechanisms for video streaming over the internet," in *IJACSA*, vol. 5, 2014, pp. 155–161.

[65] "Minstrel," https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel.

[66] 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.

[67] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "A quest for an internet video quality-of-experience metric," in *ACM HotNets'12*, 2012.

[68] 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.

[69] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *ACM SIGCOMM'10*, 2010.

[70] ——, "Tool release: Gathering 802.11 n traces with channel state information," *ACM SIGCOMM Comp. Comm. Rev*, vol. 41, no. 1, pp. 53–53, 2011.

[71] 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.

[72] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic http streaming," in *ACM CONEXT'12*, 2012.

[73] "Video dataset," https://media.xiph.org/video/derf/.

Varun Gupta completed his undergraduate studies in Electrical Engineering from the Indian Institute of Technology, Delhi in 2011 and received the M.S. degree in Electrical Engineering from Columbia University, New York in 2012. He is currently working towards his Ph.D. at Columbia University. His research interests include network performance evaluation, stochastic modeling, wireless networks, and datacenter networks.

Craig Gutterman received the B.Sc. degree in Electrical Engineering from Rutgers University, New Jersey in 2012 and received the M.S degree in Electrical Engineering from Columbia University, New York in 2014. He is currently working towards his Ph.D. at Columbia University. He received the NSF GRFP and the From Data to Solutions NSF IGERT fellowships. His interests include mobile and wireless networks, optimization algorithms, and machine learning. His research focuses on analyzing data and developing algorithms to improve the performance of networked systems.

Yigal Bejerano is currently a Visiting Research Scientist at Columbia University. He received his Ph.D. Degree in Electrical Engineering in 2000, from the Technion - Israel Institute of Technology, Haifa, Israel. After graduation, he joined Bell-Labs as a researcher until 2017, His research interests are management aspects of high-speed and wireless networks. He published over 65 peer-reviewed research papers at the leading conferences and journals of the networking community and holds over 65 US and international patents. He is a co-recipient of Outstanding Paper Award at IEEE HPSR 2012, Best Paper Award at IEEE HPSR 2013, Best Paper Award Runner-UP at IEEE Infocom 2017. Dr. Bejerano is on the technical program committee (TPC) of numerous conferences and he also serves as an Associate Editor of the IEEE/ACM Transactions on Networking.

Gil Zussman received the Ph.D. degree in electrical engineering from the Technion in 2004 and was a postdoctoral associate at MIT in 2004-2007. Since 2007 he has been with Columbia University where he is currently an Associate Professor of Electrical Engineering. His research interests are in the area of networking, and in particular in the areas of wireless, mobile, and resilient networks. He is a co-recipient of 7 paper awards including the ACM SIGMETRICS'06 Best Paper Award, the 2011 IEEE Communications Society Award for Advances in Communication, and the ACM CoNEXT'16 Best Paper Award. He received the Fulbright Fellowship, the DTRA Young Investigator Award, and the NSF CAREER Award, and was the PI of a team that won first place in the 2009 Vodafone Foundation Wireless Innovation Project competition.