

98-0150: Proposed Modifications to the Baseline Text and Living List on Multipoint ABR Behavior

**Sonia Fahmy, Raj Jain, Rohit Goyal,
and Bobby Vandalore**

Depart

**Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
<http://www.cse.wustl.edu/~jain/>**

iversity

[h](#)

[n/](#)



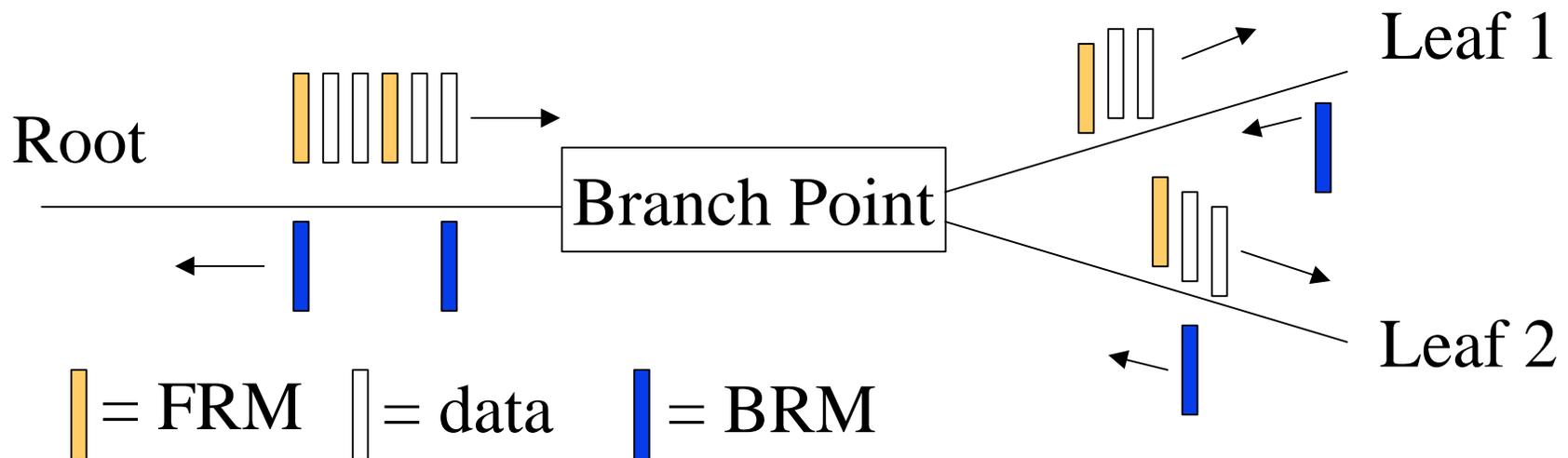
- q Pt-to-mpt ABR
- q Mpt-to-pt and mpt-to-mpt ABR

Pt-to-mpt ABR

- q There were no simulation results when Section 5.10.8 was written.
- q Now we have a better understanding.

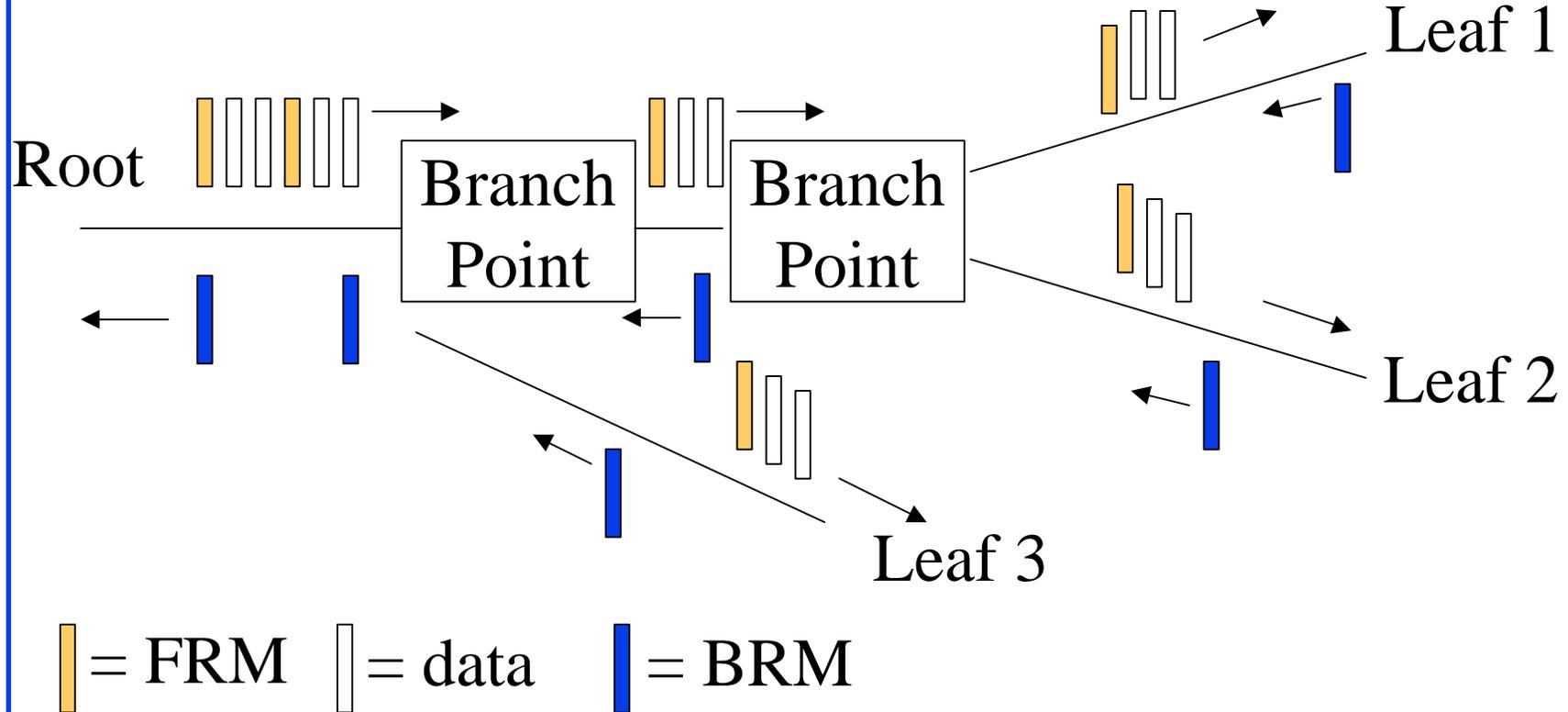
The Consolidation Operation

- q Necessary to prevent feedback implosion: too many BRMs per FRM at the root



Requirements [97-0615]

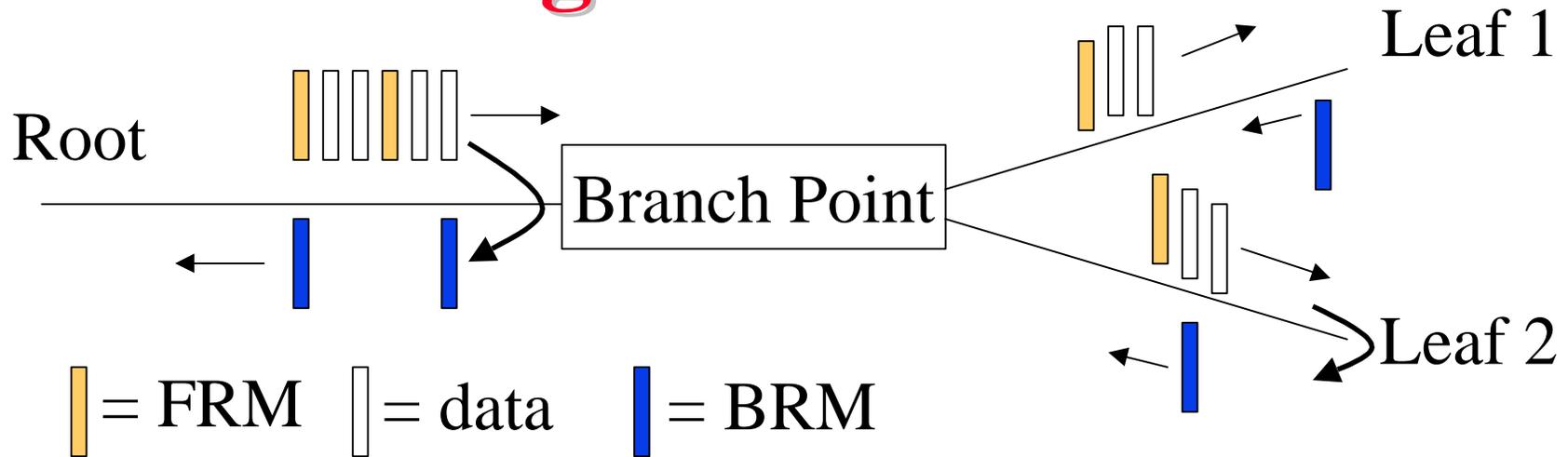
1. Scalability: Overhead and feedback delay should not increase with the number of leaves, branches or levels



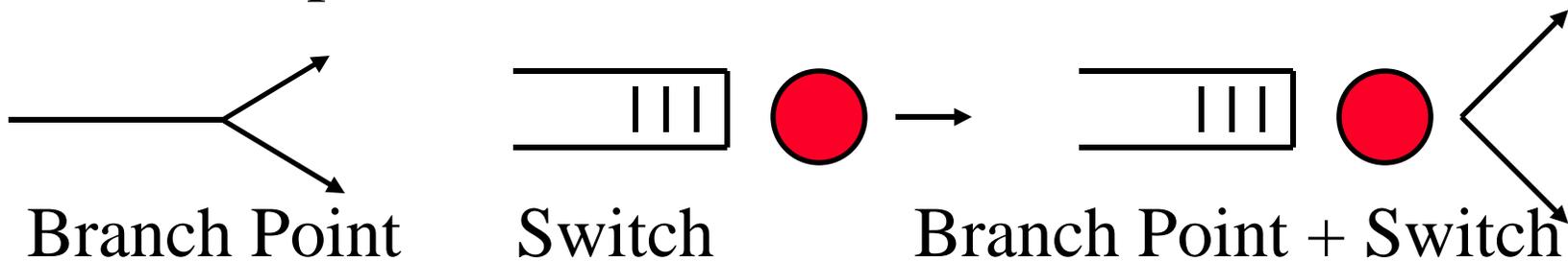
Requirements (cont)

2. Ratio of BRMs to FRMs inside the network and root should be close to 1.
3. Handling non-responsive branches and timeouts:
Algorithm should not halt nor cause
overload/underload
4. Consolidation noise, transient response, and
complexity should be minimal
⇒ May or may not want to wait for feedback from
all branches.

Design Alternatives



1. When to send BRM? On receiving FRM or BRM?
2. Interaction of branch point and switch operations if branch point is a switch?



Sample Algorithms

Algorithm	1	2	3	4	5	6	7
Complexity	High	High	Low	Med	>Med	>Med	>>Med
Transient Response	Fast	Med	Med	Slow	Fast for overload		Very fast for overld
Noise	High	Med	High	Low	Low	Low	Low
BRM:FRM	1	< 1	≤ 1	≤ 1	may>1	lim=1	lim=1
Sensitivity to branch points and levels	High	High	Low	Med	>Med	Med	Med

Pt-to-mpt ABR

- q Actions:
 - q Add general requirements of pt-to-mpt algorithms to Section 5.10.8 (Motion 1)
 - q Briefly describe sample algorithms in Appendix I (Motion 2)

Motion 1

Add this text at the end of the text for the second item in section 5.10.8.2:

The BRM consolidation method at the branch points needs to:

1. Scale well with the number of levels and with the number of branches in the multicast tree.
2. Ensure that the ratio of BRMs to FRMs in the network and at the root is maintained close to one.
3. Handle non-responsive branches such that they do not halt the consolidation operation nor cause overload or underload.
4. Exhibit: (a) minimal consolidation noise and consolidation delays, (b) fast transient response, (c) low complexity.

Motion 2

Add the following section to Informative Appendix I:

I.9 Sample Branch Point Algorithms For Multipoint ABR Flow Control:

A branch point replicates cells from the root to each branch in the responding state and consolidates their feedback. Sample consolidation algorithms are given next.

One method of consolidating information from BRM cells is to assign the ER field in returning RM cells to the minimum of the ER values indicated by the branches, the CI to the OR of the indicated CI values, and the NI to the OR of the NI values.

In a simple point-to-multipoint ABR algorithm [14] (references may be removed in the specifications), the minimum explicit rate indicated by the BRM cells received from the branches is maintained, say as MER. Whenever an FRM cell is received, it is multicast to all branches, and a BRM is returned using the MER value for the BRM explicit rate. MER is then set to PCR. A simple enhancement to reduce noise in this algorithm is to only generate the BRM cell if a BRM has been received from at least one leaf after the last BRM was sent by the branch point [16].

To reduce the complexity of the algorithm, some of the backward RM cells generated by the destinations can be forwarded, instead of turning around the RM cells at the branch points. Whenever an FRM cell is received at a branch point, the algorithm simply sets a flag indicating the receipt of the FRM cell, and multicasts it to all branches. When a BRM cell is received from a branch, it is passed back to the source (after using the minimum allocation), only if the flag was set. The flag and the MER register are then reset [10].

To reduce consolidation noise, the BRM cell can only be passed back when BRM cells from all branches have been received after the last feedback. This can be easily implemented by maintaining a separate flag for each branch to indicate if a BRM cell has been received from the branch after the last BRM cell was sent. It is necessary to handle the possible non-responsiveness of a branch by implementing timeouts in this algorithm.

In addition, the transient response of this algorithm may be slow due to waiting for feedback from possibly distant leaves. This delay can be avoided when a severe overload situation has been detected. In this case, there is no need to wait for feedback from all the branches, and the overload can be immediately indicated to the source [4].

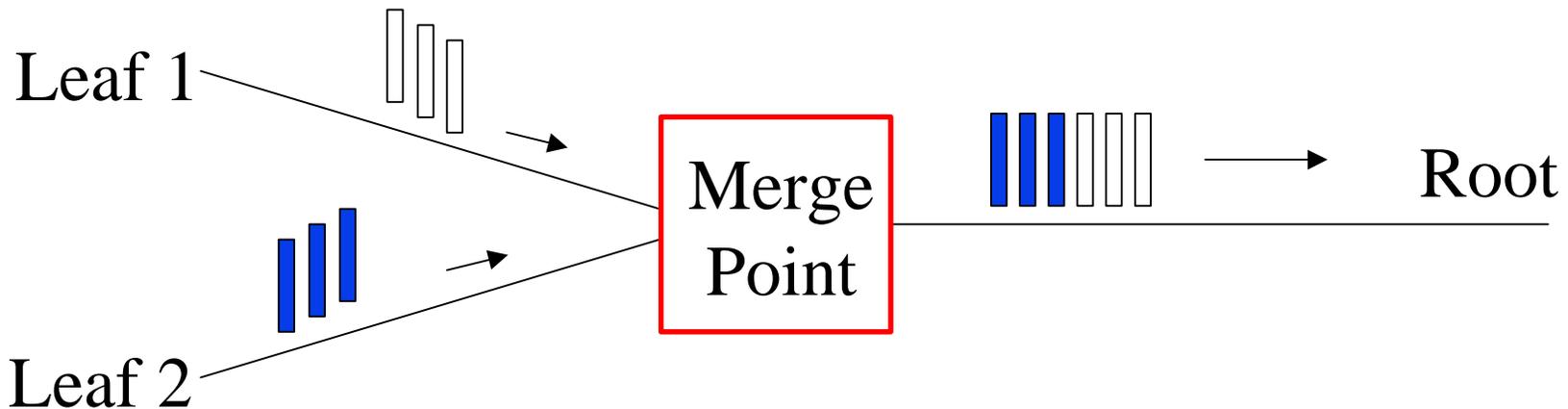
In the cases when the branch point is itself a switch and queuing point, the branch point can invoke the switch scheme whenever a BRM is received, and not just when a BRM is being sent. Hence, overload at the branch point itself can be detected and indicated according to the fast overload indication idea.

The fast overload indication idea may increase the BRM cell overhead, since the ratio of source-generated FRM cells to BRM cells received by the source can exceed one. To alleviate this problem, a counter (maintained for each multipoint VC) can be incremented whenever a BRM cell is sent before feedback from all branches has been received. When feedback from all branches indicates underload, and the value of that counter is more than zero, this particular feedback can be ignored and the counter decremented [4].

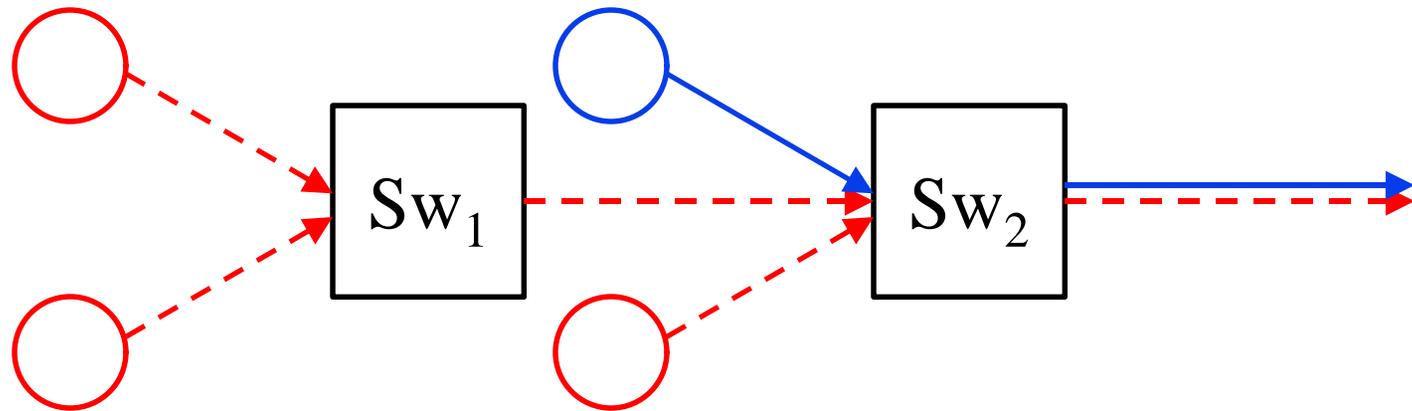
Multipoint-to-Point VCs

[97-0832]

- q A multipoint-to-point VC can have more than one concurrent sender
- q Traffic at root = Σ Traffic originating from leaves



Sources, VCs, and Flows



- q Sw₂ has to deal with
 - q Two VCs: Red and Blue
 - q Four sources: Three red sources and one blue source
 - q Three flows: Two red flows and one blue

Fairness Definitions

- q Source-based: N-to-one connection = N one-to-one connections \Rightarrow Use max-min fairness among sources
- q VC/Source-based:
 1. Allocate bandwidth fairly among VCs
 2. For each VC, allocate fairly among its sources
- q Flow-based: Flow = VC coming on an input link. Switch can easily distinguish flows.
- q VC/Flow-based:
 1. Allocate bandwidth fairly among VCs
 2. For each VC, allocate fairly among its flows

Mpt-to-pt ABR

- q Actions: (Motion 3)
 - q Create a living list item on Mpt-to-pt
 - q Add a sample merge point algorithm
(Applies to mpt-to-mpt also)
 - q Move fairness definitions from 96-004 to this item. (96-004 mixes parameter signaling with fairness)

Motion 3

Add a separate item for flow control for multipoint-to-multipoint connections as follows:

Title: Flow control for ABR multipoint-to-multipoint connections

Problem Statement: Define the desirable forms of fairness

for multipoint connections, and extend current switch algorithms for multipoint connections. Conduct a performance analysis to examine the fairness, complexity, overhead, transient response, delays, and scalability tradeoffs involved. Interoperability must also be studied.

Solution Requirements: Fairness, low overhead, fast response, scalability.

Item Introduced: February 1998

Last Updated: February 1998

Current Status: Under Study

Other Working Groups: SIG, PNNI

Contribution Log: 97-0832, 97-1085R1, 98-0150

Work To Be Done:

Baseline Text:

Multipoint ABR Flow Control

Sample merge point algorithms for multipoint-to-point connections are given below. Multipoint-to-multipoint connections can be handled by combining a point-to-multipoint (branch point) algorithm with a multipoint-to-point (merge point) algorithm.

Merge points must ensure that BRM cells are sent to the appropriate sources at the appropriate times. These algorithms should maintain the BRM to FRM ratio at the sender and inside the network close to one. They should be simple, scalable, and minimize noise and delays. With multipoint-to-point and multipoint-to-multipoint connections, the implicit assumption that each connection has only one source is no longer valid.

Fairness Definitions

Four different types of fairness can be defined for multipoint-to-point and multipoint-to-multipoint connections:

1. *Source-based fairness*, which divides bandwidth fairly among active sources as if they were sources in point-to-point connections, ignoring group memberships.
2. *VC/source-based fairness*, which first gives fair bandwidth allocations at the VC level, and then fairly allocates the bandwidth of each VC among the active sources in this VC.
3. *Flow-based fairness*, which gives fair allocations for each active flow, where a flow is a VC coming on an input link. Formally,
$$\text{NumFlows}_j, j \in \text{OutputPorts}, = \forall i, i \in \text{InputPorts}, \Sigma_i \text{ Number of VCs coming on port } i \text{ and being switched to port } j$$
4. *VC/flow-based fairness*, which first divides the available bandwidth fairly among the active VCs, and then divides the VC bandwidth fairly among the active flows in the VC.

Sample Merge Point Algorithm for Source-Based Fairness

An example merge point algorithm for source-based fairness can operate as follows [13]. The algorithm maintains a flag at the merge point for each of the flows being merged. The flag indicates that an FRM has been received from this flow after a BRM had been sent to it. Therefore, when an FRM is received at the merge point, it is forwarded to the root and the flag is set. When a BRM is received at the merge point, it is duplicated and sent to the branches that have their flag set, and the flags are then reset.

Switch Scheme Restrictions for VC Merge Switches:

Source-based fairness algorithms operating in VC merge switches need to consider the following issues:

1. Per source accounting should not be performed. For example, measuring the rates or activity for each source, or distinguishing overloading and underloading sources should not be performed. The algorithm can use the information supplied in RM cells, in addition to aggregate measurements such as load, capacity and queuing delays. If accounting is performed at the VC level or at the flow level, an additional mechanism to divide VC or flow bandwidth among sources is necessary.
2. CCR values from BRM cells should not be used in computing rate allocations for sources in multipoint connections, since the CCR value can be that of another source that does not go through the switch performing the computation. CCR values from FRM cells can be used to compute rate allocations for sources in multipoint connections, even though the CCR used to compute the rate for a source may not actually be the CCR value of the source. The maximum CCR value seen during an interval can also be used instead of the CCR of the source.