Real-Time Monitoring of Multicast Group Information

Achmad BASUKI†, Achmad Husni THAMRIN†, Nonmembers, Hitoshi ASAEDA†, and Jun MURAI††, Members

SUMMARY This paper presents a method to monitor information of a large-sized multicast group that can follow the group’s dynamics in real-time while avoiding feedback implosion by using probabilistic polling. In particular, this paper improves the probabilistic-polling-based approach by deriving a reference mean value as the reference control value for the number of expected feedback from the properties of a binomial estimation model. As a result, our method adaptively changes its estimation parameters depending on the feedback from receivers in order to achieve a fast estimate time with high accuracy, while preventing the possible occurrence of feedback implosion. Our experimental implementation and evaluation on PlanetLab showed that the proposed method effectively controls the number of feedback and accurately estimates the size of a dynamic multicast group.

key words: membership estimation, adaptive sampling, probabilistic polling, binomial probability, multicast

1. Introduction

Methods for monitoring information about a multicast group, such as group size, loss rate distribution, RTT distribution, have been studied for many years. The main issue in this problem is how to get that information in a timely fashion, while not overwhelming the network with unnecessary messages. An early implementation for counting group size in a timely manner has been demonstrated by Dutta et al. in MarconiNet [1]. Their implementation relies on the real-time transport control protocol, RTP/RTCP [2]. However, the protocol takes a longer time to finish counting and does not scale well to large multicast group size.

For large-scale multicast applications where tracking the group size population in real-time is important as well as the accuracy, it is more appropriate to implement sampling-based techniques. The sampling-based technique reduces the number of feedback through a polling query sent over multicast, where the probability \( p \) of the query is set by the sender. The sender can then estimate the number of receivers using the binomial estimation model. In this technique, the feedback from receivers can be sent in unicast and the destination does not have to be the sender itself, which depend on the defined subsequence queries from the sender (appropriate for SSM [3] operation). This technique can follow the dynamic numbers of multicast receivers by polling at intervals of \( T \), where \( T \) is the observation time. Unfortunately, feedback implosion may still occur in this approach; and there could be problems in determining the optimal \( p \) and \( T \), as discussed in [4].

The information of multicast group size can be acquired by soliciting the feedback from all receivers through either a *sender-initiated* or a *receiver-initiated* mechanism [5]. However, both mechanisms lead to the scaling problem in feedback handling. Two general approaches have been proposed to avoid the feedback implosion. The first is the hierarchical approach, in which feedback suppression through *hierarchical* trees is employed. This can be implemented in network routers [6]–[10] or representative designated nodes [11]–[13]. The second approach is the *end-to-end* approach, whose implementation is more practical and more feasible, because this approach does not require network support and it employs feedback suppression by a random timer [14]–[16] or sampling [4], [17]–[19].

In this paper, we propose a method to sample a multicast group in a fast and scalable manner that adaptively follows the dynamics of multicast group size. We achieved this by proposing an *adaptive* polling query that changes the probability value \( p \), in order to maintain the number of expected feedback around a certain reference value, and the observation time \( T \), in order to dynamically adapt to the network response times. As a result, our proposed method could estimate a very large of multicast group size in a fast estimate time with a high accuracy as well as preventing the possible occurrence of the feedback implosion.

This work is motivated by our conviction that the service of broadcasting industry and the Internet with global IP multicast will converge. Multicast TV and radio have started in live broadcast service since 2007 in UK [20] as an example. In this broadcasting industry, knowing the size of audience of the broadcasted contents in timely manner is the most important figure as it is associated with the commercial ads. With IP Multicast the size of audience, represented by the multicast group size, will be possible to be measured in real-time with a much better accuracy.

Multicast group size estimate is also an essential component in reliable multicast, since it is needed for minimizing the number NAKs. It is also useful for real-time video multicast, where it can be used to detect congestion and adjust the sender’s rate based on the collected feedback from receivers regarding the loss rate in the group, as used in [17]. We believe that in the near future, the group size estimate will be an important function for carrying out adaptive video
multicasting, where the sender achieves quality adaptation or congestion control by inferring reception status from its heterogeneous receivers.

The paper is organized as follows. We describe the behavior of the general probabilistic polling method in Sect. 2. In Sect. 3, we present our approach for dealing with adaptive estimation parameters. In Sects. 4 and 5, we provide details of the simulation and analysis of our adaptive estimation method. In Sect. 6, we describe the experimental implementation and evaluation of our approach using PlanetLab. We discuss the related work in Sect. 7. Finally, in Sect. 8 concludes this paper.

2. Probabilistic-Polling-Based Method

In the probabilistic-polling-based method, a data sender (source) sends a probabilistic polling query \( p \) to a multicast group address for soliciting feedback from receivers [21]. Each receiver receives a query and then generates a random value from the uniform distribution \( U[0 \ldots 1] \). Only those receivers whose random value is less than or equal to \( p \) will send feedback to the source via unicast. This mechanism reduces the number of feedback received by the source. Figure 1 illustrates the probabilistic-polling-based method.

This method is best modeled using binomial estimation as defined in [18]. The polling query occurs over a series of polling rounds, \( t = 1 \ldots k \), where the response of each receiver, \( n \in \{1 \ldots N\} \), follows a random variable \( X_{t,n} \sim \text{Bernoulli}(p) \). At each polling query \( p_t \), a success event, i.e., a receiver sends a feedback, is defined as \( X_{t,n} = 1 \); otherwise \( X_{t,n} = 0 \). Then, the sum of these identically independent distributed (iid) Bernoulli trials represents the number of feedback \( Y \) received by the source, as

\[
Y_t = \sum_{t=1}^{kN} X_{t,n}.
\]

By this definition, \( Y \) is considered as a binomial random variable with parameters \((N, p)\): \( Y \sim \text{binomial}(N, p) \).

![Image](image1.png)

**Fig. 1** Probabilistic-polling-based method; the left side illustrates the number of feedback \( Y \) acquired in each polling round \( t \), the right side illustrates the response of receivers.

2.1 Estimator

In order to estimate the size of a large multicast group using the probabilistic-polling-based method, the source can use a naive estimator. By this estimator, \( Y_t \), the number of feedback in each polling round \( t \), would be divided by the query probability \( p \). The naive estimate of multicast group size \( \hat{N}_t \) is given by

\[
\hat{N}_t = \frac{Y_t}{p}.
\]

In the case of a very large group size, the ratio \( Y/N \) of the number of feedback received to the group size will converge to \( p \) (follows the strong law of large numbers). However, when the query probability \( p \) is small, the estimate \( \hat{N}_t \) is very noisy, as emphasized by [4], and this estimator does not take into account the previous estimate. A better estimator that incorporates the previous estimate for making future estimates, as used by [22], is the Exponential Weighted Moving Average (EWMA) estimator, which is defined as

\[
\hat{N}_t = \alpha \hat{N}_{t-1} + (1 - \alpha) \frac{Y_t}{p}.
\]

The performance of this estimator has been analyzed in [23] and it is considerably better than the naive estimator used in their experiment. We also confirmed similar findings in a simulation (described later in Sect. 4), as shown in Fig. 2. In this simulation, we used \( p = 0.1 \) for both the estimators and \( \alpha = 0.8 \) for the EWMA estimator.

Figure 2 shows the probability \( p \) is the determining factor behind the feedback implosion. A constant \( p \) will lead to the feedback implosion problem if the value is not set appropriately. Even if \( p \) is set to an appropriate value for a certain multicast group size, significant changes in the multicast group size will make the \( p \) value no longer appropriate.

![Image](image2.png)

**Fig. 2** Naive versus EWMA estimates using static \( p = 0.1 \) and \( \alpha = 0.8 \) for the EWMA estimator. The number of feedback messages is directly proportional to the number of receivers with a sampling factor \( p \). The inset picture shows that EWMA estimates are less noisy than Naïve estimates.
2.2 Feedback Response Time

In the probabilistic-polling-based method, the estimate can only be made at the end of the observation time. In most previous studies [4], [17], researchers used a fixed observation time (polling interval) that was equal to or twice the common round trip time (RTT). However, the RTT varies depending on the location where a sender and the receivers are located in the networks.

Nonnenmacher [16] based on Wei [24] stated that receivers’ RTT distribution as seen by a multicast source roughly follows a beta distribution. We express the RTT distribution as a beta distribution. We express the receivers’ RTT distribution as seen by a multicast source as a beta distribution. We express the RTT distribution roughly follows a beta distribution as seen by a multicast source.

$$Y \sim \text{binomial}(N, p),$$

mentioned in Sect. 2, has an expected value $E[Y] = Np$ and standard deviation $SD(Y) = \sqrt{Np(1-p)}$ [25, pp.144–145]. According to the well-known De Moivre-Laplace limit theorem, when $N$ is large ($N \rightarrow \infty$), a binomial random variable with parameters $N$ and $p$ will have approximately the same distribution as a normal random variable with the same mean and standard deviation as the binomial, $Y \sim \text{normal}(Np, \sqrt{Np(1-p)})$.

This approximation, in general, will give good approximation as long as $5 \leq Np \leq N - 5$ [26].

In an estimation with a single polling round using normal approximation, the number of feedback should not be close to 5 [27, p.18]. Therefore, we propose a method to always maintain the number of feedback to a value greater than 5 by changing the probability $p$ in the subsequent polling rounds. As expressed in Eq. (2), the ratio $Y/N$ will converge to $p$; therefore, adjusting the value of $p_t$ will trigger the incoming feedback $Y$. Let $rd$ be a target value; then, $p_{t+1}$ is defined as follows:

$$p_{t+1} = p_t \frac{rd}{Y_t}. \quad (4)$$

Hence, the number of feedback on the next polling round, $Y_{t+1}$, is expected to be around $rd$. However, $Y$ is a normal random variable, thus it will always change even when $N$ does not change. Let $Np = rd$ be the expected value $E[Y]$, hence the interval estimator for $Y$ at a level of confidence $1 - \alpha$ is thus:

$$\left[ rd-(0.5+z_{\alpha/2})\sqrt{rd(1-p)}, rd+(0.5+z_{\alpha/2})\sqrt{rd(1-p)} \right]. \quad (5)$$

The constant 0.5 is a continuity correction attributed to the normal approximation when approximating a discrete distribution, such as a binomial distribution [25, p.206]. This continuity correction gives a more accurate approximation, especially for small $N$. The $z_{\alpha/2}$ ($\Phi(z_{\alpha/2}) = 1 - \alpha/2$) is a two-sided confidence interval value from standard normal. To avoid frequent changes of $p$ that could deteriorate the estimation $\hat{N_t}$, we refine Eq. (4) to include boundary conditions as follows:

$$p_{t+1} = \begin{cases} 
    p_t, & Y_{\min} \leq Y_t \leq Y_{\max} \\
    p_t \frac{rd}{Y_t}, & \text{otherwise},
\end{cases} \quad (6)$$

where $Y_{\min}$ and $Y_{\max}$ denote the lower and upper bounds in Eq. (5). From this equation, $p$ will change only when $Y_t$ crosses the boundary; with large $z_{\alpha/2}$ prevents unnecessary changes in $p$. With these $p_t$ and $Y_t$, we can then calculate $\hat{N_t}$. By means of this approach, the number of expected feedback can be maintained around the anticipated reference value regardless of the dynamics of multicast group size.

3.2 Adaptive Observation Time

As discussed in Sect. 2.2, we assumed that the delay distribution roughly follows a beta distribution and the observation time in each polling round should be long enough to
include all, or at least almost all, feedback. First we use a normal approximation to the collected RTTs of feedback samples in each polling round; below we refine the approximation. Therefore, we define the observation time of the next polling round $t + 1$ as

$$T_{t+1} = E[RTT_t] + z_α SD(RTT_t), \quad (7)$$

where $z_α$ ($Φ(z_α) = 1 - α$) should be chosen sufficiently large such that all – or at least almost all – feedback is captured by the next observation time period, $T_{t+1}$. For example $z_α = 3.54$ (obtained from the one-sided confidence interval of a standard normal table) will cover approximately 99.98% of the normal approximation distribution.

However, the normal approximation in Eq. (7) may ignore the late feedback responders when the RTT distribution tends to $beta(a, b)$ with $a < b$ as in Fig. 3. To solve this problem, we propose a correction technique in which the $RTT_t$ is still being monitored during the on-going polling round $t + 1$. Therefore, we refine Eq. (7) to be as follows:

$$T_{t+1} = \begin{cases} \max(RTT_t), & \text{if } RTT_t \text{ exists at } t + 1 \\ E[RTT_t] + z_α SD(RTT_t), & \text{otherwise}. \end{cases} \quad (8)$$

We call this the adaptive observation time. The simulation and analysis of this adaptation technique in Sects. 4.2 and 5.2 show how this adaptive observation time contributes to the estimation accuracy.

### 3.3 Initialization

We take into consideration the initialization period in this adaptive solution. We start polling conservatively with a low query probability $p_0$, assuming $N_0 \approx 10^6$, with a reasonable observation time $T_0 = 1$ s. The query probability $p$ and the observation time $T$ will gradually change according to Eq. (6) and Eq. (8).

In summary, our proposed adaptive method determines the optimal values of $p$ and $T$ that reflect the dynamics of the multicast group size and network response times, respectively. Combined with the estimator defined in Eq. (3), this adaptive method provides an estimation of a large multicast group size, prevents the possible occurrence of feedback implosion, and achieves a fast estimate time.

### 4. Simulation

In this section, we investigate the robustness of the method and try to identify the situations in which it works well/poorly. We have performed two types of simulations to show the adaptiveness of our approach.

In the first simulation, we wrote an original simulation program in C that took into account the multicast join and leave model; the selected estimator in this case was EWMA Eq. (3). The size of multicast group at a particular time can be calculated on the basis of the join and leave events of receivers. The generic representation of the number of receivers $N(τ)$, or equivalently, the size of multicast group at time $τ$ can be defined as follows:

$$N(τ) = \sum_{i=1}^{∞} 1(J_i ≤ τ ≤ J_i + D_i), \quad (9)$$

where $J_i$, $D_i$ and $J_i + D_i$, respectively, denote the join time, the duration of participation and the leave time of the receiver $i$. According to the model in [28], we consider an exponential distribution for the join time and the duration of participation of receivers used in Eq. (9). With this model we could generate the dynamics of multicast group size in the simulation.

In the second simulation, we wrote a simulation program that includes heterogeneous RTT which follows a beta distribution, as discussed in Sect. 2.2.

#### 4.1 Adaptiveness of $p$

We performed the first simulation to show how our method cope with the dynamics of multicast group size with a defined target value, rd. Through this simulation program, we were able to propose our adaptive method for solving the possible feedback implosion problem in probabilistic-polling-based method.

The target value rd must first be determined for the adaptive query probability in Eq. (6). The value of rd should be sufficiently large (more than 5) to achieve good estimation accuracy (the larger the rd value, the higher will be the estimation accuracy). Moreover, the rd value should also reflect the consideration of the feedback rate that can be handled by the sender to avoid feedback implosion. Suppose that a sender can receive feedback at rates of up to 256 kbit/s, where the size of each feedback packet is 128 bytes. Hence, the number of feedback should be less than 250 packets per second.

In our simulation, we conservatively set $rd = 160$ and the observation time $T = 1$ s. If we assume that $N_0 \leq 10^6$ (as considered in Sect. 3.3) and $Np \approx rd$ at $τ = 0$, we can begin an initial polling round with a probability polling query, $p_0 = 0.00016$. We then calculate the boundary values using Eq. (4) with $z_{0.9999} = 3.72$; thus we obtain $Y_{min} = 112$ and $Y_{max} = 207$ (less than 250 packets per second) for the first polling round. We used these values to perform our first simulation.

Figure 4 shows the estimation results obtained using the EWMA estimator with our adaptive probabilistic polling query technique, where the feedback has a mean close to $rd = 160$. As also shown in this figure, the parameter $p$ in each polling round gets adjusted accordingly whenever the number of feedback goes outside $[Y_{min}, Y_{max}]$.

#### 4.2 Adaptiveness of $T$

We performed the second simulation with a constant 20,000 receivers for 10 polling rounds. The simulation uses a constant $p = 0.008$ and an initial polling interval $T_0 = 1$ s. $RTT$
distribution is generated from $\beta(2, 6)$ random values at intervals $1 - 1500$ ms and $z_r = 3.54$ is used for the adaptive observation time as described in Eq. (8). In Fig. 5, we evaluated the use of adaptive observation time in adapting to the dynamics of RTT and also compared the effect on the estimation accuracy with the use of fixed observation time interval $T = 1$ s.

As indicated in Fig. 5, the adaptive observation time adaptively follows the dynamics of receivers latency. The fixed observation time ($T = 1$ s) results in a higher estimation error compared to the adaptive observation time, because it ignores feedback arriving after the observation time.

5. Analysis

In this section, we analyze the results of the simulations described in Sect. 4.

5.1 Feedback Rate and Estimation Accuracy

The simulation result presented in Fig. 4 shows that our method could successfully maintain the feedback average at around the reference target value. The average feedback rate is 155.027 packets per second; this value is well below the limit set in Sect. 4.1, namely, 250 packets per second. Further, only 0.602% of the polling rounds triggered the changes in $p$, in which the number of feedback was out of the boundary values.

The sender can anticipate the incoming rate of feedback, therefore feedback implosion could be avoided. At the same time, as shown in Fig. 6, $rd$ also influences the estimation accuracy. If a sender is able to handle a high feedback rate, then a high value of $rd$ could be defined to achieve a more accurate estimation.

We also performed several simulation instances using the same simulation program as in Sect. 4.1 to analyze the estimation accuracy of our adaptive probabilistic polling query under various $rd$ values and confidence levels.

We analyzed the data of 20 simulation instances using the 99.98% confidence level with various $rd$ values. The estimation accuracy analysis in Table 1 reveals that the estimation accuracy remains almost unchanged on increasing the confidence levels from 68.28% to 99.98% with $rd = 160$ and that the changes in $p$ decreases as the confidence level increases. When the
Table 1  Estimation errors vs confidence level.

| Confidence Level with rd = 160 | 68.28% | 95% | 99% | 99.98% |
|-------------------------------|-------|-----|-----|--------|
| Mean(%)                      | 2.112 | 2.118 | 2.130 | 2.124  |
| SD(%)                        | 0.356 | 0.359 | 0.360 | 0.360  |
| # p changes                  | 6506.50 | 2080.10 | 742.75 | 84.20 |

Fig. 7  Estimation accuracy using the adaptive observation time and using the fixed observation time interval on different RTT distributions. Receivers latency are distributed within an interval of 1 – 1500 ms and for fixed observation time, \( T = 1 \) s is used.

Confidence level of the two-sided interval is 99.98%, the number of changes in \( p \) is the smallest and the mean and standard deviation of the percentage of relative errors are 2.124% and 0.360%, respectively.

5.2 Observation Time and Estimation Accuracy

The simulation results in Fig. 5 show that the adaptive observation time adaptively follows the dynamics of feedback response times. We then evaluated the use of the adaptive observation time and the use of fixed observation time interval on the accuracy of estimation. Using three types of receivers latency distribution, \( \text{beta}(2, 6) \), \( \text{beta}(6, 6) \), and \( \text{beta}(6, 2) \), according to Sect. 2.2, the adaptive observation time always gives lower estimation errors compared to the fixed observation time interval, as shown in Fig. 7. In conclusion, using a fixed observation time will have low estimation accuracy if the value \( T \) does not cover the majority of samples.

The initial polling interval time \( T_0 = 1 \) s for the adaptive observation time affected the estimation accuracy whenever most of the feedback latencies are more than one second and obviously many feedback messages were ignored. Since the EWMA estimator is used, where previous estimates influenced the on-going estimate, the slope of estimation errors in the beginning of polling rounds is visible in Fig. 7 and indicates a transient state of estimation. The estimation accuracy then improved in the following polling rounds, since the use of adaptive observation time avoids ignoring feedback as much as possible.

5.3 Performance Comparison

We used the EWMA estimator along with our adaptive polling query method in all previous simulations. We now compare the estimation results between two static probabilistic-polling-based methods and our adaptive method. To obtain a representative comparison, we performed 20 simulations using adaptive probabilistic-polling-based methods with Alouf’s estimator [4, Eq. (24)] and EWMA estimator. However, in order to use Alouf’s estimator, we need to know the average group size under the steady-state condition \( \rho \) and the mean service rate \( \mu \) in advance, which is very difficult under real-case conditions. In our analysis, we extracted \( \rho \) and \( \mu \) from the real group size generated in our adaptive method simulations and then performed simulations using Alouf’s estimator. Finally, the performance of these three simulations are presented as an average of estimation relative errors in Fig. 8.

Estimation error spikes were observed in the transient state periods in the case of all estimators; moreover in the transient state periods, Alouf’s estimator performed worse in both conditions. From the relative errors table shown in the inset of Fig. 8, we infer that the estimation errors of the adaptive probabilistic-polling-based method are comparable to those of the static polling-based methods under all simulation conditions, especially in the transient state periods.

Alouf’s estimator is superior to the other estimators during steady-state period only if the parameters \( \rho \) and \( \mu \) can be determined in advance, as confirmed in Fig. 8. These two important parameters, as discussed in [23], may be estimated by the source through several probes by letting the newly arrived receivers send a “hello” message and, in a similar way, by letting receivers probabilistically send a “goodbye” message when leaving the session. However, the
best method for finding these parameters without sacrificing the convergence estimate times has still been a research question.

5.4 Parameter Setup

In this subsection, we summarize the parameter values used in our simulation. As discussed in Sect. 4.1, the boundary values are derived from Eq. (5), and using a 99.98% confidence level of the two-sided interval from the standard normal distribution table, we obtain $z_{0.9999} = 3.72$. From the simulation results described in Sect. 4.1, it was observed that only 0.602% of the polling rounds triggered the changes in $p$, in which the number of feedback was out of the boundary values; therefore, we believe that the value of $z_{0.9999}$ is suitable for our method, as $p$ does not change unless there is a significant change in $N$. Meanwhile, $p_{0}$ and $rd$ can be changed according to the feedback implosion avoidance requirement and the assumption of $N_{0}$. In this research, we used $p_{0} = 0.00016$ and $rd = 160$; thus Eq. (6) becomes

$$P_{t+1} = \begin{cases} p_{t}, & Y_{\min} \leq Y_{t} \leq Y_{\max}, \\ \frac{160}{p_{t}Y_{t}}, & \text{otherwise,} \end{cases}$$

where $Y_{\min} = 159.5 - 3.72 \sqrt{160(1-p_{t})}$ and $Y_{\max} = 160.5 + 3.72 \sqrt{160(1-p_{t})}$. To calculate the observation time $T_{t}$, we use Eq. (8) and we set $T_{0}$ to 1 s, as described in Sect. 4.2. The change in $T$ is given by

$$T_{t+1} = \begin{cases} \text{Max}(RTT), & \text{if } RTT \text{ exists at } t + 1 \\ E[RTT] + 3.54 \times SD(RTT), & \text{otherwise,} \end{cases}$$

where the constant 3.54 is obtained from the one-sided confidence interval of the standard normal table with 99.98% confidence level. These values in the adaptive probabilistic polling query Eq. (10) and the adaptive observation time Eq. (11) are defined as the reference parameters setup for our experimental implementation.

6. Experimental Implementation

We implemented and evaluated our proposed adaptive estimation method on the PlanetLab testbed [29] with following objectives:

- to evaluate our adaptive estimation method, and
- to analyze the implementation results.

6.1 Implementation

We installed a receiver daemon on 100 stable PlanetLab nodes. In order to generate a large group size (large number of receivers), each PlanetLab node run a daemon that randomly emulates up to 1000 virtual receivers. The generation of the emulated group size depends on each virtual receiver’s join and leave times. Similar to what we did in our simulation, we incorporated an exponential join and leave time distribution. Approximately 100,000 virtual multicast receivers were emulated in our PlanetLab testbed.

We ran our receiver program on each PlanetLab node. This program listened to a UDP port and responded to the sender’s polling queries. For evaluation purposes, each of the 100 PlanetLab nodes sent information to the sender about the total number of emulated receivers and the number of sampled receivers in each polling round. A random delay ranging from 0 to 600 ms was introduced to each virtual receiver before replying to a query from the sender. This delay is to emulate the heterogeneous link delay for each virtual receiver. Hence, we can expect to gain some insight into the dynamics of the response times from this implementation.

At a sender node, the reference adaptive parameters (the adaptive probabilistic polling query and the observation time) described in Sect. 5.4 were used and implemented with the EWMA estimator. A host running Linux at our lab acted as a sender and was connected to the PlanetLab network. Multicast communication from this sender was emulated by sending each query packet to all nodes simultaneously. The process of packets reaches all receivers was considered as an emulation of flooding process in the multicast routing protocol. We used the same packet size, i.e., 64 bytes, for the polling query and the polling reply. Since we used IP and UDP, the feedback packet size in the network was 92 bytes.

6.2 Evaluation

We successfully generated nearly 70,000 simultaneous receivers at the peak level for the validation of our adaptive method. Furthermore, the implementation results are in complete agreement with the simulation results.

Figures 9 and 10 show that our method can successfully control the number of expected feedback to around the defined reference value. Figure 9 shows that 98.9% of the feedback are within the boundary values. This behavior indicates that the query probability changed adaptively according to the number of feedback received by the sender. The observation times also adaptively changed according to the feedback response times, as shown in Fig. 10. This adap-

![Fig. 9](image-url)
tiveness of the observation time indicates that the estimate convergence times were achieved in the granularity of the network response times.

We repeated the experiment 10 times and analyzed the collected data. The estimation errors were below 10%, as shown in Fig. 11. The estimation error follows the relative rate of change in the real group size. Moreover, upon close inspection, we suspect that the parameter \( \alpha = 0.8 \) in the case of EWMA estimator is not optimal for the group size distribution obtained in our experiment, especially when the group size was decreased significantly in a short time periods.

Figure 12 shows a snapshot of the incoming feedback traffic as measured at the sender’s network interface. This snapshot of feedback traffic was captured during one of the experiment runs, in which it confirms the similar pattern to what we observed in the number of received feedback in Fig. 10. Thus, from these implementation results, we can infer that the adaptive parameters setup prevents feedback implosion very well.

7. Related Work

We have analyzed the problem in probabilistic-polling-based on estimating very large group size in multicast-capable network environment by considering the dynamics of multicast group size and feedback response times. The probability polling query \( p \) is an important scaling parameter that can either cause or prevent feedback implosion, while the polling interval \( T \) is an important parameter to appropriately follow the dynamics of feedback response times.

Bolot et al. [17] estimated the group size of a multicast group using the mechanism of soliciting feedback through a series of polling rounds until the sender receives the responses. Their mechanism uses an incremented probability \( p \) from \( 2^{-16} \), polling is stopped after a reply is received or when the value of \( p \) has reached 1.0. The observation timer is set to two times the largest RTT. If the timer expires, the sender initiates the next polling round with a new \( p \). It prevents feedback implosion when the number of receivers is less than or equal to \( 2^{16} \). Yet, this approach has several disadvantages, including low estimation quality, slow estimation (several polling queries), and possible occurrence of feedback implosion. Moreover, the approach does not consider the dynamic nature of the multicast group size.

Nonnenmacher and Biersack [16] proposed an intelligent feedback suppression using a timer-based technique that is applied to all receivers. This approach was proven to be scalable up to \( 10^6 \) in their simulation. They estimated the number of receivers based on the received feedback messages only from a single polling round; however, with an increase in the number of receivers, the convergence estimate times become slow. It has a low estimator quality since it is biased whenever the group size increases. Liu and Nonnenmacher [19] extended this work [16] by adding the closed form of the estimator, and they derived the estimator bounds using Poisson approximation in order to obtain a more accurate estimator for a multicast group size.

Friedman and Towsley [18] analyzed the research results of [16], [17] and proposed a refined algorithm to im-

![Fig. 10](image1.png)

Adaptive query probability and observation times.

![Fig. 11](image2.png)

Estimation accuracy.

![Fig. 12](image3.png)

A snapshot of the incoming feedback traffic as measured at the sender’s network interface.
prove the estimation quality. They mapped their analysis of the polling-based method into binomial \((N, p)\) estimation model to add a more accurate estimator by carrying out several rounds of polling before estimating the group size until a minimum number of feedback was received. While their estimator quality is good, the convergence estimate time is slow. Moreover, they did not consider the dynamic nature of the multicast group size.

Alouf et al. [4] proposed a refinement of the former research by defining an unbiased estimator and by applying Kalman filter theory to obtain more accurate estimation results. Moreover, they used a polling round at every observation step \(nT\) (larger than the largest RTT) with static probability \(p\), in which an estimate of the dynamics of multicast group size was performed at the end of each polling round. However, they have an open issue on determining the optimal values selection for the probability \(p\) and the observation time \(T\). These values were heavily influenced of a good estimation.

8. Conclusion and Future Work

In this paper, we presented our contribution to the probabilistic-polling-based approach by providing an adaptive mechanism to follow the dynamics of multicast group size and feedback response times. Our method improves the static probabilistic polling methods by relying on the properties of the binomial estimation model to derive an adaptive polling query. In addition, we have included an observation time that takes into account the network response times in each polling round. We have evaluated our proposed adaptive estimation method through simulations and by an actual implementation over a wide-scale network using the PlanetLab testbed. We conclude that our adaptive estimation method achieves a good balance between estimation convergence time, estimation accuracy and scalability.

Acknowledgments

This work was supported by National Institute of Information and Communications Technology (NICT), Japan. The authors thank Rodney Van Meter and Akimichi Ogawa for supporting this work into completion.

References

[1] A. Dutta, H. Schulzrinne, and Y. Yemini, “MarconiNet – An architecture for Internet radio and TV networks,” Proc. NOSSDAV’99, Basking Ride, NJ, USA, June 1999.
[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A transport protocol for real-time applications,” RFC 3550 (Standard), July 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt
[3] H. Holbrook and B. Cain, “Source-specific multicast for IP,” RFC 4607 (Proposed Standard), Aug. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4607.txt
[4] S. Alouf, E. Altman, and P. Nain, “Optimal on-line estimation of the size of a dynamic multicast group,” Proc. IEEE INFOCOM’02, pp.1109–1118, 2002.
[5] D. Towsley, J. Kurose, and S. Pingali, “A comparison of sender-initiated and receiver-initiated reliable multicast protocols,” IEEE J. Sel. Areas Commun., vol.15, no.3, pp.398–406, April 1997.
[6] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sunanasekera, and L. Visciano, “PJM reliable transport protocol specification,” RFC 3208 (Experimental), Dec. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3208.txt
[7] C. Papadopoulos, G. Parulkar, and G. Varghese, “Light-weight multicast services (LMS): A router-assisted scheme for reliable multicast,” IEEE/ACM Trans. Netw., vol.12, no.3, pp.456–468, 2004.
[8] R. Yavatkar, J. Grifffen, and M. Sudan, “A reliable dissemination protocol for interactive collaborative applications,” Proc. ACM MULTIMEDIA’95, pp.333–344, San Francisco, California, USA, 1995.
[9] F. Filali, H. Asaeda, and W. Dabbous, “Counting the number of members in multicast communication,” Proc. ACM NGC’02, pp.63–70, Boston, MA, USA, Oct. 2002.
[10] D. Dolev, O. Mokryn, and Y. Shavitt, “On multicast trees: Structure and size estimation,” IEEE/ACM Trans. Netw., vol.14, no.3, pp.557–567, 2006.
[11] H.W. Holbrook, S.K. Singhal, and D.R. Cheriton, “Log-based receiver-reliable multicast for distributed interactive simulation,” Proc. ACM SIGCOMM’95, pp.328–341, ACM, Cambridge, Massachusetts, USA, 1995.
[12] J.C.-H. Lin and S. Paul, “RMTTP: A reliable multicast transport protocol,” Proc. IEEE INFOCOM’96, pp.1414–1424, 1996.
[13] D. DeLucia and K. Obrazczka, “Multicast feedback suppression using representatives,” Proc. IEEE INFOCOM’97, vol.2, pp.463–470, Kobe, Japan, 1997.
[14] S. Floyd, V. Jacobson, S. McCanne, C.-G. Liu, and L. Zhang, “A reliable multicast framework for light-weight sessions and application level framing,” Proc. ACM SIGCOMM’95, pp.342–356, Cambridge, Massachusetts, USA, 1995.
[15] M. Grossglauser, “Optimal deterministic timeouts for reliable scalable multicast,” Proc. IEEE INFOCOM’96, vol.3, pp.1425–1432, March 1996.
[16] J. Nonnenmacher and E.W. Biersack, “Scalable feedback for large groups,” IEEE/ACM Trans. Netw., vol.7, no.3, pp.375–386, 1999.
[17] J.-C. Bolot, T. Turletti, and I. Wakeman, “Scalable feedback control for multicast video distribution in the Internet,” Proc. ACM SIGCOMM’94, pp.58–67, London, UK, Sept. 1994.
[18] T. Friedman and D.F. Towsley, “Multicast session membership size estimation,” Proc. IEEE INFOCOM’99, pp.965–972, 1999.
[19] C. Liu and J. Nonnenmacher, “Broadcast audience estimation,” Proc. IEEE INFOCOM’00, pp.952–960, 2000.
[20] BBC, “BBC TV and radio multicast service,” Dec. 2009. [Online]. Available: http://www.bbc.co.uk.multicast/
[21] M.H. Ammar, “Probabilistic multicast: Generalizing the multicast paradigm to improve scalability,” Proc. IEEE INFOCOM’94, pp.848–854, 1994.
[22] Jorg Nonnenmacher, Reliable multicast transport to large groups, Ph.D. Thesis, EPF, Lausanne, Switzerland, July 1998.
[23] S. Alouf, E. Altman, C. Barakat, and P. Nain, “Optimal estimation of multicast membership,” IEEE Trans. Signal Process., vol.51, no.8, pp.2165–2176, 2003.
[24] L. Wei and D. Estrin, “The trade-offs of multicast trees and algorithm,” Proc. ICCCN’94, San Francisco, CA, USA, Sept. 1994.
[25] S. Ross, A First Course in Probability, 6th ed., International Edition, Prentice-Hall, New Jersey, 2002.
[26] S.F. Arnold, Mathematical Statistics, Prentice-Hall, Englewood Cliffs, New Jersey, 1990.
[27] T. Friedman, Scalable estimation of multicast session characteristics, Ph.D. Thesis, University of Massachusetts Amherst, May 2002.
[28] K.C. Almeroth and M.H. Ammar, “Collecting and modeling the join/leave behavior of multicast group members in the MBone,”
[29] PlanetLab. “An open platform for developing, deploying, and accessing planetary-scale services,” Dec. 2007. [Online]. Available: http://www.planet-lab.org/

Achmad Basuki received B.E. from Brawijaya Univ. Indonesia in 2000 and has been joined as a full-time lecturer in computer science at Faculty of Mathematics and Natural Science from 2003 to 2006. He received master’s degree in 2008 from Graduate School of Media and Governance Keio Univ. Japan and is now a Ph.D. candidate at the same university. His current research interests include multicast and its deployment issues.

Achmad Husni Thamrin is Assistant Professor at Keio University. He is a graduate of Keio University, Graduate School of Media and Governance (Ph.D. 2005, MMG, 2002). His research interests include multicast, internet over broadcast media, and peer-to-peer networks.

Hitoshi Asaeda is Associate Professor of Graduate School of Media and Governance, Keio Univ. He received Ph.D. in Media and Governance from Keio Univ. in 2006. From 1991 to 2001, he was with IBM Japan, Ltd. From 2001 to 2004, he was a research engineer specialist at INRIA Sophia Antipolis, France. His research interests are IP multicast routing architecture and its deployment, dynamic networks and streaming applications. He is a member of ACM, IEEE, IPSJ, and WIDE Project.

Jun Murai is Professor of Faculty of Environment and Information Studies, Keio Univ. He received M.E. and Ph.D. in Computer Science from Keio Univ. in 1981 and 1987 respectively. He was a director of Keio Research Institute at SFC, the president of Japan Network Information Center (JP-NIC), board director of ICANN, Adjunct Professor at Institute of Advance Studies, United Nation Univ. He also teaches computer network and computer communication.