DyMo: Dynamic Monitoring of Large Scale LTE-Multicast Systems

Yigal Bejerano*, Chandru Raman†, Chun-Nam Yu*, Varun Gupta‡, Craig Gutterman‡, Tomas Young†, Hugo Infante†, Yousef Abdelmalek@ and Gil Zussman‡

∗ Bell Labs, Nokia, Murray Hill, NJ, USA.
† Mobile Networks, Nokia, Murray Hill, NJ, USA.
‡ Electrical Engineering, Columbia University, New York, NY, USA.
@ Verizon Wireless, Basking Ridge, NJ, USA.

***Abstract*—LTE *evolved Multimedia Broadcast/Multicast Service* (eMBMS) is an attractive solution for video delivery to very large groups in crowded venues. However, deployment and management of eMBMS systems is challenging, due to the lack of real-time feedback from the User Equipment (UEs). Therefore, we present the *Dynamic Monitoring* (*DyMo*) system for low-overhead feedback collection. *DyMo* leverages eMBMS for broadcasting *Stochastic Group Instructions* to *all UEs*. These instructions indicate the reporting rates as a function of the observed Quality of Service (QoS). This simple feedback mechanism collects very limited QoS reports from the UEs. The reports are used for network optimization, thereby ensuring high QoS to the UEs. We present the design aspects of *DyMo* and evaluate its performance analytically and via extensive simulations. Specifically, we show that *DyMo* infers the optimal eMBMS settings with extremely low overhead, while meeting strict QoS requirements under different UE mobility patterns and presence of network component failures. For instance, *DyMo* can detect the eMBMS Signal-to-Noise Ratio (SNR) experienced by the 0.1% percentile of the UEs with Root Mean Square Error (RMSE) of 0.05% with only 5 to 10 reports per second regardless of the number of UEs.**

Keywords*— *Wireless Monitoring, LTE, eMBMS, Multicast, Feedback Mechanism

I. Introduction

Wireless video delivery is an important service. However, unicast video streaming over LTE to a large user population in crowded venues requires a dense deployment of Base Stations (BSs) [1], [2]. Such deployments require high capital and operational expenditure and may suffer from extensive interference between adjacent BSs.

LTE-eMBMS (evolved Multimedia Broadcast/Multicast Service) [3], [4] provides an alternative method for content delivery in crowded venues which is based on broadcasting to a large population of User Equipment (UEs) (a.k.a. eMBMS receivers). As illustrated in Fig. 1, in order to improve the Signal-to-Noise Ratio (SNR) at the receivers, eMBMS utilizes soft signal combining techniques.[1] *Thus, a large scale Modulation and Coding Scheme (MCS) adaptation should be conducted simultaneously for all the BSs based on the Quality of Service (QoS) at the UEs.*

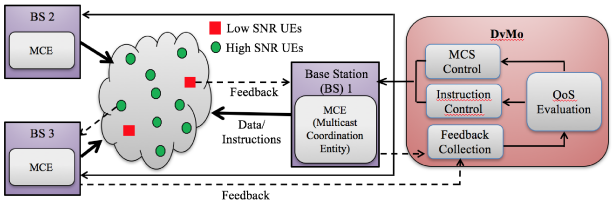


Fig. 1. The *DyMo* system architecture: It exchanges control information with the Multicast Coordination Entity (MCE) of BSs which use soft signal combining for eMBMS. The Instruction Control module uses broadcast to dynamically partition the UEs into groups, each sending QoS reports at a different rate. The reports are sent to the Feedback Collection module and allow the QoS Evaluation module to identify an SNR Threshold. It is used by the MCS Control module to specify the optimal MCS to the MCEs.

eMBMS Limitations: Unfortunately, the eMBMS standard [3] only provides a mechanism for UE QoS reporting once the communication terminates, thereby making it unsuitable for real-time traffic. Recently, the Minimization of Drive Tests (MDT) protocol [5] was extended to provide eMBMS QoS reports periodically from all the UEs or when a UE joins/leaves a BS. However, in crowded venues with tens of thousands of UEs (e.g., [1]), even infrequent QoS reports by each UE may result in high signaling overhead and blocking of unicast traffic.[2] Due to the limited ability to collect feedback, a deployment of an eMBMS system is very challenging. In particular, it is hindered by the following limitations:

(i) *Extensive and time consuming radio frequency surveys*: Such surveys are conducted before each new eMBMS deployment. Yet, they provide only limited information from a few monitoring nodes.
(ii) *Conservative resource allocation*: The eMBMS MCS and Forward Error Correction (FEC) codes are set conservatively to increase the decoding probability.
(iii) *Oblivious to environmental changes*: It is impossible to infer QoS degradation due to environmental changes, such as new obstacles or component failures.

[1] All the BSs in a particular venue transmit identical multicast signals in a time synchronized manner.

[2] A BS can only support a limited number of connections while the minimal duration for an LTE connection is in the order of hundreds of milliseconds.

eMBMS Parameter Tuning Challenge: Clearly, there is a need to dynamically tune the eMBMS parameters according to the feedback from UEs. However, a key challenge for eMBMS parameter tuning for large scale groups is *obtaining accurate QoS reports with low overhead*. Schemes for efficient feedback collection in wireless multicast networks have recently received considerable attention, particularly in the context of WiFi networks (e.g., [6]–[9]). Yet, WiFi feedback schemes cannot be easily adapted to eMBMS since unlike WiFi, where a single Access Point transmits to a node, transmissions from multiple BSs are combined in eMBMS. Efforts for optimizing eMBMS performance focus on periodically collecting QoS reports from all UEs (e.g., [10]) but such approaches rely on extensive knowledge of the user population (for more details, see Section II).

***DyMo* System:** This paper presents the *Dynamic Monitoring* (*DyMo*) system designed to support efficient LTE-eMBMS deployments in crowded and dynamic environments by providing accurate QoS reports with low overhead. *DyMo* identifies the maximal eMBMS *SNR Threshold* such that only a small number of UEs with SNR below the SNR Threshold may suffer from poor service[3]. To identify the SNR Threshold accurately, *DyMo* leverages the broadcast capabilities of eMBMS for fast dissemination of instructions to a large UE population.

Each instruction is targeted at a sub-group of UEs that satisfies a given condition. It instructs the UEs in the group to send a QoS report with some probability during a reporting interval.[4] We refer to these instructions as *Stochastic Group Instructions*. For instance, as shown in Fig. 2, *DyMo* divides UEs into two groups. UEs with poor or moderate eMBMS SNR are requested to send a report with a higher rate during the next reporting interval. In order to improve the accuracy of the SNR Threshold, the QoS reports are analyzed and the group partitions and instructions are dynamically adapted such that the UEs whose SNR is around the SNR Threshold report more frequently. The SNR Threshold is then used for setting the eMBMS parameters, such as the MCS and FEC codes.

From a statistics perspective, *DyMo* can be viewed as a practical method for realizing *importance sampling* [11] in wireless networks. Importance sampling improves the expectation approximation of a rare event by sampling from a distribution that overweighs the important region. With limited knowledge of the SNR distribution, *DyMo* leverages Stochastic Group Instructions to narrow down the SNR sampling to UEs that suffer from poor service and consequently obtains accurate estimation of the SNR Threshold. To the best of our knowledge, this is the first realization of using broadcast instructions for importance sampling in wireless networks.

The *DyMo* system architecture is illustrated in Fig. 1. It operates on an independent server and exchanges control information with several BSs supporting eMBMS. The Instruction Control module instructs the different groups to send reports at different rates. The reports are sent via unicast to the Feedback Collection module and allow the QoS Evaluation module to identify an accurate SNR Threshold. The SNR Threshold is determined such that only a predefined number of UEs with SNR below the threshold, termed as *outliers*, may suffer from poor service. The MCS Control module utilizes the SNR Threshold to configure the eMBMS parameters (e.g., MCS) accordingly. Finally, the QoS Evaluation module continually refines group partitions based on the reports.

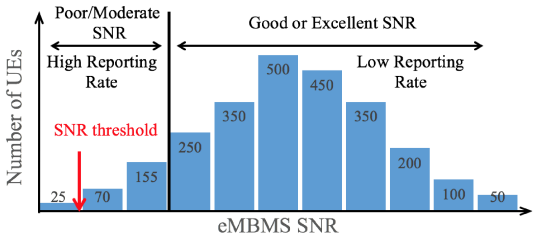


Fig. 2. Operation of *DyMo* for a sample UE QoS distribution: UEs are partitioned into two groups based on their SNR and each group is instructed to send QoS reports at a different rate. The partitioning is dynamically adjusted based on the reports to yield more reports from UEs whose SNR is around the estimated SNR Threshold.

We focus on the QoS Evaluation module and develop a *Two-step estimation* algorithm which can efficiently identify the SNR Threshold as a one time estimation. We also develop an *Iterative estimation* algorithm for estimating the SNR Threshold iteratively, when the distribution changes due to UE mobility or environmental changes, such as network component failures. Our analysis shows that the *Two-step estimation* and *Iterative estimation* algorithms can infer the SNR Threshold with a small error and limited number of QoS reports. It is also shown that they outperform the *Order-Statistics estimation* method, a well-known statistical method, which relies on sampling UEs with a fixed probability. For instance, the *Two-step estimation* requires only 400 reports when estimating the 1% percentile to limit the error to 0.3% for each re-estimation. The *Iterative estimation* algorithm performs even better and the maximum estimation error can be bounded according to the maximum change of SNR Threshold.

We conduct extensive at-scale simulations, based on real eMBMS radio survey measurements from a stadium and an urban area. It is shown that *DyMo* accurately infers the SNR Threshold and optimizes the eMBMS parameters with low overhead under different mobility patterns and even in the event of component failures. *DyMo* significantly outperforms alternative schemes based on the *Order-Statistics estimation* method which rely on random or periodic sampling.

Our simulations show that both in a stadium-like and urban area, *DyMo* detects the *eMBMS SNR value of the* 0.1% *percentile with Root Mean Square Error (RMSE) of* 0.05% with only 5 messages per second. This is at least 8 times better than *Order-Statistics estimation* based methods. *DyMo* also infers the optimal SNR Threshold with RMSE of 0.3 dB regardless of the UE population size, while the error of the best *Order-Statistics estimation* method is above 1 dB. *DyMo* violates the outlier bound (of 0.1%) with RMSE of at most 0.35 while the best *Order-Statistics estimation* method incurs RMSE of over 4 times. The simulations also show that after

[3]While various metrics can be used for QoS evaluation, we consider the commonly used eMBMS SNR, referred to as SNR, as a primary metric.

[4]A higher probability results in a higher reporting rate, and therefore, we will use rate and probability interchangeably.

a failure, *DyMo* converges instantly (i.e., in a single reporting interval) to the optimal SNR Threshold. Thus, *DyMo* is able to infer the maximum MCS while preserving QoS constraints.

Our Main Contributions: To summarize, the main contributions of this paper are three-fold:

(i) We present the concept of Stochastic Group Instructions for efficient realization of importance sampling in wireless networks.

(ii) We present the system architecture of *DyMo* and efficient algorithms for SNR Threshold estimation.

(iii) We show via analysis and extensive simulations that *DyMo* performs well in diverse scenarios.

The principal benefit of *DyMo* is its ability to infer the system performance based on a low number of QoS reports. It converges very fast to the optimal eMBMS configuration and it reacts very fast to changes in the environment. Hence, it eliminates the need for service planning and extensive field trials. Further, *DyMo* is compatible with existing LTE-eMBMS deployments and does not need any knowledge of the UE population. We note that, due to space constraints, the proofs and some simulation results are omitted and appear in [12].

II. Related Work

Wireless multicast control schemes received considerable attention in recent years (see survey in [6] and references therein). Below we briefly review the most relevant papers.

LTE-eMBMS: Most previous work on eMBMS (e.g., [13]–[16]) assumes individual feedback from all the UEs and proposes various MCS selection or resource allocation techniques. Yet, extensive QoS reports impose significant overhead on LTE networks, which are already highly congested in crowded venues [1]. An efficient feedback scheme was proposed in [10] but it relies on knowledge of path loss (or block error) of the entire UE population to form the set of feedback nodes.

Recently, [17] proposed a multicast-based anonymous query scheme for inferring the maximum MCS that satisfies *all UEs* without sending individual queries. However, the scheme cannot be implemented in current LTE networks, since it will require UEs to transmit simultaneous beacon messages in response to broadcast queries.

WiFi Multicast: Most of the wireless multicast schemes are designed for WiFi networks. Some rely on *individual feedback* from all nodes for each packet [8], [9]. *Leader-Based Schemes* [18]–[20] collect feedback from a few selected nodes with the weakest channel quality. *Cluster-Based Feedback Schemes* in [7], [21] balance accurate reporting with minimization of control overhead by selecting nodes with the weakest channel condition in each cluster as feedback nodes.

However, WiFi multicast solutions cannot easily be applied to LTE-eMBMS systems. First, in WiFi, each node is associated with an Access Point, and therefore, the Access Point is aware of every node and can specify the feedback nodes. In LTE, eMBMS UEs could be in the idle state and *the network may not be aware of the number of active UEs*. Second, eMBMS is based on simultaneous transmission from various BSs. Thus, unlike in WiFi where MCS adaptation is done at each Access Point independently, a common MCS adaptation should be done at all BSs.

III. Model and Objective

A. Network Model

We consider an LTE-Advanced network with multiple BSs providing eMBMS service to a very large group of m UEs in a given large venue (e.g., sports arena, transportation hub).[5] Such venues can accommodate tens of thousands of users. The eMBMS service is managed by a single *DyMo* server as shown in Fig. 1 and all the BSs transmit identical multicast signals in a time synchronized manner. The multicast flows contain FEC code that allows the UEs to tolerate some level of losses ℓ (e.g., up to 5% packet losses).

All UEs can detect and report the eMBMS QoS they experience. More specifically, time is divided into short *reporting intervals*, a few seconds each. We assume that the eMBMS SNR distribution of the UEs does not change during each reporting interval.[6] We define the *individual SNR value* $h_v(t)$, such that at least a given percentage $1-\ell$ (e.g., 95%) of the eMBMS packets received by an UE v during a reporting interval t have an SNR above $h_v(t)$. For a given SNR value, $h_v(t)$, there is a one-to-one mapping to an *eMBMS MCS* such that a UE can decode all the packets whose SNR is above $h_v(t)$ [15], [16]. The remaining packets ℓ can be recovered by appropriate level of FEC assuming ℓ is not too large.

B. Objective

We aim to design a scalable efficient eMBMS monitoring and control system for which the objective is outlined below and that satisfies the following constraints:

(i) *QoS Constraint* – Given a *QoS Threshold* $p \ll 1$, at most a fraction p of the UEs may suffer from packet loss of more than ℓ. This implies that, *with FEC, a fraction* $1-p$ *of the UEs should receive all of the transmitted data*. We refer to the set UEs that suffer from packet loss after FEC as *outliers* and the rest are termed *normal* UEs.

(ii) *Overhead Constraint* – The average number of UE reports during a reporting interval should be below a given *Overhead Threshold* r.

Objective: *Accurately identify at any given time* t *the maximum SNR Threshold,* $s(t)$ *that satisfies the QoS and Overhead Constraints.*

Namely, the calculated $s(t)$ needs to ensure that a fraction $1-p$ of the UEs have individual SNR values $h_v(t) \geq s(t)$.

The network performance can be maximized by using $s(t)$ to calculate the maximum eMBMS MCS that meets the QoS constraint [15], [16]. This allows reducing the resource blocks allocated to eMBMS. Alternatively for a service such as video, the video quality can be enhanced without increasing the bandwidth allocated to the video flow.

[5] In this paper, we consider only the UEs subscribing to eMBMS services.

[6] The SNR of each individual eMBMS packet is a random variable selected from the UE SNR distribution. We assume that this distribution does not change significantly during the reporting interval.

TABLE I
EXAMPLE OF THE *DyMo* FEEDBACK REPORT OVERHEAD.

Group	No. of UEs	Report Prob.	Avg. reports per interval	Avg. per sec	Rate per min
H	250	20%	50	5	$\approx 100\%$
L	2250	2%	45	≈ 5	$\approx 12\%$

IV. THE *DyMo* SYSTEM

We now briefly present the *DyMo* system architecture, shown in Fig. 1. The details can be found in [12].

Feedback Collection: This module operates in the *DyMo* server and in a DyMo *Mobile-Application* on each UE. At the beginning of each reporting interval, the Feedback Collection module broadcasts *Stochastic Group Instructions* to all the UEs. These instructions specify the QoS report probability as a function of the observed QoS (i.e., eMBMS SNR). In response, each UE independently determines whether it should send a QoS report at the current reporting interval.

QoS Evaluation: The UE feedback is used to estimate the eMBMS SNR distribution, as shown in Fig. 2. Since the system needs to determine the SNR Threshold, $s(t)$, the estimation of the low SNR range of the distribution has to be more accurate. To achieve this goal, the QoS Evaluation module partitions the UEs into two or more groups, according to their QoS values. This allows *DyMo* to accurately infer the optimal value of $s(t)$, by obtaining more reports from UEs with low SNR. We elaborate on the algorithms for $s(t)$ estimation in Section V.

MCS Control: The initial eMBMS MCS is determined from unicast SNR values reported by the UEs during unicast connections. Then, after each reporting interval, the QoS Evaluation module infers the SNR Threshold, $s(t)$, and the MCS Control module determines the desired eMBMS settings, mainly the eMBMS MCS and FEC, according to commonly used one-to-one mappings [15], [16]. This iterative process is demonstrated in the following example.

Example: Consider an eMBMS system that serves $2,500$ UEs with the QoS Constraint that at most $p = 1\% = 25$ UEs may suffer from poor service. Assume a reporting interval of 10 seconds. To infer the SNR Threshold, $s(t)$, that satisfies the constraint, the UEs are divided into two groups:

- *High-Reporting-Rate* (H): 10% (250) of UEs that experience poor or moderate service quality report with probability of 20%, i.e., an expected number of 50 reports per interval.
- *Low-Reporting-Rate* (L): 90% (2250) of the UEs that experience good or excellent service quality report with probability of 2%, implying about 45 reports per interval.

Table I presents the reporting probability of each UE and the number of QoS reports per reporting interval by each group. It also shows the number of QoS reports per second and the reporting rate per minute (i.e., the expected fraction of UEs that send QoS reports in a minute). Since the QoS Constraint implies that only 25 UEs may suffer from poor service, these UEs must belong to group H. Although only 10 QoS reports are received at each second, all the UEs in group H send QoS reports at least once a minute. Thus, the SNR Threshold can be accurately detected within one minute.

V. ALGORITHMS FOR SNR THRESHOLD ESTIMATION

This section describes the algorithms utilized by *DyMo* for estimating the SNR Threshold, $s(t)$, for a given QoS Constraint, p and Overhead Constraint r. In particular, it addresses the challenges of partitioning the UEs into groups according to their SNR distribution as well as determining the group boundaries and the reporting rate from the UEs in each group, such that the overall estimation error of $s(t)$ is minimized. We first consider a static setting where the SNR values of UEs are fixed and then extend to the case of dynamic environments and UE mobility. The proofs are omitted due to space constraints and can be found in [12].

A. Order Statistics

We first briefly review a known statistical method in quantile estimation, referred to as *Order-Statistics estimation*. It provides a baseline for estimating $s(t)$ and is also used by *DyMo* for determining the initial SNR distribution in its first iteration assuming a single group. Let $F(x)$ be a Cumulative Distribution Function (CDF) for a random variable X, the quantile function $F^{-1}(p)$ is given by, $\inf\{x \mid F(x) \geq p\}$.

Let $X_1, X_2, \ldots, X_r$ be a sample from the distribution F, and F_r its empirical distribution function. It is well known that the empirical quantile $F_r^{-1}(p)$ converges to the population quantile $F^{-1}(p)$ at all points p where F^{-1} is continuous [22]. Moreover, the true quantile, $S_p = F(F_r^{-1}(p))$, of the empirical quantile estimate $F_r^{-1}(p)$ is asymptotically normal [22] with mean p and variance

$$Var[S_p] = p(1-p)/r. \tag{1}$$

For SNR Threshold estimation, F is the SNR distribution of all UEs. A direct way to estimate the SNR Threshold $s(t)$ is to collect QoS reports from r randomly selected UEs, and calculate the empirical quantile $F_r^{-1}(p)$ as an estimate.[7]

B. The Two-Step Estimation Algorithm

We now present the *Two-step estimation* algorithm that uses two groups for estimating the SNR Threshold, $s(t)$, in a static setting. We assume a fixed number of UEs, m, and a bound r on the number of expected reports. By leveraging *Stochastic Group Instructions*, *DyMo* is not restricted to collecting reports uniformly from all UEs and can use these instructions to improve the accuracy of $s(t)$. One way to realize this idea is to perform a two-step estimation that approximates the shape of the SNR distribution before focusing on the low quantile tail. The *Two-step estimation* algorithm works as follows:

Algorithm 1: Two-Step Estimation for the Static Case

1) Select p_1 and p_2 such that $p_1 p_2 = p$. Use p_1 as the percentile boundary for defining the two groups.
2) Select number of reports r_1 and r_2 for each step such that $r_1 + r_2 = r$.

[7]Note that F can have at most m points of discontinuity. Therefore, we assume p is a point of continuity for F^{-1} to enable normal approximation. If the assumption does not hold, we can always perturb p by an infinitesimal amount to make it a point of continuity for F^{-1}.

3) Instruct all UEs to send QoS reports with probability r_1/m and use these reports to estimate the p_1 quantile $\hat{x}_1 = F^{-1}_{r_1}(p_1)$.
4) Instruct UEs with SNR value below $\hat{x}_1$ to send reports with probability $r_2/(p_1 \cdot m)$ and calculate the p_2 quantile $\hat{x}_2 = G^{-1}_{r_2}(p_2)$ as an estimation for $s(t)$ (G is the CDF of the subpopulation with SNR below $\hat{x}_1$).

Upper Bound Analysis of the Two-Step Algorithm: To simplify the notation, we use r_1 and r_2 to denote the expected number of reports at each step. From (1) we know that $\hat{p}_1 = F^{-1}(\hat{x}_1)$ and $\hat{p}_2 = G^{-1}(\hat{x}_2)$ are unbiased estimators of p_1 and p_2 with variance $p_1(1-p_1)/r_1$ and $p_2(1-p_2)/r_2$. Our estimate $\hat{x}_2$ has true quantile $\hat{p}_1\hat{p}_2$. Assume $\hat{p}_1$ is less than $p_1+\epsilon_1$ and $\hat{p}_2$ is less than $p_2+\epsilon_2$ with high probability (for example, we can take ϵ_1 and ϵ_2 to be 3 times the standard deviation for $> 99.8\%$ probability). Then, the over-estimation error is bounded by $(p_1+\epsilon_1)(p_2+\epsilon_2) - p \approx \epsilon_1 p_2 + \epsilon_2 p_1$, after ignoring the small higher order term $\epsilon_1\epsilon_2$.

The case for under-estimation is similar. By using symmetry arguments, we show in [12] that the error is minimized by taking $p_1 = p_2 = \sqrt{p}$, and $r_1 = r_2 = r/2$ so that $\epsilon_1 = \epsilon_2 = 3\sqrt{\sqrt{p}(1-\sqrt{p})/(r/2)}$. This leads to proposition 1.

Proposition 1. *The distance between p and the quantile of the Two-Step estimator $\hat{x}_2$, $\hat{p} = F^{-1}(x_2)$, is bounded by*

$$6\sqrt{2}\sqrt{\frac{p\sqrt{p}(1-\sqrt{p})}{r}}$$

with probability at least $1 - 2(1-\Phi(3)) > 99.6\%$, where Φ is the normal CDF.

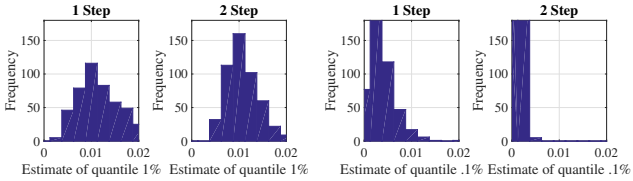
We now compare this result against the bound of 3 standard deviations in the Order Statistics case, which is $3\sqrt{p(1-p)/r}$. With some simple calculations, it can be easily shown that if $p \leq 1/49 \approx 2\%$, the *Two-step estimation* has smaller error than the *Order-Statistics estimation* method. Essentially the *Order-Statistics estimation* method has an error of order $\sqrt{p}/\sqrt{r}$, while the *Two-step estimation* has an error of order $p^{3/4}/\sqrt{r}$. Since $p \ll 1$, the difference can be significant.

Example: We validated the error estimation of the *Two-step estimation* algorithm and the *Order-Statistics estimation* method by numerical analysis. We considered the cases of $p = 1\%$ and $p = 0.1\%$ of uniform distribution on $[0,1]$ using $r = 400$ samples over population size of 10^6. The *Two-step estimation* algorithm has smaller standard error compared to the *Order-Statistics estimation*, as shown in Fig. 3. Its accuracy is significantly better for very small p.

The *Two-step estimation* algorithm can be generalized to 3 or more telescoping group sizes, but p will need to be much smaller for these sampling schemes in order to reduce the number of samples.

C. The Iterative Estimation Algorithm

We now turn to the dynamic case in which *DyMo* uses the SNR Threshold estimation $s(t-1)$ from the previous reporting interval to estimate $s(t)$ at the end of reporting interval t. Assume that the total number of UEs m is known initially.



(a) (b)

Fig. 3. Estimates of (a) $p = 1\%$ and (b) $p = 0.1\%$ quantiles for 500 runs for the *Order-Statistics estimation* (1-step) method and the *Two-step estimation* algorithm.

Suppose that *DyMo* has a current estimate $\hat{x}$ of the SNR threshold, $s(t)$, and $s(t)$ changes over time. We assume that the change in SNR of each UE is bounded over a time period. Formally, $|h_v(t_1) - h_v(t_2)| \leq L|t_1 - t_2|$, where L is a Lipschitz constant for SNR changes. For example, we can assume that the UEs' SNR cannot change by more than 5dB during a reporting interval. [8] This implies that within the interval, only UEs with SNR below $\hat{x} + 5$dB affect the estimation of the p quantile (subject to small estimation error in $\hat{x}$).

DyMo only needs to monitor UEs with SNR below $x_L = \hat{x} + L$. Denote the true quantile of x_L, defined by $F^{-1}(x_L)$, as p_L. To apply a process similar to the second step of the *Two-step estimation* algorithm by focusing on UEs with SNR below x_L, first an estimate of p_L is required. *DyMo* uses the previous SNR distribution to estimate p_L and instructs the UEs to send reports at a rate $q = r/(p_L \cdot m)$. Let Y be the number of reports received during the last reporting interval, then $Y/m \cdot q$ can be used as an updated estimator for p_L. This estimator is unbiased and has variance $\frac{p_L}{m}\frac{1-q}{q}$. As a result, the Iterative Estimation algorithm works as follows:

Algorithm 2: Iterative Estimation for the Dynamic Case

1) Instruct UEs with SNR below $\hat{x} + L$ to send reports at a rate q. Construct an estimator $\hat{p}_L$ of p_L from the number of received reports Y.
2) Set $p' = p/\hat{p}_L$. Find the p' quantile $x' = G^{-1}_Y(p')$ and report it as the p quantile of the whole population (G is the CDF of the subpopulation with SNR below $\hat{x} + L$).

Upper Bound Analysis of the Iterative Algorithm: Suppose the estimation error of p_L is bounded by ϵ_1, and the estimation error of $p' = p/\hat{p}_L$ is bounded by ϵ_2 with high probability. Then, the estimation error is

$$(\frac{p}{\hat{p}_L} \pm \epsilon_2)p_L - p = (\frac{p}{p_L \pm \epsilon_1} \pm \epsilon_2)p_L - p.$$

The over-estimation error is bounded by

$$\frac{p}{p_L - \epsilon_1}\epsilon_1 + p_L\epsilon_2. \tag{2}$$

If we assume $p_L - \epsilon_1 \geq p$ (we know $p_L \geq p$ by the Lipschitz assumption), then the bound can be simplified to $\epsilon_1 + p_L\epsilon_2$. The same bound also works for the under-estimation error.

[8] In our simulations, each reporting interval has a duration of 12s.

If r denotes also the expected number of samples collected, $r = p_L \cdot m \cdot q$. The standard deviation of $\hat{p}_L$ can be written as:

$$\sqrt{\frac{p_L}{m}\frac{1-q}{q}} = \sqrt{\frac{p_L^2}{r}(1-\frac{r}{p_L m})} \leq \frac{p_L}{\sqrt{r}}.$$

If we assume $\epsilon_1 = 3p_L/\sqrt{r}$, the error of $\hat{p}_L$ is less than ϵ_1 with probability at least $\Phi(3)$. Since we assume $p_L - \epsilon_1 \geq p$ above, this implies $(1–3/\sqrt{r})p_L \geq p$. If $r \geq 100$, then $p < 0.7p_L$ will satisfy our requirement.

The standard deviation of estimating the $p' = p/\hat{p}_L$ quantile is

$$\sqrt{\frac{1}{Y}\frac{p}{\hat{p}_L}(1-\frac{p}{\hat{p}_L})} \leq \frac{1}{2\sqrt{Y}}, \tag{3}$$

by using the fact that $x(1-x) \leq 1/4$ for $x \in [0,1]$ and Y is the number of reports received (a random variable). If the expected number of reports r is reasonably large (≥ 100, say), then Y can be well approximated by a normal and $Y \geq 0.7r$ with high probability $\Phi(3) = 99.8\%$. Then, (3) is bounded by $2/(3\sqrt{r}) \geq 1/(2\sqrt{0.7r})$ with high probability ($\Phi(3) = 99.8\%$), and we can set $\epsilon_2 = 2/\sqrt{r}$. Substituting these back into (2), gives us the following proposition.

Proposition 2. *The distance between p and the quantile of the estimator x, $\hat{p} = F^{-1}(x)$, is approximately bounded by*

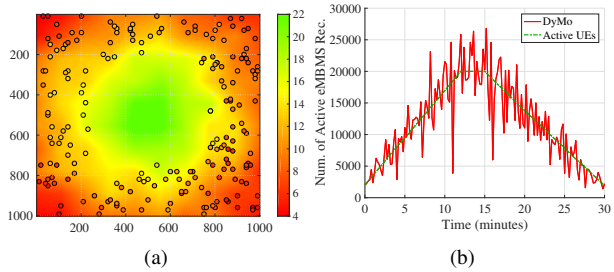
$$5\frac{p_L}{\sqrt{r}}$$

with probability at least $1 - 2(1 - \Phi(3)) > 99.6\%$, if the expected sample size $r \geq 100$ and $p \leq 0.7p_L$.

This shows that the error is of order $p_L/\sqrt{r}$. We can see that the estimation error can be smaller compared to the error of order $p^{3/4}/\sqrt{r}$ in the static *Two-step estimation* if p_L is small (i.e., the SNR of individual users does not change much during a reporting interval).

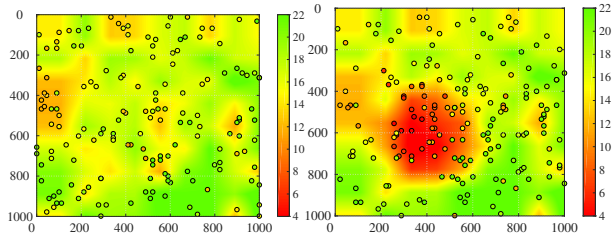
Exponential Smoothing: *DyMo* applies exponential smoothing by weighing past and current reports to smooth the estimates of the SNR Threshold, $s(t)$, and take older reports into account. It estimates the SNR Threshold as $s(t) = \alpha\hat{x}(t) + (1-\alpha)s(t-1)$, where $\hat{x}(t)$ is the new raw SNR Threshold estimate using the *Iterative estimation* above and $s(t-1)$ is the SNR Threshold from the previous reporting interval. We set $\alpha = 0.5$ to allow some re-use of past reports without letting them have too strong an effect on the estimates (e.g., samples older than 7 reporting intervals have less than 1% weight). *DyMo* also uses the exponential smoothing method for estimating the SNR distribution while taking into account QoS reports from previous reporting intervals.

Unknown Number of UEs: If the total number of UEs, m, is unknown or changes dynamically, *DyMo* can estimate m by requiring UEs above the threshold $\hat{x} + L$ to send reports. These UEs can send reports at a lower rate, since m is not expected to change rapidly. Similar to the *Two-step estimation* algorithm, *DyMo* allocates $r_1 = r_2 = r/2$ reports to each group. The errors in estimating the total number of UEs m will contribute to the error ϵ_1 in the estimation of p_L in (2). The error analysis in this case is largely similar.



(a) (b)

Fig. 4. (a) The heatmap of UE SNR distribution in a stadium area of $1000 \times 1000m^2$ and (b) the evolution of the number of active UEs over time compared to the number estimated by *DyMo* for a stadium environment.



(a) (b)

Fig. 5. The heatmap of the SNR distribution of UEs (a) before a failure and (b) after a failure.

VI. Performance Evaluation

A. Methodology

We perform extensive simulations to evaluate the performance of *DyMo* with various values of QoS Constraint, p, Overhead Constraint, r, and number of UEs, m. Our evaluation considers dynamic environments with UE mobility and a changing number of *active UEs*, dynamically selected from the given set of m UEs. In this paper, we present a few sets of simulation results in which the *SNR Threshold, $s(t)$, changes significantly over time*. Additional results can be found in [12].

We consider a variant of *DyMo* where the number of active UEs is unknown and is estimated from its measurements. We compare the performance of *DyMo* to four other schemes. To demonstrate the advantages of *DyMo*, we augment each scheme with additional information, which is hard to obtain in practice. The evaluated benchmarks are the following:

- ***Optimal*** – Full knowledge of SNR values of the UEs at any time and consequently accurate information of the SNR distribution. *This is the best possible benchmark although impractical, due to its high overhead.*
- ***Uniform*** – Full knowledge of the SNR characteristics at any location while assuming uniform UE distribution and static eMBMS settings. In practice, *this knowledge cannot be obtained even with rigorous field trial measurements.*
- ***Order-Statistics*** – It is based estimation of the SNR Threshold using random sampling. The active UEs send reports with a fixed probability of $r/\mathbb{E}[m(t)]$ per second, assuming that the expected number of active UEs, $\mathbb{E}[m(t)]$, is known. We assume that the UEs are configured with this reporting rate during initialization. In practice, $\mathbb{E}[m(t)]$ *is not available. We also ignore initial configuration overhead in*

our evaluation. *Order-Statistics* is the best possible approach when not using broadcast messages for UE configuration. We consider two variants of *Order-Statistics*. The first is ***Order-Statistics w.o. History*** which ignores SNR measurements from earlier reporting intervals. The second variant ***Order-Statistics w. History*** considers the history of reports.

Both *DyMo* and *Order-Statistics w. History* perform the same exponential smoothing process for assigning weights to the measurements from previous reporting intervals with a smoothing factor of $\alpha = 0.5$. We use the following metrics to evaluate the performance of the schemes:

(i) *Accuracy* – The accuracy of the *SNR Threshold* estimation, $s(t)$. After calculating $s(t)$ at each reporting interval, we check the actual *SNR Threshold Percentile* in the accurate SNR distribution. This metric provides the percentile of active UEs with individual SNR values below $s(t)$.

(ii) *QoS Constraint violation* – The number of outliers above the QoS Constraint p.

(iii) *Overhead Constraint violation* – The number of reports above the Overhead Threshold r.

The total simulation time for each instance is 30mins with 5 reporting intervals per minute (each is 12s). During each reporting interval, an active UE may send its SNR value at most once. The accuracy of each SNR report is 0.1dB.

B. Simulated Environments

We simulated a variety of environments with different SNR distributions and UE mobility patterns. Although the simulated environments are artificial, their SNR distributions mimic those of real eMBMS networks obtained through field trial measurements. To capture the SNR characteristics of an environment, we divide its geographical area into rectangles of $10m \times 10m$. For each reporting interval, each UE draws its individual SNR value, $h_v(t)$, from a Gaussian-like distribution which is a characteristic of the rectangle in which its located. The rectangles have different mean SNR, but the same standard deviation of roughly 5dB (as observed in real measurements). Thus, the SNR characteristics of each environment are determined by the mean SNR values of the rectangles at any reporting interval.

In some environments, typically where the SNR variations are small, the SNR Threshold, $s(t)$, barely changes over time. For such scenarios, the *Uniform* scheme, based on rigorous field trial measurements, is an appropriate solution. In such a situation, *DyMo* can efficiently infer the SNR Threshold and reduce the need for expansive field trails. The results for these simulations are in [12]. In this paper, we discuss two types of environments in which $s(t)$ changes significantly over time.

• **Stadiums:** In a stadium, the eMBMS service quality is typically significantly better inside the stadium than in the surrounding vicinity (e.g., the parking lots). To capture this, we simulate several stadium-like environments, in which the stadium, in the center of the venue, has high eMBMS SNR with mean values in the range of $15-25$dB. On the other hand, the vicinity has significantly lower SNR with means values of $5-10$dB. An example of a stadium is shown in Fig. 4(a).

We assume a mobility pattern in which, the UEs move from the edges to the inside of the stadium in 12mins, stay there for 3mins, and then go back to the edges.[9] As shown in Fig. 4(b), as the UEs move toward the center, the number of active UEs gradually increases from 10% of the UEs to 100%, and then declines again as they move away.

• **Failures:** In the case of a malfunctioning component, the QoS in some parts of a venue can degrade significantly. To simulate failures, we consider cases in which the eMBMS SNR is high with a mean between $15-25$dB. During the simulation, (around the 10^{th} minute), we mimic a failure by reducing the mean SNR values of some of the rectangles by over 10dB to the range of $5-10$dB. The mean SNR values are restored to their original values after a few minutes. Figs. 5(a) and 5(b) provide an example of the mean SNR values of such a venue before and after a failure, respectively.

In such instances, we assume random mobility pattern, in which each UE moves back and forth between two uniformly selected points. During the simulation, 50% of the UEs are always active, while the other 50% join and leave at some random time.

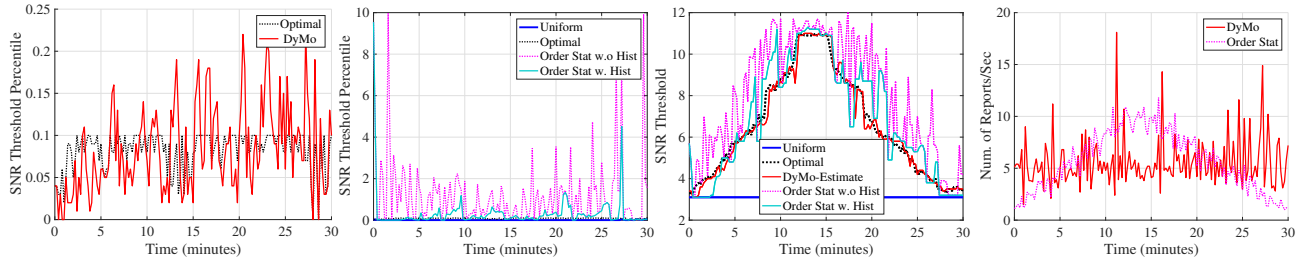
C. Performance over time

We first illustrate the performance of the different schemes over time for two given instances, a stadium and a failure scenario, with $m = 20,000$ UEs, QoS Constraint $p = 0.1\%$, and Overhead constraint $r = 5$ reports/sec, i.e., 60 messages per reporting interval. The *number of permitted outliers can be at most* 20 *at any given time*. These values correspond to typical situations in dense eMBMS environments. The key difference between the two instances is the rate at which the SNR Threshold changes. In the case of the stadium, the SNR Threshold gradually change as the UEs change their locations. In the failure scenario, the SNR Threshold is roughly fixed but it drops instantly by 10dBs for the duration of the failure.

The results of the stadium and failure case are shown in Figs. 6 and 7, respectively. Figs. 6(a), 6(b), 7(a), and 7(b) show the actual SNR Threshold percentile over time. From Figs. 6(a) and 7(a), we observe that *DyMo* can accurately infer the SNR Threshold with an estimation error of at most 0.1%. Fig. 7(a) shows slightly higher error of 0.25% at the time of the failure (at the 7^{th} minute). The *Order-Statistics* variants suffer from much higher estimation error to the order of a few percentage points, as shown by Figs. 6(b) and 6(b).[10] This performance gap results in different estimation accuracy of the SNR Threshold for *DyMo* and *Order-Statistics* schemes as illustrated in Figs. 6(c) and 7(c), respectively. These figures show that the performance of *DyMo* and *Optimal* is almost identical. Even in the event of a failure, *DyMo* reacts

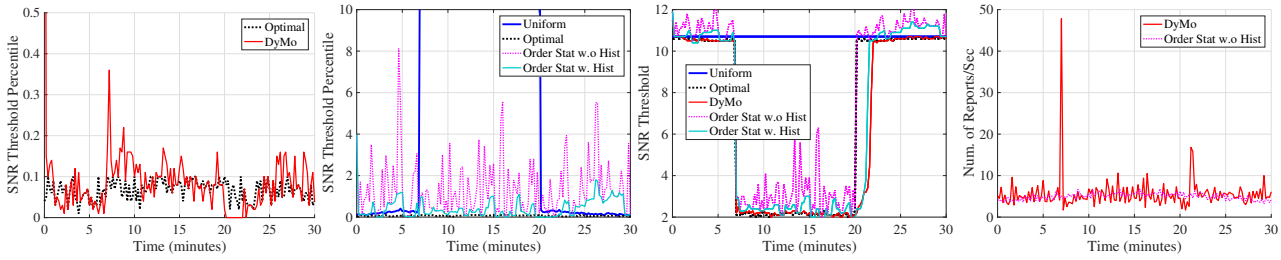
[9]While significant effort has been dedicated to modeling mobility (e.g., [23], [24] and references therein), we use a *simplistic mobility model* since our focus is on the multicast aspects rather than the specific mobility patterns.

[10]Notice that Figs. 6(a) and 6(b) as well as Figs. 7(a) and 7(b) use different scales for the Y axes.



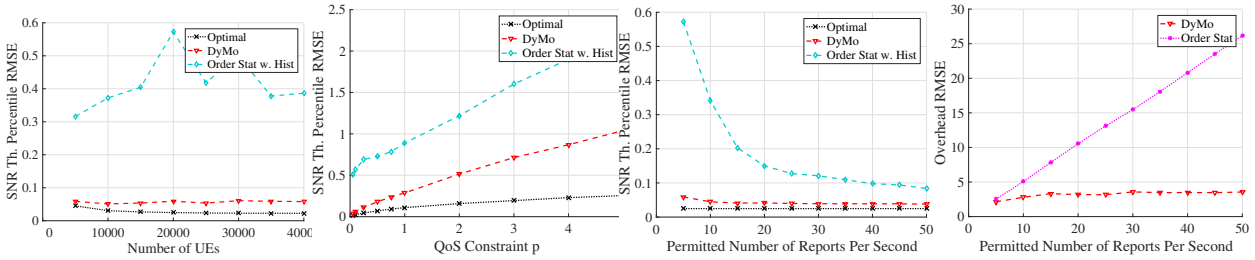
(a) (b) (c) (d)

Fig. 6. Simulation results from a single simulation instance lasting for 30mins in a stadium environment with $20,000$ UEs moving from the edges to the center and back, with $p = 0.1$ and $r = 5$ messages/sec. (a) The actual percentile of the SNR Threshold estimated by *DyMo*, (b) the actual percentile of the SNR Threshold estimated by *Order-Statistics*, (c) the SNR Threshold estimation, and (d) the QoS report overhead.



(a) (b) (c) (d)

Fig. 7. Simulation results from a single simulation instance lasting for 30mins in a component failure environment with $20,000$ UEs moving side to side between two random points, with $p = 0.1$ and $r = 5$ messages/sec. (a) The actual percentile of the SNR Threshold estimated by *DyMo*, (b) the actual percentile of the SNR Threshold estimated by *Order-Statistics*, (c) the SNR Threshold estimation, and (d) the QoS report overhead.



(a) (b) (c) (d)

Fig. 8. The Root Mean Square Error (RMSE) of different parameters averaged over 5 different simulation instances lasting for 30mins each in a stadium environment with different SNR characteristics and UE mobility patterns. (a) SNR Threshold percentile RMSE vs. the total number of UEs in the system, (b) SNR Threshold percentile RMSE vs. the QoS Constraint p, (c) SNR Threshold percentile RMSE vs. the number of permitted reports, and (c) Overhead RMSE vs. the amount of permitted reports.

immediately and detects the SNR Threshold accurately. The *Order-Statistics* variants react quickly to a failure but not as accurately as *DyMo*. After the recovery, both *DyMo* and *Order-Statistics w. History* gradually increase their SNR Threshold estimates, due to the exponential smoothing process.

The SNR Threshold estimation gap directly impacts the number of outliers as well as the network utilization, i.e., the spectral efficiency. Figs. 6(d) and 7(d) indicate only mild violation of the Overhead Constraint by both the *DyMo* and *Order-Statistics* variants. The detailed results for the spectral efficiency appear in [12]. We observe that accurate SNR Threshold estimation allows *DyMo* to achieve near optimal spectral efficiency with negligible violation of the QoS Constraint. The other schemes suffer from sub-optimal spectral efficiency, excessive number of outliers, or both. Given that the permitted number of outliers is at most 20, the *Order-Statistics w. History* and *Order-Statistics w.o. History* schemes exceed this value sometimes by a factor of 10 and 40, respectively. Among these two alternatives, *Order-Statistics w. History* leads to lower number of outliers. We observe that in this stadium example, the *Uniform* scheme yields a very conservative eMBMS MCS setting, which causes low network utilization. In the failure scenario, the conservative eMBMS MCS of *Uniform* is not sufficient to cope with the low SNR Threshold and it leads to excessive number of outliers.

D. Impact of Various Parameters

We now turn to evaluate the quality of the SNR Threshold estimation and the schemes ability to preserve the QoS and Overhead Constraints under various settings. We use the same configuration of $m = 20,000$ UEs, $p = 0.1\%$ and $r = 5$ reports/sec and we evaluate the impact of changing the values of one of the parameters. The results are shown in Fig. 8, where each point in the figure is the average of 5 different stadium simulation instances of 30mins each with different SNR characteristics and UE mobility patterns. The error bars are small and not shown. In these examples, we compare *DyMo* only with *Optimal* and *Order-Statistics w. History*

which is the best performing alternative. The results for failure scenarios are similar and can be found in [12].

Fig. 8(a) shows the Root Mean Square Error (RMSE) in SNR Threshold percentile estimation vs. m. The non-zero values of RMSE in *Optimal* are due to quantization of SNR reports. The RMSE in the SNR Threshold estimation of *DyMo* is close to that of *Optimal* regardless of the number of UEs. Fig. 8(b) shows the RMSE in SNR Threshold estimation as the QoS Constraint p changes. *DyMo* outperforms the alternative schemes as p increases. As p increases, we observe an increasing quantization error, which impacts the RMSE of all the schemes including the *Optimal*.

Fig. 8(c) illustrates the SNR Threshold percentile RMSE as the Overhead Constraint is relaxed. The SNR Threshold percentile RMSE of *DyMo* is 0.05% even with Overhead Constraint of 5 reports/sec while *Optimal* RMSE due to quantization is 0.025%. *DyMo* error slightly reduces by relaxing the Overhead Constraint (*Optimal* error stays 0.25%). Even with 10 times higher reporting rate, *DyMo* significantly outperforms the *Order-Statistics* alternatives. The RMSE in SNR Threshold percentile for *Order-Statistics* is in the order of the required average value of 0.1 even with a permitted overhead of 50 reports/sec, i.e,. 3000 reports per reporting interval. This is a very high overhead on the unicast traffic, since in LTE networks each connection lasts several hundred msecs even for sending a short update. Unlike the downlink, uplink resources are not reserved for eMBMS systems and utilize the unicast resources. The RMSE of number of outliers is qualitatively similar to the SNR Threshold percentile results.

We also compute the overhead RMSE for different UE population sizes, m, QoS Constraint p, and Overhead Constraints r. In each case, the overhead RMSE of *DyMo* is between $2-4$. We notice an interesting case when the permitted overhead is allowed to increase as shown in Fig. 8(d). While the *DyMo* RMSE is consistently small, the RMSE of *Order-Statistics* scales almost linearly with the permitted overhead. This is due to the static reporting rate of *Order-Statistics* despite changing number of active UEs.

VII. Conclusion

This paper presents a *Dynamic Monitoring* (*DyMo*) system for large scale monitoring of eMBMS services, based on the concept of *Stochastic Group Instructions*. Our extensive simulations show that *DyMo* achieves accurate, close to optimal, estimation of the SNR Threshold even when the number of active UEs is unknown. It can improve the spectral efficiency for eMBMS operation while adding a low reporting overhead.

VIII. Acknowledgment

This work was supported in part by NSF grants CNS-16-50669 and CNS-14-23105.

References

[1] J. Erman and K. K. Ramakrishnan, "Understanding the super-sized traffic of the super bowl." in *Proc. ACM IMC'13*, 2013.

[2] A. Kaya, D. Calin, and H. Viswanathan, "On the performance of stadium high density carrier Wi-Fi enabled LTE small cell deployments," in *Proc. IEEE WCNC'15*, 2015.

[3] "3GPP TS 26.346 V13.1.0, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs (Release 13)," June 2015. [Online]. Available: http://www.3gpp.org/DynaReport/26346.htm

[4] D. Lecompte and F. Gabin, "Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: overview and rel-11 enhancements," *IEEE Comm. Mag.*, vol. 50, no. 11, pp. 68–74, 2012.

[5] "3GPP TS 37.320 V12.2.0 , 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Universal Terrestrial Radio Access (UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRA); Radio measurement collection for Minimization of Drive Tests (MDT); Overall description; Stage 2 (Release 12)," Sept. 2014. [Online]. Available: http://www.3gpp.org/DynaReport/37320.htm

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

[7] 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.

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

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

[10] Y. Cai, S. Lu, L. Zhang, C. Wang, P. Skov, Z. He, and K. Niu, "Reduced feedback schemes for LTE MBMS," in *Proc. IEEE VTC'09*, 2009.

[11] A. B. Owen, *Monte Carlo theory, methods and examples*, 2013.

[12] Y. Bejerano, C. Raman, C.-N. Yu, V. Gupta, C. Gutterman, T. Young, H. Infante, Y. Abdelmalek, and G. Zussman, "DyMo: Dynamic monitoring of large scale LTE-multicast systems," in *arXiv:1701.02809 [cs.NI]*, 2017.

[13] J. Yoon, H. Zhang, S. Banerjee, and S. Rangarajan, "MuVi: a multicast video delivery scheme for 4G cellular networks," in *Proc. ACM MOBICOM'11*, 2012.

[14] R. Sivaraj, A. Pande, and P. Mohapatra, "Spectrum-aware radio resource management for scalable video multicast in LTE-advanced systems," in *Proc. IFIP Networking'13*, 2013.

[15] L. Militano, D. Niyato, M. Condoluci, G. Araniti, A. Iera, and G. M. Bisci, "Radio resource management for group-oriented services in LTE-A," *IEEE Trans. Veh. Technol.*, vol. 64, no. 8, pp. 3725–3739, 2015.

[16] J. Chen, M. Chiang, J. Erman, G. Li, K. Ramakrishnan, and R. K. Sinha, "Fair and optimal resource allocation for LTE multicast (eMBMS): group partitioning and dynamics," in *Proc. IEEE INFOCOM'15*, 2015.

[17] 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 *Proc. ACM MOBIHOC '16*, 2016.

[18] 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.

[19] 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 *Proc. IEEE ICNP'09*, 2009.

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

[21] V. Gupta, C. Gutterman, Y. Bejerano, and G. Zussman, "Dynamic rate adaptation for WiFi multicast to very large groups design and experimental evaluation," in *Proc. IEEE INFOCOM'16*, 2016.

[22] A. W. Van der Vaart, *Asymptotic statistics*. Cambridge university press, 2000.

[23] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy-walk nature of human mobility," *IEEE Trans. Netw.*, vol. 19, no. 3, pp. 630–643, 2011.

[24] S. Scellato, I. Leontiadis, C. Mascolo, P. Basu, and M. Zafer, "Evaluating temporal robustness of mobile networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 105–117, 2013.