Performance Analysis of Feedback Synchronization for Multicast ABR Flow Control

Xi Zhang† and Kang G. Shin‡

Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, MI 48109-2122, USA
Tel: +1 734 647 6977 Fax: +1 734 763 4617
†*Corresponding author. Email: xizhang@eecs.umich.edu* ‡*Email: kgshin@eecs.umich.edu*

Designated GLOBECOM '99 Symposium: High Speed Networks

Abstract—**We present a balanced and unbalanced binary-tree model to explore the delay performance of feedback synchronization algorithms for multicast ABR flow control. Using this model, we analyze the feedback-delay performance scalability of the *Soft-Synchronization Protocol* (SSP) [1], which derives a single "consolidated" RM (Resource Management) cell at each multicast branch-point from feedback RM-cells of different downstream branches that are not necessarily responses to the same forward RM-cell. In contrast with the other existing schemes, SSP is shown to be able to effectively support synchronization of feedback RM-cells and make the effective RM-cell roundtrip time (RTT) virtually independent of the multicast-tree's topology. Also derived is the optimal RM-cell update interval for SSP to minimize RM-cell RTTs for a given multicast tree.**

Index Terms—**ATM, ABR, flow control, multicast, scalability, feedback consolidation, feedback soft-synchronization, feedback delay.**

I. Introduction

In multicast ABR, simultaneous congestion feedback from all branches can cause a *feedback implosion* [2] at the source, especially when the multicast tree is large. Hence, it is important to consolidate the congestion feedback at each branch-point and only the consolidated feedback is sent upstream. Since different downstream-branches' feedback RM-cells may arrive at the branch point at significantly different times, the consolidation of feedback RM-cells must be *synchronized* at the branch-point before the consolidated RM-cell can be forwarded upstream.

The first-generation feedback consolidation algorithms [3], [4], [5] employ a simple hop-by-hop feedback mechanism to deal with the feedback implosion problem. On receipt of a forward RM-cell, the consolidated feedback is propagated upwards by a *single* hop. While hop-by-hop feedback is very simple, it does not scale well because the RM-cell round-trip time (RTT) is proportional to the height of the multicast tree. Additionally, since feedback RM-cells from downstream nodes are *freely* synchronized at branch nodes, the source may be misled by the incomplete feedback information, which can cause the *consolidation noise* problem [6].

To reduce the RM-cell RTT and eliminate the consolidation noise, the authors of [7], [6] proposed feedback synchronization at branch-points by accumulating feedback from *all* branches. The main problem with this scheme is its slow transient response since the feedback from the congested branch may have to needlessly wait for the feedback from the longer paths, which may not be congested at all. The authors of [8] proposed an improved consolidation algorithm to speed up the transient response by sending the fast congestion feedback without waiting for all branches' feedback during the transient phase.

One of the critical deficiencies of the schemes described above is that they do not detect and remove non-responsive branches from the feedback synchronization process. One or more non-responsive branches may detrimentally impact end-to-end performance by providing either stale congestion information or by stalling the entire multicast connection. In [1], we proposed a *Soft-Synchronization Protocol* (SSP) which derives a single consolidated RM-cell at each branch-point from feedback RM-cells of different downstream branches that are not necessarily responses to the same forward RM-cell in each synchronization cycle. The SSP not only scales well with the multicast-tree topology, but also can readily detect and remove non-responsive branches.

All of the above-referenced work only focused on the various protocols' design and implementation issues. However, the feedback-delay properties of various feedback synchronization algorithms are neither well understood nor thoroughly studied. In this paper, we develop a balanced and unbalanced binary-tree model to characterize the feedback-delay properties of a class of feedback synchronization algorithms in terms of RM-cell RTTs. In Section II, we overview the proposed SSP. In Section III, using the binary-tree model we derive the analytical properties of SSP and hop-by-hop feedback synchronization algorithms. Our analytical results show SSP to not only be able to support efficient feedback synchronization, but also make the effective RM-cell RTT virtually independent of the multicast-tree's height and path-length variations. In Section IV, we derive the optimal RM-cell interval for SSP to minimize RM-cell RTTs for a given multicast tree. The paper concludes with Section V.

II. Description of SSP

We first present an overview of SSP, the switch feedback synchronization algorithm [1]. At the center of SSP is a pair of connection-update vectors: (i) $conn\_patt\_vec$, the connection pattern vector where $conn\_patt\_vec(i)$ = 0 (1) indicates the i-th output port of the switch is (not) a downstream branch of the multicast connection. Thus, $conn\_patt\_vec(i)$ = 0 (1) implies that a data copy should (not) be sent to the i-th downstream branch and

The work reported in this paper was supported in part by the U.S. Office of Naval Research under Grant N00014-99-1-0465.

```
On receipt of a feedback RM cell from $i$-th branch:
  if ($conn\_patt\_vec(i) \neq 1$) { ! Only process connected branches
    $resp\_branch\_vec(i) := 1$; ! Mark connected and responsive branch
    $MCI := MCI \vee CI$; ! Bandwidth-congestion indicator processing
    $MER := \min\{MER, ER\}$; ! ER information processing
    if ($conn\_patt\_vec \oplus resp\_branch\_vec = \underline{1}$) { ! soft synchronization
      send RM cell ($dir := back$, $ER := MER$,
                $CI := MCI$); ! Send fully-consolidated RM-cell upstream
      $no\_resp\_timer := N_{nrt}$;  ! Reset non-responsive timer
      $resp\_branch\_vec := \underline{0}$);  ! Reset responsive branch vector
      $MCI := 0$;  $MER := ER$;}}; ! Reset RM-cell control variable
On receipt of a forward RM cell:
  multicast RM cell based on $conn\_patt\_vec$; ! Multicast RM cell
  $no\_resp\_timer := no\_resp\_timer - 1$; ! No-responsive branch checking
  if ($no\_resp\_timer = 0$) { ! There is a non-responsive branch
    $conn\_patt\_vec := resp\_branch\_vec \oplus \underline{1}$; ! update connect. pattern vec.
    if ($resp\_branch\_vec \neq \underline{0}$) { ! There is at least one responsive branch
      send RM cell ($dir := back$, $ER := MER$,
              $CI := MCI$); ! Send partially-consolidated RM-cell up-stream
      $no\_resp\_timer := N_{nrt}$; ! Reset non-responsive timer
      $resp\_branch\_vec := \underline{0}$; ! Reset responsive branch vector
      $MCI := 0$;  $MER := ER$; }}; ! Reset RM-cell control variables
```

Fig. 1. Pseudocode for Switch Feedback Synchronization Algorithm.

a feedback RM-cell is (not) expected from the i-th downstream branch;[1] (ii) $resp\_branch\_vec$, the responsive branch vector is initialized to $\underline{0}$ and reset to $\underline{0}$ whenever a consolidated RM-cell is sent upward from the switch. $resp\_branch\_vec(i)$ is set to 1 if a feedback RM-cell is received from the i-th downstream branch. The connection pattern specified in $conn\_patt\_vec$ is updated by $resp\_branch\_vec$ each time when the non-responsive branch is detected or a new connection request is received from a downstream branch.

A simplified pseudocode of the switch RM-cell processing algorithm is given in Fig. 2. On receipt of a feedback RM-cell returned from a receiver or a connected downstream branch, the switch first marks its corresponding bit in the $resp\_branch\_vec$ and then conducts RM-cell consolidation operations. If the modulo-2 addition (the soft-sychcronization operation), $conn\_patt\_vec \oplus resp\_branch\_vec$ equals $\underline{1}$, an all 1's vector, indicating all feedback RM-cells synchronized, then a fully-consolidated feedback RM-cell is generated and sent upward. But, if the modulo-2 addition is not equal to $\underline{1}$, the switch needs to await other feedback RM-cells for synchronization. Notice that since the synchronization algorithm allows feedback RM-cells corresponding to different forward RM-cells to be consolidated, the feedback RM-cells are "softly-synchronized" at branch nodes.

Upon receiving a forward RM-cell, the switch first multicasts it to all the connected branches specified by $conn\_patt\_vec$. Then, decrease the non-responsive timer for this connection by one. The no_resp_timer is initialized to a threshold N_{nrt} and reset to N_{nrt} whenever a consolidated RM-cell is sent upward. The predetermined time out value N_{nrt} for non-responsiveness is determined by such factors as the difference between the maximum and minimum RM-cell RTTs in a multicast tree. We use the forward RM-cell arrival time as a natural clock for detecting/removing non-responsive branches (such that it will still work even in the presence of faults in the downstream branches). Each time a switch receives a forward RM-cell, the multicast connection's no_resp_timer is decreased by one. If $no\_resp\_timer = 0$ (time out) and $resp\_branch\_vec \neq \underline{0}$ (i.e., there is at least one downstream branch responsive), then the switch will stop awaiting arrival of feedback RM-cells and immediately generate a partially-consolidated RM-cell, and send it upward. Whenever $no\_resp\_timer = 0$, at least one non-responsive downstream branch is detected and will be removed by the simple complementary operation: $conn\_patt\_vec := resp\_branch\_vec \oplus \underline{1}$, which updates $conn\_patt\_vec$. Therefore, a downstream branch which has not sent any feedback RM-cell for N_{nrt} forward RM-cell time units will be removed from the multicast tree.

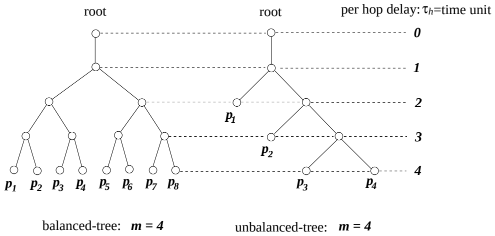


Fig. 2. Balanced and unbalanced binary multicast trees.

III. RM-cell RTT Analysis

It is well-known that feedback delay plays a crucial role in determining the effectiveness of any feedback-based flow-control scheme [1]. In this section, we analyze the properties of RM-cell RTT for different feeback synchronization algorithms.

A. The Binary-Tree Model

To simplify the analysis of RM-cell RTT, we *quantize* the network feedback-delay by assuming each switch-hop to have a uniform delay (including processing and propagation delays). This assumption can be readily relaxed because the difference in switch processing delay and link-propagation delay of different switch-hops can be translated into different numbers of switch-hops with the same delay. We use the *hop-delay*, τ_h, which is the sum of the switch-processing delay and link-propagation delay taken in each hop, as the *time unit* in our delay analysis. To study the worst case and enable performance comparison, we only consider two types of multicast trees: *balanced* and *unbalanced binary trees*. Since we are only concerned with a path's RM-cell RTT which is determined by its length, it suffices to consider binary trees. Notice that in an unbalanced binary tree, the number of paths, denoted by n, from the root to all leaves is equal to the height of the tree, denoted by m, while in a balanced binary tree $n = 2^{m-1}$. Fig. 2 illustrates these two types of trees with height $m = 4$.

As discussed in [1], [8], for ABR services only the feedback from the *most-congested* path in a multicast connection governs the flow-control operations at the source. However, the RM-cell RTT of different paths in a multicast tree may vary significantly since the path lengths differ from each other. Thus, we need to analyze each individual path's RM-cell RTT in a multicst tree. The individual path's RM-cell RTT is also affected by the feedback synchronization algorithms used. In addition, the RM-cell RTT for a given path may vary at the beginning of the flow-control operation in an initial state, during which feedback RM-cells are not yet "regularly" synchronized. The RM-cell RTT becomes stable

[1] Note that the negative logic is used for convenience of implementation.

after feedback RM-cells are regularly synchronized and enters a steady state. In the following, we analyze the RM-cell feedback-delay properties, in both initial and steady states, of each path in a multicast tree, which is flow-controlled by hop-by-hop and SSP schemes, respectively. We omit all the proofs in the following lemmas, theorems, and corollaries for lack of space, but refer the interested readers to [9] (available on-line) for their detailed proofs.

B. Feedback-Delay Properties for Hop-by-Hop Scheme

The following theorem gives a set of formulas for calculating all paths' RM-cell RTTs in an unbalanced-tree for the hop-by-hop scheme.

Theorem 1: If an unbalanced multicast-tree of height $m \geq 2$ is flow-controlled by the hop-by-hop scheme with an RM-cell interval $\Delta \geq 1$ (τ_h), then the RM-cell RTT, $\tau_u(j, \Delta)$, of the j-th (counting from left to right) path, P_j, remains the same in both steady and initial states, and is given by:

$$\tau_u(j, \Delta) \;=\; \begin{cases} 2 + j\,\Delta; & \text{if } 2 \leq \Delta \leq \tau_{max} \\ 2(j+1); & \text{if } \Delta = 1 \end{cases} \qquad (1)$$

where $1 \leq \Delta \leq \tau_{max}$,[2] $\tau_{max} = 2m$, and $1 \leq j \leq m-1$. □

The following corollary, providing the equations to compute the all paths' RM-cell RTTs in a balanced-tree for the hop-by-hop scheme, is the direct result from *Theorem 1* by letting $j = m-1$ in Eq. (1).

Corollary 1: If a balanced multicast-tree connection of height $m \geq 2$ is flow-controlled by the hop-by-hop feedback scheme with the RM-cell interval $\Delta \geq 1$, then RM-cell RTTs of all paths, $\tau_b(j, \Delta)$, are the same in both steady and initial states, and are determined by:

$$\begin{aligned} \tau_b(j, \Delta) \;&=\; \max_{j \in \{1, 2, \cdots, m-1\}} \{\tau_u(j, \Delta)\} \\ &=\; \begin{cases} \tau_{max} + (m-1)(\Delta - 2); & \text{if } 2 \leq \Delta \leq \tau_{max} \\ \tau_{max}; & \text{if } \Delta = 1 \end{cases} \end{aligned} \qquad (2)$$

where $\tau_{max} = 2m$, $1 \leq j \leq 2^{m-1}$, and $\tau_u(j, \Delta)$ is defined by Eq. (1) for an unbalanced multicast tree of the same height. □

C. Feedback-Delay Properties for SSP Scheme

The following lemma characterizes the fundamental synchronization relationships between paths under SSP, which lays the foundation for *Lemma 2*.

Lemma 1: Consider an unbalanced multicast-tree of height $m > 2$. Let P_i be a relatively shorter path than another path $P_{\tilde{i}}$ such that $1 \leq i < \tilde{i} \leq m-1$. If the multicast-tree is flow-controlled by SSP with the RM-cell interval $\Delta \geq 1$, then $P_{\tilde{i}}$'s feedback RM-cell does not have to wait for P_i's feedback RM-cell to synchronize feedback RM-cells at any branch-node. □

The lemma given below reveals four *iff* conditions for a path's RM-cell RTT to attain its limiting minimum, which consists of prepagation and processing delays only (i.e., no synchronization waiting-time delay).

Lemma 2: Let P_j be the j-th path in an unbalanced-tree as defined in *Lemma 1* with $1 \leq j \leq m-1$. Then, the following four claims are equivalent for the steady-state RM-cell RTT:

Claim 1. P_j's feedback RM-cell doesn't wait for a longer path $P_{\tilde{j}}$'s $(\tilde{j} > j)$ feedback RM-cell to achieve feedback synchronization at the first branch-node from P_j's leaf;

Claim 2. P_j's feedback RM-cell doesn't wait for feedback RM-cells for synchronization at *any* branch-node on P_j;

Claim 3. $\exists\, k \in \{0, 1, 2, \cdots\}$ such that $2(m - j - 1) - k\Delta = 0$, where $1 \leq j \leq m-1$ and $1 \leq \Delta \leq \tau_{max} = 2m$;

Claim 4. P_j's steady-state RM-cell RTT $\tau_u(j, \Delta)$ attains its minimum and is given by:

$$\tau_u(j, \Delta) = \min_{\Delta}\{\tau_u(j, \Delta)\} = 2(j+1) \qquad (3)$$

where $1 \leq j \leq m-1$ and $1 \leq \Delta \leq \tau_{max} = 2m$. □

Based on *Lemma 1* and *Lemma 2*, we obtain the following theorem, which gives a set of formulas to calculate all paths' RM-cell RTTs during both initial and steady states in an unbalanced-tree under SSP.

Theorem 2: Let P_j be the j-th path of an unbalanced-tree as defined in *Lemma 1* $(1 \leq j \leq m-1)$. If the multicast tree is flow-controlled by SSP with the RM-cell update interval Δ $(1 \leq \Delta \leq \tau_{max} = 2m)$,[3] then the following claims hold for $j = 1, 2, \cdots, m-1; \tau_{max} = 2m; 1 \leq \Delta \leq \tau_{max}$:

Claim 1. The number of P_j's feedback RM-cells going through initial state is determined by:

$$k_j^* \overset{\triangle}{=} \max_{k \in \{0, 1, 2, \cdots\}} \{k \mid 2(m - j - 1) - k\Delta \geq 0\}; \qquad (4)$$

Claim 2. P_j's RM-cell RTT in steady state is determined by:

$$\tau_u(j, \Delta) = \tau_{max} - k_j^* \Delta; \qquad (5)$$

Claim 3. The i-th RM-cell RTT during P_j's initial state is determined by:

$$\tau_u(j, \Delta, i) = \begin{cases} \tau_{max} - (i-1)\Delta; & \text{if } k_j^* \geq 1 \wedge 1 \leq i \leq k_j^* \\ \tau_u(j, \Delta); & \text{if } k_j^* \geq 1 \wedge i > k_j^* \\ \tau_{max}; & \text{if } k_j^* = 0. \end{cases}$$

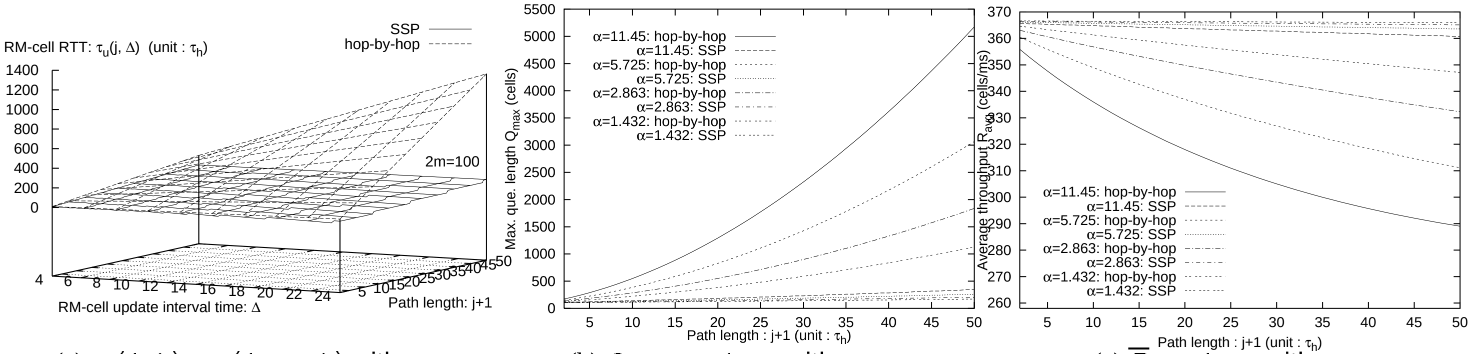
□

The corollary described below, giving the equations for calculating all paths' RM-cell RTTs in a balanced tree under SSP, follows directly from *Theorem 2* by letting $j = m-1$ in Eq. (4) which leads to $k_{m-1}^* = 0$ and thus $\tau_b(j, \Delta) = \tau_u(m-1, \Delta) = \tau_{max}$ by Eq. (5).

Corollary 2: If a balanced-tree multicast connection of height $m \geq 2$ is flow-controlled by SSP with the RM-cell interval $\Delta \geq 1$,

[2] *Theorem 1* still holds even when $\Delta \geq \tau_{max} = 2m$. But the RM-cell update interval Δ is usually a fraction of the maximum RM-cell RTT. So, we do not consider the case of $\Delta \geq \tau_{max} = 2m$ even if it is analytically correct.

[3] *Theorem 2* still holds for $\Delta > \tau_{max} = 2m$, but Δ is typically a fraction of the maximum RM-cell RTT $\tau_{max} = 2m$.



(a) $\tau_u(j,\Delta)$ vs. $(j+1,\Delta)$ with $m=50$. (b) Q_{max} vs. $j+1$ with $m=50$. (c) $\overline{R}$ vs. $j+1$ with $m=50$.

Fig. 3. Impact of P_j's path length $j+1$, tree height m, RM-cell interval Δ on P_j's RM-cell RTT $\tau_u(j,\Delta)$, max. queue length Q_{max}, avg. throughput $\overline{R}$.

then all paths' RM-cell RTTs, $\tau_b(j,\Delta)$, are the same in both steady and initial states and are determined by:

$$\tau_b(j,\Delta) = \max_{j\in\{1,2,\cdots,m-1\}} \{\tau_u(j,\Delta)\} = \tau_{max} \quad (6)$$

where $\tau_{max} = 2m$, $1 \le j \le 2^{m-1}$, and $\tau_u(j,\Delta)$ given by Eq. (5) is P_j's RM-cell RTT for an unbalanced multicast tree of the same height. □

Remark 1: Comparing *Theorem 2* and *Theorem 1*, we observe the following: (1) for the hop-by-hop scheme, RM-cell RTT in initial-state is the same as that in steady-state. In contrast, for the SSP scheme, RM-cell RTT in initial-state, if any, is larger than, and lower bounded by, RM-cell RTT in steady-state. For SSP, the initial-state acts like a "warm-up" period for feedback RM-cells to be synchronized at each branch-node, during which the initial-state RM-cell RTTs converge to their corresponding steady-state values. The "warm-up" periods for P_j $(1 \le j \le m-1)$ are determined by k_j^* values given in Eq. (4). (2) for SSP in both initial and steady states, the RM-cell RTT $\tau_u(j,\Delta)$ is upper-bounded by $\tau_{max} = 2m$ (see Claim 2 and Claim 3 of *Theorem 2* and Eq. (6)). The increase rate of $\tau_u(j,\Delta)$, as a function of m, is $O(m)$ in the worst case. In contrast, for the hop-by-hop scheme, the RM-cell RTT $\tau_u(j,\Delta)$ is not upper-bounded by $\tau_{max} = 2m$ (see Eqs. (1) and (2)). Also, $\tau_u(j,\Delta)$ is very sensitive to path length j and RM-cell update interval Δ, and increases at a rate up to $O(m^2)$ in the worst case.

D. Numerical Comparison between SSP and Hop-by-Hop

We present the numerical results derived from *Theorem 1* and *Theorem 2*. We only focus on the unbalanced multicast tree to study the worst case of RM-cell RTT variations. Since P_j's length equals $j+1$ for $j = 1, 2, \cdots, m-1$ (see the unbalanced tree shown in Fig. 2), $\tau_u(j,\Delta)$ is the RM-cell RTT for P_j with a length of $j+1$ in an unbalanced tree. Fig. 3(a) plots P_j's RM-cell RTT $\tau_u(j,\Delta)$ vs. P_j's length $j+1$ and RM-cell interval Δ with tree height $m = 50$ for the two different schemes. We observe that for both hop-by-hop and SSP schemes RM-cell RTTs $\tau_u(j,\Delta)$'s increase monotonically with path length $j+1$, RM-cell interval Δ, and tree-height m. However, $\tau_u(j,\Delta)$ for the hop-by-hop scheme increases much faster, and is always larger, than that for the SSP scheme, and tends to blow up (as high as 1200 τ_h) as $j+1$, Δ, and m increase. In contrast to the hop-by-hop scheme, the increase of $\tau_u(j,\Delta)$ for SSP is very limited as $j+1$, Δ, and m get larger. In addition, $\tau_u(j,\Delta)$ for SSP is upper-bounded by $2m = 100 = \tau_{max}$ as shown in Fig. 3(a), which verifies *Theorem 2*. Thus, as shown in Fig. 3(a), the RM-cell RTT for SSP is virtually independent of path length, RM-cell interval, and multicast-tree height, as compared to the hop-by-hop scheme. This is because (1) the synchronization waiting-time is much longer for hop-by-hop than SSP; (2) the number of forward RM-cells required for a feedback RM-cell to return from the leaf node to the root in the hop-by-hop scheme is proportional to m, while in SSP, any single RM-cell can return from the leaf node back to the root by itself.

As analyzed in [1], RM-cell RTTs, or the path lengths, have a significant impact on both the bottleneck maximum queue length Q_{max} and the average throughput $\overline{R}$. Due to space limitation, we omit the derivations (based on the fluid modeling) of closed-form expressions for Q_{max} and $\overline{R}$ as functions of RM-cell RTT (which are available on-line in [1]). Instead, we present the numerical solutions of Q_{max} and $\overline{R}$ as the functions of P_j's path length $j+1$ in an unbalanced multicast tree to compare the performance between the hop-by-hop and SSP schemes. Assume the multicast-tree bottleneck bandwidth $\mu = 155$ Mbps ≈ 367 cells/ms, τ_h = 0.1 ms, $\Delta = 4\tau_h = 0.4$ ms, and $m = 50$. Fig. 3(b) and Fig. 3(c) plot Q_{max} and $\overline{R}$ vs. path length $j+1$ with different rate-gain parameter α values [1] for the two different schemes. For the hop-by-hop scheme, maximum queue length Q_{max} is observed to increase dramatically (see Fig. 3(b)) while the average throughput $\overline{R}$ drops significantly (see Fig. 3(c)) as P_j's path length $j+1$ and tree height m (the maximum for $j+1$) increase. This undesirable trend worsens as α gets larger. In contrast, for SSP with the same parameter settings, both Q_{max}'s increase and $\overline{R}$'s drop are very small when $j+1$ and m (even as α varies) increase. Again, Q_{max} and $\overline{R}$ for SSP are found to be virtually independent of the path-length and tree-height variations. Hence, SSP is more scalable than the hop-by-hop scheme in terms of maximum buffer requirement and average throughput when the multicast-tree topology changes.

IV. On Selection of RM-cell Update Interval Δ

Even though the RM-cell RTT for SSP is much smaller than the hop-by-hop scheme, its $\tau(j,\Delta)$ value can be reduced further by properly selecting RM-cell interval Δ. We now focus on how Δ affects $\tau(j, \Delta)$ and discuss how to select Δ to reduce SSP's RM-cell RTT.

A. Analytical Relationships between RM-cell RTTs and Δ

Unlike unicast, the selection of value for RM-cell interval Δ makes a significant impact on all paths' RM-cell RTTs in a multicast-tree. To analytically quantify this impact, we introduce the following definitions.

Definition 1: If P_j's feedback RM-cell is only synchronized with the feedback RM-cells which correspond to the same forward RM-cell, then P_j is said to be *strictly-synchronized*. □

Obviously, P_{m-1} is always strictly-synchronized since it is synchronized only with P_m. The following theorem describes the *iff* condition, as a function of Δ, for identifying strictly-synchronized paths.

Theorem 3: Let P_j be the j-th path of an unbalanced multicast-tree as defined in *Lemma 1* $(1 \leq j \leq m-1)$. If this multicast tree is flow-controlled by SSP, then the following three claims are equivalent.

Claim 1. The number of P_j's RM-cells going through the initial-state, $k_j^* = 0$, where k_j^* is defined in *Theorem 2*;

Claim 2. P_j is strictly-synchronized;

Claim 3. P_j's RM-cell RTT attains the maximum: $\tau_u(j, \Delta) = \tau_{max} = 2m$. □

Remark 2: (1) The strictly-synchronized path has the largest RM-cell RTT, and hence, the number of strictly-synchronized paths should be minimized. (2) A larger Δ results in a larger number of strictly-synchronized paths, so a smaller Δ is desired.

Definition 2: Let W_j be the net waiting time for P_j's feedback RM-cell to synchronize with feedback RM-cells via the other paths at all consolidating branch-nodes along P_j. If $W_j = 0$, then P_j is said to be a *wait-free-synchronized* path. □

Clearly, P_{m-1} is always a wait-free-synchronized path since according to *Lemma 1*, a longer path never waits for feedback RM-cells via shorter paths for synchronization. Since P_{m-1} is both strictly-synchronized and wait-free-synchronized, we exclude P_{m-1} from all the following theorems and treat P_{m-1} separately. The theorem given below provides formulas to determine W_j and establishes an *iff* condition to identify wait-free-synchronized paths, all of which are affected by the value of Δ.

Theorem 4: Let P_j be the j-th path of an unbalanced multicast-tree as defined in *Lemma 1* $(1 \leq j \leq m-2)$ and W_j be the net waiting time for P_j's feedback RM-cell to synchronize with feedback RM-cells at all consolidating branch-nodes along P_j. If this multicast tree is flow-controlled by SSP, then for $1 \leq j \leq m-2$ the following claims hold:

Claim 1. P_j's net waiting time W_j for synchronization is upper bounded by Δ, and W_j is determined by:

$$W_j = 2(m-j-1) - k_j^* \Delta < \Delta; \tag{7}$$

where k_j^* is defined by Eq. (4) in *Theorem 2*;

Claim 2. If P_j is strictly-synchronized, then $W_j = 2(m-j-1) > 0$;

Claim 3. P_j is a wait-free-synchronized path, i.e., $W_j = 0$ iff $2\,(m-j-1)\ mod\ \Delta = 0$. □

Remark 3: (1) According to *Lemma 2*, the wait-free-synchronized path has the minimum RM-cell RTT. Thus, the number of wait-free-synchronized paths should be maximized. (2) A smaller Δ will lead to a larger number of wait-free-synchronized paths. So, a small Δ is desirable.

The theorem below classifies the entire multicast-tree path set into three exclusive categories, and provides the explicit expressions (as functions of Δ) for calculating the number of paths for each path-category.

Theorem 5: Let P_j be the j-th path of an unbalanced multicast-tree as defined in *Lemma 1* $(1 \leq j \leq m-2)$. If this multicast tree is flow-controlled by SSP, then the entire path set $\mathcal{P} \stackrel{\triangle}{=} \{P_1, P_2, P_3, \cdots, P_{m-3}, P_{m-2}\}$ is partitioned into a strictly-synchronized path subset $\mathcal{P}_S$, a wait-free-synchronized path subset $\mathcal{P}_N$, and a non-strictly-synchronized and non-wait-free-synchronized path subset $\mathcal{P}_W$, i.e., $\mathcal{P} = \mathcal{P}_S \oplus \mathcal{P}_N \oplus \mathcal{P}_W$, and furthermore, for $1 \leq \Delta \leq \tau_{max} = 2m$ the following claims hold:

Claim 1. The number of strictly-synchronized paths, denoted by S_Δ, is determined by: $S_\Delta \stackrel{\triangle}{=} \|\mathcal{P}_S\| = \lceil \frac{\Delta}{2} \rceil - 1$, where $\|\cdot\|$ denotes the cardinality of a set;

Claim 2. The number of wait-free-synchronized paths, denoted by N_Δ, is determined by:

$$N_\Delta \stackrel{\triangle}{=} \|\mathcal{P}_N\| = \begin{cases} \lfloor \frac{2(m-2)}{\Delta} \rfloor, & \text{if } \Delta = \text{even}; \\ \lfloor \frac{(m-2)}{\Delta} \rfloor, & \text{if } \Delta = \text{odd}; \end{cases} \tag{8}$$

Claim 3. The number of paths which are neither wait-free-synchronized, nor strictly-synchronized, denoted by W_Δ, is determined by:

$$\begin{aligned} W_\Delta &\stackrel{\triangle}{=} \|\mathcal{P}_W\| \\ &= \begin{cases} m - \lfloor \frac{2(m-2)}{\Delta} \rfloor - \lceil \frac{\Delta}{2} \rceil - 1, & \text{if } \Delta = \text{even}; \\ m - \lfloor \frac{(m-2)}{\Delta} \rfloor - \lceil \frac{\Delta}{2} \rceil - 1, & \text{if } \Delta = \text{odd}. \end{cases} \end{aligned} \tag{9}$$

□

Remark 4: (1) The number of strictly-synchronized paths is proportional to Δ. (2) The number of wait-free-synchronized paths is proportional to $\frac{1}{\Delta}$. (3) If $\Delta = 1$ or 2, then P_j is always a wait-free-synchronized path for all $j = 1, 2, \cdots, m-2$. (4) Taking $\Delta =$ even is preferable in terms of the number of wait-free-synchronized paths.

B. Discussion and Numerical Evaluation

According to *Theorem 5*, S_Δ is proportional to Δ while N_Δ is inversely proportional to Δ. Thus, a smaller Δ is desired since strictly-synchronized paths maximize RM-cell RTTs while wait-free-synchronization paths minimize RM-cell RTTs. Consider two extreme cases: Case 1: $\Delta = 1$ (i.e., there is an RM-cell

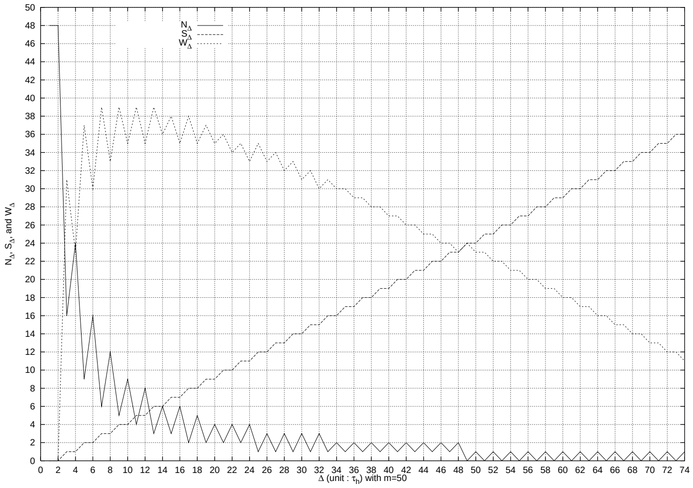


Fig. 4. N_Δ, S_Δ, and W_Δ vs. Δ with $m = 50$.

traversing per switch-hop) or 2, by *Theorem 5*, $S_\Delta = 1$ (P_{m-1} is always strictly-synchronized) and $N_\Delta = m - 1$ (P_{m-1} is always wait-free-synchronized), i.e., all paths of interest are wait-free-synchronized paths with minimal $\tau_u(j, \Delta) = 2(j+1)$; Case 2: $\Delta = \tau_{max} = 2m$, by *Theorem 5*, $S_\Delta = m - 1$ and $N_\Delta = 1$ (P_{m-1} is always wait-free-synchronized). However, the benefits of having larger N_Δ and smaller S_Δ do not come for free; the price paid for this is a large bandwidth cost for multicasting RM-cells at a higher RM-cell transmitting frequency $\frac{1}{\Delta}$. This introduces a trade-off between $\tau_u(j, \Delta)$ and bandwidth cost for RM-cells.

Theorem 5 suggests that selecting Δ to increase N_Δ is related to tree height m. As indicated by Eq. (8), to be able to take advantage of SSP, Δ should not be larger than $m-2$ in which case only P_{m-1} and possibly P_1 (when Δ = even) are wait-free-synchronized paths and about more than a half of paths are strictly-synchronized. In Fig. 4, N_Δ, S_Δ, and W_Δ are plotted against Δ with $m = 50$. We observe that (1) N_Δ decreases as Δ increases; S_Δ is proportional to Δ; W_Δ is not monotonic and reaches its peak value when $N_\Delta = S_\Delta$ and $\Delta \in [1, m-2]$. (2) When $\Delta > m - 2$, N_Δ becomes very small and flat fluctuating between 0 and 1; and on the other hand, when Δ decreases from $m - 2$ to 1, N_Δ increases dramatically. If τ_h is large enough, then taking Δ = 2 will result in the optimal case where all paths become wait-free-synchronized paths. In addition, we also observe that an *even* Δ is preferred since an even Δ gives a larger N_Δ than its neighbor values of *odd* numbers, which is consistent with Eq. (8). Thus, in general, Δ should be taken as an even number within the range of $[2, m-2]$.

Fig. 5 plots synchronization waiting-time W_j vs. path number j ($j + 1$ is P_j's length) while varying Δ. Although W_j is not a monotonic function of j for a given Δ, W_j increases on average as Δ rises. Thus, a smaller Δ is desired to minimize RM-cell RTTs on all paths. We also observe that W_j is a periodic function of j with the amplitude upper bounded by Δ, which verifies the Claim 1 of *Theorem 4*. In addition, for a given Δ, there are always some wait-free-synchronized paths ($W_j = 0$). For example, if $\Delta = 6$, there are $N_\Delta = 16$ wait-free-synchronized paths, which is consistent with *Theorem 5* and numerical results shown in Fig. 4 with $m = 50$. Furthermore, Fig. 5 also shows that a smaller Δ results in a larger number of wait-free-synchronized ($W_j = 0$) paths, N_Δ, which also verifies *Theorem 5*.

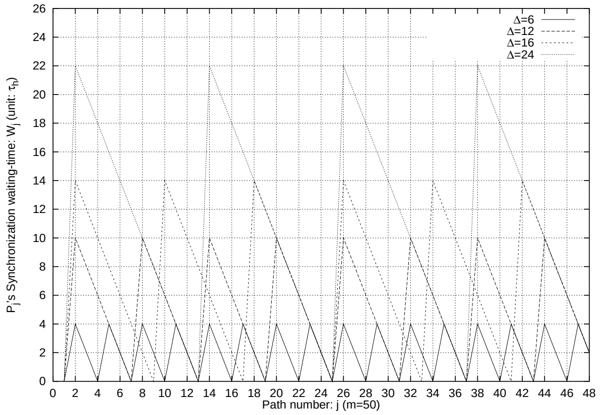


Fig. 5. P_j's synchronization waiting-time: W_j vs. path number: j ($m = 50$).

V. Conclusion

We presented an analytical technique to quantitatively characterize the delay performance of feedback synchronization algorithms. This technique was applied to analyze the feedback-delay properties of SSP and compare it with the hop-by-hop scheme. The analytical results showed that SSP outperforms the hop-by-hop scheme in terms of feedback-delay performance scalability in both balanced and unbalanced binary multicast-tree cases. We also derived the optimal RM-cell update interval for SSP to minimize RM-cell RTTs for a given multicast tree. The analytical results have been verified by the simulation for a number of simple cases. We are currently conducting extensive simulations to evaluate the feedback-delay performance of various feedback synchronization algorithms in more general multicast-network scenarios.

References

[1] X. Zhang, K. G. Shin, D. Saha, and D. Kandlur, "Scalable flow control for multicast ABR services," in *Proc. of IEEE INFOCOM*, March 1999. URL http://www.eecs.umich.edu/∼xizhang/papers/mcast.ps.

[2] J. Crowcroft and K. Paliwoda, "A multicast transport protocol," in *Proc. of ACM SIGCOMM*, pp. 247–256, August 1988.

[3] L. Roberts, *Rate Based Algorithm for Point to Multipoint ABR Service*, ATM Forum contribution 94-0772, September 1994.

[4] K.-Y. Siu and H.-Y. Tzeng, "On max-min fair congestion control for multicast ABR services in ATM," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 545–556, April 1997.

[5] H. Saito, K. Kawashima, H. Kitazume, A. Koike, M. Ishizuka, and A. Abe, "Performance issues in public ABR service," *IEEE Communications magazine*, vol. 11, pp. 40–48, November 1996.

[6] Y.-Z. Cho and M.-Y. Lee, "An efficient rate-based algorithm for point-to-multipoint ABR service," in *Proc. of GLOBECOM*, November 1997.

[7] W. Ren, K.-Y. Siu, and H. Suzuki, "On the performance of congestion control algorithms fo multicast ABR in ATM," *Proc. of IEEE ATM WORKSHOP*, August 1996.

[8] S. Fahmy, R. Jain, R. Goyal, B. Vandalor, and S. Kalyanaraman, "Feedbackback consolidation algorithms for ABR point-to-mulipoint connections in ATM networks," in *Proc. of IEEE INFOCOM*, April 1998.

[9] X. Zhang and K. G. Shin, "Feedback soft-synchronization for multicast ABR flow-control in ATM networks," *Technical Report, Real-Time Computing Laboratory, EECS Dept., The University of Michigan, Ann Arbor*, URL http://www.eecs.umich.edu/∼xizhang/papers/ssp.ps, January 1999.