PERFORMANCE ANALYSIS ON MULTICAST FEEDBACK CONTROL WITH UNSYNCHRONIZED CLOCKS

Shuju Wu, Central Connecticut State University; Xiaobing Hou, Central Connecticut State University

Abstract

To ensure reliable multicast delivery, packet loss needs to be reported to the sender by sending feedback messages. Feedback implosion problems arise when too many multicast group members generate and send duplicate feedback messages which overwhelm the sender processing capability or cause network congestion near the multicast sender. Proposed feedback control mechanisms use timers, representative or network layer support to reduce the number of feedback messages. Comparative studies show that representative schemes with timers provide better feedback suppression performance where the timer length is determined by assuming synchronized clocks among the group members. However, synchronized clocks may not be available network-wide and there is a systematic synchronization bias when link delays between the time client and the time server are asymmetric. This paper extends the study of previous representative schemes and analyzes their performance when unsynchronized clocks are used. A simulation study was performed in order to verify the effectiveness of suppression, and the number and the location of representatives were observed based on dynamic network situations. The analysis and simulation results showed that the representative schemes with deterministic timers maintain good feedback suppression performance even if the clocks of multicast group members are unsynchronized.

Introduction

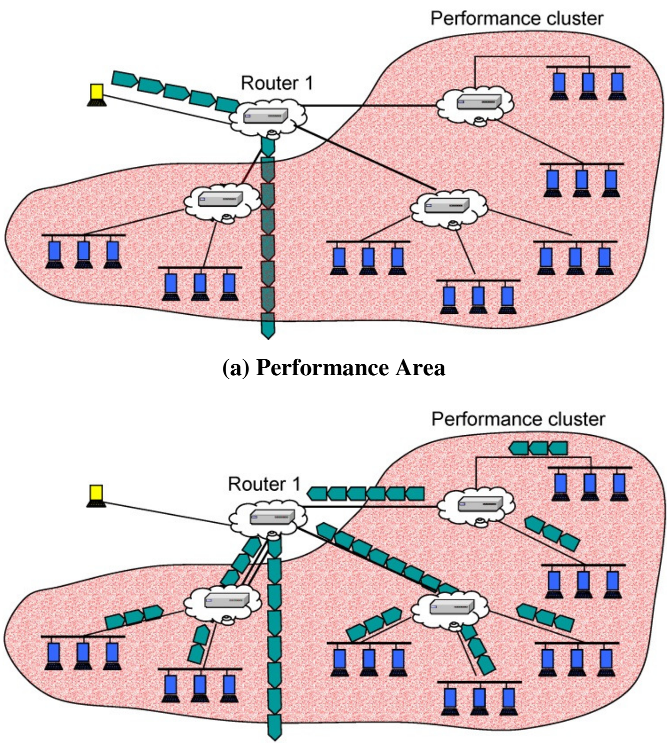
Communications using multicasting, be it IP- or overlay-based, attain efficiency over using unicasts by building an efficient multicast tree. However, faults in the Internet result in performance areas because group members share a multicast tree. Members in a performance area suffer from the same packet loss and, if the loss needs to be reported, duplicate feedback messages are sent by group members, which can overwhelm the sender processing capability and cause network congestion. This is called feedback implosion. For example, in Figure 1(a), router 1 keeps dropping data packets due to network congestion causing all of the group members in the same performance area behind it to suffer the same losses. The consequence is that all of the affected members will send the same feedback message and cause new congestion close to the source, as shown in Figure 1(b).

The Internet is dynamic and unpredictable in nature. Dynamic events such as changing node failures, link failures or network congestion can cause packet loss or faults. Faults resulting from network congestion in the Internet could last a long time and cause end-to-end performance degradation without a total loss of connection. This in turn generates a large number of feedback messages. Experiments on the MBone [1] show that even for a small multicast group of 11 members, each member experiences a very long consecutive loss of up to a few minutes, a loss that occurs in almost every trace. According to C'aceres et al. [2], link loss rates in a MBone group of 8 members are measured in one-hour long intervals and shown to vary between 2% and 35%. On a specific link, loss rates higher than 15% happen frequently and often last about 10 minutes. Also, from the results of Internet measurements [3-4], it is not unusual to find long-lasting, high-loss periods between Internet nodes, although the average loss rate over a day could be low. When such faults happen in a multicast tree and are close to the multicast source, the size of the performance cluster will be large which adversely affects most of the group members.

Feedback implosion is one of the issues that hinder the large-scale Internet implementation of IP multicast. Overlay multicast does not suffer this problem as the tree consists of unicast connections; rather, how to construct an efficient multicast tree using unicast connections is the key issue [5]. Many Internet communications can utilize multicast but are loss intolerant. Examples of such applications include video server replication and data dissemination (news, stock market quotes, collaborations, software distributions, etc.), in both wired and wireless networks [6-7]. Feedback implosion problems can impede the scalability and large-scale implementation of such applications. Therefore, feedback control is important for multicast communications.

Current feedback control mechanisms can be characterized by their use of timers, hierarchies or representatives. Timer-based mechanisms [8-10] require each member to wait a period of time, either random or deterministic, before it sends feedback. If a member receives another member's feedback before the timer expires, its feedback is suppressed. Hierarchy-based mechanisms [11-14] organize the group members into a domain hierarchy and restrict feedback messages within the local domain. Representative-based schemes [15-16] use representatives, members that can send feedback messages immediately, to suppress the other members' feedback.

It is well recognized that random timer-based approaches are simpler but less responsive and suppressive. The reason is that every feedback is delayed for a random period of time resulting in random suppression performance. Hierarchy-based schemes are more complex due to hierarchy management and utilize random timers. Using representatives to send immediate feedbacks improves the response and suppression performance. Wu et al. [17] have shown that feedback control schemes using representatives and deterministic timers provide superior overall performance.



(a) Performance Area

(b) Feedback implosion

Figure 1. Example of Feedback Implosion

Wu et al. further assumed that all of the group members have synchronized clocks to calculate the timer length. There are two concerns in assuming synchronized clocks. First, nowadays, many of the Internet end-hosts (used by end users) do not have the synchronized time service. Even though service and protocol such as Network Time Protocol NTP [18] is available, the end hosts may not be configured or have an NTP server to access. Second, in reality, there is a systematic synchronization bias when link delays between the time client and the time server are asymmetric. Since the feedback suppression timer length itself is related to network link delays, such a bias could severely affect suppression performance.

This study extends the work of Wu et al. by analyzing the feedback control performance when unsynchronized clocks are used. Simulation study is performed to verify the effectiveness of suppression in which the number and location of representatives are observed based on dynamic network fault situations.

The Representative-based Multicast Feedback Control Scheme

The work by Wu et al. used representatives and deterministic timers calculated on perfectly synchronized clocks for feedback control purposes. The proposed scheme is called Loss Pattern matched Area Based Feedback Control (LPABFC). The comparative study in the paper by Wu et al. [17] shows that LPABFC provides better performance than the other representative schemes such as REP [15] by adaptively changing the number of representatives and the locations of the representatives based on network fault situations. The REP scheme uses a constant number of representatives, which means that if the number of performance clusters is larger than the number of representatives, some clusters will have no representative to provide immediate feedback. A sensible solution is to dynamically identify the performance clusters and allow each cluster to have its own representative.

Here a few issues arise: first, how to accurately and dynamically identify and group the members into clusters without sacrificing simplicity; second, the number of representatives should be neither constant nor limited by a maximum number—ideally, it should be equal to the current number of performance clusters; third, before a representative is assigned to a cluster, suppression is still needed among the cluster members to prevent feedback implosion; fourth, operations by group members should be minimized—therefore, the members should not be involved in representative selection and management; fifth, interactions and cooperation among members normally incur overhead and increase the complexity—therefore, they should be minimized, too.

Unlike previous schemes that focus on timer selection, LPABFC tries to identify performance areas according to current fault conditions by performing loss pattern matching. Figure 2 shows the general model of feedback suppression in LPABFC. Each area has a representative (AR) that multicasts feedback messages without delay. Every receiver stores an *n*-bit bitmap indicating its receiving status of the most recent n data packets. For the purpose of loss pattern matching, the representative attaches its own n-bit loss pattern in the feedback message, so every group member keeps the loss pattern for each representative and updates it upon receiving a new one. Let LP_i be the loss pattern of receiver i, R be the receiver set, and $S\{LP_i; LP_j\}$ be the number of bits that have the same value in the bitmap corresponding to the same data packet. Receiver j is the closest matched receiver to receiver i if:

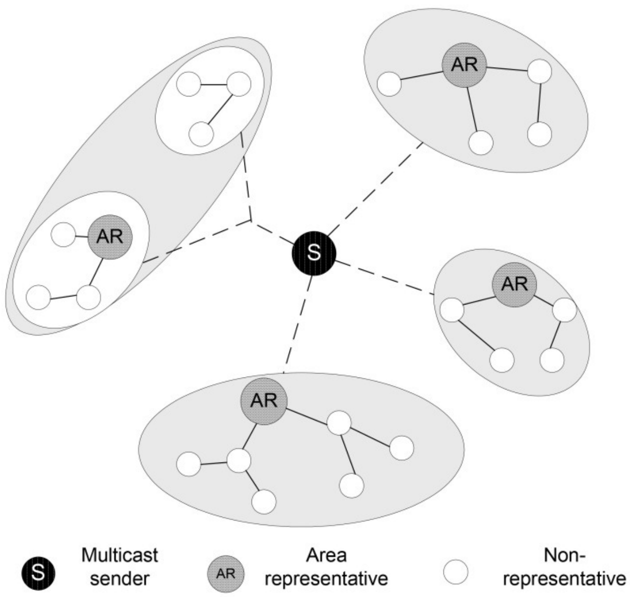


Figure 2. The Area-based Feedback Control Model

$$S\{LP_i, LP_j\} > S\{LP_i, LP_k\}, \forall k \in R, k \neq i, j$$

A receiver selects the closest matched representative as its representative. This way, representative selection is more accurate and stable, so feedback suppression is improved. The loss pattern can also provide local congestion information for multicast congestion control.

In LPABFC, timers are used by the non-representative members, called representees. The timer of a representee serves two purposes: it should be long enough to allow the AR's feedback to suppress the representee's feedback, and before a new cluster has a representative, the timer should be able to provide a certain degree of suppression among the representees. To meet these conditions, each representee calculates the length of its timer as follows. A representative attaches a time-stamp, t_{rep}—the receiving time of the DATA packet that triggers the feedback—in its feedback message. Suppose that at time $t_{feedback}$ this feedback arrives at a representee, which received the same DATA packet at t_{member}. The timer length, d, of the representee is then:

if $t_{member} < t_{rep}$,

$$d = t_{feedback} - t_{member}$$

otherwise,

$$d = t_{feedback} - t_{rep}$$

This way, deterministic suppression exists not only between the AR and representees, but also among the representees when an AR cannot represent the loss. This is because the timer length is decided by the one-way delay between the representative and representees. A representee having smaller one-way delay can suppress the other representees that have larger one-way delays. Since network dynamics may cause d to change from time to time, a representee actually keeps m most recent measurements of d to each representative, and uses the largest one of them.

The value of d includes the delay between a representative and a representee. Therefore, the timer lengths of the representees are differentiated by their delays to a representative. The timer length of a representee with shorter delay to a representative is smaller than that of a representee with a larger delay. Therefore, if a new packet loss occurs, which is not representable by the current AR, the member that is closest to the AR will timeout first and suppress further members. This decreases the chance of feedback implosion before a new representative is selected.

In LPABFC, group members are assumed to use synchronized clocks to calculate the timer length. Synchronized clocks are easy to implement in simulations and the results from work by Wu & Banerjee [16] have proved the effective performance of LPABFC. However, considering real-life Internet implementation, many of the Internet end hosts nowadays still do not have the synchronized time service. Even though service and protocol such as Network Time Protocol NTP [17] is available, the end hosts may not be configured or have an NTP server to access. Second, in reality, there is a systematic synchronization bias when link delays between the time client and the time server are asymmetric. Since the feedback suppression timer length itself is related to network link delays, such a bias could severely affect suppression performance. Therefore, it is important to validate the effectiveness of LPABFC in an unsynchronized clock environment.

Using Unsynchronized Clocks in LPABFC

In this section, an analysis and discussion is presented on suppression performance when Round Trip Time (RTT) between a representative and a representee is used to calculate the timer length without assuming synchronized clocks and symmetric link delays. First, different scenarios are analyzed to see whether the representative can suppress the representees, and then deterministic suppression among the representees is discussed.

The analysis assumes that a representee knows its RTT to the AR. How to measure the RTT is introduced at the end of this section. A representee sets its timer length to $s \cdot RTT$ with $s > 1.0$ in order to ensure suppression by the representative.

Figure 3 shows the location scenarios of the multicast source, S, the area representative, R, an intermediate member, m, and a representee, 1, used in the following analysis. Figure 3(a) shows the logical connection between the multicast source *S*, a representative *R* and a general representee 1. Figures 3(b-e) show the possible real relative positions between representative R and member 1.

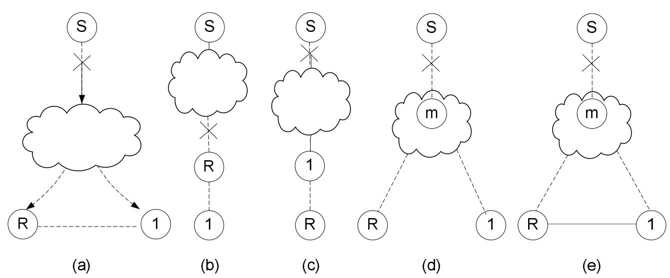


Figure 3. Suppression between a Representative and the Representees in Different Location Scenarios
(a) Logical Connections between Group Members Loss Affects Both Representative *R* and Member 1
(b) Real Location Same Branch Representative at the Upper Level
(c) Real Location Same Branch Representative at the Lower Level
(d) & (e) Real Location—The Representative and Representees in Different Branches

Suppression between the AR and the Representees

Referring to Figure 3(a)—note that this is not a real network topology—when a packet is lost by both representative *R* and representee 1, the feedback of *R* can suppress that of representee 1 if it arrives before 1's timer expires. Let:

d_{ij} = one-way delay between member *i* and member *j*
$RTT_{ij} = d_{ij} + d_{ji}$, round trip time between *i* and *j*
$T_{Ri} = d_{SR} + d_{Ri}$, time for feedback of *R* to arrive *i*
$T_i = d_{Si} + s \cdot RTT_{iR}$, time for representee *i* to send feedback
$T_{ij} = T_i + d_{ij}$, time for feedback of representee *i* to arrive *j*

For R to suppress representee 1, the following condition must be satisfied:

$$T_1 - T_{R1} = d_{S1} + s \cdot d_{1R} + (s-1) \cdot d_{R1} - d_{SR} > 0 \quad (1)$$

Since a representative or a representee could be anywhere in the network topology, the following studies consider different location scenarios in order to analyze whether a representative can effectively suppress the representees if RTT is used to calculate the timer length.

Same Branch Representative at the Upper Level:
In Figure 3(b), the representative *R* and the representee are in the same multicast tree branch, but *R* is closer to the source and, therefore, always detects packet loss earlier. Since $d_{S1} = d_{SR} + d_{R1}$, Equation (1) becomes:

$$\begin{aligned} T_1 - T_{R1} &= d_{SR} + d_{R1} + s \cdot d_{1R} + (s-1) \cdot d_{R1} - d_{SR} \\ &= s \cdot (d_{R1} + d_{1R}) = s \cdot RTT_{1R} > 0 \end{aligned} \quad (2)$$

This means that a representative can always suppress those lower-level representees.

Same Branch, Representative at the lower Level:
In Figure 3(c), the representative *R* and representee 1 are in the same tree branch, but *R* always detects packet loss later than the representee. Since $d_{SR} = d_{S1} + d_{1R}$, Equation (1) becomes:

$$\begin{aligned} T_1 - T_{R1} &= d_{S1} + s \cdot d_{1R} + (s-1) \cdot d_{R1} - (d_{S1} + d_{1R}) \\ &= (s-1) \cdot (d_{R1} + d_{1R}) = (s-1) \cdot RTT_{1R} > 0 \end{aligned} \quad (3)$$

Equation (3) shows that a representative can always suppress those upper-level representees in the same branch.

Different Branches:
When the representative and the representee are located in different multicast tree branches, two scenarios exist as shown in Figures 3(d) and 3(e). In Figure 3(d), the feedback of *R* travels some links in the multicast tree before arriving at representee 1. In Figure 3(e), however, the feedback of *R* does not pass the links in the multicast.

In the case of Figure 3(d), Equation (1) becomes:

$$\begin{aligned} T_1 - T_{R1} &= d_{Sm} + d_{m1} + s \cdot d_{1R} + (s-1) \cdot d_{R1} - (d_{Sm} + d_{mR}) \\ &= s \cdot RTT_{1R} - RTT_{Rm} > 0 \end{aligned} \quad (4)$$

Since RTT_{1R} is larger than RTT_{Rm} and s is larger than 1, Equation (4) shows that representative R can always suppress representee 1.

In the case of Figure 3(e), Equation (1) becomes:

$$T_1 - T_{R1} = d_{m1} + s \cdot d_{1R} + (s-1) \cdot d_{R1} - d_{mR} \quad (5)$$

According to Equation (5), whether representative R can suppress member 1 is decided by the one-way delays among members R, m and 1. When d_{mR} is over large compared to d_{m1}, there is a possibility that the representative R cannot suppress the representee.

From the above analysis, it can be seen that, in most cases, a representative can suppress the representees in its area if RTT is used to calculate the timer length. Although, in the case of Figure 3(e), a representative may not be able to suppress a representee, two solutions exist. First, in LPABFC, such a representative could be an inferior initial selection (e.g., dynamic membership) far away from the fault location, in which case the representative selection algorithm will react accordingly. Second, a representee can adjust its timer to prevent further early timeouts if R continues to be appointed as the representative by the source.

Suppression among Representees

In the previous section, suppression between a representative and a representee if a packet loss affects both of them was presented. When packet loss only affects representees, deterministic suppression is desirable so that the number of timeout feedback messages is small. In this section, the authors analyze whether using RTT to calculate the timer length provides this property. Similar to the previous analysis, different location scenarios are used.

Figure 4(a) shows the logical connections between the multicast source S, a representative R and representees 1 and 2. Assume a packet loss only affects members 1 and 2. To decrease the number of feedbacks, the timeout feedback of one representee should arrive at another representee before its timer expires. In other words, member 1 can suppress member 2 if:

$$T_{12} - T_2 < 0$$

which is equivalent to

$$(d_{S1} - d_{S2}) + s \cdot (RTT_{1R} - RTT_{2R}) + d_{12} < 0 \quad (6)$$

The following scenario considers different location scenarios for analyzing how the above condition affects feedback suppression among representees.

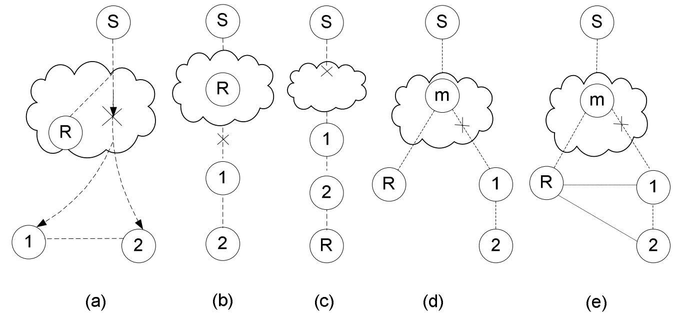


Figure 4. Suppression among Representees
(a) Logical Connections between Group Members where the Loss Affects Two Representees: Members 1 and 2
(b) Real Location
Same Branch Representative at the Upper Level
(c) Real location
Same Branch Representative at the Lower Level
(d) & (e) Real Location, Different Branches

Same Branch Representative at the Upper Level:
In Figure 4(b), representees 1 and 2 are in the same multicast tree branch. Since $d_{S2} = d_{S1} + d_{12}$, Equation (6) becomes:

$$T_{12} - T_2 = -s \cdot (d_{12} + d_{21}) < 0 \quad (7)$$

This means that an upper-level representee can always suppress those lower-level representees.

Same Branch Representative at the Lower Level:
This case is shown in Figure 4(c). In fact, in this case, packet loss also affects the representative which can suppress both member 1 and member 2. There is no timeout feedback in this case.

Different Branches:
When the representative and the representees are located in different multicast tree branches, two scenarios exist, as shown in Figures 4(d) and 4(e). In Figure 4(d), the feedback of R always travels some links in the multicast tree before arriving at representees 1 and 2. In Figure 4(e), however, representees 1 and 2 receive feedback messages from R from different paths and their RTTs to R are independent.

For the case of Figure 4(d), the result is same as Equation (7) and representee 1 can still suppress representee 2. In the case of Figure 4(e), the suppression between member 1 and member 2 depends on their RTTs to representative R. According to Equation (6), member 1 can suppress member 2 if:

$$T_{12} - T_2 = s \cdot (RTT_{1R} - RTT_{2R}) < 0 \quad (8)$$

Similarly, member 2 can suppress member 1 if:

$$T_{21} - T_1 = (d_{S2} - d_{S1}) + s \cdot (RTT_{2R} - RTT_{1R}) + d_{21} < 0 \quad (9)$$

which is equivalent to:

$$2 \cdot d_{21} + s \cdot RTT_{2R} < RTT_{1R} \quad (10)$$

From Equations (8)-(10), it can be seen that the only condition for no suppression between two representees is when $RTT_{2R} < RTT_{1R} < 2 \cdot d_{21} + s \cdot RTT_{2R}$.

Similar to using a synchronized clock, deterministic suppression among the representees exists in most cases. It should be noted that there is no guarantee of a single timeout feedback message, either by using the synchronized clock or by measuring the RTT.

RTT Measurement and Timer Adjustment

To measure the RTT, a member could unicast probe messages infrequently to the representative. When the representative sends feedback, it puts the time information (aggregated for all of the members that have probed it during that period) in the feedback. Then, the representees can calculate their RTTs after receiving the feedback.

In the case of synchronized clocks, representative R can always suppress representee 1 because both d_{SR} and d_{R1} are considered in the timer length calculation, thus the timer length is long enough for suppression. However, when using RTT to calculate the timer length, d_{SR} is not considered and the representative may not be able to suppress the representees, as in the case of Figure 3(e). In this case, the representee needs to adjust its timer length. For this purpose, if a representee timeouts (at t_1) before the feedback of its area representative arrives (at t_2), it should increase its RTT by at least $t_2 - t_1$ to avoid further early timeouts.

Simulation Results

Timer selection affects the responsiveness of the feedback and location of the selected representative because the member with the smallest timer will send out a feedback message first and could be selected as a representative. This section introduces the simulation results that show how the area representatives are selected and managed based on the number of network faults and dynamics. Unsynchronized clocks are used to measure RTTs between members. The results show that using unsynchronized clocks in LPABFC still achieves effective representative selection, which is similar to the original scheme that uses synchronized clocks.

The simulation study is implemented with Network Simulator-2 (ns-2) [19] with random transit-stub-type network topology generated by GT-ITM [20]. The transit-stub graph model is widely used in network simulation research to model today's Internet. The transit domains represent the backbone networks and the stub domains represent the edge networks. The new ns-3 is not used due to its incompatibility with GT-ITM. Since the synchronized-clock approach in LPABFC was studied in ns-2 and GT-ITM, to be consistent, this study was carried out in the same simulation environment.

The study on the representative management is carried in a 100-node transit-stub network topology. Packet loss is randomly introduced on the transit-stub links. The length of a loss period is uniformly selected between 20 seconds and 40 seconds, and the length of a loss-free period is uniformly selected from between 30 seconds and 60 seconds.

Figure 5 shows how the number of representatives changes according to loss conditions from simulation times of 0 to 200. It can be seen that during the first half of the time, the number of representatives increases as new losses appear on different links; then, it fluctuates as the old loss periods terminate and new loss periods start.

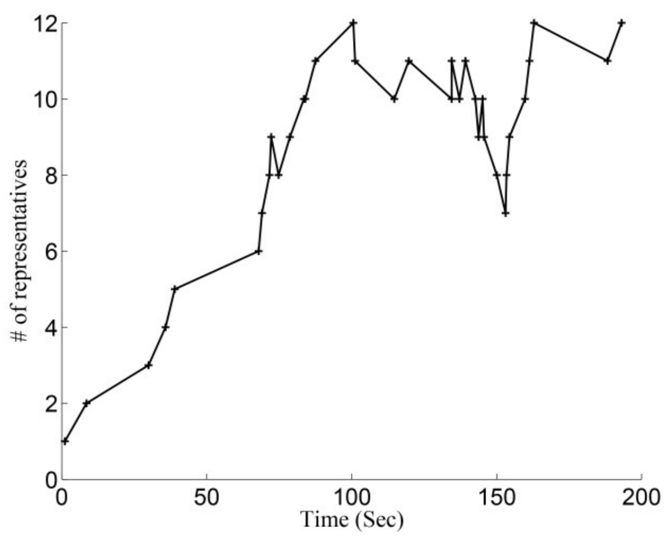


Figure 5. The Number of Representatives versus Simulation Time

The number of clusters, the members in a cluster, and the creation and merge of the cluster is shown in the following graphs and are based on the simulation time when such an even occurs.

- Simulation time of 1.290: Figure 6 shows the network topology. The multicast sender is in red and the other group members are in blue. The group density is 75%; i.e., about 75 nodes become the multicast group members. Node 17 is selected as the representative at simulation time 1.290 and there is only one area including all of the group members.

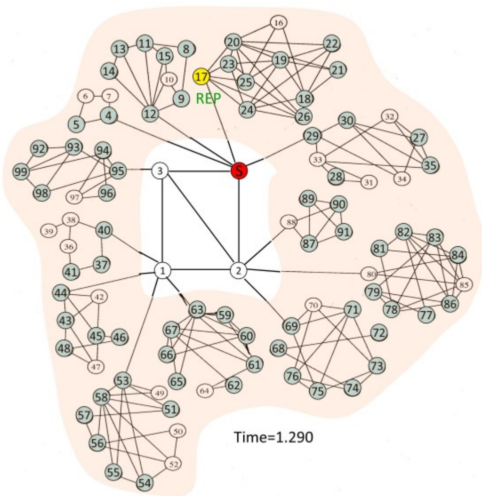


Figure 6. Representative Management at a Time of 1.290

- Simulation time of 8.654: Figure 7 shows that at time 8.654, node 40 is added as a new representative and a new area is formed accordingly. The rest of the nodes still choose REP 17 as the AR. Therefore, two areas exist. After this time, area number continues to increase when new loss occurs, and nodes 84, 63, 95 and 12 become representatives.

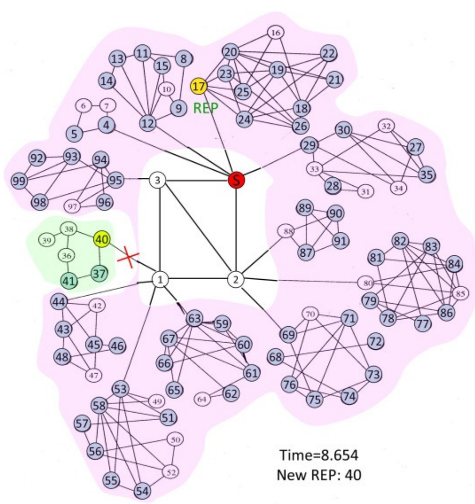


Figure 7. Representative Management at a Time of 8.654

- Simulation times of 39.099 and 68.024: It can be seen in Figures 8 and 9 that new areas are formed at times 39.099 and 68.024, respectively, in which nodes 95 and 12 are selected as ARs. In addition, $link_{1\text{-}40}$ is recovered from packet loss but node 40 remains as a representative that will be deleted if it does not send feedback for a sufficiently long time.

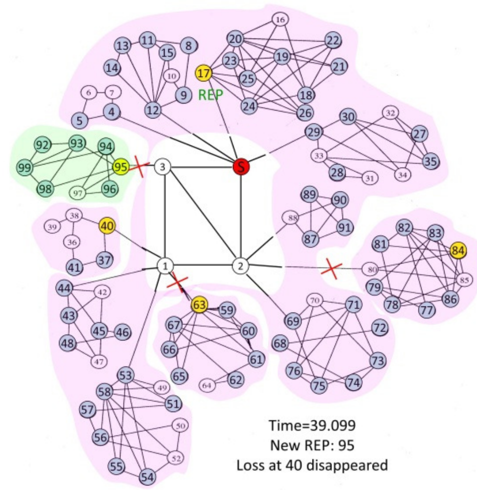


Figure 8. Representative Management at a Time of 39.099

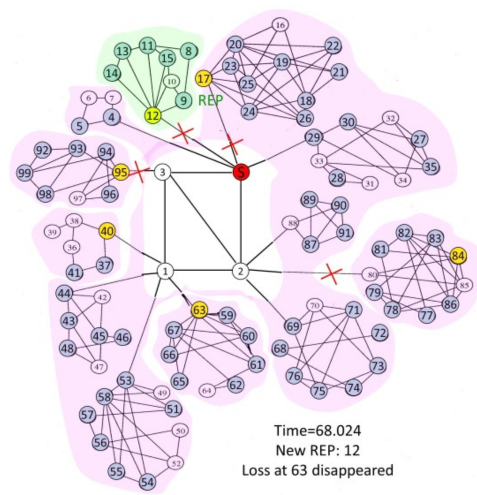


Figure 9. Representative Management at a Time of 68.024

- Simulation time of 78.687: Figure 10 shows three things at time 78.687. First, representative 40 is deleted by the sender, and its area merges with another area represented by node 63. Second, four new clusters are added with nodes 4, 29, 53 and 90 being the representative. Third, the loss on $link_{s\text{-}17}$ and $link_{3\text{-}95}$ terminates.

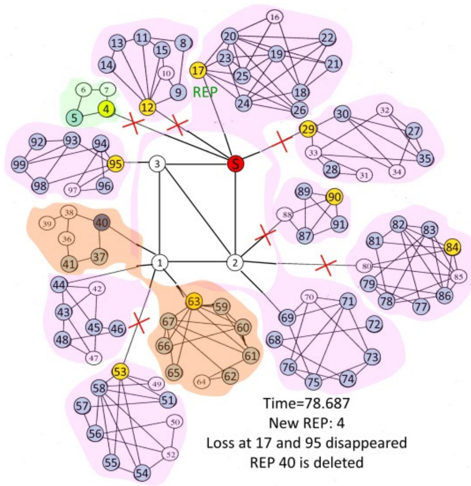


Figure 10. Representative Management at a Time of 78.687

- Simulation time of 83.994: In Figure11, loss happens on $link_{1\text{-}40}$ again and results in a new area. In addition, by monitoring the loss detection time, node 87 finds that it detects packet loss earlier than its AR, node 90. Therefore, it requests that node 90 be replaced and becomes the new representative.

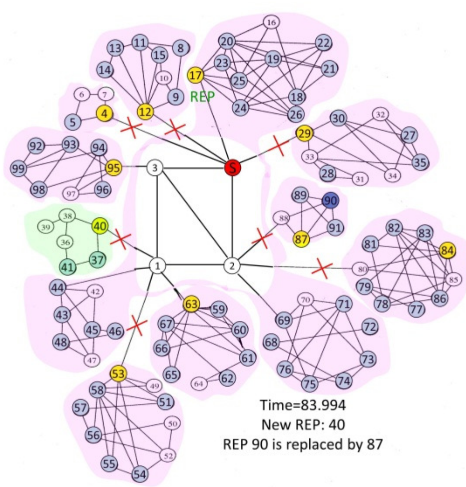


Figure 11. Representative Management at a Time of 83.994

From the above simulation results and analysis, it can be seen that the number of representatives and areas in LPABFC still adaptively changes according to the dynamic network conditions and the scheme selects the appropriate members to be the representatives.

Conclusion

Next-generation networks will support a much higher data transmission rate than current networks and provide more advanced services. In particular, the demand for Internet group communications such as live media broadcasting, content distributions and group collaborations has grown rapidly with the increase in commercial usage of the Internet. IP multicasting has to overcome some critical problems in order to be safely deployed on the Internet, and one of them is feedback implosion.

Previous studies show that LPABFC provides effective feedback implosion control. This study extends the previous studies and analyzed feedback control performance when unsynchronized clocks are used in timer calculations. The analytical and simulation results show that effective feedback suppression is retained with unsynchronized clocks, and the number and location of representatives are managed based on dynamic network fault situations.

For future studies, LPABFC can be extended to multicast congestion control because the representatives provide responsive feedback. It will also be studied in wireless communications which, by nature, supports multicast and broadcast. One example is cellular networks where handoff, i.e., user switches between cells or networks [21], must be taken into consideration. Another example is wireless mesh networks [22], where fewer available resources, such as power and computation capability, could be a concern.

References

[1] Yajnik, M., Kurose, J., & Towsley, D. (1996). Packet Loss Correlation in the mbone Multicast Network. *Proceeding of IEEE Global Conference on Communications*.

[2] C´aceres, R., Adams, A., Bu, T., Duffield, N., Horowitz, J., Friedman, T. et al. (2000). The Use of End-to -End Multicast Measurements for Characterizing Internal Network Behavior. *IEEE Communications*, *38*(5).

[3] Network Research Group. Retrieved August 21, 2012 from http://www.wand.net.nz/. Original site retrieved June 2006, from http://watt.nlanr.net/.

[4] Labovitz, C., Ahuja, A., Bose, A., & Jahanian, F. (2000). Delayed Internet Routing Convergence. *Proceeding of ACM SIGCOMM*.

[5] Hou, X., & Wu, S. (2011). On the Performance of an Application Layer Multicast Protocol. *International Journal of Modern Engineering*, *12*(1), 20-28.

[6] Yu, Y., & Ni, L. (2011). Survey of Reliable and Efficient MAC Layer Multicast Protocols in MANETs. *Second International Conference on Mechanic Automation and Control Engineering (MACE).*

[7] Vella, J., & Zammit, S. (2012). A Survey of Multicasting over Wireless Access Networks. *Communica-*

tions Surveys & Tutorials, PP(99), 1-36.

[8] Grossglauser, M. (1997). Optimal Deterministic Timeouts for Reliable Scalable Multicast. *IEEE Journal on Selected Areas of Communications*, *15*(3), 422 -433.

[9] Nonnenmacher, J., & Biersack, E. W. (1999). Scalable Feedback for Large Groups. *IEEE/ACM Transactions on Networking*, *7*(3), 375-386.

[10] Floyd, C., Jacobson, V., Liu, C., McCanne, S., & Zhang, L. (1997). A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. *IEEE/ACM Transactions on Networking*, *5* (6), 784-803.

[11] Burget, R., Komosny, D., & Ganeshan, K. (2012). Topology Aware Feedback Transmission for Real-time Control Protocol. *Journal of Network and Computer Application, 35*(2)*,* 723-730.

[12] Holbrook, H. W., Singhal, S. K., & Cheriton, D. R. (1995). Log-based Receiver-reliable Multicast for Distributed Interactive Simulation. *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication,* (pp. 328-341).

[13] Xiao, Z., & Birman, K. P. (2001). A Randomized Error Recovery Algorithm for Reliable Multicast. *Proceedings of IEEE INFOCOM*.

[14] Paul, S., Sabnani, K. K., Lin, J. C., & Bhattacharyya, S. (1997). Reliable Multicast Transport Protocol (RMTP). *IEEE Journal on Selected Areas of Communications*, *15*(3), 407-421.

[15] DeLucia, D., & Obraczka, K. (1997). Multicast Feedback Suppression Using Representatives. *Proceedings of IEEE INFOCOM*.

[16] Wu, S., & Banerjee, S. (2002). Multicast Feedback Control Using Loss-Pattern Matching. *Proceedings of the IEEE International Conference on Communications*, (pp. 1253-1258).

[17] Wu, S., Banerjee, S., & Hou, X. (2006). Performance Evaluation and Comparison of Multicast Feedback Control Mechanisms. *SIMULATION: Transactions of the Society for Modeling and Simulation International, 82*(5), 345-362.

[18] Mills, D. (1996). Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI. *RFC2030*.

[19] The Network Simulator - ns-2. Retrieved September 5, 2012, from http://www.isi.edu/nsnam/ns/.

[20] GT-ITM: Georgia Tech Internetwork Topology Models. Retrieved September 20, 2012, from http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html.

[21] Arabmakki, N., Rashad, S., & Krijestorac, S. (2012). A Comparison of Different Vertical Handoff Algorithms between WLAN and Cellular. *International Journal of Research and Innovation*, *4*(1), 5-12.

[22] Zhu, J., & Pecen, R. (2009). A Wireless Mesh Network Based Automatic Utility Data Collection System. *International Journal of Modern Engineering*, *10*(1), 29-34.

Biographies

SHUJU WU is currently an Associate Professor at the Computer Electronics and Graphics Technology Department at Central Connecticut State University. She received her Ph.D. degree in Information Science from the University of Pittsburgh in 2004. Dr. Wu's teaching and research interests include computer communications and networks, multimedia systems, performance modeling and evaluation, and network applications. Dr. Wu may be reached at swu@ccsu.edu.

XIAOBING HOU is currently an Assistant Professor at the Computer Electronics and Graphics Technology Department at Central Connecticut State University. He received his Ph.D. degree in Information Science from the University of Pittsburgh in 2006. Dr. Hou's teaching and research interests are in the areas of computer networking and information security. He is a member of IEEE and ACM. Dr. Hou may be reached at xhou@ccsu.edu.