Keio University Master's Thesis Academic Year 2007

Estimation Method of Multicast Group Size for Large Networks

Keio University Graduate School of Media and Governance

Achmad Basuki

修士論文要旨 2007 年度 (平成 19 年度)

Estimation Method of Multicast Group Size for Large Networks

論文要旨

遠隔教育などを支援するために必要となる IP マルチキャストの展開には，広域で利用可能な IP マルチキャストアプリケーションとそれを提供する広帯域マルチキャストネットワークの普及が欠かせない．この普及を促進するため，コンテンツ受信者数を適宜把握する手法が必要であるが，全ての受信者情報を収集する仕組みは「フィードバックインプロージョン」を引き起こし，多量の返信データによるネットワーク負荷を増大させてしまう．このフィードバックインプロージョンを回避するために，返信タイミングをランダムあるいは定期的にずらして返信を分散させるという手法が提案されているが，この方法では情報の集約に時間を要してしまう．

本論文では,「ポーリングによる受信者数推測手法」の提案を行う．この手法では，データ送信者がポーリング要求を行う際,「推測パラメーター」を受信者全員に送信し，その推測パラメーターに基づいてランダムに選ばれた受信者が返信を行う．推測パラメーターは受信者数やネットワーク状態によって適切に変化するため，フィードバックインプロージョンを抑制させるだけでなく，受信者数の推測性能も維持することができる．

本論文では，様々なネットワーク環境を想定したシミュレーターを用いて最適な推測パラメーターを得た．推測パラメーターとポーリングを行う間隔は，推測した受信者数と RTT の平均値と標準偏差によって得られる．

シミュレーションで得られた推測パラメーターとこの提案手法を検証するため，実通信環境下で稼働する実装を開発し，PlanetLab を用いて検証を行った．その結果，提案手法はフィードバックインプロージョンを効果的に回避でき，受信者数もほぼ正確に推測できた．提案手法における推測パラメーターを変化させることで，推測誤りを減少させた．また，受信者情報の集約時間も短縮できた．

キーワード:

1. 受信者数計測, 2. 推測手法, 3. IP マルチキャスト
4. フィードバックサンプリング, 5. フィードバックインプロージョン

慶應義塾大学大学院 政策・メディア研究科

アフマド　バスキ

Abstract of Master's Thesis Academic Year 2007

Estimation Method of Multicast Group Size for Large Networks

Summary

Emerging large-scale IP Multicast applications and the availability of broadband multicast enabled networks are two important factors driving the growth of IP Multicast beyond the current base of users on education networks. To accommodate that growth, solutions are required to monitor the number of participating receivers for some specific purposes. However in assuring the service model, we need to solicit information from receivers but could easily trigger a feedback implosion. The available solutions are mostly focused on delaying the feedbacks with some random or static timer, which has the undesired effect of creating significant counting convergence delay.

We propose a method to estimate the number of receivers using a polling-based approach, where the sender sends a polling request consisting of an estimation parameter and randomly selected receivers respond upon receiving it. While previous research focused on finding a good quality estimator, our research focuses on applying a method to adaptively change the estimation parameters in order to avoid possible feedback implosion and achieve estimation accuracy.

We obtained the optimum parameters setup for probability factor and observation time of estimation from network simulation results with various scenarios. Both the probability factor and the observation time are changed accordingly based on the estimated number of receivers and the average RTT and its standard deviation of responses.

We developed the actual implementation and evaluated in the PlanetLab testbed to recognize the adaptiveness of our method in the real communication environment. The results show that our method is sufficient to estimate the number of receivers and efficiently avoid feedback implosion. The adaptive nature of the method helps to improve estimation error. It also gives the fast estimate convergence time.

Keywords:
1. Membership Counting, 2. Estimation Method, 3. IP Multicast,
4. Feedback Sampling, 5. Feedback Implosion

Keio University Graduate School of Media and Governance
Achmad Basuki

Contents

Abstract **i**

List of Figures **iv**

List of Tables **v**

1 Introduction **1**
1.1 Background 1
1.2 Motivation 3
1.3 Problem Statements 4
1.4 Research Goals 4
1.5 Thesis Outline 5

2 Issues on Membership Monitoring **6**
2.1 Feedback Implosion 6
2.1.1 Hierarchical Approach 7
2.1.2 End-to-end Approach 9
2.2 Estimation Convergence Time 12
2.2.1 Hierarchical Support 12
2.2.2 End-to-End Support 12
2.3 Summary 14

3 Related Work **16**
3.1 Real-Time Control Protocol - RTCP 16
3.2 Probabilistic Polling-based Approach 17
3.3 Summary 19

4 Requirements **20**
4.1 Scalability 20
4.2 Accuracy 20
4.3 Estimation Time 21
4.4 Summary 22

5 Approach **23**
5.1 Generic Polling-based Method 23
5.2 Adaptive Parameters 24
5.2.1 Probability factor p 24
5.2.2 Observation period T 25
5.3 Initialization Method 25
5.4 Estimation Algorithm 26
5.5 Summary 27

6 Simulation **28**
6.1 Overview 28
6.2 Method of Simulation 28
6.3 Simulation Scenario 30
6.4 Simulation Assumptions 31
6.5 Results and Analysis 31
6.5.1 Estimation Errors 31
6.5.2 Feedbacks Throughput 33
6.5.3 Estimation Convergence Times 34
6.6 Optimum estimation parameters 36

7 Evaluation **38**
7.1 Overview 38
7.2 Implementation 38
7.2.1 Adaptive Parameters Changes 39
7.2.2 Initialization Method 40
7.3 Evaluation Results 41
7.4 Summary 45

8 Conclusion and Future Work **46**
8.1 Conclusion 46
8.2 Future Work 47

Acknowledgments **48**

Bibliography **49**

List of Figures

1.1 Estimated multicast group size in MBone based on RTCP feedbacks . . . 2
1.2 Distance Learning over global multicast network . . . 3

2.1 Feedback implosion . . . 6
2.2 structure-based approach . . . 8
2.3 Representative-based approach . . . 9
2.4 End-to-end approach . . . 10
2.5 Timer-based approach . . . 10
2.6 Sampling-based approach . . . 11
2.7 RTCP with Unicast Feedback . . . 13
2.8 Multicast size estimation using polling on high degree router . 13
2.9 Counting convergence time of RTP/RTCP . . . 14

4.1 Near future possible IPv6 Multicast reachability of our distance learning stream . . . 21

5.1 Polling-based method . . . 23
5.2 Estimation Algorithm in sender . . . 26
5.3 Algorithm in Receiver . . . 26

6.1 Simulation building block . . . 29
6.2 Model of Simulation . . . 30
6.3 Estimation errors on p between 10% to 50% . . . 32
6.4 Estimation errors on p between 60% to 100% . . . 33
6.5 Feedback Throughput . . . 34
6.6 Illustration of Observation Time, T . . . 35
6.7 Estimation Convergence Time . . . 36

7.1 Distributed nodes in PlanetLab testbed . . . 39
7.2 The method of adaptive parameter changes applied at sender side . . . 40

7.3 Comparison of Real and Estimate of dynamic multicast membership 42
7.4 Dynamic changes of probability p, and the effect on number of feedbacks 42
7.5 Dynamic changes of observation time, T 43
7.6 Estimation errors on every polling round 44
7.7 Feedback throughput rate received at sender side 44

List of Tables

2.1 Taxonomy of Feedback Implosion Control and Membership Monitoring 15

3.1 Estimation Trade-off in Polling-based approach 18
3.2 Comparison of Related Work 19

6.1 Estimation Errors in percentage(%) 32
6.2 Feedbacks Throughput in towards sender (in kbit/s) 33
6.3 Estimates convergence time (ms) 35
6.4 Maximum RTT in each simulation scenario (ms) 35
6.5 Optimum p with estimated N receivers 36

Chapter 1

Introduction

This chapter introduces the background of the research and comprises of the general important issues explained in problem statements section. We define our motivation, followed by research goals and the outline of this thesis.

1.1 Background

Internet has been performed an evolution on the way people enjoying several entertainments, news or any other group communication. Broadcasting media like TV, Radio or any other live events intended to get the advantages from Internet to get out from constraints such as; limited coverage, limited frequency spectrum, and no receiver feedback from conventional broadcasting media receiver. These conditions trigger the acceleration of media convergences towards IP infrastructure.

Moreover, with the acceleration of global IP Multicast infrastructure deployment, the above constraints will no longer be problem for such media above. IP Multicast [1] provides efficient distribution when one media sender delivers data to a large number of destinations simultaneously. By means of multicast routing functionality, the network will copy the streams as necessaries according to receiver's interest. Although previously the multicast was more on "Many-to-Many" communication model, called Any-Source Multicast (ASM), to support wide coverage real-time online multi-party video conference, but nowadays "One-to-Many" model, called Source-Specific Multicast (SSM), becomes the preference and standardize in RFC 4607 [2]. The enabler services that could benefited from SSM service model are could be in such area as real-time distance learning system, TV and Radio online broadcast, real-time stock quotes dissemination, live events broadcast and etc.

Despite the above appealing feature, the lesson and learned we got from [3], indicates the growth of receivers (group size) in the Internet Multicast Backbone (MBone) was seen so slow. Figure 1.1 shows up to date estimated number of IP Multicast receivers derived from [4]. Lack of compelling content and motivation to deploy the further multicast infrastructure becomes one of barriers.

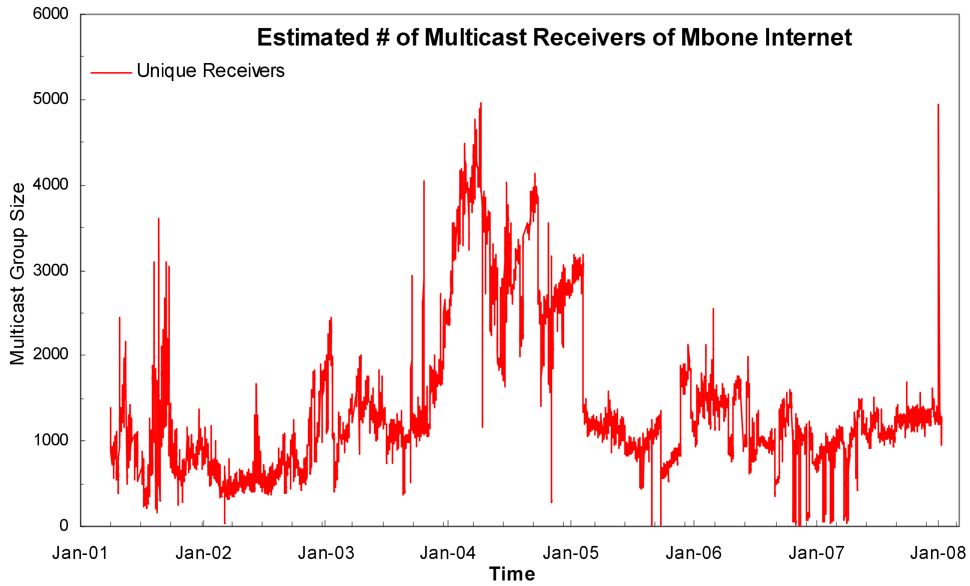

Figure 1.1: Estimated multicast group size in MBone based on RTCP feedbacks

On the other hand, Uni-Directional Link (UDL) technology has demonstrated its functionality of widespread data delivery with a low cost operation over broadcast infrastructure by a combination of satellite and terrestrial network in order to address the limitation on frequency spectrum. With their advances combine with IPv6 Multicast, make them an ideal platform for distance learning system such as done in SOI-Asia Project (School on the Internet for Asia Project) [5, 6] that could reach wider geographical coverage compare to the terrestrial.

Recent advances, a wide coverage broadband network infrastructure of Research and Education Network such as INTERNET2, ORION, JANET, GEANT2, TEIN2, APAN, RENATER, CERNET, KOREN, WIDE, and many others have enabled the IPv4 and IPv6 Multicast features in their network which open a great number of potential receivers to take advantages from such above application. One good example of them has been demonstrated by SOI Asia Project to deliver their live class or event to a number of receivers with underlying IPv6 Multicast platform as illustrated in Figure 1.2. Other example is Open Student Television Network (OSTN) [7], the emerging student-produced entertainment, educational and news contents provider whose eagerly trying to reach millions of student receivers by IP Multicast across those Research and Education Network as the claimed in their web site.

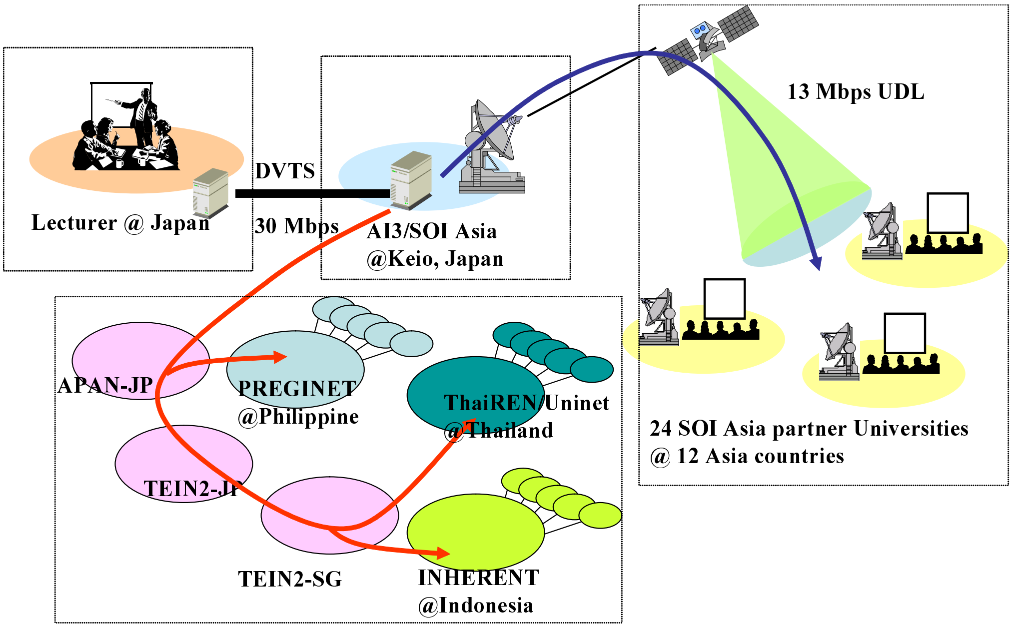

Figure 1.2: Distance Learning over global multicast network

Distance learning system is one of perfect case in utilizing the IPv4 or IPv6 Multicast network to reach very large number of receivers, however current available solution does not sufficiently support of knowing the number of receivers which is important for application level specific that support administrative purposes, receiver's participation characteristic and future capacity planning. Meanwhile, in reliable multicast application, knowing the number of receivers play importance roles to do feedback suppression, such as used in Scalable Reliable Multicast (SRM) [8] protocol. The available solution, such as Real-time Transport Protocol with its control protocol (RTP/RTCP) [9] consumes longer time to conclude a counting for very large number of receivers as experienced by MarconiNet in [10]. It is because the mechanism of the protocol that uses delay timers to regulate on how often receivers send control or session messages. This is undesirable condition for the importance of monitoring multicast group size.

1.2 Motivation

IP Multicast applications in general have difficulties to know the number of participating receivers (group size). To be able to monitor the group size we need to solicit feedback from all receivers in periodic manner, but it is not feasible and scale for very large number of receivers. Scalable operation has become the issues when we need the feedback information from receivers and has been addressed by many researchers [11, 8, 12, 13, 14, 15] from more than a decade ago since its emerging.

The above drawbacks effect in such a large-scale application such a distance learning, whose currently emerges running on top of IP Multicast network such as demonstrated in [5]. In natural sense, the lecturer or seminar presenter or live event organizer need to know the existence of remote sites who really participate in the session over the time. This information becomes so important for administrative, evaluation and capacity planning purposes when it is possible to be provided. By having such kind of information, we will be able to characterize participant's favorites participation over the time and positively beneficial to determine and enhance planning of future live lectures, seminars, and events.

Meanwhile, IP Multicast has seen slow deployment in the Internet, especially in commercial services, since its service model and architecture do not efficiently address many features required for robust implementation of multicast as discussed in [16]. By having the knowledge of the membership information, it may contribute an alternative charging model that based on membership size instead of input-rate basis.

All of the above issues are our strong motivation that lead us to find the solution to the above problem.

1.3 Problem Statements

In order to fulfill the motivation objectives we need to understand common problem in this research topic which is the *feedback implosion* and more specific issue on *estimation convergence time* for membership monitoring purposes. The details issues are discussed in Chapter 2. Hence we define the problem statements as follows:

- How can we estimate the group size of a multicast session?
- How can we avoid feedback implosion problem during the estimation?
- How can we have estimator that has faster convergence estimate than existing approaches?

1.4 Research Goals

This research is motivated by the conviction that large-scale multicast applications are emerging and will emerge much more in near future as triggered by more and more globally deployed multicast network such as in the Research and Education Network. Hence, the motivation leads to the objective on having the estimation method of multicast group size for large networks, that has a balance on achieving scalability and accuracy. We believe the membership monitoring will be an essential component for widespread deployment of scalable multicast.

1.5 Thesis Outline

This thesis organizes in seven chapters. The next Chapter 2 describes problems in more details by discussing the issues on membership monitoring. In Chapter 3 covers the related work in this field. Chapter 4 provides the requirements and continues with Chapter 5 that presents the approach. In Chapter 6 and 7, we mainly analyze and evaluate our approach based on various network simulation scenarios and implementation in PlanetLab testbed. This thesis finalizes with Chapter 8 that concludes our work.

Chapter 2

Issues on Membership Monitoring

In this chapter we emphasize the detail problems with the existing solution in regard to objective of this research. **Feedback implosion** and **estimation convergence time** are the central issues to be solved.

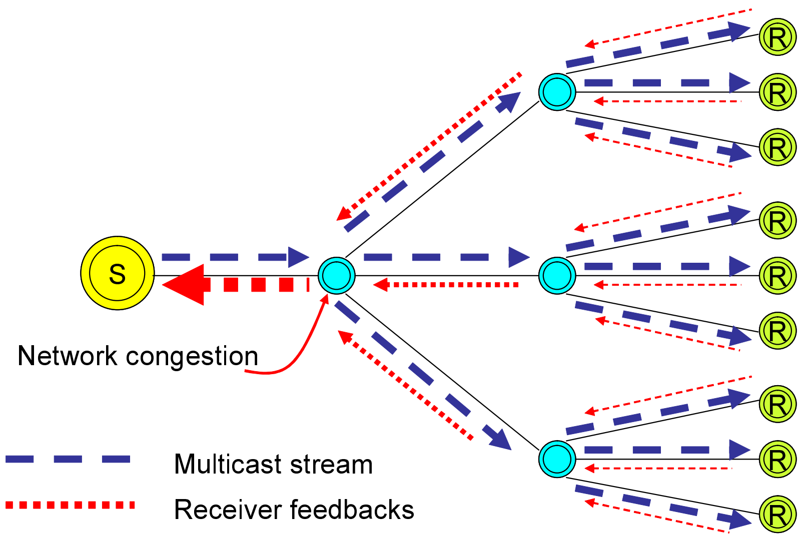

Figure 2.1: Feedback implosion

2.1 Feedback Implosion

IP Multicast provides un-reliable and best-effort delivery at network layer. It is designed for efficient data distribution mechanism for many-to-many or one-to-many communication model. For reliable delivery, feedback channel being used to provide feedback information or even controlling the packets delivery. It is obvious, the existence of feedback is important whether it is for guaranteeing reliability or even only for control information of specific sender adjustment or membership monitoring.

As we mentioned earlier in Section 1.2, that IP Multicast basically does not define how to monitor the feedback from receivers. Hence, we need to implement process level application to be able to monitor the membership from all receivers whether by sender initiates feedback request, *sender-initiated* approach or each receiver periodically reports to source, *receiver-initiated* approach. Since we need to solicit feedback from all multicast receivers then it poses common scaling problem whenever a very large number of receivers send their feedbacks all together.

Figure 2.1 illustrates how the feedback from receivers may cause network congestion close to the source node, which is known as *feedback implosion problem*. Suppose we have source of media streamer, with link capacity full-duplex 256 kbit/s, multicasts to 10,000 receivers, while the size of each receiver's feedback is 90 bytes. Whenever all receivers simultaneously join the multicast group and send receiver ACKs towards the source, apparently it will give about 7.2 Mbit/s of total feedbacks traffic. This condition becomes completely undesirable congestion toward the source at its first hop. That is why current standard RFC 3550 [9] applies 5% of session bandwidth of source stream as limitation of total feedback traffic from receivers in order to avoid implosion and schedules the feedback transmission interval between receivers. However, this condition sacrifices undesirable condition for membership monitoring discussed in next Section 2.2.

Feedback implosion was firstly analyzed more than a decade ago when designing reliable multicast transport protocol and has been extensively addressed by many researchers up to now. As we illustrated in previous paragraph, feedback implosion is inherited in the area of real-time multicast communication that support for collecting feedback from receivers. Hence, to get better understanding, we try to classify solution into two big categories, namely *hierarchical approach* and *end-to-end approach* as described in the following sections.

2.1.1 Hierarchical Approach

This classification applied to solution that conceptually applying hierarchical trees feedback suppression, whether it needs network components support, *structured-based* or directly gathering the information of the trees from designated nodes, *representative-based.*

Structure-based

Structure-based as illustrated in Figure 2.2, reduces the implosion and provides low overhead to the source because intermediate nodes can summarize or aggregate feedback information or even counting the session size quicker. It highly depends on congruent of feedback tree and apparently poses high cost implementation and maintenance since it needs total changes to all network components.

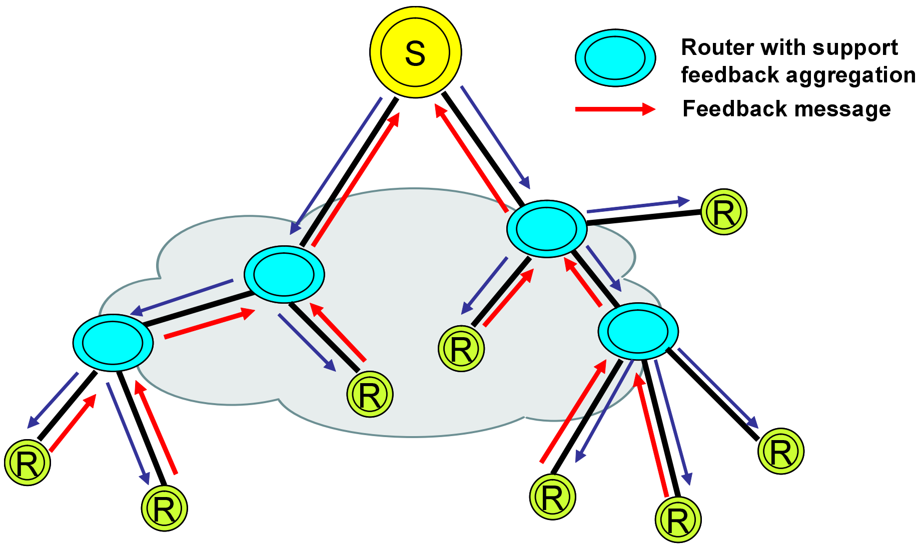

Figure 2.2: structure-based approach

Pragmatic General Multicast (PGM) [17], Light-weight Multicast Services (LMS) [15] and Tree-based Multicast Transport Protocol (TMTP) [18] are fall into structure-based category.

Representative-based

Representative-based solution as illustrated in Figure 2.3 alleviates feedback implosion by sender dynamically select representatives set of receivers from most congested multicast sub-trees, based of feedback information from receivers. These representative nodes then become sender representative on processing feedback. If no similar feedbacks have been received by other receivers during a waiting random timer, then receivers will send feedback directly towards upstream representative or even to the sender. This solution does not need network processing feedback, since everything performed by dynamically constructing hierarchical tree based on receiver's feedback information such as, RTT (round-trip time) and TTL(time-to-live). However, since it is only intended for reliable multicast to minimize and processing the feedback and guarantee the reliability, the representative nodes can not infer session size. In general for representative-based approach principle, it is possible to gather the information status of receivers by consolidating all DR(Designated Receiver) or representative nodes.

Log-Based Receiver-reliable Multicast (LBRM) [19], Reliable Multicast Transport Protocol (RMTP) [14] and multicast feedback suppression using REPresentative (REP) [20] are fall into representative-based category.

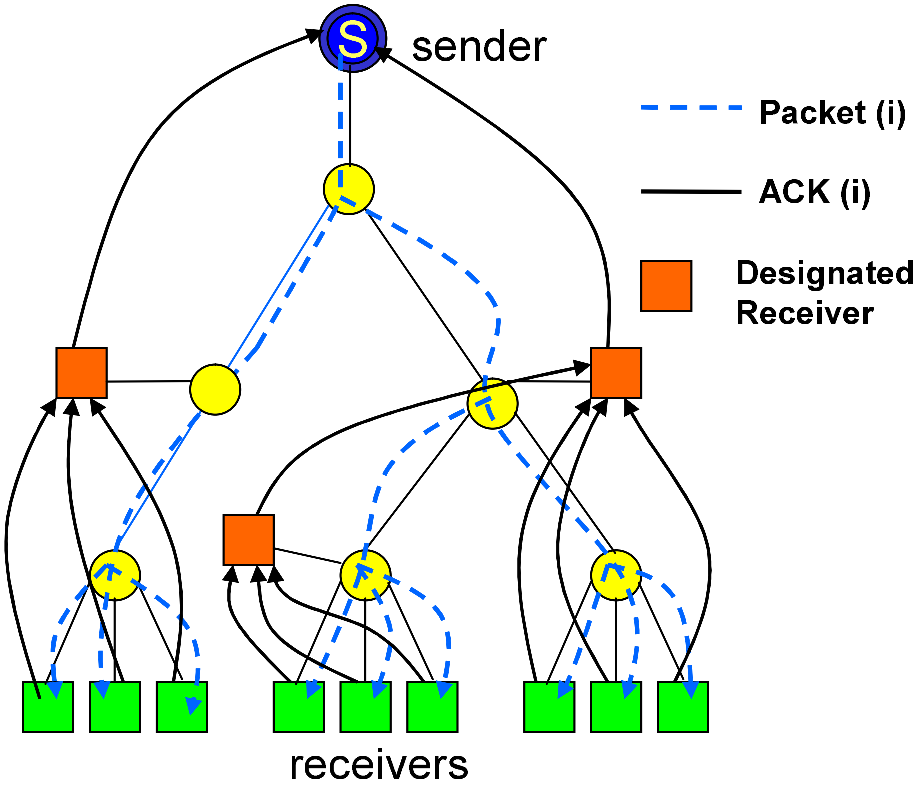

Figure 2.3: Representative-based approach

2.1.2 End-to-end Approach

End-to-end approach basically faces scaling problem of obvious feedbacks implosion whenever sender receives feedbacks from large number of receivers at simultaneously as illustrated in Figure 2.4. On other hand, structure-based approach looks very much appealing. However, it does not fit with the current Internet for several reason such practical and high cost implementation reason. It is because we need to do much more work that include standardizing the specification and applying new features in every network router functionality. Moreover, network owners usually will not easily provide network router functionality to all, because it might expose some degree of their network integrity. Hence while waiting the improved network functionalities, current most appealing and feasible cost in the development are end-to-end approach. It is also suitable for applications designed to work with the present-day Internet. In order to avoid feedbacks sent simultaneously from all receivers, the most logical ways are about delaying or sampling the receivers who sent the feedback as explained in the followings.

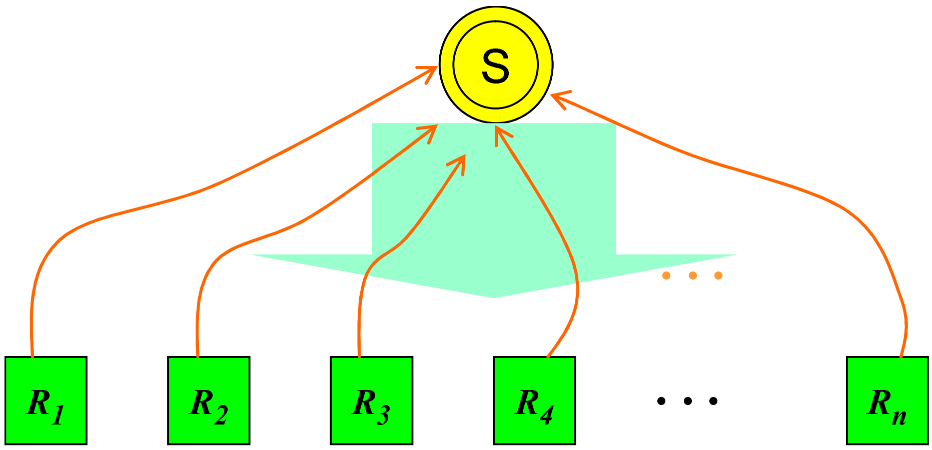

Figure 2.4: End-to-end approach

Timer-based

Timer-based approach in general can be illustrated in Figure 2.5. When a receiver on a multicast group receives others feedback message, it will immediately suppress its own feedback. Then, it will wait using random back-off timers. It can send feedback after the timer expires and no other feedbacks being received. In general, this approach applying feedback suppression algorithm by determining of feedback timers from receivers, where the timer can be set randomly or deterministically.

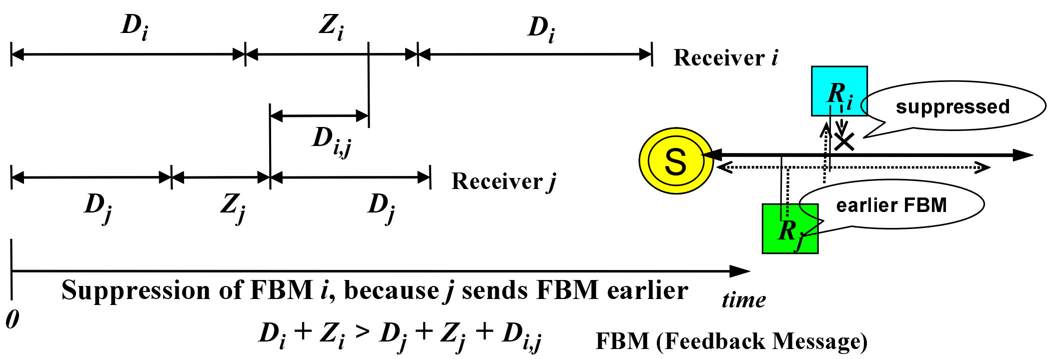

Figure 2.5: Timer-based approach

This solution does not need modification to the current IP multicast routing protocol and also no network processing feedback, entirely between the end-to-end application. It is simple and flexible to be implemented since it just need to modify the behavior of the multicast application. However, it costs a higher application delay, relies heavily on the availability of multicast feedback channel to be able efficiently distribute suppression and membership group size information can only be estimated as the effect of suppression. The Scalable Reliable Multicast (SRM) in [8] and other work DTRM (Deterministic Timeouts for Reliable scalable Multicast) and NB (Nonnenmacher and Biersack) in [12, 13] are fall into this category. SRM sets random timers weighted with the RTT between sender and each receivers. DTRM sets the timer based on assumption of bounded (deterministic) delay jitter and also uses RTT between all receivers and sender. While NB independently set the random timers (using exponential distribution) on each receiver without previous knowledge of RTT between senders and every receiver, but depends on estimation of group size.

Sampling-based

Sampling-based solution applied by sender probabilistically sampling the receivers feedback through polling request as illustrated in Figure 2.6. Probability factor sends through a polling request that multicast-ed to all participating receivers. On each receivers, upon receiving the polling request, receivers that have match requested parameter will send the feedback. This solution is purely centralized feedback control by the source, instead of distributed control like in timer-based. It poses a challenge problem which is the difficulty on determining best probability factor. The work from Bolot et al. [11] and Friedman and Towsley [21] are fall into this category.

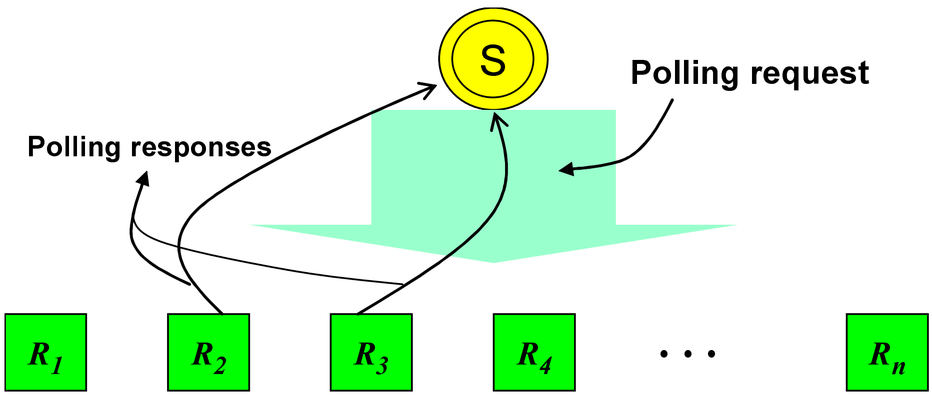

Figure 2.6: Sampling-based approach

2.2 Estimation Convergence Time

In multicast membership counting there are several existing research results, which are in general supports for *direct-counting* of received reports or through polling requests to sampling the reports, *polling-based*. We categorize them as in the followings sections.

2.2.1 Hierarchical Support

In this category, it can be represented by routers, or representative nodes as reporter instead of directly from receivers, hence we categorized it in this class. Filali, Asaeda and Dabbous in [22] and extension of RTP in [23] are using *direct-counting* technique, while Dolev et all. [24] define OMT(On Multicast Trees) by *polling-based* technique.

Filali, Asaeda and Dabbous in [22] propose membership counting through a modification of multicast routing protocol, specifically PIM-SM, by adding multicast counting table which will record the status or join/leave of receivers to a multicast session. The records will be process hop-by-hop toward the sender's DR. By this solution, intermediate node routers are able to count the membership of the active multicast session.

RTCP extension with unicast feedback are proposed in [23] to overcome the problem when we have to run RTP in SSM environment. Feedback responses are illustrated in illustrated in Figure 2.7, where receivers are not transmitting their feedback to multicast channel, instead unicastly towards the feedback target. The feedback target then make summarization or aggregation to be delivered multicastly to original sender and all other receivers.

Dolev et all., use combination approach to avoid the implosion by structure-based and membership estimation by polling-based. Instead of polling to all receivers, he proposed fast algorithm estimation which polls the high degree router based in the multicast trees structure. Since it polls only on high degree router, obviously the network response time slightly faster than all other receivers, which is resulting faster estimation convergence time. An illustration to that can be depicted in Figure 2.8

2.2.2 End-to-End Support

Real-Time Control Protocol (RTCP) defined in RFC 3550 [9] are basically *direct-counting* solution, which mainly governs the feedback transmission interval that scale linearly to the number of participating receivers as explained in previous paragraph. While the work in [11, 13, 21, 25, 26] are *polling-based*, which we will discuss further in next Chapter 3, altogether they are fall

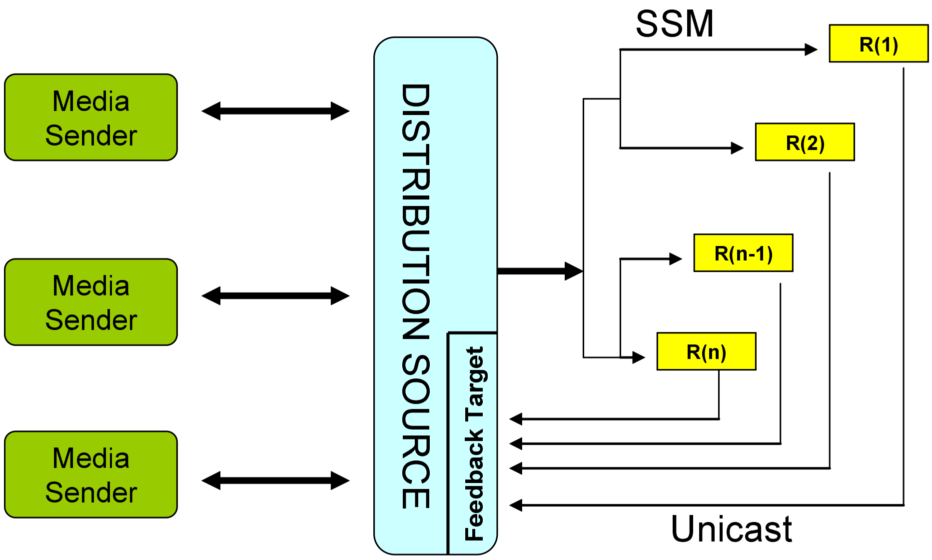

Figure 2.7: RTCP with Unicast Feedback

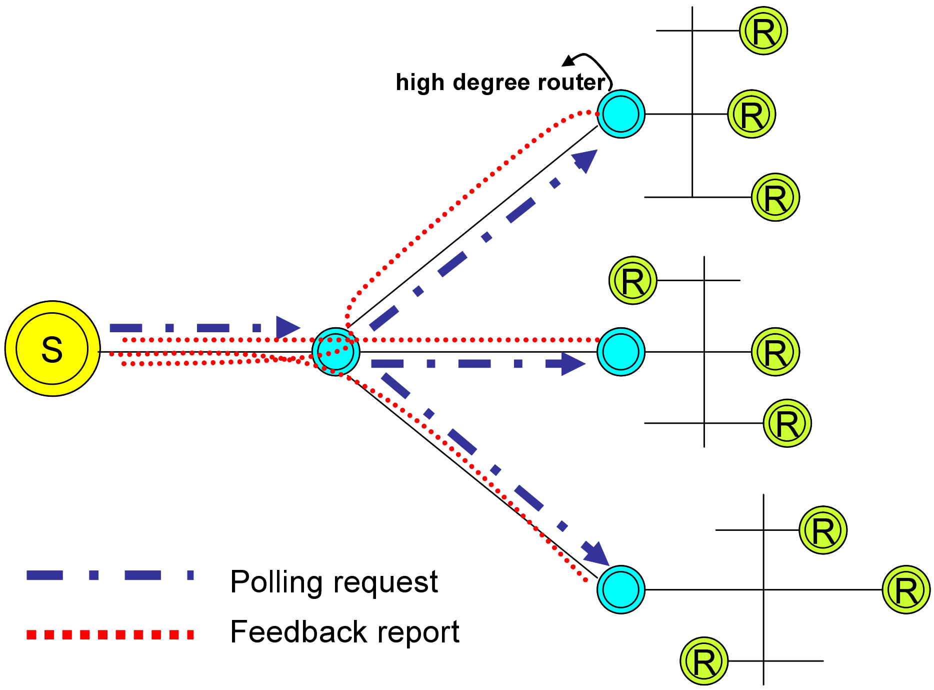

Figure 2.8: Multicast size estimation using polling on high degree router

into this End-to-end approach category which are not dependent on network supports.

Current multimedia network application such as VoIP, video conferencing and other multicast application use RTP as defined in RFC 3550 [9]. This standard provides mechanism to solicit the feedbacks by applying timing rules

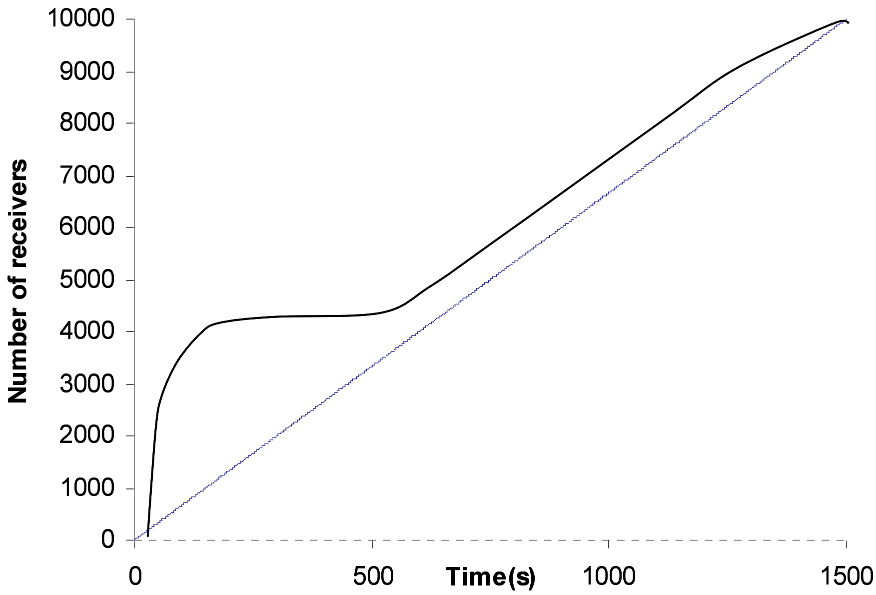

Figure 2.9: Counting convergence time of RTP/RTCP

for each receiver to transmit their feedbacks. The more receivers exist in a session group the longer report interval time. Hence, the receiver's report interval will scale linearly with the number of members in the group. Although it reduces the feedback implosion problem, but introduce new condition which is counting convergence time became too long as we can see on the Figure 2.9, where to know the existence of 10,000 receivers took 1500 seconds to conclude (suppose 128 kbit/s source stream with 10,000 receivers join simultaneously). This fact implies that for very large multicast group size, counting is become impractical and indicates that alternative sampling method may solve the problem.

2.3 Summary

On concluding this chapter, we provides the taxonomy of feedback implosion control and membership monitoring in Table 2.1. Multicast membership monitoring poses significant issues on how to deal with the trade-off between avoiding feedback implosion and achieving faster convergence time of estimation. Hierarchical approach may fulfill the our research objective, however the high costs in implementation leads us to see the alternatives on end-to-end approach. On the other hand, the current existing standard RFC 3550 implies alternative solution that may use feedback sampling method instead of direct-counting.

Table 2.1: Taxonomy of Feedback Implosion Control and Membership Monitoring

Solution	**Feedback Control**	**Feedback**	**Counting Time**	**Membership Counting**	**Targeted Application**
PGM[17]	structure-based	Unicast	Fast	hierarchical	Data Distribution
LMS[15]	structure-based	Multicast	Fast	hierarchical	Data Distribution
TMTP[18]	structure-based	Unicast	Fast	hierarchical	Data Distribution
REP[20]	representative-based	Unicast	-	-	Data Distribution
LBRM[19]	representative-based	Unicast	-	-	Data Distribution
RMTP[14]	representative-based	Unicast	-	-	Data Distribution
SRM[8]	timer-based	Multicast	Slow	end-to-end – polling-based	Any
DTRM[12]	timer-based	Multicast	Slow	-	Any
NB[13]	timer-based	Multicast	Slow	end-to-end – polling-based	Real-Time Multimedia
BTW[11]	sampling-based	Unicast	Slow	end-to-end – polling-based	Real-Time Multimedia
FT[21]	sampling-based	Unicast	Slow	end-to-end – polling-based	Real-Time Multimedia
RTP[9]	timer-based	Multicast	Slow	end-to-end – direct-counting	Real-Time Multimedia
FHW[22]	structure-based	Multicast	Fast	hierarchical – direct-counting	Any
DOLEV[24]	sampling-based	Unicast	Fast	hierarchical – polling-based	Any
RTCP-SSM[23]	representative-based	Unicast	Slow	hierarchical – direct-counting	Real-Time Multimedia
LN[25]	timer-based	Multicast	Slow	end-to-end – polling-based	Real-Time Multimedia
SARA[26]	sampling-based	Unicast	Slow	end-to-end – polling-based	Real-Time Multimedia

Chapter 3

Related Work

There are at least two general categories to handle feedbacks in scalable manner; end-to-end approach and hierarchical approach which has been discussed in Chapter 2. At the first ones, is empowering the application to have knowledge of the group size membership with the possible feedbacks implosion. Second ones, provides information feedback aggregation support by network components hierarchically to help reducing the possible feedbacks implosion effectively. The applicability of those feedback implosion avoidance technique used in two different implementation objectives; reliable multicast transport and real-time transport. Reliable multicast transport needs total reliability hence it tries to achieve the reliable transmission from source to receiver by handling effectively the receiver feedbacks, suitable for bulk data distribution application. While in the real-time multicast transport needs semi reliability or loss-tolerant where feedback management can used for knowing reception quality feedback, participant identification, and synchronization between media streams or others. It is mostly applicable in multimedia application e.g: VIC[27], RAT[28], VLC[29], DVTS[30], etc.

In this chapter we focus to emphasize an end-to-end approach that applicable in real-time multimedia transport area, specifically with probabilistic polling-based approach. This method is considered to be the appropriate one to handle a very large number of receivers.

3.1 Real-Time Control Protocol - RTCP

RTCP is a sister's protocol of RTP and both define in RFC 3550[9]. It uses different multicast channel to convey the information to all nodes, designed to operate in ASM service model in the beginning. However, non-normative suggestion exist in the specification that in order to operate in SSM service model, receivers's RTCP is entirely turn-off. Current working progress in [23] is proposing specification to be able to run on SSM environment without sacrificing RTCP information.

RTCP basically has mainly has 4 functions that can be describe as follows:

1. Provide information to application
   e.g: feedback on quality of transmission

2. Identify RTP source and receivers
   e.g: canonical name for easier recognition

3. Control RTCP transmission interval
   e.g: scaling purpose to govern feedback traffic only 5% session bandwidth

4. Minimal session control information
   e.g: participant identification displayed in user interface application

RTCP runs alongside RTP and provides periodic reporting of these information. In order to control the feedback to avoid such implosion they use transmission control based on the numbers of receivers. It has the knowledge on how many receivers participating in a session, and then change the RTCP transmission interval from each node. The information sends multicastly, hence every node knows about the number of members.

However, as we have discussed in Chapter 2 section 2.2, since the RTCP transmission interval scales linearly by the number of members, hence for monitoring purposes on very large multicast session, it failed to convey the accurate size of session as also experience by Dutta, Schulzrinne and Yemini on proposing Internet radio and TV called MarconiNet [10]. It is also stated in the specification section 6.2.1 of [9] that for a very large number of receivers it should use the sampling-based method.

3.2 Probabilistic Polling-based Approach

Polling-based approach on handling the feedback implosion has long been researched and the summary of polling-based approach can also be seen in Table 3.2 and common trade-off issues are tabulated in Table 3.1.

The first one was proposed by Bolot et al.[11], they use mechanism to solicit feedbacks using a series of polling rounds, until sender gets the responses then estimates the group size. It has low estimation quality, slow estimates (several polling queries), and prone to implosion. They used incremented probability p from 2^{-16} and will stop after receiving reply or the p is already reaching value of 1.0. Timer of observation is set to two times of larger RTT between receivers group to source. If the timer timeouts, it will launch next polling round with new p. It is obvious that it can avoid the feedback implosion if the receivers are more than 2^{16}. Moreover, it does not follow the dynamic nature

Table 3.1: Estimation Trade-off in Polling-based approach

Small p, Large T	**Small p, Large T**
avoid implosion slow estimate low estimator quality	prone to implosion fast estimate high estimator quality
Small p, Large T	**Small p, Large T**
avoid implosion fast estimate low estimator quality	prone to implosion slow estimate high estimator quality

of multicast session membership, because their intention was to estimate the congested receivers as needed in their video conferencing system IVS.

Nonnenmacher and Biersack in [13] were proposed intelligent feedbacks suppression using timer-based technique to avoid the implosion that is applied to all receivers. It proved to be scalable up to 10^6 and mainly contribute to reliable multicast communication. They estimate the number of receivers based on the received feedback messages from a single polling round. It has low estimator quality since it is biased whenever membership size increases, which will lead to over-estimate result. It is highly dependent to the availability of multicast feedback channel in order to distribute parameters and inform suppression to all receivers, which is undesirable for operational Source-Specific Multicast. This approach can follow the dynamic of membership but the delay is assumed to be homogeneous delay.

Friedman and Towsley were the first ones who did analysis in [21] for the former research results of [11, 13] and proposed refinement algorithm to improve estimation quality feedback implosion suppression. They mapped their analysis of polling method into binomial (N, p) distribution add more accurate estimator by doing more than one polling before estimates the membership until a minimum amount of feedback is reached. It costs on convergence estimation time and failed to follow the dynamic natures of multicast group size.

Liu and Nonnenmacher in [25] continue the work with addition on making closed-form to estimates the group size, but still inherited the possible feedback implosion.

The latest research by Sara et al. in [26] propose the refinement of the former research on unbiased estimator and efficiently usage of the previous estimates in order to know far better membership estimates while avoiding the feedback implosion. This approach did refinement of the work in [13] and add better assumption that delay is not homogeneous. They used periodic polling for estimation and at each end of polling, with static value of probability, p and observation time, nT and set the observation timer at nT larger than largest RTT. In their simulation analysis they use 1s for the T and p to lowest one in order to avoid implosion. However, they have open issues on determining a priori information needed for estimation that related to what the optimal value of probability p and observation time, T.

Table 3.2: Comparison of Related Work

Approach	#ACKs needed	Previous Estimate	Feedback Implosion	Estimate Convergence	Estimator Quality	Initialization
Bolot, Turletti, and Wakeman	at least one	no	$> 2^{16}$	slow	low	Static
Nonnenmacher and Biersack	at least one	yes	no	fast	low	Static
Friedman and Towsley	at least one	no	no	fast	high	Static
Liu and Nonnenmacher	at least one	no	possible	slow	low	Static
Sara et al.	more than one	yes	no	fixed	high	Static

3.3 Summary

On concluding this section, polling-based method is considered to be appropriate one to handle a very large number of receivers in end-to-end approach. The above discussed works have their own trade-off problem between the estimator quality, avoiding the feedback implosion and also estimate convergence time. Those issues need to solve for what we believe as solution for scalable multicast membership monitoring.

Chapter 4

Requirements

We have studied several issues on dealing with feedback in appropriate manners from what we introduced in previous chapters. In this chapter, we define the requirements of the multicast group size monitoring technique in large network that should fulfill the following aspects.

4.1 Scalability

Hierarchical and End-to-end approach on handling the receiver's feedbacks could scale from several hundreds to millions, according to what we learned of the related work. Current reality, as we can depict from Figure 1.1 in Chapter 1, the growth of IP Multicast receivers on Internet IP Multicast backbone (MBone) is around 7,000 at maximum. By the consideration of current environment availability and future expectation, we choose end-to-end approach that utilized probabilistic polling-based method (sampling-based). In order not to inherit disadvantages from previous research, the proposed approach should have a mechanism to change dynamically the probabilistic parameter to avoid possible implosion.

4.2 Accuracy

Estimation is an approximate calculation of quantity or degree based of uncertain input data which is still usable. Hence, it is likely prone to quantity estimation error, which usually caused by inadequate numbers of sample data. According to our operational experience for a distance learning system of SOI-Asia Project [5, 31], even though we now have 27 partners across Asian countries, but the representation of multicast receivers were less than 10 receivers in every regular class basis. This condition caused by current facts, that most of our partners only deployed the multicast capable network limited to specific network devices that connected to our system (most of them only in one distance learning classroom). Another facts in MBone operation around 1997 to 1999 [3, 32], the finding was only about 100 to 200 active IP Multicast group receivers. So in reality the group size membership could go to high number of receivers as well as lower number of receivers. Hence, the proposed approach should have a mechanism to control the accuracy by adjusting the probability factor in order to adapt to lower number of receivers condition as well as higher numbers. With these features, the proposed solution will have a balance between scalability and accuracy aspects. The technique in implementing this should have a mechanism to control the probability factor in order to have better accuracy in low number of receivers.

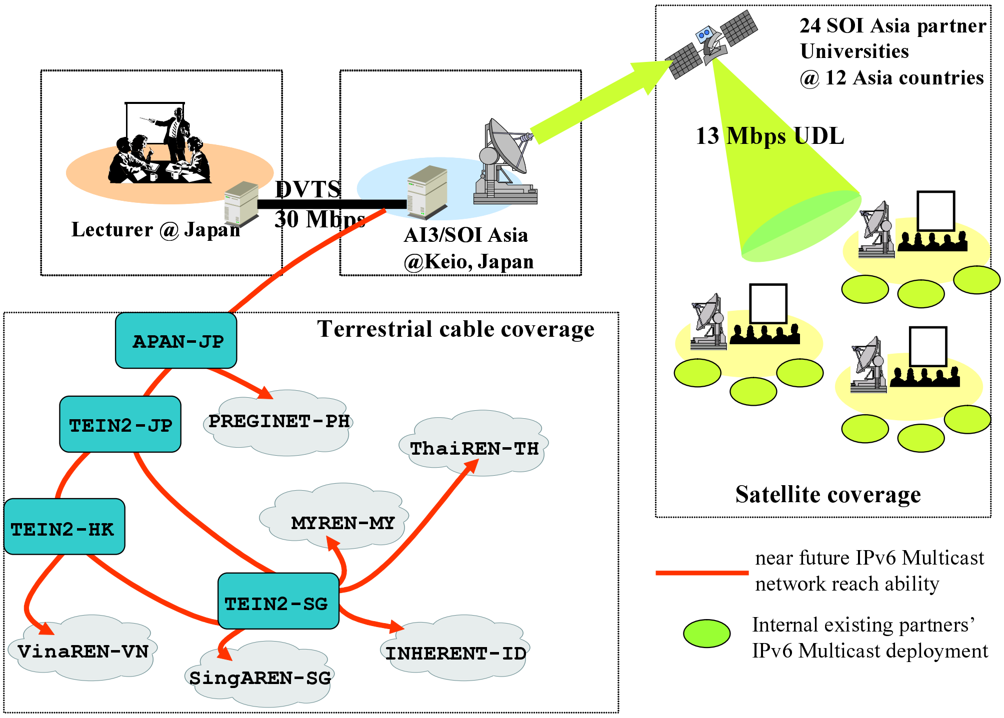

Figure 4.1: Near future possible IPv6 Multicast reachability of our distance learning stream

4.3 Estimation Time

Multicast group size membership could be very dynamic based on join and leave characteristic of receivers as analyzed in [32]. Moreover, receiver's response delay could be very diverse due to network connection characteristic, which can be under terrestrial cable coverage or satellite coverage such illustrated in Figure 4.1. Hence, we need a technique using this sampling-based method that could adapt to those dynamics, which mean the time granularity that follows the feedback response time. This will affect on the adaptable setup of observation time based on network latency.

4.4 Summary

Those three aspects become the consideration on designing the proposed technique on sampling-based method and will be used as verification aspects in concluding this research. The balance condition amongst those three aspects should be anticipated whenever near future broader deployment of IPv6 Multicast becomes true.

Chapter 5

Approach

5.1 Generic Polling-based Method

The generic polling-based method as we derived from several previous work in [11, 13, 21, 26] can be illustrated in Figure 5.1 and described with the simple analytical model in Equation 5.1. Source sends polling query to multicast group address whose all receivers join to it. Each receiver generates random value $U[0..1]$, then compares with requested probability sent by the source, unicastly. Those receivers whose have match value to requested probability will respond, while others remain silence. That describes how we can sample the receiver's feedbacks to avoid implosion towards the source.

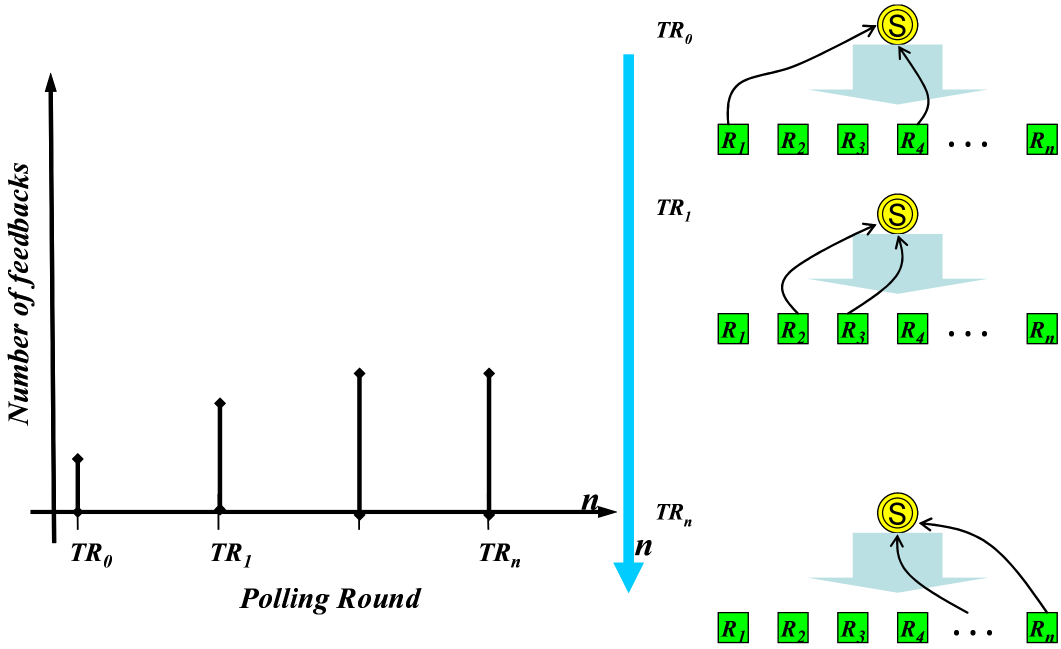

Figure 5.1: Polling-based method

The analytical model representation explains as follows. Source(S) periodically queries receivers with probability p every T time. Each Receiver(R) will send a feedback to source whose has match probability p. Source(S) receives Y_n sample of feedbacks during the next T time. At each end of observation period do the estimate using the Y_n solicited feedbacks.

$$\hat{N}_n(T) = \frac{\sum_{T} Y_n}{p} \tag{5.1}$$

where $\hat{N}_n(T)$ is the estimated number of receivers within T period, Y_n is the number of sample of receiver responses, p is the probability factor set by the sender, and n is the estimation round.

The above naïve estimator lacks of incorporating the real world parameters and also has central issues on determining the optimal value of p and T. Large p value will lead to feedback implosion problem while large T value will lead to slow convergence estimation time. Those factors are also the drawbacks in other previous research [11, 21, 13, 25, 26] that all used static values p and T that lead to low accuracy whenever it is implemented in lower number of receivers and still possible to have implosion whenever it is extremely large receivers join the multicast group.

In this chapter we describe our proposed approach that apply the technique on determining the optimal value of p and T that really based on network and members condition and doing the estimate using the available analytical model, which is a naïve estimator.

5.2 Adaptive Parameters

The adaptive parameters in here are probability factor, p, that will control the number of feedback samples for estimation and the observation period, T, that governs how long we need to wait before concluding estimation at each end of polling round.

5.2.1 Probability factor p

This parameter determines number of sample expected to come, determined by Source, while initial value of it, comes from the our proposed initialization technique in section 5.3. The probability factor, p range between 0 to 1, but never zero to avoid process divided by zero. This parameter basically will start conservatively based on initialization process from low value like 0.01 up to equal to 1, for having responses from all receivers instead of sampling whenever the membership condition is still low, e.g. below 100 receivers. On each receiver, generates uniform random variable $U[0..1]$, this function will be executed whenever received a polling query from Source. Random value generation process have to be as random as possible by using random seed. Receivers with generated random value $\leq p$ will respond to the source, while others remain silence. By having the changeable parameter p, we can control the accuracy and still avoiding the implosion.

5.2.2 Observation period T

This parameter determines how long should we wait for expecting the feedback responses. The longer we wait will affect on the longer estimation convergence time and also the time granularity of monitoring purposes to follow the dynamic membership condition.

Time interval, T, as experienced in [11, 26] is set to twice of the largest RTT statically. Our argumentation to that, supposes we have composition of receivers that closer and faraway from the source, it is obviously feedbacks will arrive at source at slightly different time. If somehow we set the T $1.0s$ while most of feedbacks come at $1.2s$, then it will lead to estimation error and highly possible to have feedback implosion on the second round of polling round. Similar condition, if most of the feedbacks come at $500ms$, then we waste the time for estimation convergence time. It can be true when we deliver the multicast stream to the heterogeneous network that have composition of receivers in terrestrial cable and satellite network coverage. By having adaptive T, it will improves the estimate convergence times and also give better time granularity of monitoring on following real network condition and dynamics of membership. It will have better result from current standard RTP [9], since it does not sacrifice the time consume to conclude the membership size as the number of receivers increased.

5.3 Initialization Method

Conceptually, adaptive parameter changes on probability factor, p and observation period, T are possible whenever we have reference values to start the normal operation of estimation. It means we need initial value of p and T.

In order to achieve it, we use a bootstrap mechanism by doing several polling rounds to collect information in short period. Those information collected during the bootstrap period will then being summarized to get reasonable initial values of expected $E[p]$ and $E[T]$. This should be done at sender side before doing the normal estimation process, we called it a *transient* condition. So conservatively, we can start with low probability of p with reasonable estimation time of T.

5.4 Estimation Algorithm

We define the algorithm for sender agent in Figure 5.2 and receiver agent in Figure 5.3. This algorithm will be executed by sender in a multicast session within the same IP multicast group address as data stream but different port numbers. The receivers, later on, will respond unicastly towards sender who initiated polling request.

Evaluator (p_i, T_i) {
   if $(initial)$ {
      bootstrap(); /*$p_0 \rightarrow p_n$, $T_0 \rightarrow T_n$ */
      p_i, $T_i \leftarrow$ bootstrap($E[p]$, $E[T]$);
   } **else** {
      p_i, $T_i \leftarrow p_{(i-1)}$, $T_{(i-1)}$;
   }
}
SendPoll (p_i, T_i) {
   $i++, n++$;
   while $(t \leq T_i)$ {
      ReportSolicitation(Y_n, RTT_n);
   }
}
EstimateSize(p_i, Y_n);

Figure 5.2: Estimation Algorithm in sender

PollMonitor (p_i) {
   if $(true)$ {
      $p \leftarrow random :: uniform(0, 1); / * U[0..1] * /$
      if $(p \leq p_i)$ {
         SendReport();
         PollMonitor(p_i);
      }
   } **else** {
      PollMonitor(p_i);
   }
}

Figure 5.3: Algorithm in Receiver

5.5 Summary

Our contribution is more on applying the adaptive technique to determine optimal value of p and T that reflect to network and membership condition and estimate using simple sampling based analytical form, which is a naïve estimator. We believe by applying this approach appropriately on handling the receiver feedbacks as controlled by the request from source, will give result that overcomes the trade-off problem between the estimator qualities, avoiding the feedback implosion and also convergence estimation time.

Chapter 6

Simulation

In this chapter we present the details of implementation of the approach defined in Chapter 5. This chapter explains on how we design the implementation in order to verify the method we proposed using network simulator.

6.1 Overview

We implement our multicast group size estimation method in network simulator from UCB/LBL/VINT Project, called ns-2 [33]. ns-2 is discrete event simulator, where the advance time depends on the timing of events maintained through a scheduler. It has rich library of network and protocols objects. Since our method classified as end-to-end approach in which is more on applying the method in the application layer, then we need to use existing available network and protocol objects in order to get working with multicast network environment.

The purposes of this simulation are to quantify the behavior of our adaptive estimation parameters, the scalability on avoiding the implosion, while try to achieve accurate of estimation and fast estimation time as has been defined in the requirement Chapter 4.

6.2 Method of Simulation

We need to do experimentation using network simulator in order to understand the effect on differing probability factor, p and estimation observation time, T over the several simulated network nodes. We implement sender and receiver agents in ns-2 by modifying the existing RTCP/RTCP agents to behave like method we defined previously in Chapter 5 section 5.4. In general, building block of simulation is shown in Figure 6.1.

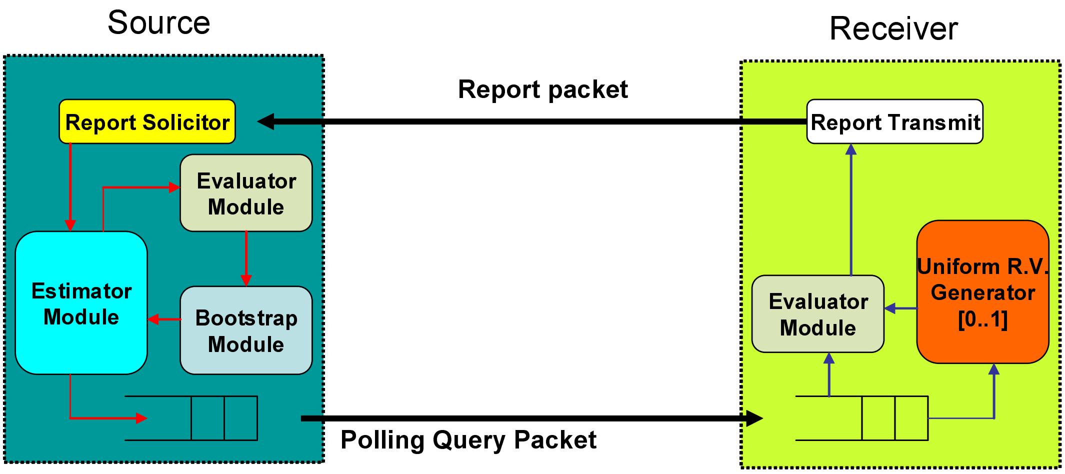

Figure 6.1: Simulation building block

Sender has mechanism to transmit the polling request message consist of p values to multicast address that all receivers were joined and listen for the feedback replies. After receiving all the sampled feedbacks, the sender estimates the total numbers of receivers using estimator as defined in Chapter 5 equation 5.1. Other calculation for the total feedbacks throughput and average RTT are also executed to determine p and T for the next polling round.

In the receiver block, we define functions for monitoring polling request, generate uniform random variable $U[0..1]$ and transmitting the feedback. In ns-2, we utilized heuristic seed assignment of uniform random variables generator in order to have random values that independent to each other. This is for the purpose of not to have the same receivers who response to each polling request from the sender. Upon receiving polling request, it triggers to random value generation, then compare with received p value. If the generated random value less than or equal to p, it will trigger feedback transmit function and other condition remains silent.

In order to get closer to the real Internet, we generate network topology and nodes using topology generator, called BRITE[34]. By using this utility, we generate topology and the required nodes numbers and import topology and link assignment between each nodes to ns-2 simulator. The general view of this, can be illustrated in Figure 6.2, where there are one sender and many nodes represent as receivers.

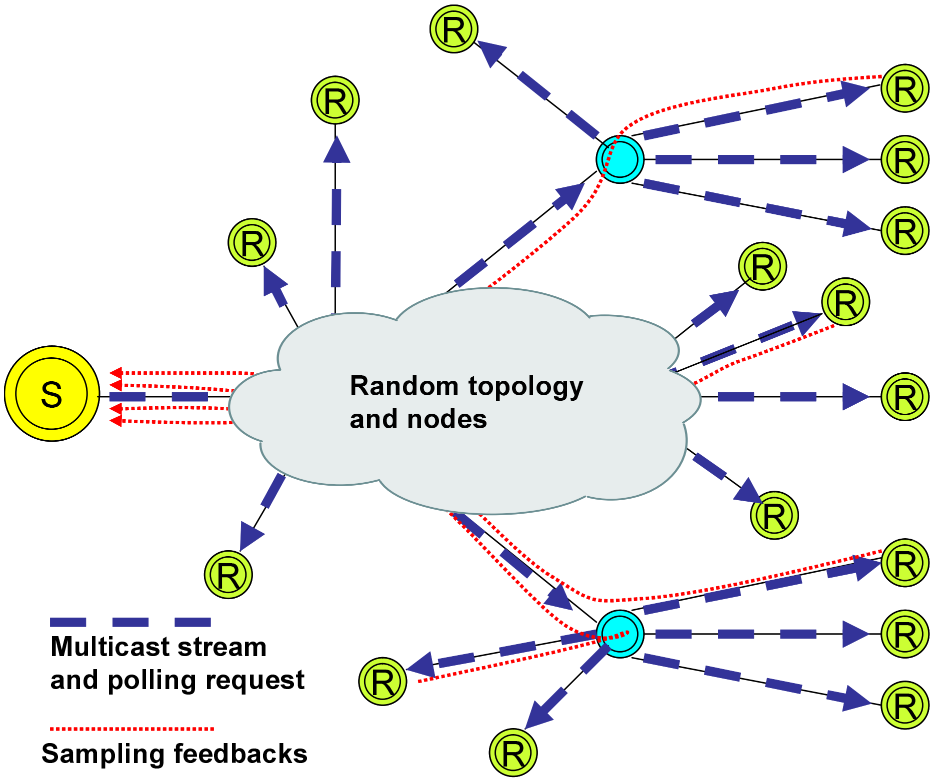

Figure 6.2: Model of Simulation

6.3 Simulation Scenario

We define the scenario to evaluate the performance of our method using several metrics. We generate several network topologies using BRITE that consist of 20 to 200 nodes. In all simulation configuration we added one extra node behave as sender that stream data to multicast group where all receivers joined to.

For each node's scenario, we conducted changing polling probability parameter, p ranging from 10% up to 100%. Then for every scenario we did 10 times simulation instances and computed the average of metric values. The metrics we mainly interested are;

- **Estimation errors**,
  To know the accuracy of this method and find what is the optimal values of p by increasing the number of receivers.
- **Feedbacks throughput**,
  To predict the scalability aspects by considering the threshold of feedback throughput, in which sender could handle.
- **Estimation convergence time**, To predict the optimal estimate con-

vergence time for the next polling round based on RTT.

6.4 Simulation Assumptions

Several assumption are used in all simulation instances. We assume all nodes are simultaneously join at $t = 0$ to the multicast group address where the sender sends the constant rate of RTP stream 128 kbit/s. Bandwidth setup on each link were high enough with minimum 10 Mbit/s up to 1 Gbit/s, since we would like to know how big the feedback throughput converges towards sender.

For simplification, multicast join latencies are ignored, since we used flooding type multicast routing protocol such as DVMRP [35], in which initial packets are flooded too all nodes that are simultaneously joined. It facilitates both data stream and polling request packet that multicast to reach all receivers.

Link delay is also introduced between nodes in order to represent cumulative end-to-end delay from sender point of view. We generated random link delay values between 10 - 600 ms applied randomly using uniform random variable generator to each link between nodes. Combined with the node degrees of generated topology by BRITE, give *end-to-end* delay that will vary in each of simulation instance which represent the dynamic behavior of real world Internet.

6.5 Results and Analysis

Based on the scenarios and assumptions explained in the previous section, we did the simulations and represent the result in sequence order; on estimation errors, feedback throughput and also estimate convergence time. The results were taken from the average result of 10 times simulation instances of each probability factor p in 10, 25, 40, 50, 60, 75, 80 and 100 percentage. Each of them conducted in each 20, 40, 60, 80, 100, 120, 140, 160, 180, and 200 nodes scenarios. In total we have conducted 800 times simulation instances.

6.5.1 Estimation Errors

Table 6.1 represents data for evaluating the accuracy of estimation. We compute the estimation errors as follows:

$$\% \quad of \quad est.error = \left| \frac{Est.Count - ExactCount}{ExactCount} \right| \quad * \quad 100$$

Table 6.1: Estimation Errors in percentage(%)

p	Number of Multicast receivers									
	20	40	60	80	100	120	140	160	180	200
10	15.00	7.50	10.00	3.75	6.00	5.00	7.14	7.50	3.89	3.00
25	6.00	3.00	1.33	2.00	5.60	1.33	5.43	2.50	0.89	3.80
40	4.50	5.00	3.33	2.50	3.90	2.50	6.14	2.31	3.33	1.15
50	6.00	5.50	2.00	3.00	3.40	3.50	4.29	0.75	4.67	1.30
60	6.00	5.00	4.50	1.37	1.60	1.58	1.79	0.12	2.17	1.25
75	7.00	2.25	3.00	0.25	0.60	0.25	1.79	1.31	1.11	1.10
80	6.00	3.25	0.83	0.37	0.10	0.17	0.21	1.69	2.11	1.25
100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

We plot the data in Table 6.1 into two graph in Figure 6.3 and Figure 6.4. We see the estimation error for p less than 50%, the estimation errors are so noisy compare to p greater than 50%. Moreover refer to related data, we can see by the increasing number of receivers, the estimation error become lower and lower.

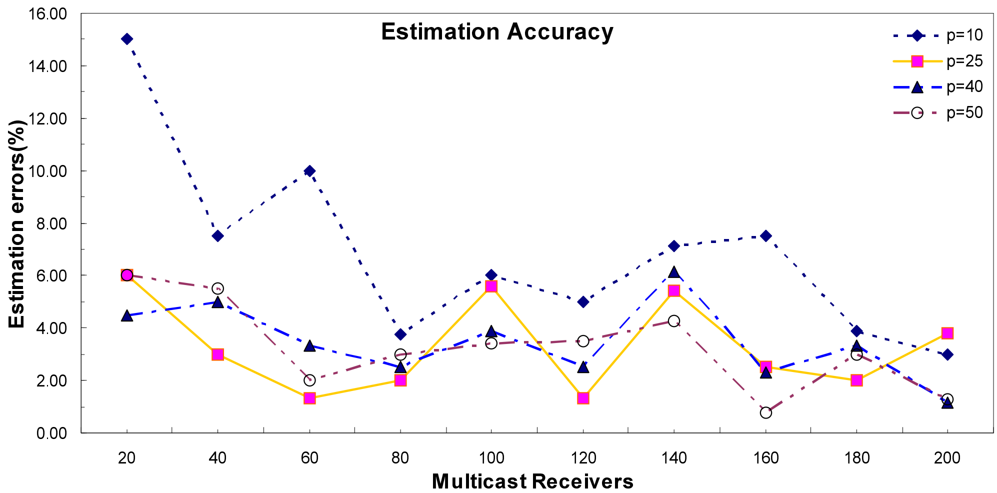

Figure 6.3: Estimation errors on p between 10% to 50%

On the other hand, if the real number of receivers become low less than 40, the estimation will be useless. It is obvious because based on our raw data, whenever we use $p = 10\%$ on actual membership of 20 nodes, the results become so noisy sometime *underestimate* to 0 or *overestimate* to 40. This is because the number of samples below an adequacy of doing statistical sampling. In this case we may change the $p = 100\%$ to get accurate results, but need to check the feedback throughput to avoid possible implosion.

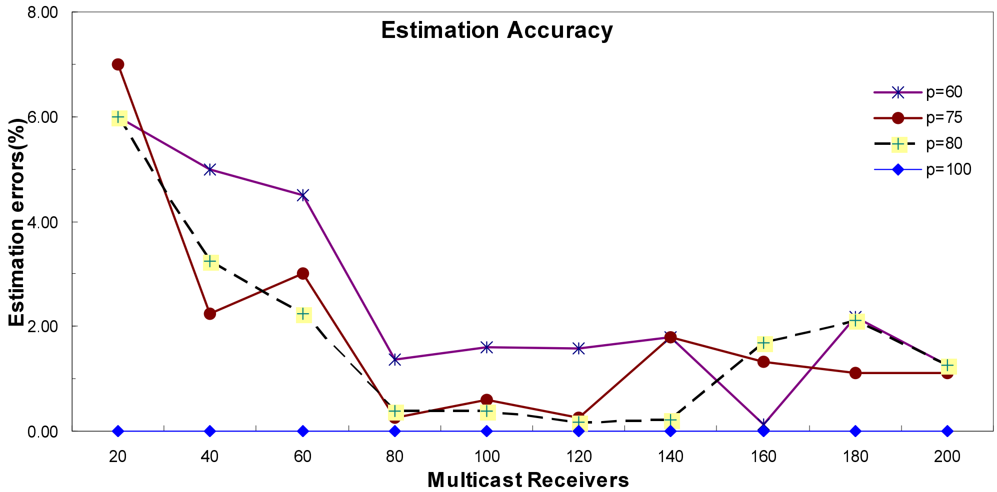

Figure 6.4: Estimation errors on p between 60% to 100%

6.5.2 Feedbacks Throughput

Table 6.2 shows the feedback throughput on each estimation in every nodes setup. We plot the data into graphic for easier understanding in Figure 6.5. As indicated in previous section, we are interesting to increase the estimation accuracy for low numbers of actual multicast group size.

Table 6.2: Feedbacks Throughput in towards sender (in kbit/s)

p	Number of Multicast receivers									
	20	40	60	80	100	120	140	160	180	200
10	0.60	0.41	0.76	0.89	1.11	1.17	1.50	1.55	1.74	1.98
25	0.70	1.35	1.98	2.15	2.90	2.86	3.54	3.82	4.23	4.90
40	1.39	1.91	2.70	3.61	4.58	5.20	5.57	6.27	6.62	7.69
50	1.42	2.30	3.88	3.86	4.90	5.91	6.50	7.61	7.90	9.80
60	1.72	3.15	4.87	4.75	5.95	7.26	8.33	9.20	10.54	11.01
75	2.08	3.81	5.07	6.38	7.63	8.49	10.58	12.00	12.82	14.83
80	2.32	4.46	5.36	6.79	8.21	9.12	10.50	13.28	13.42	15.29
100	3.09	4.95	7.38	7.87	9.93	11.70	13.76	15.31	16.49	19.44

We use 128 kbit/s RTP constant stream to all receivers, with assumption that link capacity is full-duplex and could handle feedbacks traffic up to the capacity as they could transmit the data stream. But in order to avoid further congestion in the network we can refer to RTP/RTCP [9] that use 5% of *session bandwidth*(amount of bandwidth used by the stream's sender) to be shared among participating receivers. If we apply in our condition, 5% of 128 kbit/s is 6.4kbit/s. We can see from the table with $p = 100\%$ up to 50 nodes membership, the total throughput are still below 6 kbit/s, which mean it is sufficient enough to handle all response to increase the estimation counting result. Even receiving all 200 feedbacks our results indicate about 15% of

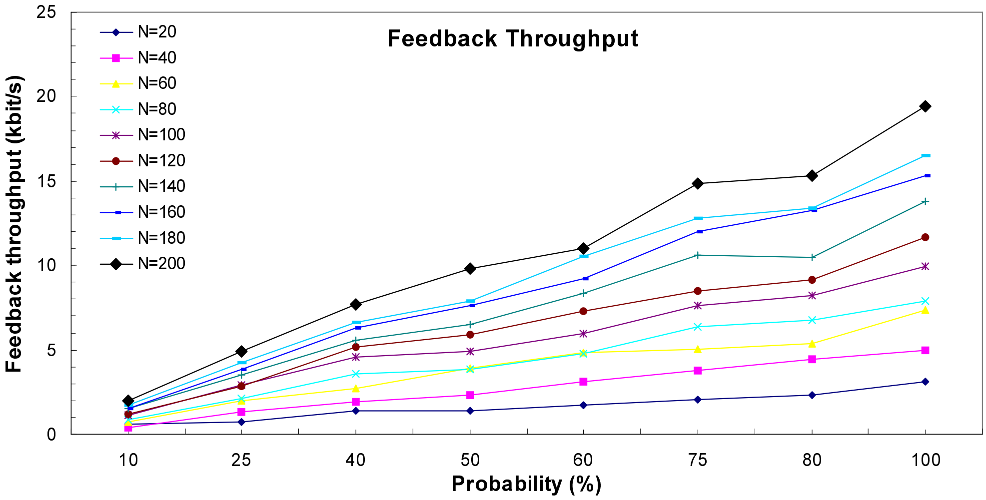

Figure 6.5: Feedback Throughput

session bandwidth. Since our method are controlled probability by the sender, hence receiving such amount of traffic are considered still sufficient without causing the implosion.

6.5.3 Estimation Convergence Times

The collected RTT on each simulation instances are basically all response times between the time when sender sends the polling request to each individual receiver's response is received. It consists from the fastest response (minimum RTT) until the later response (maximum RTT). Good observation time should cover up to the later polling responses in each polling round to guarantee that all or most responses have arrived. At the end of observation time, T indicates the time sender do the estimation counting, which means the estimate convergence time comes at the end of each polling round. The illustration shown in Figure 6.6.

The observation time, T can be estimated from the average RTT and its spread of RTT data. The spread of RTT can be measured from its standard deviation. In order to cover up to the maximum RTT, we calculate the T based on the average RTT plus the statistical one-sided confidence interval 99.98% (3.54) of its RTT standard deviation. The calculation results shown in Table 6.3 and maximum RTT data shown Table 6.4 can be compared. The results indicate it is sufficient enough to use that calculation in order to cover up to the maximum RTT of the feedbacks replier.

We see on Figure 6.7, the estimation time increases as of increased multicast group size. In actual condition, it will highly depends on the network condition. In our simulation case, since we use generated topology with ran-

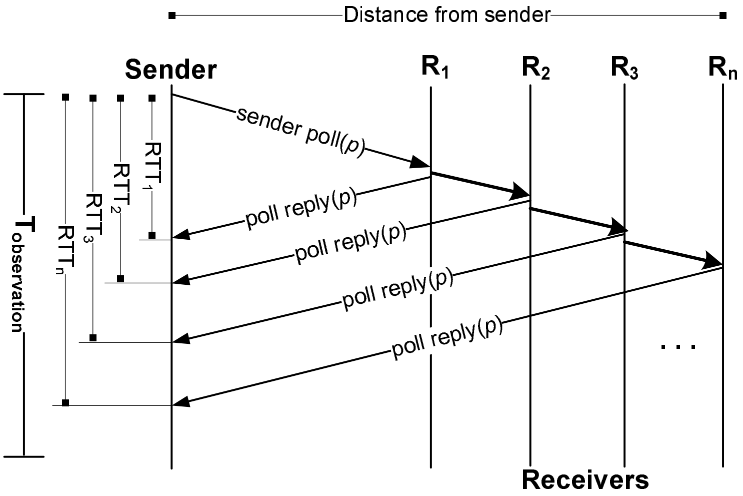

Figure 6.6: Illustration of Observation Time, T

Table 6.3: Estimates convergence time (ms)

p	Number of Multicast receivers									
	20	40	60	80	100	120	140	160	180	200
10	1233.91	1910.44	2085.05	2164.54	2195.08	2494.77	2442.32	2442.30	2562.58	2523.68
25	1629.82	2127.76	2223.41	2249.38	2298.35	2422.40	2351.70	2396.25	2249.38	2543.42
40	2218.80	2235.93	2109.24	2325.20	2170.07	2323.77	2438.91	2255.81	2549.82	2356.36
50	1996.58	2114.18	2039.52	2197.27	2365.46	2344.92	2458.21	2255.37	2197.27	2477.86
60	1874.84	2235.93	2185.74	2276.06	2302.84	2433.97	2403.52	2283.13	2528.50	2465.93
75	2041.61	1994.42	2214.83	2236.36	2381.17	2329.21	2390.32	2302.88	2504.80	2389.94
80	1951.72	2224.44	2048.64	2303.38	2202.42	2374.88	2384.02	2232.63	2516.46	2429.92
100	2193.25	2145.70	2107.45	2275.37	2261.29	2301.33	2403.68	2300.19	2560.03	2508.49

Table 6.4: Maximum RTT in each simulation scenario (ms)

p	Number of Multicast receivers									
	20	40	60	80	100	120	140	160	180	200
10	1298.02	1972.05	1582.03	1855.03	1873.99	1860.02	1838.03	1804.02	1946.03	2148.02
25	1614.02	1478.03	1950.02	1842.03	2043.75	2116.04	1894.03	1992.04	2268.03	2054.01
40	1944.03	1632.05	1744.03	1846.04	1896.76	2066.04	2158.04	1944.04	2160.05	2099.02
50	1998.03	1760.06	1850.03	1854.03	1948.15	2230.03	2195.05	1832.04	2464.04	2264.02
60	1652.03	1632.05	1808.03	1906.04	2077.32	1962.04	2204.04	2010.04	2096.06	2154.02
75	2082.04	1561.05	1788.03	1862.03	2220.14	2112.04	2187.05	2140.03	2218.04	1992.02
80	1734.04	1758.05	1662.03	2108.04	1995.55	2142.04	2064.04	1872.05	2252.04	2234.01
100	1870.04	1780.06	1910.04	1948.04	1897.75	1941.05	2116.04	2120.03	2776.04	2180.02

dom link delay assignment, the further a node location from sender (several hop away) most likely the larger end-to-end latency will be, as indicated by RTT.

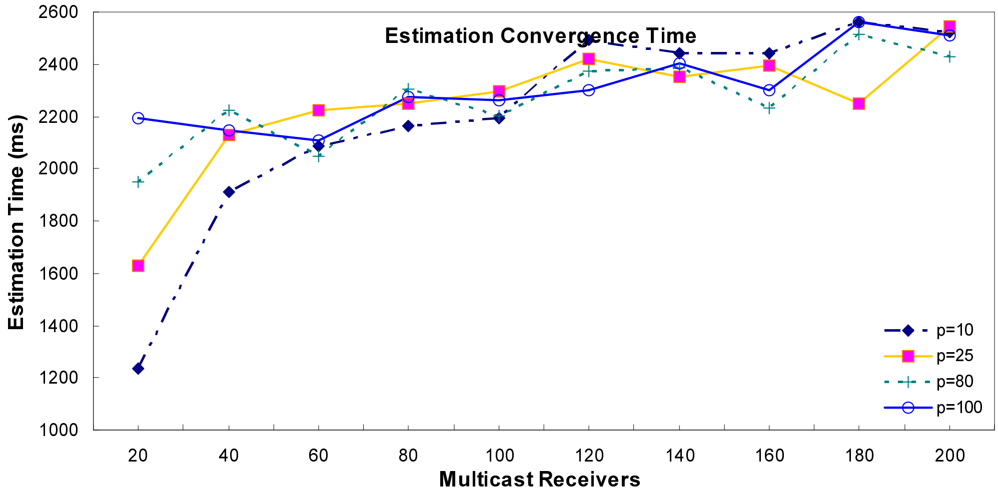

Figure 6.7: Estimation Convergence Time

Table 6.5: Optimum p with estimated N receivers

$\widehat{N}$	p	$\sum Y_n$
100	100.000	100
200	80.000	160
300	53.333	160
400	40.000	160
500	32.000	160
600	26.667	160
700	22.857	160
800	20.000	160
900	17.778	160
1000	16.000	160
2000	8.000	160
3000	5.333	160
4000	4.000	160
5000	3.200	160
6000	2.667	160
7000	2.286	160
8000	2.000	160
9000	1.778	160
10000	1.600	160

6.6 Optimum estimation parameters

The analysis of simulation results presented in previous sections from several simulation scenarios, indicate higher number of nodes the lower estimation error in all p trials. Using lower p for small number of nodes (20 to 60) is not sufficient and gives quite noisy estimation, especially using p 10% to 25%. Referring to Table 6.1, the lowest estimation error achieved with p around

60% to 100%.

We choose estimation error around or below 2% which means conservatively we conclude from that table, the estimation below 100 receivers the optimum p will be 100% and between 100 to 200 receivers optimum p is 80%. When sender polls the request with $p = 80\%$ in environment with 200 receivers, ideally the expected feedbacks will be 160. If we cross check with Table 6.2 and its explanation, handling 160 feedbacks is not considered to have feedback implosion. Hence, we predict and construct Table 6.5 for the purpose to build the reference optimum value of p to be used for adaptiveness of p during estimates multicast group size.

For the adaptiveness of observation time, as discussed in the explanation of Table 6.3, the observation time trends is not really obvious but significantly refers to actual receiver response times. The results indicated sufficiency to use that calculation in order to cover up to the maximum RTT of the feedbacks replier.

Chapter 7

Evaluation

This chapter mainly evaluates our proposed approach on applying adaptive estimation parameter changes on network testbed based on preliminary analysis of simulation in Chapter 6.

7.1 Overview

We implement our actual running code in an emulated network created on top of PlanetLab [36] environment. PlanetLab is a network research testbed geographically distributed, as in Figure 7.1, over hundreds of real Internet nodes that supports development or evaluation of new network services or protocols. Essentially each developer or project given "slice" of PlanetLab, a set of allocated resources distributed across PlanetLab nodes, to their own experimentation. Those PlanetLab nodes are mostly hosted by research institutions connected to the Internet.

In general, the experimentation on top of PlanetLab testbed is trying to take the advantages of implementation under real-world network conditions. Specifically, we define our purposes as follows:

- Implement actual running code referring to simulation result
- Evaluate the estimation method on top of PlanetLab testbed.
- Analyze the implementation results and compare with simulation results

7.2 Implementation

Our PlanetLab slice consists of 50 stable nodes where each of node emulates 100 nodes receivers given in total 5,000 emulated number of multicast receivers. At each node we run our receiver code which listens and responds to polling request from the sender. At sender we determine the initialization mechanism and also adaptive estimation parameters p and T accordingly during normal estimation process. Before we go in details of initialization and adaptive estimation parameters. We need to clarify statements we used for the whole emulation experiment in the PlanetLab testbed.

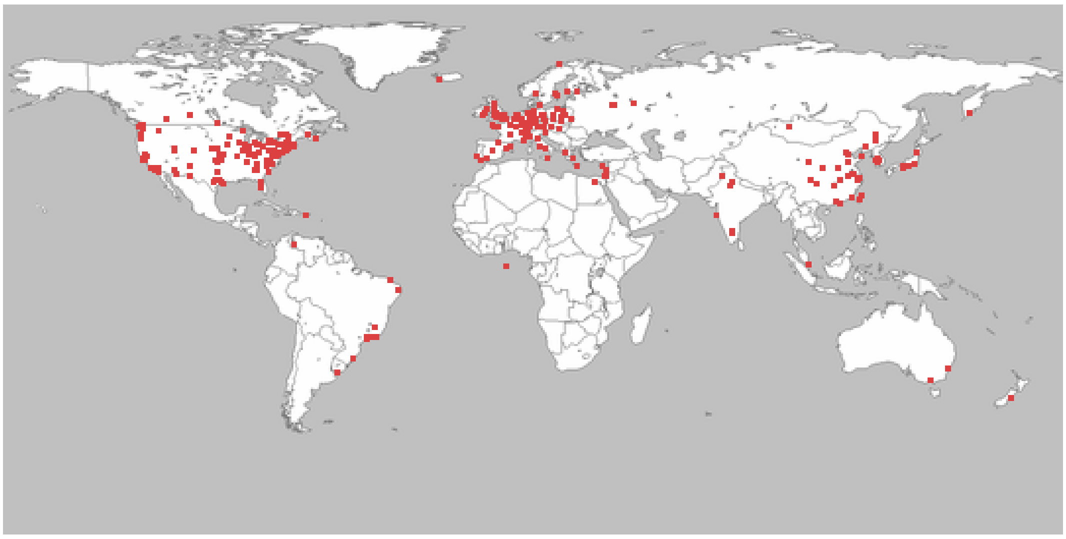

Figure 7.1: Distributed nodes in PlanetLab testbed

1. Since it is difficult to set up a native IP multicast routing environment in the current PlanetLab testbed, hence we emulated one Linux host in our laboratory to send directly to all 50 PlanetLab nodes by unicast, which means multi-unicast. We consider it sufficient to emulate flooding process in multicast routing protocol to reach all receivers.

2. In each node of 50 PlanetLab nodes, we employ random delay of 100 emulated nodes before replying the request. The purpose of this to give diverse delay of total 5,000 emulated receivers within PlanetLab nodes. The delays are generated randomly between 0 to 600 ms.

3. We use same data size of packet for polling request and also the polling reply 32 bytes. Since we use IP and UDP, the size of each packet in the network is 60 bytes.

7.2.1 Adaptive Parameters Changes

We have discussed in the beginning of Chapter 5 that we will use naïve estimator defined in Equation 5.1, by determining optimal parameters for doing the estimation. On avoiding the feedback implosion, we need to have an optimal value of p and T that refer to multicast membership condition. We use controlling method defined in Figure 7.2 to improve the quality of the naïve estimator. This will give a dynamic setup of p and T during dynamic multicast membership estimation counting to the optimal value we defined as controlling point.

We define the controlling point of probability factor by the number of expected feedbacks. We set the feedback threshold to 160 based on value defined in Table 6.5 of Chapter 6. For observation time, we use consideration as we discussed in Section 6.5.3 of Chapter 6 to setup the T. It is based on calculation of average RTT plus confidence interval 99.98% (3.54) from its RTT standard deviation to accommodating the possible maximum late reply feedbacks in each polling round. The illustration of the method could be depicted from Figure 7.2.

AdaptiveParameterChanges (p_0, T_0) {
  if (feedbacks=true) {
    $p_{new} \leftarrow p_{old} * \frac{threshold(160)}{feedbacks}$;
    if ($p_{new} \geq 100\%$) {
      $p_{new} \leftarrow 100\%$;
    }
  }
  if (feedbacks=false) {
    $T_{new} \leftarrow 1sec$;
    $p_{new} \leftarrow 2 * p_{old}$;
  } **else** {
    $T_{new} \leftarrow E[RTT] + 3.54 * \sqrt{Var(RTT)}$;
  }
}

Figure 7.2: The method of adaptive parameter changes applied at sender side

In conclusion of the previous paragraphs, basically we use the adaptive method defined in Figure 7.2 and at the end of every polling round we do the estimation counting of the multicast group size based on the expected number of feedbacks divide by the p as we set like the one defined in Equation 5.1.

7.2.2 Initialization Method

As we defined in Chapter 5 Section 5.3, we use a bootstrap mechanism to determine the initial values of p and T, by doing several polling queries at first.

We choose 5 polling rounds with each of it is within of 1 second observation period, given in total initialization time within 5 seconds. During each end of polling round, we collect RTT responses and adjusting the p based on expected feedback threshold. At the end of initialization process, we then proceed the estimation of multicast group size using the adaptive method we defined before.

By considering the optimum value of p defined in Chapter 6 Table 6.5, then as initial setup values, we use $p = 1.6\%$ and $T = 1sec.$. We use $p = 1.6\%$, which means at the first trial, we are not expecting more than 10,000 receivers join simultaneously within time period $T = 1sec$. We use this assumption based on data of estimated multicast group size in [4]. We believe this assumption quite relevant, because receivers tends to joining a newly announced multicast stream not from the beginning. Instead, mostly join right after it, in the order of minutes. For the initial observation period $T = 1sec$ is based on the assumption that current real world RTT are mostly around $600ms$. We consider $1sec$ as initial value is sufficient observation time, before we got real knowledge of the real response time from the participating receivers. In fact, many previous researchers [11, 26] use conservatively two times of maximum RTT.

7.3 Evaluation Results

As we mentioned earlier a Linux host PC in our laboratory acts as sender sends polling queries directly to all emulated receivers in PlanetLab testbed. We choose an iteration of 100 polling rounds launched to all emulated receiver with an initialization parameter $p = 1.6\%$ and $T = 1sec$. The behavior of our running code was smoothly running in doing 5 seconds initialization setup then followed by normal estimation process. At each end of individual polling round, it successfully counts the p and T parameters for next polling round.

We then collect the data of estimation parameters and the estimated numbers of receivers during the process for further analysis. The results of the experiment are all presented in Figure 7.3, Figure 7.4 and Figure 7.5.

We see in Figure 7.3 our method could follow the dynamic membership of multicast group size, and the estimator successfully set the p accordingly to avoid the feedbacks whenever it detects the increasing numbers of receivers. There are several spikes of estimation result depicted from that Figure, and we will analyze it further by looking at estimation errors.

In Figure 7.4 we see how can our method limit the expected feedbacks by controlling the p. It shows that our method successfully keep the amount of feedbacks around the threshold 160 feedbacks as we set from the starting of estimation process. The result indicates that it still have higher spikes up to

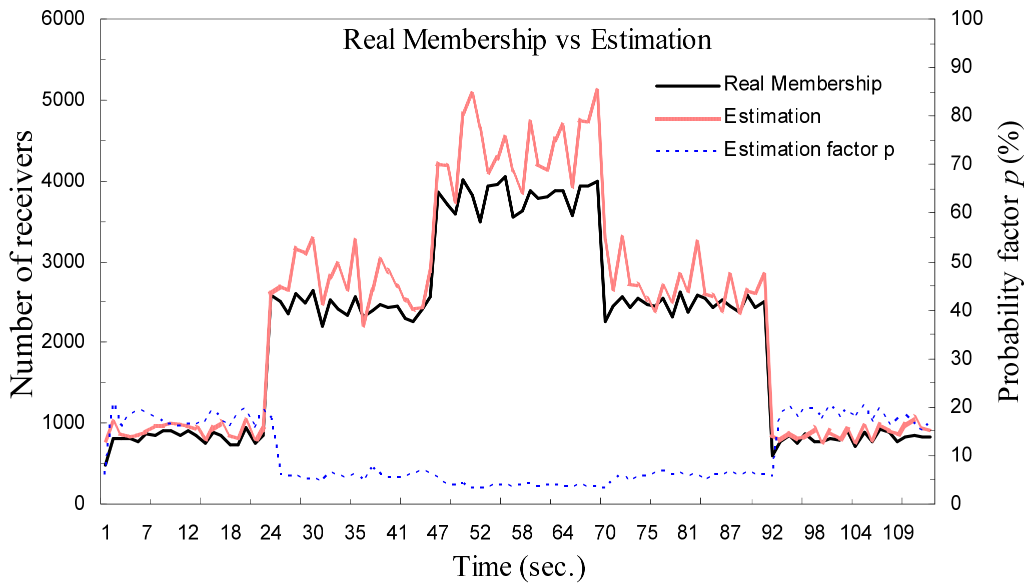

Figure 7.3: Comparison of Real and Estimate of dynamic multicast membership

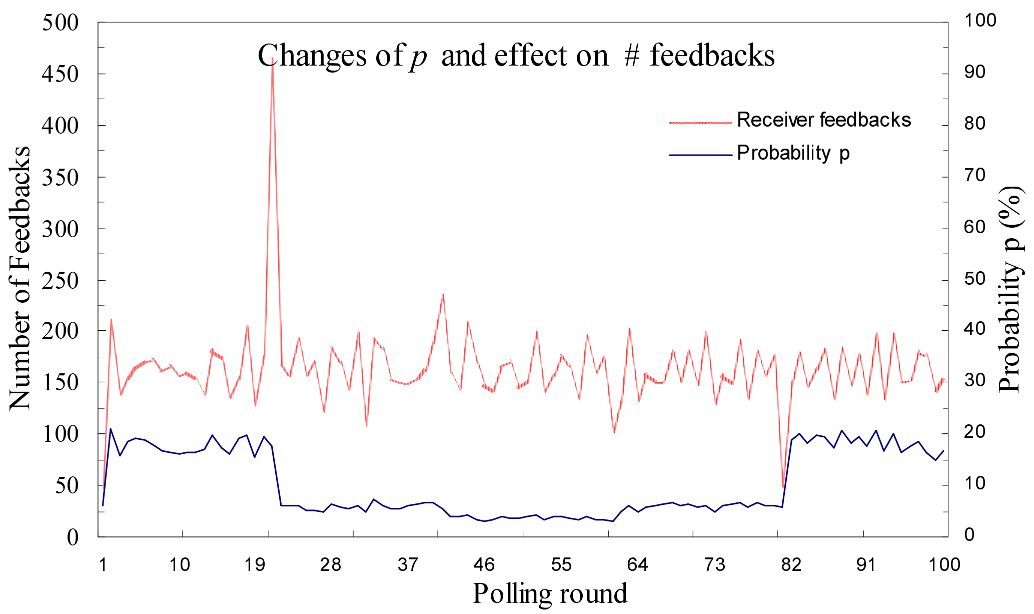

Figure 7.4: Dynamic changes of probability p, and the effect on number of feedbacks

460 feedbacks by the time the real membership slightly increased from around 1,000 receivers to about 2,600 receivers. But it also do the correction of p

right after that spike too. It proves the our method as soon as possible deliver the new p for next following polling round as correction whenever we detect the possible feedback implosion.

While in Figure 7.5, we can see the observation parameter T is also dynamically changed based on receivers response time. It fluctuates during the all polling rounds. If we refer in details, it has average estimate convergence time about 1.14 seconds and its standard deviation is very small about 37.32 milliseconds. So our method could drive the estimator to have granularity of estimation convergence in the order around 1 seconds, which is consider much fast if we compare to the problem posed by existing standard like in RTP/RTCP we discussed before in Chapter 2.

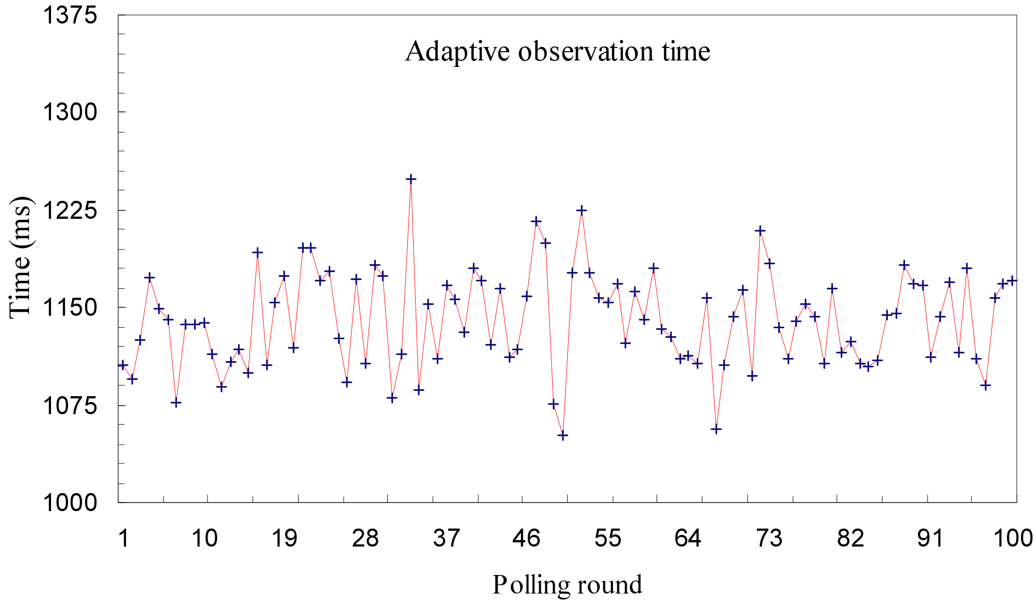

Figure 7.5: Dynamic changes of observation time, T

By extracting the previous results we present the analysis in Figure 7.6. It describes the estimation error at each polling round during the experiment. We found that the estimation errors were quite high especially during polling rounds right after the initialization period and also when the size of multicast group was increasing or decreasing very slightly. But our method did the correction soon after that for the next polling round. At a condition where the dynamic multicast group size was not so slightly changes, the estimation errors could go below 10% or even 0% as we rely on the data.

We also extracted the data presented in Figure 7.7 to analyze the feedback throughput rate received at sender side. From the point of view of feedback traffic it is quite in line with the amount of feedbacks, but it is around 68.5 kbit/s in average, while at the increasing membership, it raises to around

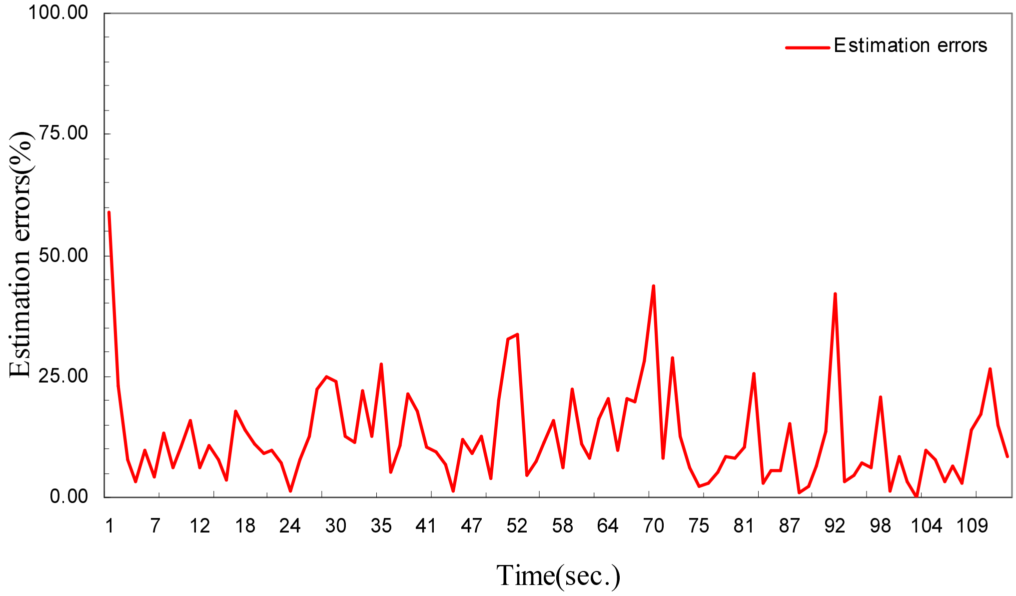

Figure 7.6: Estimation errors on every polling round

168 kbit/s. At the time the membership slightly decreasing it falls to around 20 kbit/s. So suppose we assume we have 128 kbit/s multicast session at sender, then the possible feedback throughput rate is about a half of it session bandwidth.

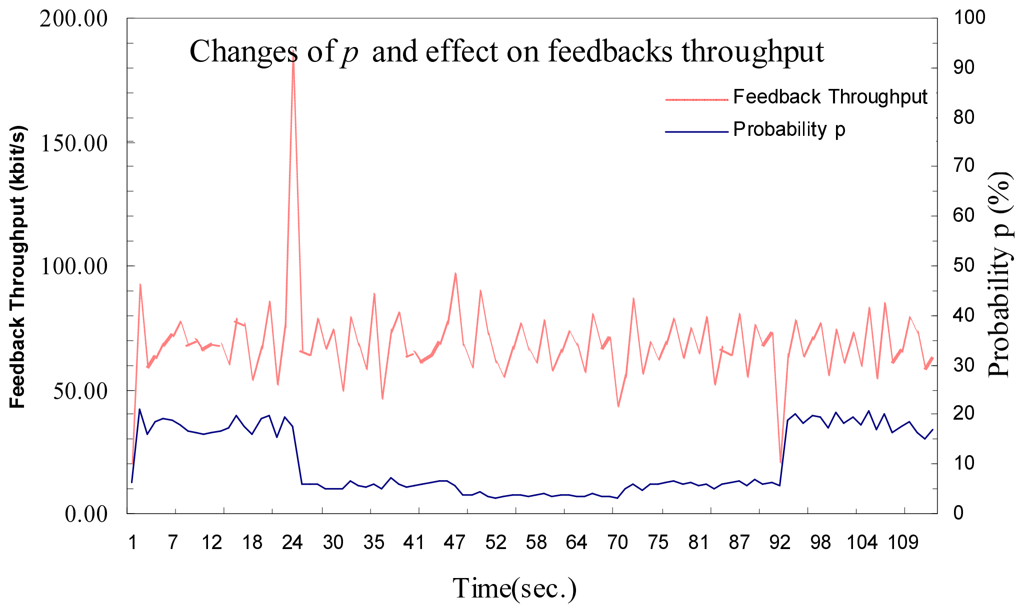

Figure 7.7: Feedback throughput rate received at sender side

7.4 Summary

In summarizing this Chapter, we conclude that our method successfully implemented in a PlanetLab testbed. The result shows indication that the adaptive method could alleviates feedback implosion. It reacts as soon as possible they detect the increased of expected feedbacks and delivers the new p to avoid possible feedback implosion while in moving forward in the estimation polling rounds. It can scales up to thousands receivers membership, while still try to keep the amount of feedbacks at the targeted threshold that we defined. From the point of view accuracy, in general, it still posses a high estimation error. It indicates we need to improve the quality of estimator. Other important point is the estimate convergence time was really fast and successfully set the observation time based on the receivers response time and have the good time granularity to follow the dynamic multicast group size.

Chapter 8

Conclusion and Future Work

We conclude our work in this chapter followed by consideration for future work.

8.1 Conclusion

We have discussed how the polling-based technique could contribute to the estimation of multicast group size. It will be useful if applied in such area of real-time live distance learning system or even global live events. We define the approach to achieve the goal on having a good balance of scalability, estimation accuracy and the estimate convergence time for monitoring method of multicast group size in large networks.

We use network simulator to study on how the method would performed with various network scenarios to predict the scalability and the accuracy. We found that for membership numbers below or equal 100 receivers, the estimator needs to use highest probability, $p = 100\%$ to achieve higher estimation quality. For the large numbers, we apply the proportion based on possible estimation error and the numbers of the feedbacks to avoid feedback implosion and also maintain the accuracy. For the estimate convergence time, we use the statistical calculation from receivers response time to the polling request. It then applies for the observation time of next polling round. From this we can have better observation time that adapts to the network condition.

We also verify our implementation in PlanetLab testbed to characterize the method in more real life network condition and also larger number of receivers. We found that our method could scale up to thousands number of receivers and controlling the number of expected feedbacks. The estimation convergence time is very fast and we can have good time granularity of monitoring the dynamic of multicast group size membership. It can then sufficient to be implemented in real-time distance learning application system running on top of IP Multicast to estimate the numbers of participating receivers.

We conclude that the method we proposed could adaptively adjust to the optimal estimation parameters by the threshold of expected amount of feedbacks. It proves that the method could scale up to thousands receivers while avoiding the feedback implosion. The estimation convergence time is fast enough since it does not need to delay the receiver's responses.

8.2 Future Work

The status of our estimation method has successfully implemented in research environment platform by PlanetLab. Throughout our simulation study, we found lower probability p value prone to higher estimation error because of insufficient number of samples successfully coming to sender node for estimation. But for larger multicast group size it behaves quite well since the number of feedback samples are adequate to do statistical sampling calculation.

The result of implementation in PlanetLab testbed does not really exhibit what we expected in term of estimation accuracy. The analysis done in Chapter 7 concludes that our method does not behave well in a condition that the multicast group size goes increasing or decreasing very slightly. For sure this issues need to be addressed in near future for further refinement of the method.

Throughout this paper we use the naïve estimator as our main estimator. Complemented with our method, it gives significant quality in following the dynamic multicast group size. But it is also proved analytically becomes very noisy. In near future we would like to study the other possible refinements from the point of view of mathematical analysis. Validation of the estimator in more real environment is also important to find the good and realistic estimator that has better estimation noise filtering. Together with the other above issues are the consideration we would like to include in our near future research.

Acknowledgments

First and foremost, I would like to thank Professor Jun Murai, Professor Osamu Nakamura, Associate Professor Hiroyuki Kusumoto, and Assistant Professor Hitoshi Asaeda for their insight guidance and advice throughout the process on completing the master thesis.

I would like to express deeper thank to Professor Keiko Okawa, Dr. Achmad Husni Thamrin, Dr. Shoko Mikawa, Patcharee Basu, Kotaro Kataoka and Haruhito Watanabe for their great support since my first arrival in Japan. Special thanks to all faculty members of Murai Laboratory for their insight research questions that lead me towards this completion and also a very generous help from Kazuhisa Matsuzono and Kazuhiro Mishima. This is not yet even the end, I want to continue this research further.

A special thanks addressed to all members of the SOI-Asia and AI3 Project and INSAT for their wonderful supports. I thank to my friends for sharing pleasure and pain on completing master thesis; Yuki Oyabu, Masayoshi Mizutani, Kosuke Osaka and Yuichi Nakamura. You are all have excellent capability that could make your life even brighter. Not forget to express gratitude too Indonesian students who study at Keio University; Dikshie, Agung, Fitriana, Setyo, Salim, Dadan, Alfi and Fithra, a Keio SFC faculty member Totok Suhardiyanto and others who could not all to be mentioned in here for their great companion during my study.

Finally, I would like to express a very deep thanks to whom play important role in my life; my beautiful wife Iva Rusdiana, lovely daughter Putri Zahratul Wahida, my parents, my parents in law and younger sister - brother for keeping me in high motivation to move forward and never give up.

January 2008, Japan.

Achmad Basuki

Bibliography

[1] S. Deering, "Host extensions for IP multicasting." RFC 1112 (Standard), Aug. 1989. Updated by RFC 2236.

[2] H. Holbrook and B. Cain, "Source-Specific Multicast for IP." RFC 4607 (Proposed Standard), Aug. 2006.

[3] K. C. Almeroth, "A long-term analysis of growth and usage patterns in the multicast backbone (MBone)," in *INFOCOM*, pp. 824–833, 2000.

[4] M. Eubanks, "Measures of the Multicast Enabled Internet," last accessed in Dec 2007. Available at http://www.multicasttech.com/status/.

[5] S. Mikawa, K. Okawa, and J. Murai, "Establishment of a lecture environment using internet technology via satellite communication in asian countries," in *2003 Symposium on Applications and the Internet Workshops (SAINT 2003), 27-31 January. Proceedings*, (Orlando, FL, USA), pp. 60–66, IEEE Computer Society, 2003.

[6] P. Basu, A. H. Thamrin, S. Mikawa, K. Okawa, and J. Murai, "Internet technologies and infrastructure for asia-wide distance education," in *2007 International Symposium on Applications and the Internet (SAINT 2007), 15-19 January. Proceedings*, (Hiroshima, Japan), p. 3, IEEE Computer Society, 2007.

[7] OSTN, "Open Student Television Network," last accessed in Dec 2007. Available at http://www.ostn.tv/.

[8] S. Floyd, V. Jacobson, S. McCanne, C.-G. Liu, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," in *SIGCOMM '95: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, (Cambridge, Massachusetts, USA), pp. 342–356, ACM, 1995.

[9] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications." RFC 3550 (Standard), July 2003.

[10] A. Dutta, H. Schulzrinne, and Y. Yemini, “MarconiNet - An Architecture for Internet Radio and TV Networks,” in *Proceeding of NOSSDAV’99*, (Basking Ridge, New Jersey, USA), June 1999.

[11] J.-C. Bolot, T. Turletti, and I. Wakeman, “Scalable feedback control for multicast video distribution in the Internet,” in *SIGCOMM ’94: Proceedings of the conference on Communications architectures, protocols and applications, August 31 - September 02*, (London, United Kingdom), pp. 58–67, 1994.

[12] M. Grossglauser, “Optimal deterministic timeouts for reliable scalable multicast,” *INFOCOM ’96. 15th Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, vol. 3, pp. 1425–1432 vol.3, 24-28 Mar 1996.

[13] J. Nonnenmacher and E. W. Biersack, “Scalable feedback for large groups,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 375–386, 1999.

[14] J. C.-H. Lin and S. Paul, “RMTP: A reliable multicast transport protocol,” in *INFOCOM ’96. 15th Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, pp. 1414–1424, 1996.

[15] C. Papadopoulos, G. Parulkar, and G. Varghese, “Light-weight multicast services (LMS): a router-assisted scheme for reliable multicast,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 456–468, 2004.

[16] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, “Deployment issues for the IP multicast service and architecture,” *Network, IEEE*, vol. 14, no. 1, pp. 78–88, Jan/Feb 2000.

[17] 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. Sumanasekera, and L. Vicisano, “PGM Reliable Transport Protocol Specification.” RFC 3208 (Experimental), Dec. 2001.

[18] R. Yavatkar, J. Griffoen, and M. Sudan, “A reliable dissemination protocol for interactive collaborative applications,” in *MULTIMEDIA ’95: Proceedings of the third ACM international conference on Multimedia*, (San Francisco, California, USA), pp. 333–344, ACM, 1995.

[19] H. W. Holbrook, S. K. Singhal, and D. R. Cheriton, "Log-based receiver-reliable multicast for distributed interactive simulation," in *SIGCOMM '95: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, (Cambridge, Massachusetts, USA), pp. 328–341, ACM, 1995.

[20] D. DeLucia and K. Obraczka, "Multicast feedback suppression using representatives," in *INFOCOM '97. 16th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE.*, vol. 2, (Kobe, Japan), pp. 463–470, 1997.

[21] T. Friedman and D. F. Towsley, "Multicast session membership size estimation," in *INFOCOM '99. 18th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, pp. 965–972, 1999.

[22] F. Filali, H. Asaeda, and W. Dabbous, "Counting the number of members in multicast communication," in *Proceedings of Networked Group Communication, Fourth International COST264 Workshop, October 23-25*, (Boston, MA, USA), pp. 63–70, ACM, 2002.

[23] J. Chesterfield and E. Schooler, "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback." Internet Draft, Nov 2007. draft-ietf-avt-rtcpssm-14.txt.

[24] D. Dolev, O. Mokryn, and Y. Shavitt, "On multicast trees: structure and size estimation," *IEEE/ACM Transactions on Networking*, vol. 14, no. 3, pp. 557–567, 2006.

[25] C. Liu and J. Nonnenmacher, "Broadcast audience estimation," in *INFOCOM 2000. 19th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, pp. 952–960, 2000.

[26] S. Alouf, E. Altman, C. Barakat, and P. Nain, "Estimating membership in a multicast session," in *SIGMETRICS '03: Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, (San Diego, CA, USA), pp. 250–260, ACM, 2003.

[27] VIC, "Videoconferencing Tool," last accessed in Dec 2007. Available at http://mediatools.cs.ucl.ac.uk/nets/mmedia/wiki/VicWiki.

[28] RAT, "Robust Audio Tool," last accessed in Dec 2007. Available at http://mediatools.cs.ucl.ac.uk/nets/mmedia/wiki/RatWiki.

[29] VLC, "VideoLAN media player," last accessed in Dec 2007. Available at http://www.videolan.org/vlc/.

[30] DVTS, "Digital Video over IP," last accessed in Dec 2007. Available at http://www.sfc.wide.ad.jp/DVTS.

[31] SOI-Asia, "School on Internet Asia Project," last accessed in Dec 2007. Available at http://www.soi.wide.ad.jp/soi-asia/.

[32] K. C. Almeroth and M. H. Ammar, "Collecting and modeling the join/leave behavior of multicast group members in the MBone," in *HPDC '96: 5th IEEE International Symposium on High Performance Distributed Computing. Proceedings IEEE*, (Syracuse, New York, USA), pp. 209–216, IEEE Computer Society, 1996.

[33] NS-2, "The network simulator," last accessed in Dec 2007. Available at http://www.isi.edu/nsnam/ns/.

[34] BRITE, "An Approach Towards Universal Topology Generation," last accessed in Dec 2007. Available at http://www.cs.bu.edu/brite/.

[35] D. Waitzman, C. Partridge, and S. Deering, "Distance Vector Multicast Routing Protocol." RFC 1075 (Experimental), Nov. 1988.

[36] PlanetLab, "An open platform for developing, deploying, and accessing planetary-scale services," last accessed in Dec 2007. Available at http://www.planet-lab.org/.