97-0615: Feedback Consolidation Algorithms for ABR Point-to-Multipoint Connections

Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, Shivkumar Kalyanaraman

Department of CIS, The Ohio State University

Sastri Kota, Lockheed Martin Telecommunications

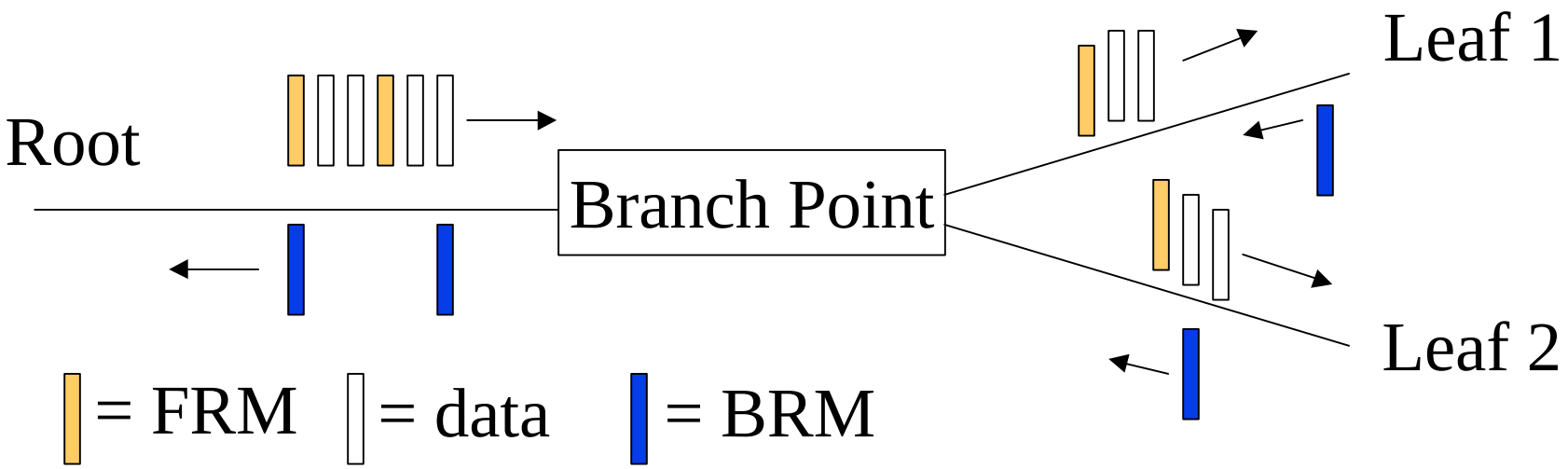
Pradeep Samudra, Samsung Telecom America, Inc.

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

- Consolidation algorithm design
- Previous 4 algorithms
- Proposed 3 algorithms
- Simulation results
- Performance comparison

The Consolidation Operation

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

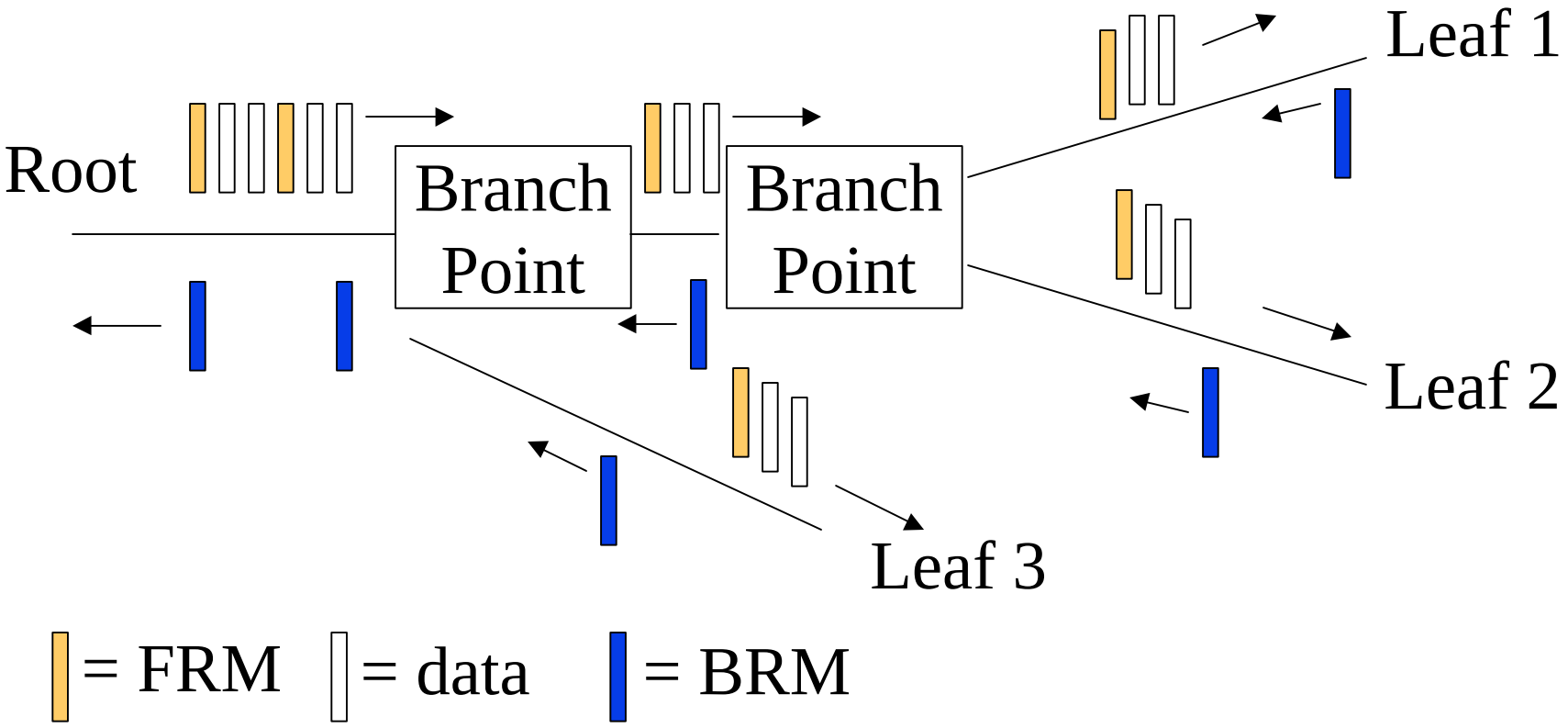


Design Issues

- Who generates BRMs: branch points or leaves?
- Wait for feedback from all branches?
- Control of ratio of BRMs to FRMs at the root?
- Ratio of BRMs to FRMs inside the network?
- Interaction of branch point and switch operations if branch point is a switch?
- Which values are stored per VC and which per branch?
- Handling non-responsive branches and timeouts? Algorithm should not halt nor cause overload/underload
- Consolidation delay and scalability?

Scalability

- Overhead (# BRMs at root and inside network) and feedback delay should not increase with the number of leaves, branches or levels



Previous Algorithms

- **Algorithm 1:** Simply turn around RM cells with the current minimum and reset minimum
- **Algorithm 2:** Turn around FRM only if at least one BRM has been received since last BRM was sent
- **Algorithm 3:** Do not turn around RM cells. Simply flag the receipt of the FRM, and return the first BRM (with modified fields) to arrive after that
- **Algorithm 4:** Wait till BRMs are received from all branches after last BRM was sent, and return the last one (with modified fields)

New Algorithms

- **Goals:**
  - Eliminate consolidation noise, but not at the expense of a very slow transient response
  - Transient response must be fast in the case of overload
- **Algorithm 5:** If the ER in the BRM is *much less* than the last ER sent (or CCR), do not wait ⇒ send the BRM, but do not reset the values: reset when feedback from all leaves is received
- **Problem:** BRM to FRM ratio at the root may exceed one

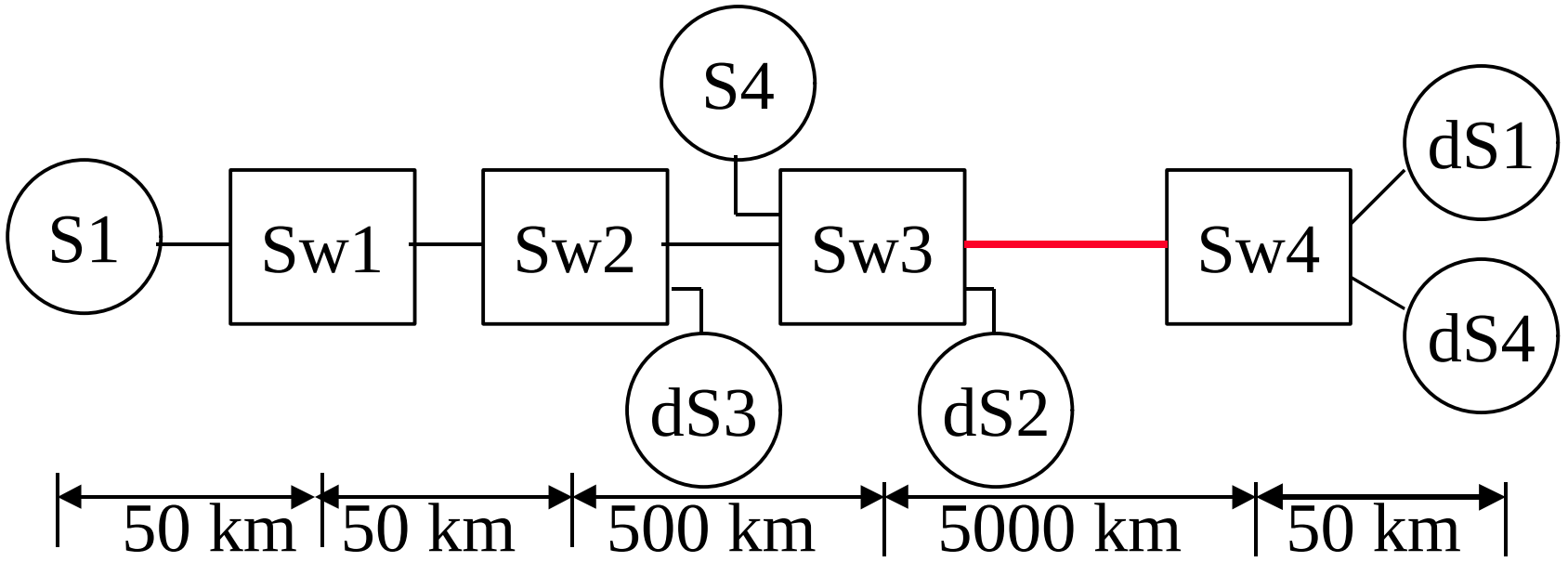
New Algorithms (Cont)

- **Solution ⇒ Algorithm 6:** For every premature RM cell, increment a counter. Decrement the counter the next time an RM giving a higher rate than the last sent is to be returned, but do not return the RM
- **Another Problem:** What if the branch point is a switch, and it is overloaded?
- **Solution ⇒ Algorithm 7:** When a BRM is received at the branch point, invoke the switch algorithm for the branches before checking if there is overload or not

Simulation Parameters

- Links: WAN, 155.52 Mbps (149.76 Mbps after SONET)
- Traffic: unidirectional; bursty, persistent and with and without (on/off) VBR background
- Source: Parameters selected to *maximize* ACR
  Initial Cell Rate = PCR
  Rate Increase Factor = 1 $\Rightarrow$ ACR is not limited
  TBE = very large
- Switch: ERICA algorithm
  Target utilization = 90%
  Averaging interval = min{100 cells, 1 ms}

Configuration 2

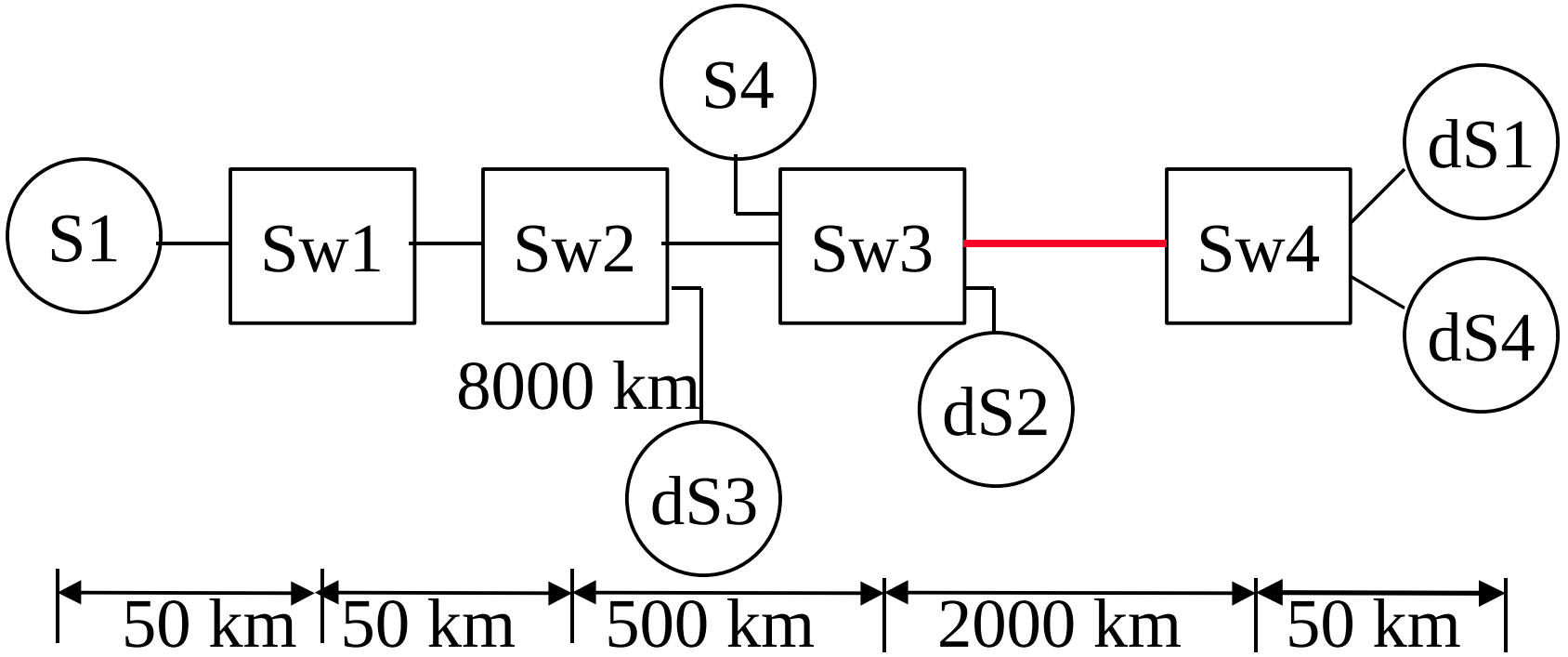


- Persistent S1 sends to dS1, dS2, dS3. S4 sends to dS4
- Chain configuration: Bottleneck link only on route to distant leaf leaves ⇒ all branches except longest branch (to dS1) give PCR as ER

Simulation Results 2

- Algorithms 1, 2, 3: noise, unfair, unstable
- Algorithms 4, 5, 6: no noise, but slow response
- Algorithm 7: no noise and fast response

Configuration 3



- Modified chain configuration: Bottleneck feedback is closer than other leaves. Non-bottlenecked feedback comes from far away

Simulation Results 3

- Algorithm 4: slow transient response
- Algorithms 5, 6: much faster response
- Algorithm 7: fastest
- Similar results with configurations with 10 leaves at different switches

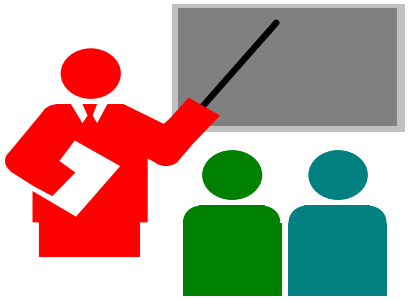
Performance Comparison

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

Performance Comparison (Cont)

- Algorithms 1 and 2 do not perform well and are complex
- Algorithm 3 suffers from consolidation noise
- Algorithm 4 has a slow transient response
- Algorithms 5, 6, and especially 7 overcome this problem

Conclusions



- Consolidation algorithms offer tradeoffs between complexity, transient response, noise, overhead and scalability
- The new algorithms 6 and 7 speed up the transient response, while eliminating consolidation noise and controlling overhead