Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-avt-rtpsample-00.ps

Audio/Video Transport wg
J. Rosenberg, H. Schulzrinne
Bell Laboratories/Columbia U.

August 1, 1998
Expires: February 1, 1999

Issues with RTP Sampling

Status of this Memo

1 Abstract

In order to control the flow of RTCP packets in large multicast groups, session participants are required to transmit their packets with a period proportional to the group size. Obtaining a group size estimate generally requires a participant to maintain a list of group members. As this can require significant memory, particularly for embedded systems, RTP has been revised to allow for a statistical sampling procedure which allows the memory size to be bounded for all group sizes. However, this sampling algorithm can interact with other aspects of RTP. In particular, we have identified three problems. First, RTP sampling has an adverse affect on the performance of BYE reconsideration. Secondly, it can cause inaccurate estimates with dynamic group sizes that decrease rapidly. Thirdly, the current SSRC sampling algorithm grossly overestimates the group size when there are a few senders. We discuss these problems in detail and present simple solutions.

2 Introduction

In order to control the flow of RTCP packets in large multicast groups, session participants are required to transmit their packets with a period proportional to the group size. Obtaining a group size estimate generally requires a participant to maintain a list of group members. As this can require significant memory, particularly for embedded systems, RTP [1] has been revised to allow for a statistical sampling procedure which allows the memory size to be bounded for all group sizes.

This algorithm operates by applying an SSRC mask with m bits equal to one to each packet received. Initially, m starts at 0. If the bitwise AND of the SSRC of the incoming packet, and the mask, equals the bitwise AND of some key and the mask, the packet is accepted and the SSRC counted in the group size estimate. When the memory requirements for the list reach some threshold B , m is incremented. Those SSRC in the list which no longer equal the key under the masking operation are discarded. If there are n SSRC in the table, the group size estimate is equal to $n * 2^m$.

This sampling algorithm can interact with other aspects of RTP. In particular, we have identified two problems. First, RTP sampling has an adverse affect on the performance of BYE reconsideration, and secondly, it can cause inaccurate estimates with dynamic group sizes that decrease rapidly. We discuss these problems in detail and present simple solutions.

3 Interaction with Reconsideration

The impact of the sampling algorithms is to effectively dampen the changes in the membership table. Changes occur less frequently, but with greater amplitudes. Consider the case where at some receiver, the mask is 3 bits, and a surge of 16 new members join the group, all over a 16 second interval. These new members will send RTCP packets, but only one in 8 of them will be seen by the sampling member ($2^3 = 8$). If we assume that the 16 members each send their first RTCP packets uniformly over the 16 second interval, it will take the sampling member 8 seconds, on average, to see one of them. When it does, its group size estimate jumps by 8. For a non-sampling member, its group size estimate increases smoothly by 16 over the 16 second interval.

Thus, the sampling algorithms tend to cause members to see group size changes in bursts instead of smoothly. Furthermore, the amount of time it takes to see a change increases. In the previous example, the sampling member had to wait 8 seconds to see a change, whereas the non-sampling member had to wait just 1 second. Its almost as if the network delay for a sampling member increases. It is this increase in delay which is troublesome. The reconsideration algorithms (both forward, BYE, and reverse) in the RTP specification operate well in cases where the network delays are small in comparison to the transmission intervals. Since the sampling algorithm increases the effective delay, the performance of the algorithms may be worse. We therefore consider each in turn.

3.1 Forward Reconsideration

Fortunately, forward reconsideration is not generally affected by the sampling algorithms. This is because forward reconsideration is effective in cases where a large number of users simultaneously join a multicast group. When these members join, each of them has an initial group size estimate of 1. As a result, each of them should have sampling off initially ($m = 0$). By the time enough of a group size estimate has been obtained to require sampling, the impact of reconsideration has already worn off.

We can demonstrate this analytically as follows. We have postulated that the impact of reconsideration is small when the ratio of the tranmission interval at a sender, and the network delays, is large [2]. After the initial spike of packets in forward reconsideration, packets are sent at a rate of $1/(1-alpha)C$, where alpha is .5, and C represents the average RTCP packet size divided by 5 percent of the session bandwidth. With sampling enabled, this flow of packets will appear as if 2^m packets arrive instantaneously every $2^m(1-alpha)C$ seconds (same average rate, but different burstiness). Thus, the effective network delay for sampling members is $2^m(1-alpha)C$. The period of packet transmissions from a group member is, on average, $n * 2^m * C$. Thus, the quotient of these represents the interval/delay value. This quotient is $n/(1-alpha)$. Here, n represents the number of entries in the sampled SSRC table. In the worst case, this quantity should be half the size of the memory available. This is because the sample mask is increased when the memory fills, resulting in a discard of half the contents of the table. If the table filled the memory before the sample

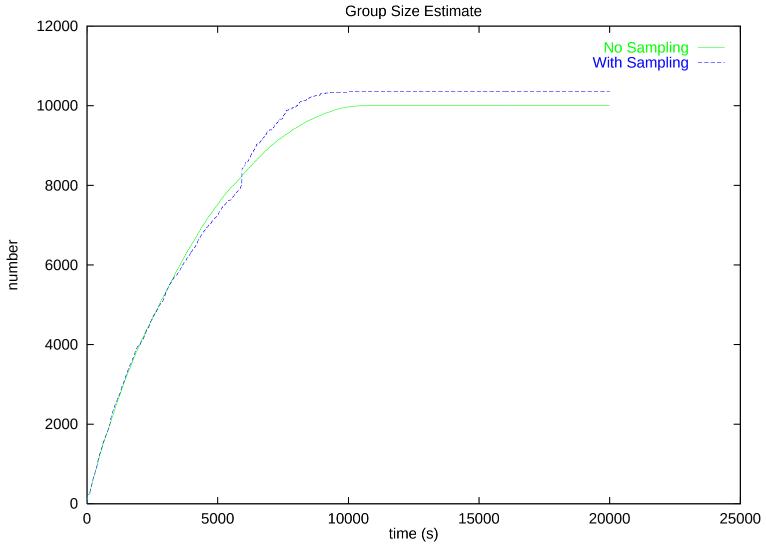

Figure 1: Group Size Estimate

increase, it will occupy half of the memory afterwards. As the specification recommends that B (the mimimum memory size) should be 100 SSRC values, the mimimum value of the quotient is 100. This is sufficient to have no impact on forward reconsideration.

These results are verified by simulation in Figures 1 and 2 (present in postscript version only). Figure 1 depicts the group size estimate as seen by a single member in a session where 10,000 users simultaneously join the group at time 0. All session members implement unconditional reconsideration. The figure depicts two curves, one where the all users sample, and another where they do not. Note that the sampling algorithm performs relatively well. Figure 2 depicts the cumulative number of RTCP packets sent to the multicast group across all users. Also note that the sampling algorithm has little impact on the RTCP traffic.

3.2 Reverse Reconsideration

Reverse reconsideration is a minor optimization that allows a session participant to more rapidly decrease its transmission interval when the group size decreases. It does this by “moving in” the transmission time of the next packet when a BYE or timeout occurs.

When used in conjunction with SSRC sampling, the net effect is as if BYE’s were delayed and occurred in greater bursts. The impact on performance of reverse reconsideration is the same as with forward. So long as the delay increases are small compared with the transmission period, little performance degradation should occur. The previous section has demonstrated that this is the case.

3.3 BYE Reconsideration

Unfortunately, SSRC sampling can have a serious impact on BYE reconsideration, depending on how the algorithm in RTP is interpreted. When a large number of users leave the group, they initialize a variable called members to 0. As BYE packets are received, the members count is incre-

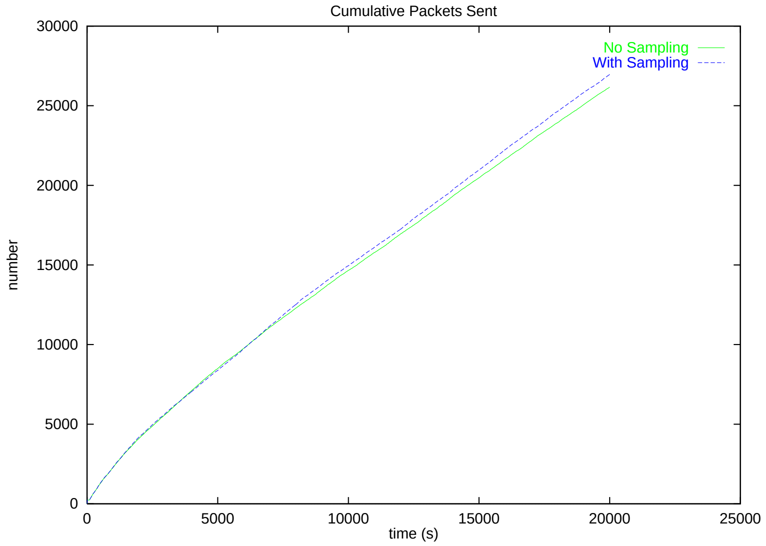

Figure 2: Cumulative Packets Sent

mented. A user sends a BYE packet according to the standard forward reconsideration algorithm, but using the variable members as a group size estimate.

If the SSRC sampling algorithm is applied against the incrementing of the members variable (in other words, the variable is increased by 2^m when a BYE matching the mask is received), BYE reconsideration can fail. Consider the case where a large number of users simultaneously leave the group at t=0. All initialize their members variables to 0. Assume further that all are applying an SSRC sampling algorithm with a mask of m bits. The implication is that none of them will increase their members counter until, on average, 2^m packets are received. At that point, the member counter jumps rapidly to 2^m, pushing off further BYE packet transmissions substantially. However, until this happens, a large number of BYE packets may have already been transmitted.

An analytical treatment of the impact of the problem is quite complex. However, the effect is demonstrated via simulations. Figure 3 depicts the cumulative number of packets sent when 10,000 users simultaneously join at t=0, and all but one leave at t=10,000. Note that when sampling is in use, there is a spike of BYE packets, even with BYE reconsideration, when the users all leave. The spike is not present under normal operation.

The fix to this problem is fairly simple. The SSRC sampling mask must not be applied to counting BYE packets after a user leaves. For every BYE packet received, the counter is incremented, no matter what kind of sampling is in use.

This means that when a BYE packet is received, it should be counted even if it doesn't appear in the membership table. As long as each user that leaves sends at most one BYE packet, there are no problems. However, the implication is that a malicious user sending multiple BYE packets can cause other users to see an abnormally high number of users leaving. The result is that correctly behaving users will take longer to send their BYE. Fortunately, this will cause a reduction in the volume of BYE traffic, not an increase. This overestimation of the leaving user count is therefore safe as far as network traffic is concerned.

A user who is not implementing sampling can check if a user is in the membership list before accepting their BYE packet and incrementing the counter.

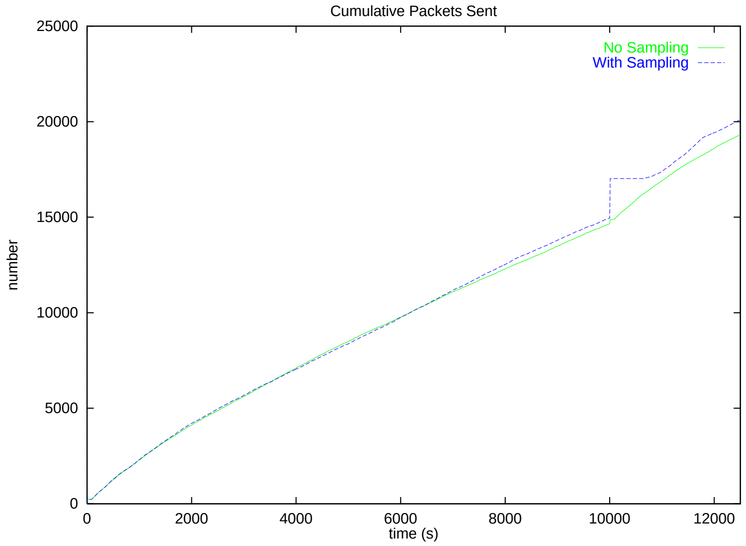

Figure 3: Interaction of BYE Reconsideration and Sampling

Once this fix is applied, the BYE spike problem disappears. This is demonstrated in Figure 4. The figure depicts the cumulative number of packets sent when 10,000 users simultaneously join at t=0, and all but one leave at t=10,000s. Notice that there are no spikes when the sampling is not applied to BYE packets.

We therefore propose that the text in section 6.3.7 be modified, so that the second bullet item reads:

"Every time a BYE packet from another user is received, members is incremented by 1. members is NOT incremented when other RTCP packets or RTP packets are received, but only for BYE packets. The variable members is incremented by 1 even if SSRC sampling is in use, and the SSRC does not match the SSRC of any current group member in the subsampled list."

4 Dynamic Group Sampling

With dynamic groups, it is possible for a large number of users to join the group, and then for a sizeable portion to later leave. Those members which remain throughout may make use of SSRC sampling. In this case, as the other group members leave, the number of entries in the sampled membership table may become small. The result is that the group size estimate may become extremely inaccurate. To combat this problem, it is desirable to compensate by decreasing the number of bits in the sample table when the memory usage falls below some threshold.

When the number of bits in the mask increases, the user compensates by removing those SSRC which no longer match. When the number of bits decreases, the user should theoretically add back those users whose SSRC now match. However, these SSRC are not known, since the whole point of sampling was to not have to remember them. Therefore, if the number of bits in the mask is just reduced without any changes in the membership table, the group estimate will instantly drop by exactly half.

To compensate for this, some kind of algorithm compensation is needed. Initially, the use of

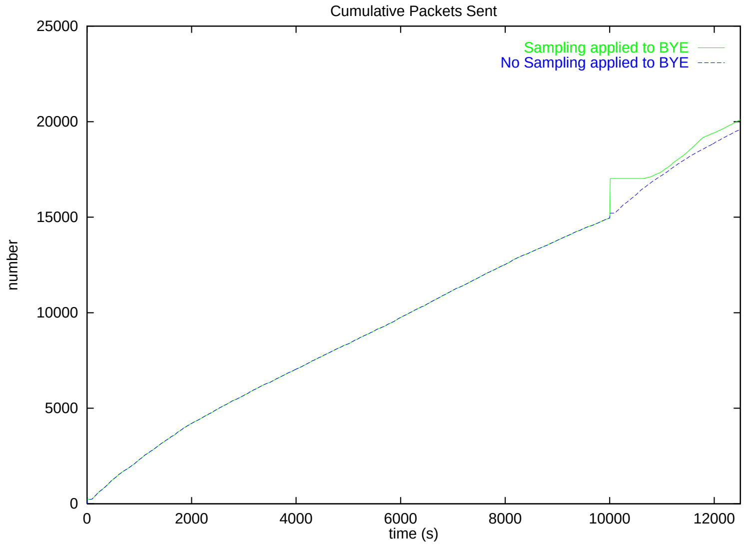

Figure 4: BYE Reconsideration Fix

corrective "fudge factors" was proposed - both an additive and a multiplicative. We have also devised a binning based mechanism which performs better than the corrective factors.

4.1 Corrective Factors

The idea with the corrective factors is to add in or multiply the group size estimate by some corrective factor to compensate for the change in sample mask. The corrective factors should decay as the fudged members are eventually learned about and actually placed in the membership list.

The multiplicative factor starts at 2, and gradually decays to one. The additive factor starts at the difference between the group size estimate before and after the number of bits in the mask is reduced, and decays to 0 (this is not always half the group size estimate, as the corrective factors can be compounded, see below). Both factors decay over a time of $c * L(t_s)$, where c is the average RTCP packet size divided by 5% of the session bandwidth, and $L(t_s)$ is the group size estimate just before the change in the number of bits in the mask at time t_s. The reason for this constant is as follows. In the case where the actual group membership has not changed, those members which were forgotten will still be sending RTCP packets. The amount of time it will take to hear an RTCP packet from each of them is the average RTCP interval, which is $cL(t_s)$. Therefore, by $cL(t_s)$ seconds after the change in the mask, those users who were fudged by the corrective factor should have sent a packet and thus appear in the table. We chose to decay both functions linearly. This is because the rate of arrival of RTCP packets is linear.

What happens if the number of bits in the mask is reduced once again before the previous corrective factor has expired? In that case, we *compound* the factors by using yet another one. Let $f_i()$ represent the ith correction function. If t_s is the time when the number of bits in the mask is reduced, we can describe the additive correction factor as:

$$f_i(t) = \begin{cases} 0 & t < t_s \\ (L(t_s^-) - L(t_s^+))\,\frac{t_s + cL(t_s^-) - t}{cL(t_s^-)} & t_s < t < t_s + cL(t_s^-) \\ 0 & t > t_s + cL(t_s^-) \end{cases} \quad (1)$$

and the multiplicative factor as:

$$f_i(t) = \begin{cases} 1 & t < t_s \\ \frac{t_s + 2cL(t_s^-) - t}{cL(t_s^-)} & t_s < t < t_s + cL(t_s^-) \\ 1 & t > t_s + cL(t_s^-) \end{cases} \quad (2)$$

Note that in these equations, $L(t)$ denotes the group size estimate obtained including the corrective factors except for the new factor.

Finally, the actual group size estimate $L(t)$ is given by:

$$L(t) = \sum f_i(t) + N2^m \quad (3)$$

for the additive factor, and:

$$L(t) = N2^m \prod f_i(t) \quad (4)$$

for the multiplicative factor.

Our simulations showed that both algorithms performed equally well, but both tended to seriously underestimate the group size when the group membership was rapidly declining. This is demonstrated in the performance curves below.

4.2 Binning Algorithm

In order to more correctly estimate the group size even when it was rapidly decreasing, we developed a binning based algorithm. The algorithm works as follows. There are 32 bins, same as the number of bits in the sample mask. When an RTCP packet from a new user arrives whose SSRC matches the key under the masking operation, it is placed in the bin with the current value of the mask, and otherwise is discarded.

When the number of bits in the mask is to be increased, those members in the bin who still match after the new mask are moved into the next higher bin. Those who don't match are discarded. When the number of bits in the mask is to be decreased, nothing is done. Users in the various bins stay where they are. However, when an RTCP packet for a user shows up, and the user is in a bin with a higher value than the current number of bits in the mask, it is moved into the bin corresponding to the current number of bits in the mask. Finally, the group size estimate L(t) is obtained by:

$$L(t) = \sum_{i=0}^{i=31} B(i)2^i \quad (5)$$

Where $B(i)$ are the number of users in the ith bin.

The algorithm works by basically keeping the old estimate when the number of bits in the mask drops. As users arrive, they are gradually moved into the lower bin, reducing the amount that the

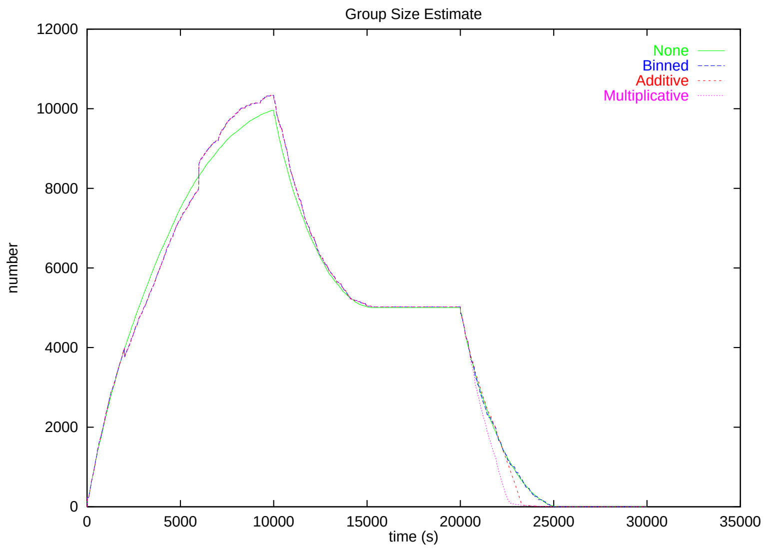

Figure 5: Correction Mechanisms

higher bin contributes to the total estimate. However, the old estimate is still updated in the sense that users which timeout are removed from the higher bin, and users who send BYE packets are also removed from the higher bin. This allows the older estimate to still adapt, while gradually phasing it out. It is this adaptation which makes it perform much better than the corrective algorithms. The algorithm is also extremely simple.

4.3 Comparison

The algorithms are all compared via simulation in Figure 5. In the simulation, 10,001 users join a group at t=0. At t=10,000, 5000 of them leave. At t=20,000, another 5000 leave. All implement an SSRC sampling algorithm, unconditional forward and BYE reconsideration. The graph depicts the group size estimate as seen by the single user present throughout the entire session. In the simulation, a memory size of 1000 SSRC was assumed. The performance without sampling, and with sampling with the additive, corrective, and bin-based correction are depicted.

As the graph shows, the bin based algorithm performs particularly well at capturing the group size estimate towards the tail end of the simulation.

4.4 Sender Sampling

The current text mandates that senders always be kept in the list, even when their SSRC do not match:

"Note that this sampling algorithm MUST NOT be applied to SSRC identifiers that correspond to senders because otherwise the calculation of the RTCP bandwidth when we_sent is true would be inaccurate. The SSRC identifiers for senders MUST always be added to the table when first received and not removed from the table when the mask is extended."

We have observed that this can cause overstimations in the group estimate when the number of senders is even a small percentage of the total group size. This is easily demonstrated analytically.

Let N_s be the number of senders, and N_r be the number of receivers. The membership table will contain all N_s senders and $1/2^m$ of the receivers. The total group size estimate in the current draft is obtained by 2^m times the number of entries in the table. Therefore, the group size estimate becomes:

$$L(t) = 2^m N_s + N_r \tag{6}$$

which exponentially weights the senders.

This is easily compensated for in the binning algorithm. A sender is always placed in the 0th bin. When a sender becomes a receiver, it is moved into the bin corresponding to the current value of m, if its SSRC matches the key under the masked comparison operation.

4.5 Proposed Text

Based on these results, we propose that the following text be used in place of paragraph 6 and 7 of section 6.3.3:

"The group size estimate is then computed according to Annex A.9"

Paragraph 2 of Section 6.3.4 should be stricken.

Paragraph 3 of Section 6.3.5 should be stricken.

We also propose that Annex A.9 be added:

Annex A.9 SSRC Sampling Mechanisms

When the group memberships for large sessions becomes substantial, a group member may find that the storage requirements for storing the SSRC listing may be excessive. A session participant MAY apply SSRC sampling, as described here, to reduce the storage requirements. A session participant MAY use any other algorithm with similar performance. A key requirement is that any algorithm considered SHOULD NOT substantially underestimate the group size, although the MAY overestimate.

The algorithm operates by applying a mask with m bits to the SSRC of each member. If the SSRC matches the SSRC of the sampling user under the masking operation, the member is added to the table, otherwise they are ignored. Initially, the mask starts with m=0, so that every SSRC is accepted and placed into the table. When the storage requirements reach some threshold, B, the member increases m by 1 bit, and discards all SSRC in the table which no longer match under the mask. This will reduce the table size by roughly half. As the group size continues to increase, the user MAY further increase the mask size by an additional bit when the table size once again approaches the threshold. A client MUST maintain a table which can accomodate at least B=100 users, for reasonable statistical accuracy. Members who are senders MUST be maintained in the table, even when their SSRC does not match under the masking operation.

Each user also maintains a set of 32 bins, numbered 0 through 31. When a new user shows up whose SSRC matches the key under the current mask (with m bits), the user is placed in bin m. Additionally, when a the mask is increased from m to m+1 bits, all users who still match under the m+1 bit mask are moved from bin m to bin m+1. Note, however, that users who are senders are always placed in the 0th bin. When a sender stops sending, it is placed in the bin corresponding to the current mask length m if its SSRC matches the key under the masking operation, otherwise it is discarded.

Just as the group size may grow, the group size may decrease. In these cases, the number of entries in the table may become too small to provide a reasonable statistical estimate. At such times, it may become necessary to decrease the number of bits in the mask. If the group size estimate is L, and there are m bits in the mask, and the total amount of storage available for the table is B, it is recommended that the mask be decreased by one when:

$$L/2^m < B/4 \tag{7}$$

When the mask is reduced from m to m-1, any users in bins m or higher are NOT MOVED into the lower bins. They stay in their current bin.

When a packet arrives from a user who is currently in some bin x, and the number of bits in the mask is currently m, for some m < x, the user is then moved from bin x to bin m. When a user times out, or a BYE packet is received for it, and it exists in the membership table, it is removed from its bin. Note that senders are always in bin 0 and never move as the SSRC mask changes.

Finally, to obtain a group size estimate, the following formula is used:

L = SUM from i=0 to i=31 of B(i) * (2**i)

Where $B(i)$ is the number of users in the ith bin.

5 Conclusion

We have presented some issues related to the SSRC sampling algorithms present in the latest RTP draft. We have identified and fixed one problem related to the interaction of BYE reconsideration and sampling. We have also identified the need for corrective factors in the sampling algorithms, and presented and compared three algorithms for it. Based on performance and simplicity, we recommended that one of them, the bin sampling algorithm, be recommended.

References

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, Jan. 1996.

[2] J. Rosenberg and H. Schulzrinne, "Timer reconsideration for enhanced RTP scalability," (San Francisco, California), March/April 1998.

6 Full Copyright Statement

7 Authors' Addresses

Jonathan Rosenberg
Bell Laboratories, Lucent Technologies
101 Crawfords Corner Rd.
Holmdel, NJ 07733
Rm. 4C-526
email: jdrosen@bell-labs.com

Henning Schulzrinne
Columbia University
M/S 0401
1214 Amsterdam Ave.
New York, NY 10027-7003
email: schulzrinne@cs.columbia.edu