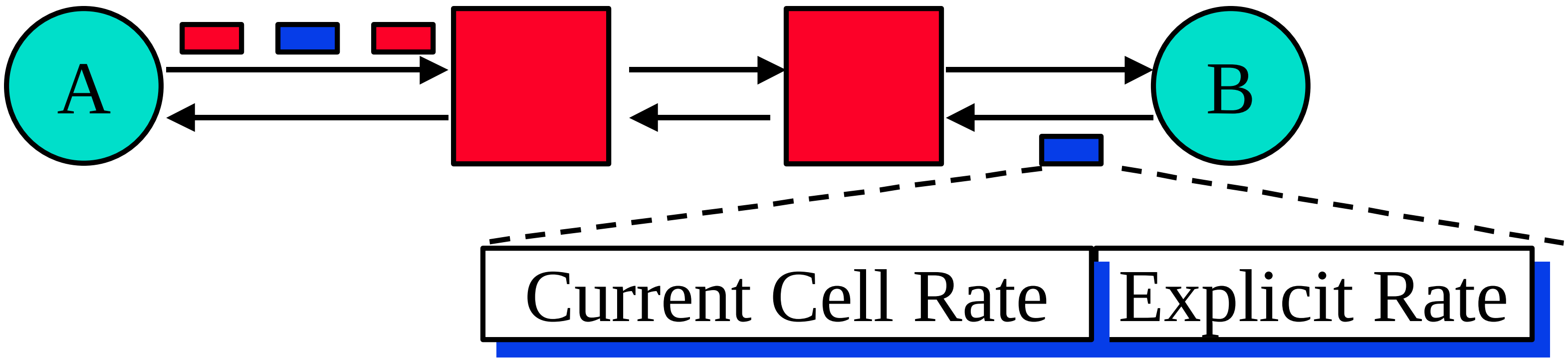
Switch Algorithms for Multipoint ABR Service over ATM Networks

Raj Jain

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

- Introduction to point-to-multipoint ABR
- Basic ABR pt-mpt Resource Allocation
- Extension/optimization of pt-mpt algorithms
- Mpt-pt: What should be the goal of allocation?
- Extension of ERICA to mpt-pt

Point-to-Point ABR

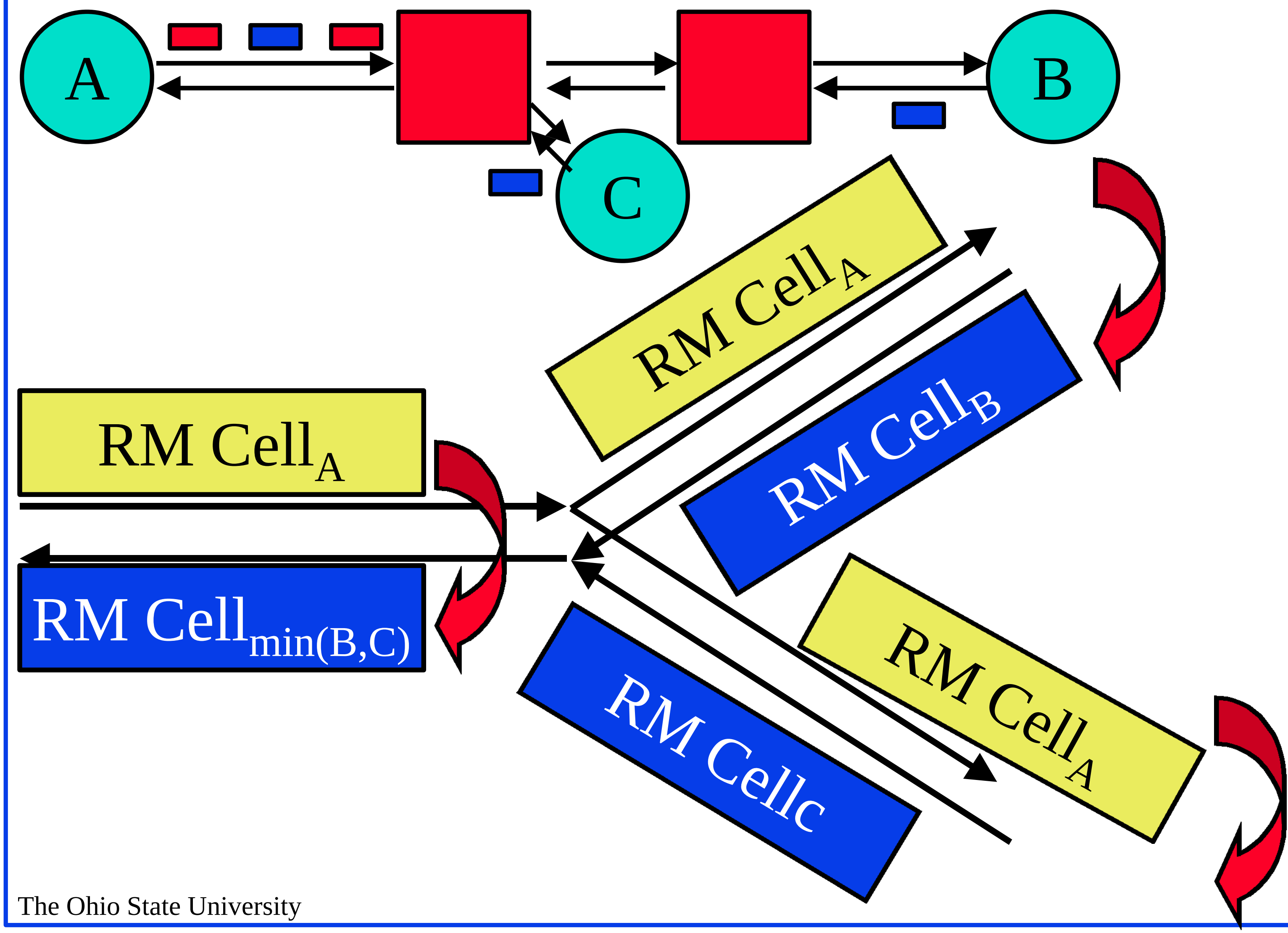


- Sources send one RM cell every n cells
- The RM cells contain "Explicit rate"
- Destination returns the RM cell to the source
- The switches adjust the rate down
- Source adjusts to the specified rate

ERICA+

- Time is slotted into averaging intervals
- ABR capacity = [link capacity – (VBR + CBR load)] × f(queue length)
- Estimate input rate = Σ CCRj
- overload = input rate/ABR capacity
- ERj_efficiency = CCRj/overload
- ER_fairshare = ABR capacity/# of active sources
- IF overload ≤ 1+ δ THEN ERj =
  max (ERj_efficiency, ER_fairshare, maxERprevious)
  ELSE ERj = max(ERj_efficiency, ER_fairshare)
- maxERcurrent = max(maxERcurrent, ERj)
- ER in BRMj = min(ER in BRMj, ERj)

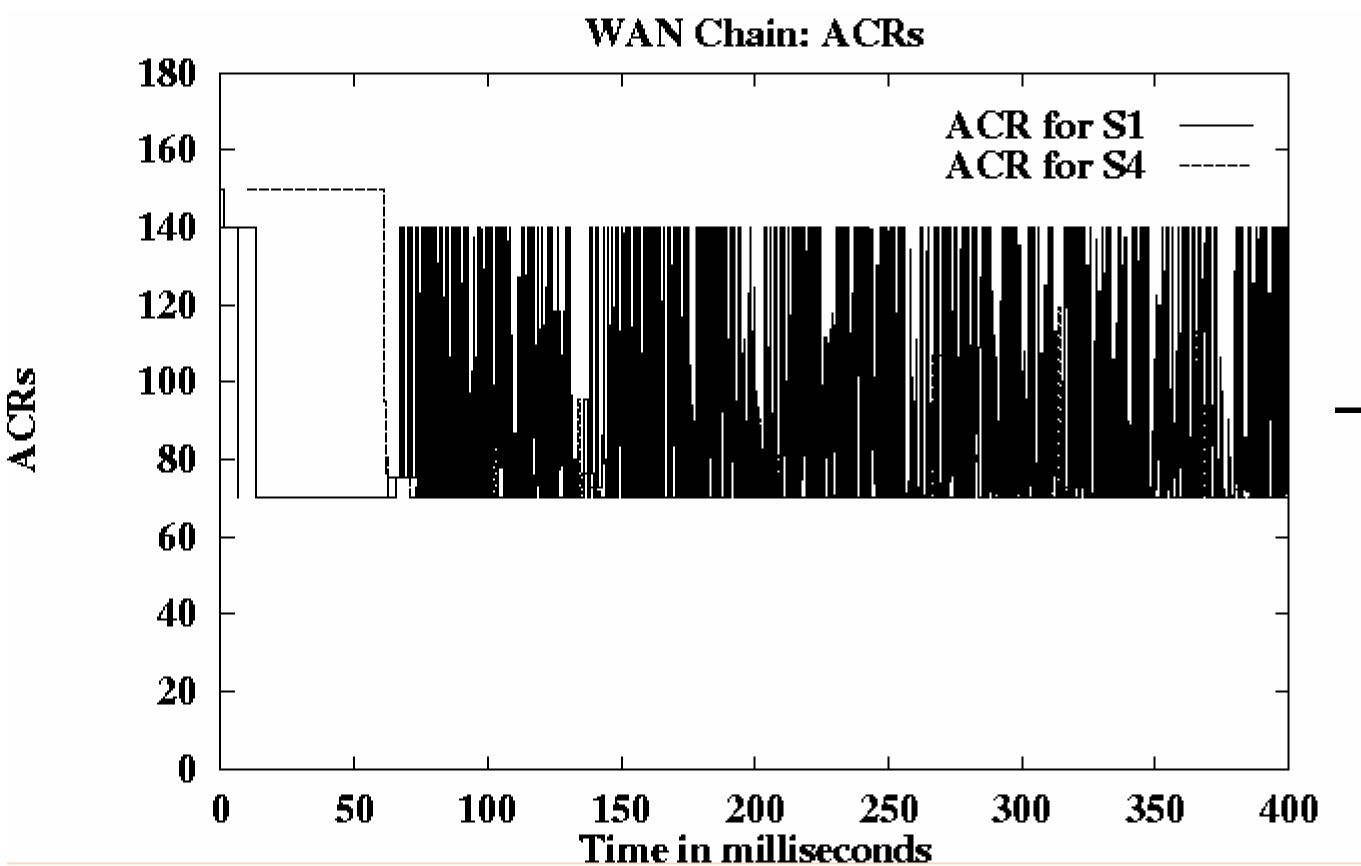
Point-to-Multipoint ABR

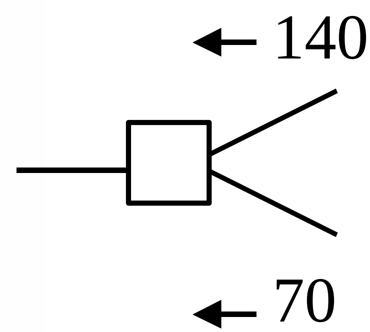


Basic Pt-Mpt: Results

- ABR with ERICA (extended for multipoint) works ok
- Efficiency, fairness, responsiveness is maintained
- Consolidation noise due to asynchronous arrival of feedback from different leaves appears as oscillations
- Additional delay due to FRM wait and BRM consolidation
  $\Rightarrow$ slower transient response than point-to-point
- Minimum of all paths is allocated
  $\Rightarrow$ Some links are underutilized
- Queue control (ERICA+) is required for stability

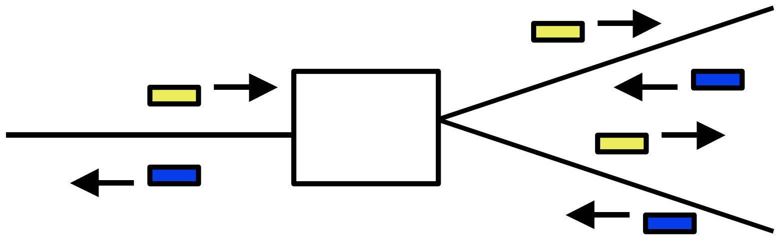
Consolidation Noise





- Feedback oscillates between 70 and 140.

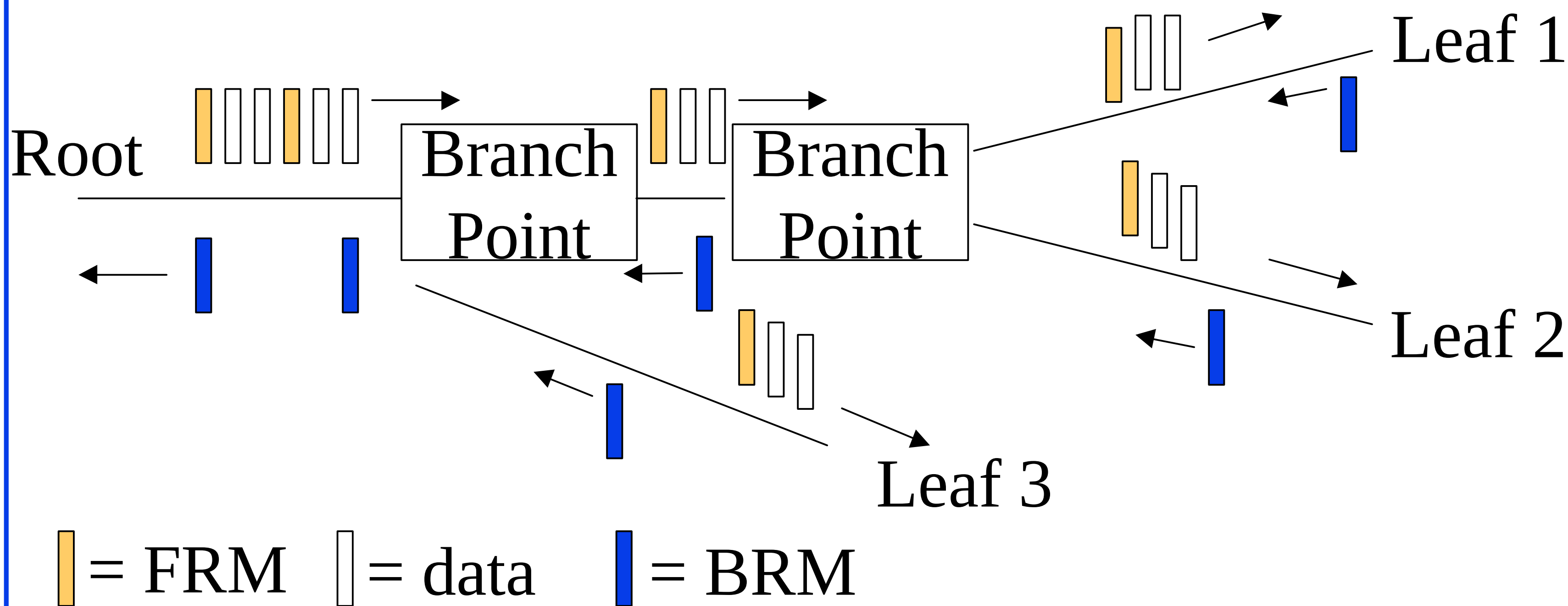
Point-to-Multipoint Connections: Issues



- If you send BRM on every FRM, you may give feedback without receiving any
  ⇒ Need to ensure that at least one feedback has been received before sending a BRM.
  Otherwise, you may give PCR
- Not all downstream feedbacks in an upstream feedback ⇒ consolidation noise
- Conclusion: Feedback should not be FRM driven

Scalability

- If the feedback is BRM driven:
  Should we wait for BRMs from all branches?
  Yes ⇒ Delay may be long. Non-responsive branches?
  No ⇒ Number of BRMs >> FRMs



Previous Algorithms

- **Algorithm 1:** Simply turn around FRM cells with the current minimum and reset minimum
  - Feedback may be sent without receiving any
  - Partial feedback ⇒ Noise
- **Algorithm 2:** Turn around FRM only if at least one BRM has been received since last BRM was sent
  - Solves "no feedback problem" but has noise
- **Algorithm 3:** Do not turn around FRM cells. Simply flag the receipt of the FRM, and return the first BRM (with modified fields) to arrive after that
  - Solves "no feedback problem" but has noise

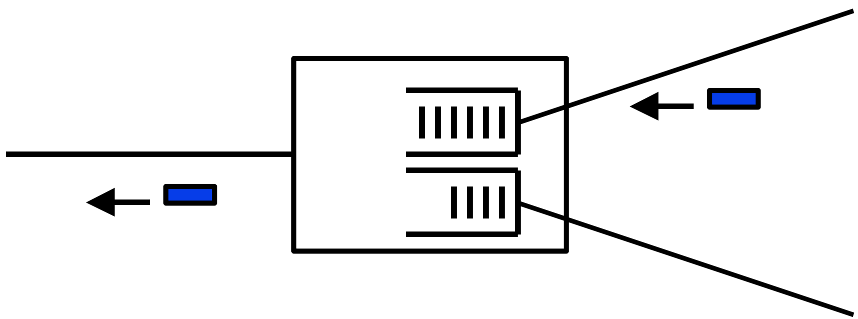
- **Algorithm 4:** Wait till BRMs are received from all branches after last BRM was sent, and return the last one (with modified fields)
  - Transient response too slow

New Algorithms

- **Algorithm 5 (new):** 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
  - BRM to FRM ratio may exceed one
- **Algorithm 6:** For every premature BRM cell, increment a counter. Decrement the counter the next time a BRM giving a higher rate than the last sent is to be returned, but do not return the BRM
  - Overload at the current switch may not be fedback in a timely manner

New Algorithms (Cont)

- **Algorithm 7:** When a BRM is received, invoke the switch algorithms for all outgoing branches before deciding whether to send feedback



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

Performance Comparison

- Studied 4 existing and 3 new algorithms.

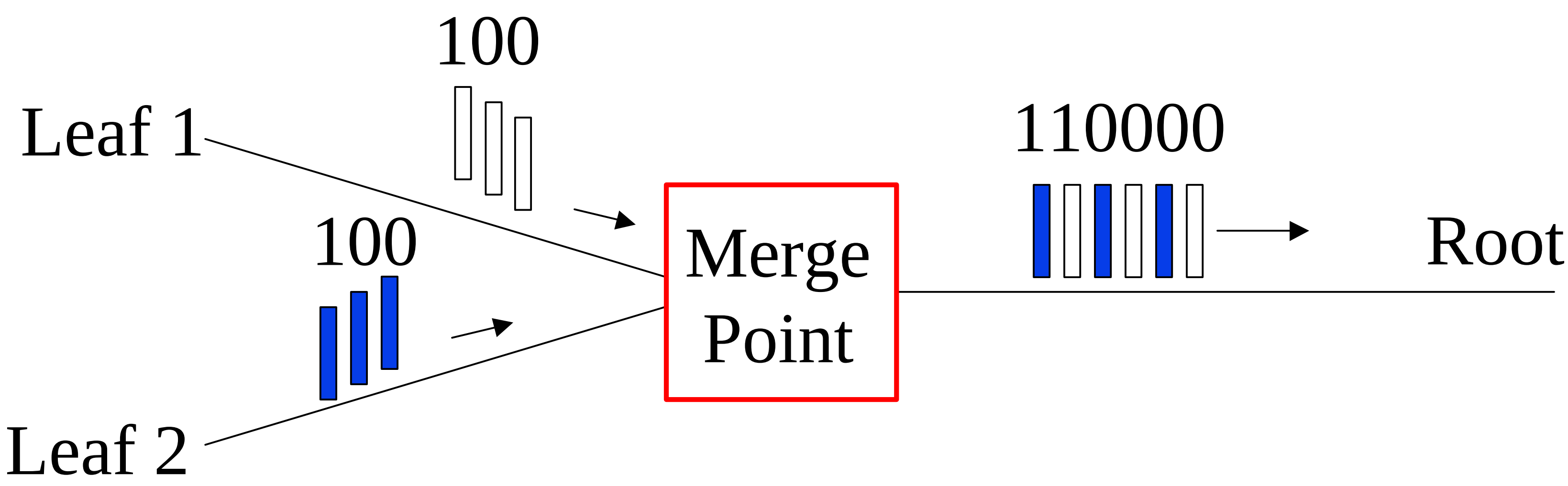
Algorithm	1	2	3	4	5	6	7
Complexity	**High**	**High**	Low	Med	>Med	>Med	>>Med
Transient Response	Fast	Med	Med	**Slow**	Fast for overload		Very fast for overld
Noise	High	Med	**High**	Low	Low	Low	Low
BRM:FRM	1	< 1	≤ 1	≤ 1	**may>1**	lim=1	lim=1
Sensitivity to branch points and levels	High	High	Low	Med	>Med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

Multipoint Consolidation: Results

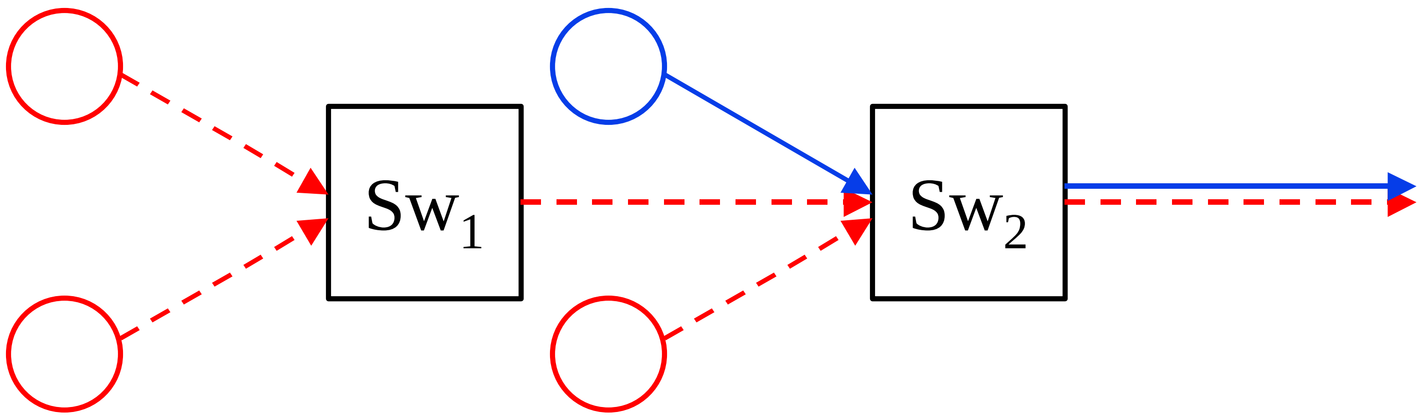
- Consolidation algorithms offer tradeoffs between complexity, transient response, noise, overhead and scalability
- The new algorithms 6 and 7 speed up the transient response, while eliminating consolidation noise and controlling overhead

Multipoint-to-Point VCs

- Problem with AAL5: Cell interleaving.
- VP merge: VCI = sender ID
  VPs are used for other purposes.
- VC merge: Buffer at merge point till EOM bit = 1.



Sources, VCs, and Flows



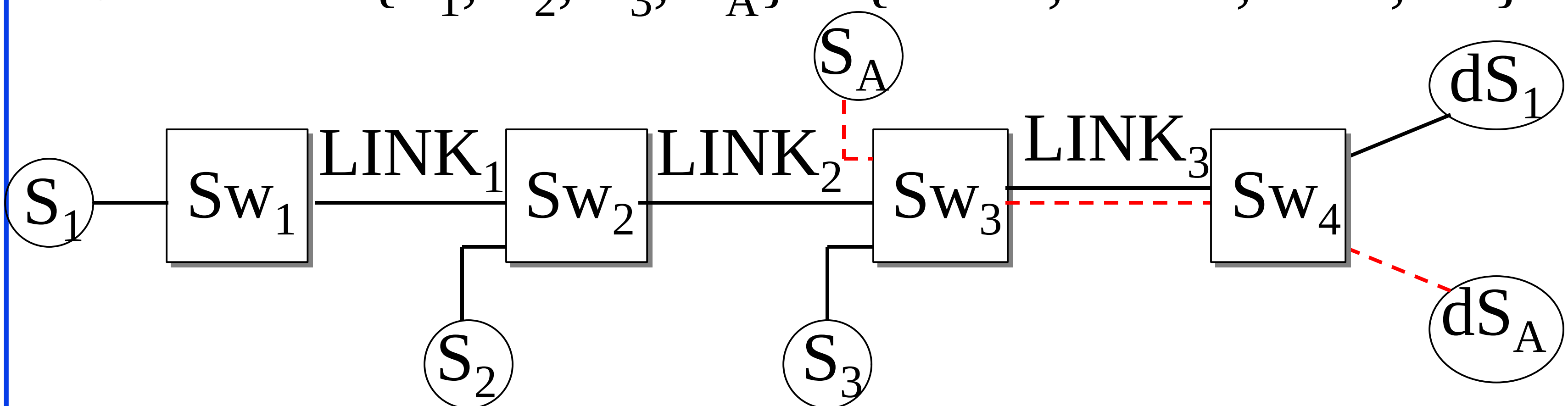
- Sw_2 has to deal with
  - Two VCs: Red and Blue
  - Four sources: Three red sources and one blue source
  - Three flows: Two red flows and one blue

Fairness Definitions

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

Example

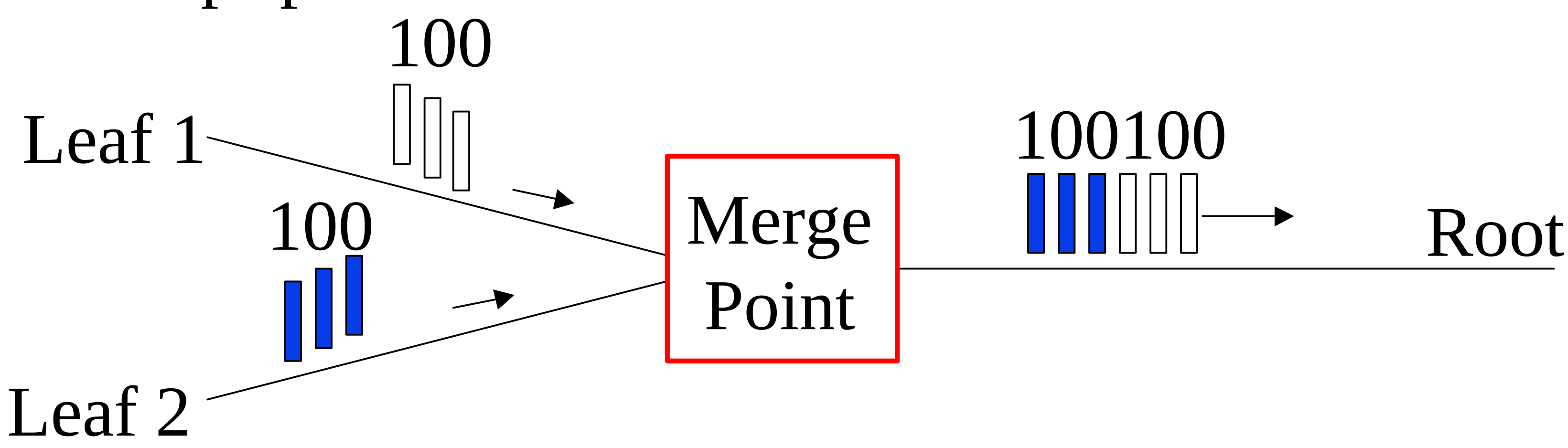
- How is the bandwidth of $LINK_3$ allocated?
- Source: $\{S_1, S_2, S_3, S_A\} \leftarrow \{37.5, 37.5, 37.5, 37.5\}$
- VC/Source: $\{S_1, S_2, S_3, S_A\} \leftarrow \{25, 25, 25, 75\}$
- Flow: $\{S_1, S_2, S_3, S_A\} \leftarrow \{25, 25, 50, 50\}$
- VC/Flow: $\{S_1, S_2, S_3, S_A\} \leftarrow \{18.75, 18.75, 37.5, 75\}$



All links are 150 Mbps

Mpt-pt Issues

- Cells of senders in the same multipoint-to-point VC cannot be distinguished
- Question: Can we achieve source-based fairness? Answer: Yes!
- We extended ERICA to achieve source based fairness for mpt-pt VCs

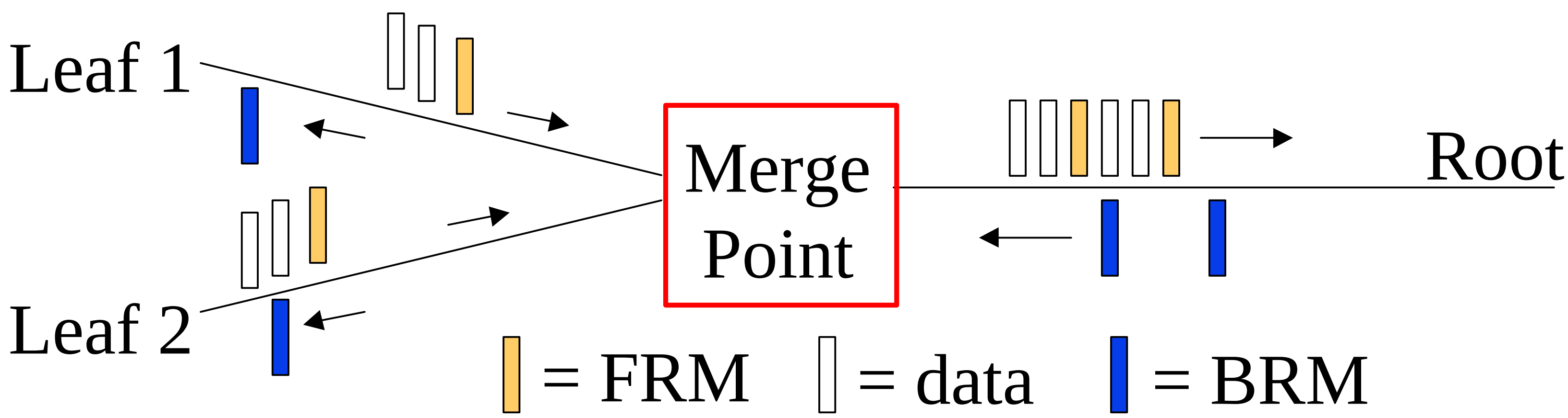


Changes to ERICA+

- Remove fair share term (# active sources)
- Options:
  - Use CCRjmax instead of CCRj
    Maximum is calculated in successive intervals
  - To minimize oscillations, use exponential averaging options for:
    - Input rate
    - ABR capacity
    - maxERprevious

Merging Point Algorithm

- Maintain a bit at the merging point for each flow being merged
  Bit = 1 ⇒ FRM received from this flow after BRM sent to it
- BRMs are duplicated and sent to flows whose bits are set, then bits are reset

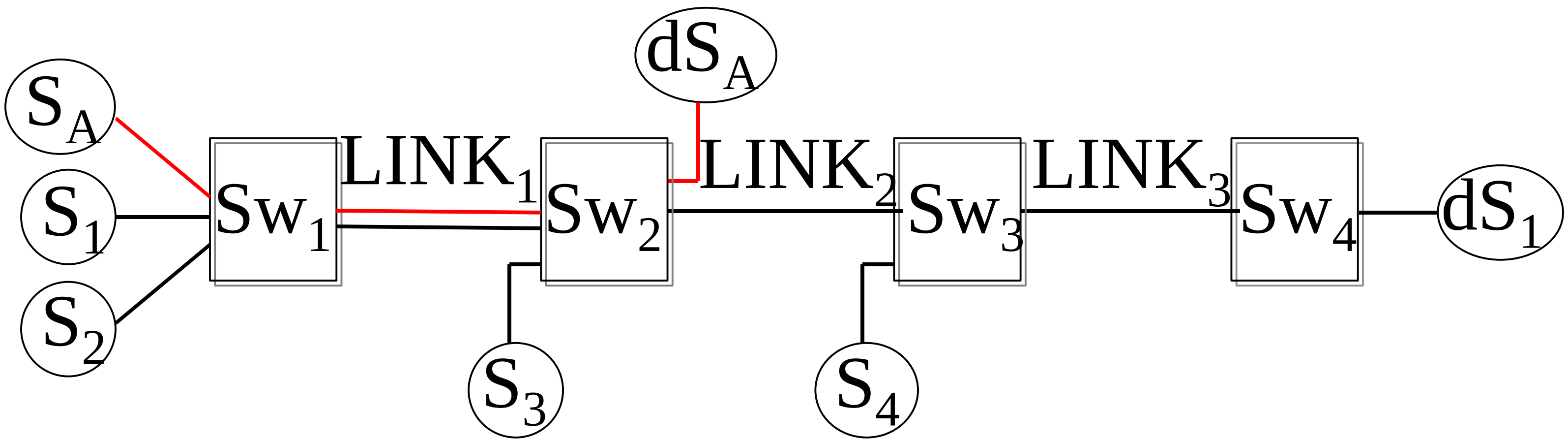


Simulation Parameters

- Unidirectional traffic
- RIF = 1/32, 1
- Rule 6 disabled
- Queue control: a = 1.15, b = 1, drain limit = 50%, target queuing delay = 1.5 s
- Measurement interval = 5 ms, 200 μs
- One cell long packets (Avoids VC merging issues)
- Max CCR and averaging maxERprevious used
- Link lengths in kms: {LINK1, LINK2, LINK3} = {50, 500, 5000}, {5000, 500, 50}

Upstream Bottleneck

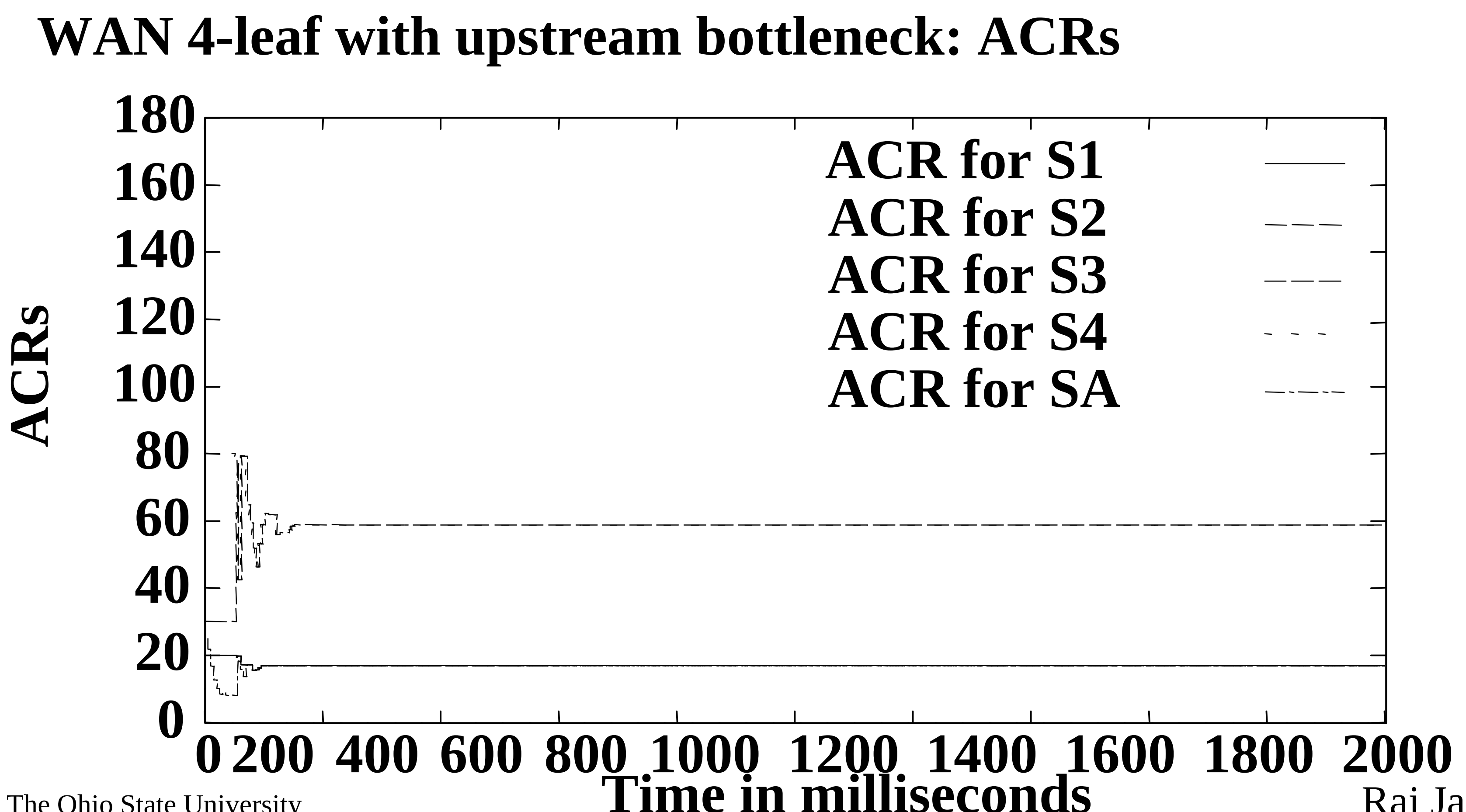
- Goal:{S1,S2,S3,S4,SA} ←{16.7,16.7,58.3,58.3,16.7}
- ICRs:{S1,S2,S3,S4,SA}←{20,20,30,80,10}
- Results are similar with different link lengths, RIF = 1/32, 1, interval length = 5 ms, 200 μs (no RMs for S1,S2 ,SA for 4 intervals; for S3,S4 for 1 interval)



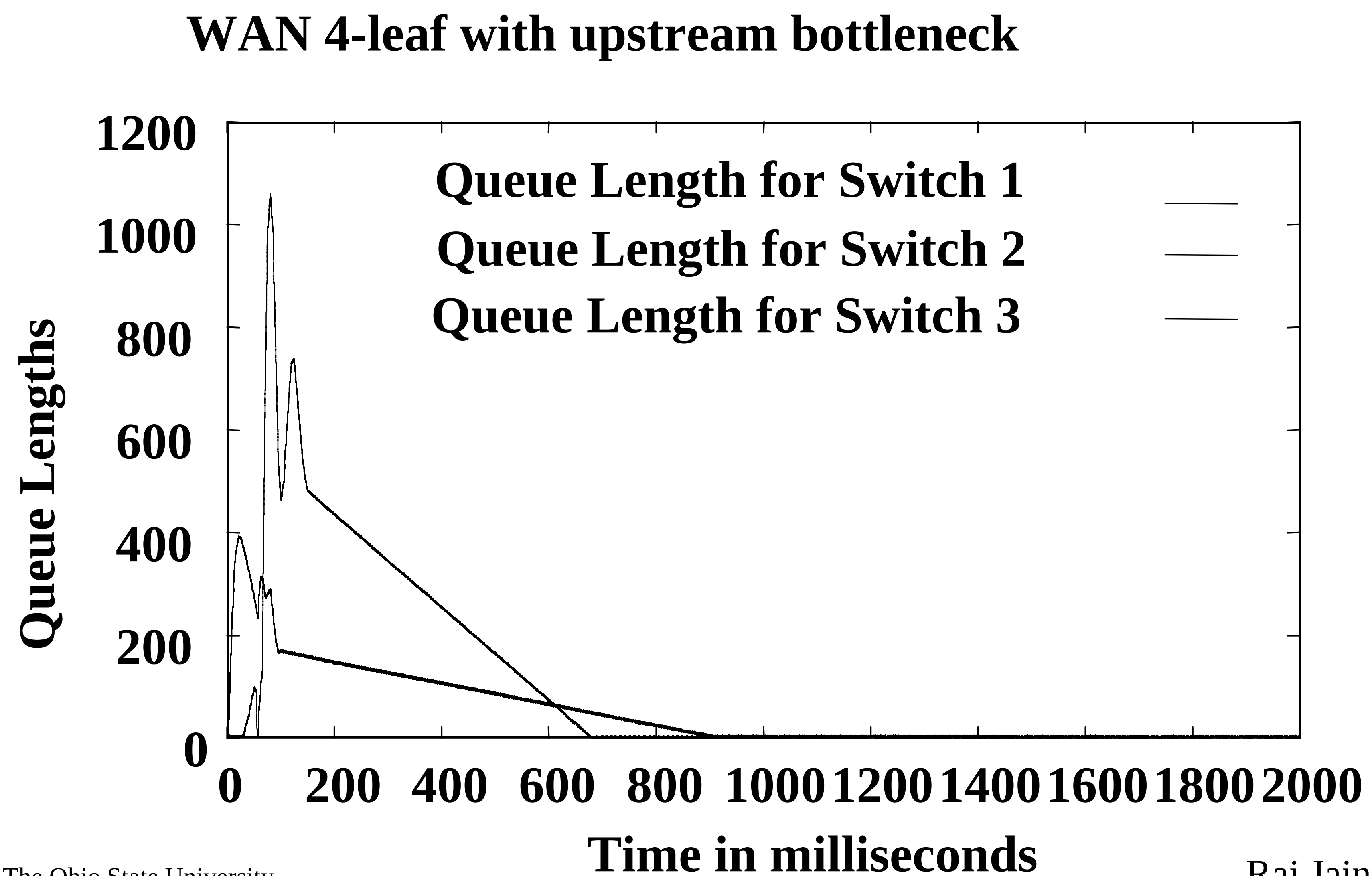
All links are 150 Mbps, except $LINK_1$ which is 50 Mbps

Simulation Results

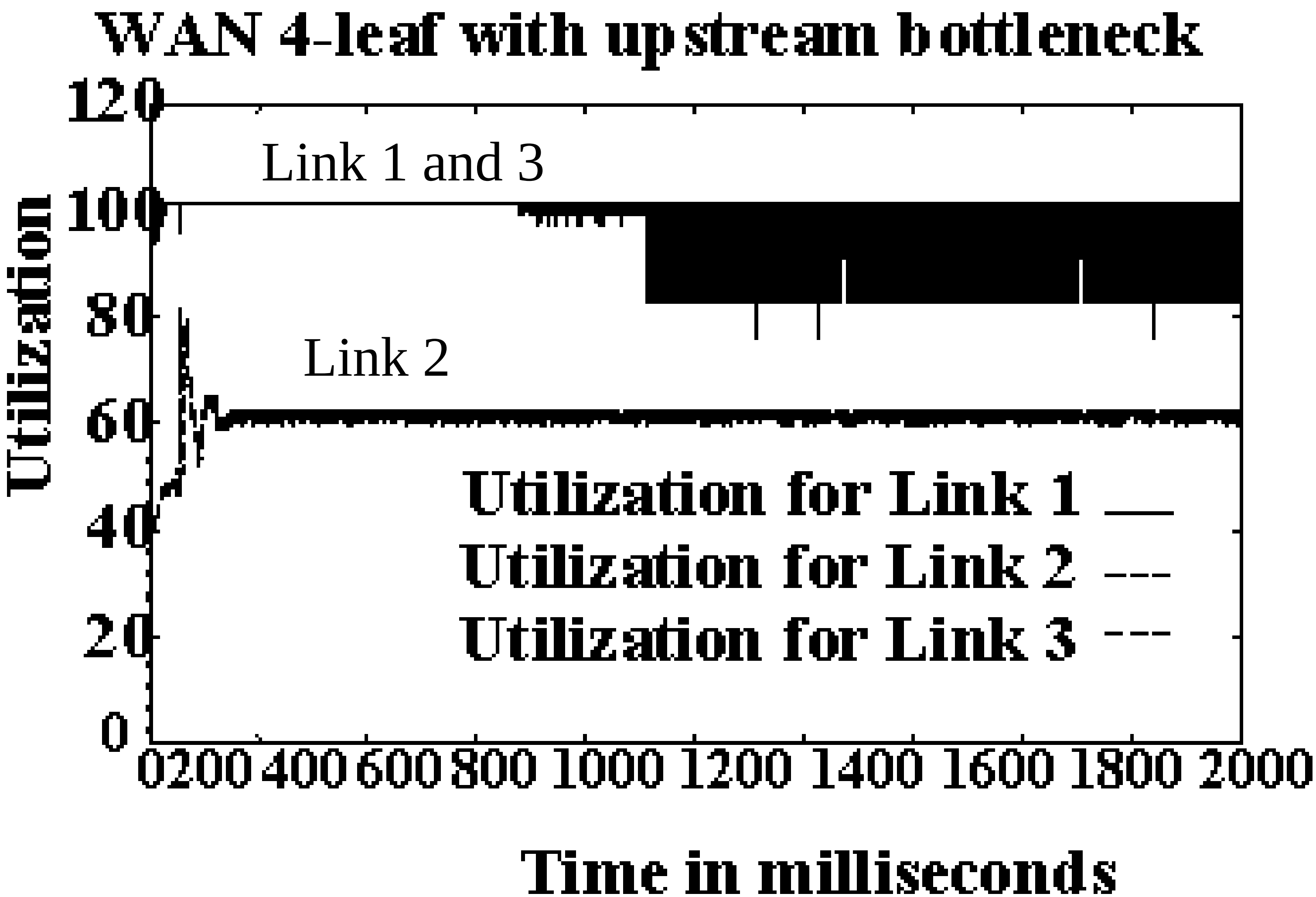
- Upstream Bottleneck, LINK3 = 5000 km, RIF = 1, interval = 5 ms



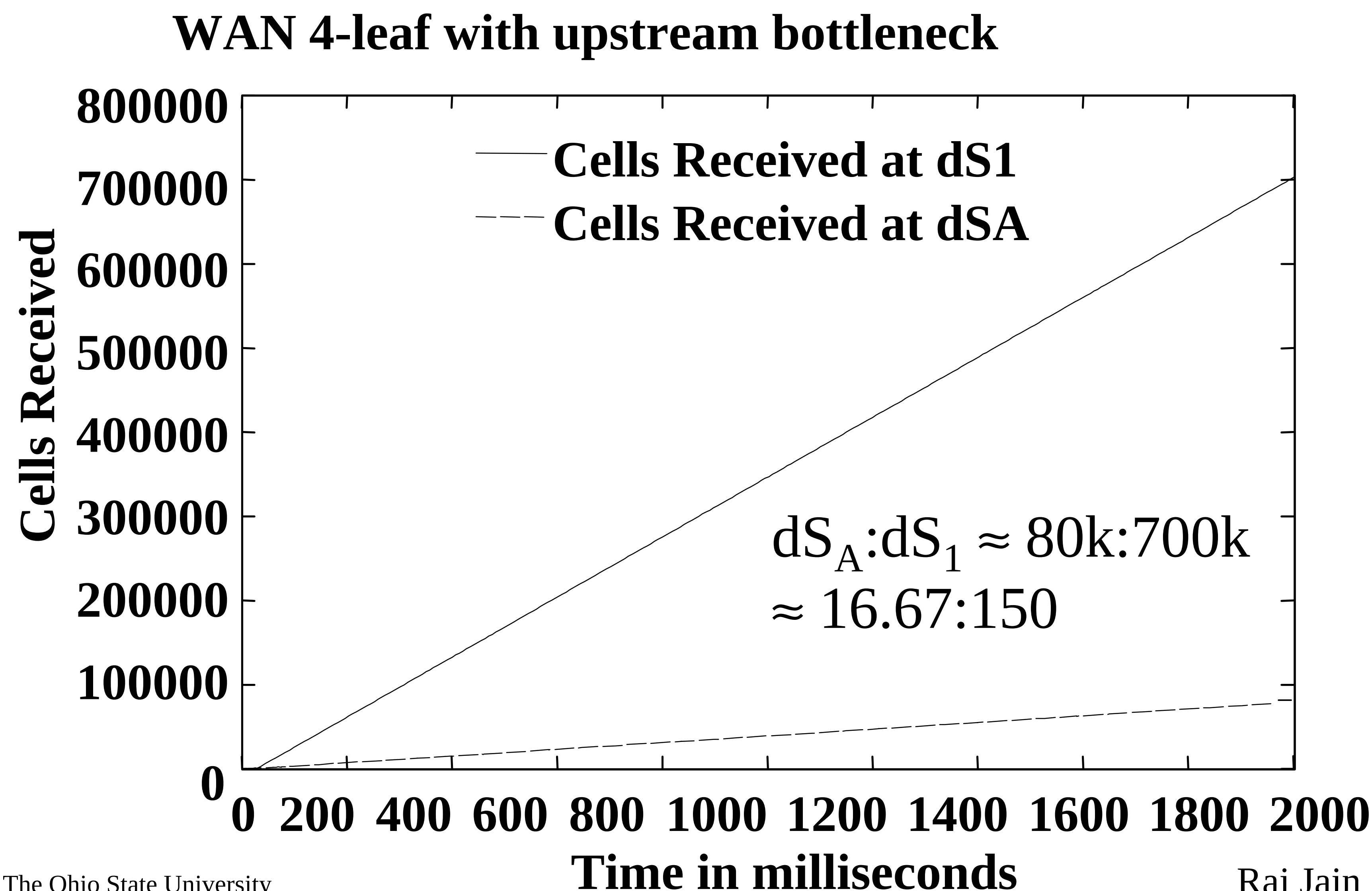
Queue Lengths



Link Utilization

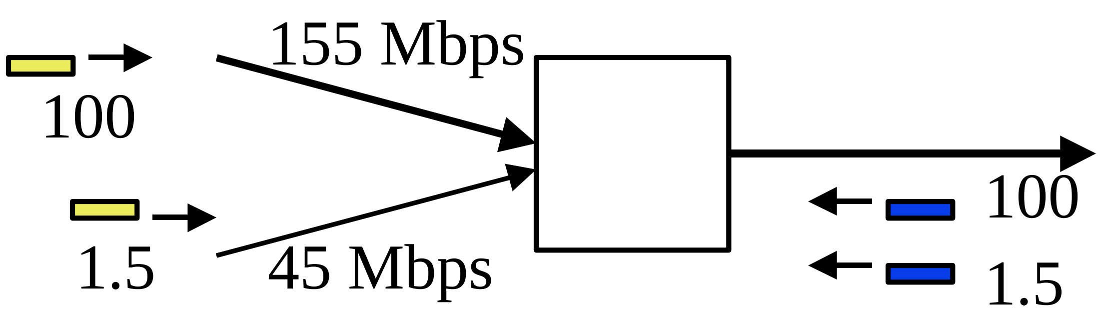


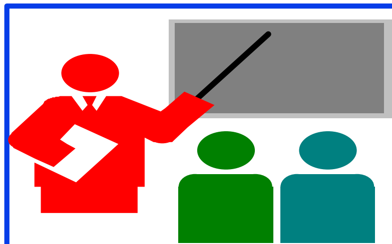
Cells Received



Lessons Learnt

- Avoid determining the effective number of active sources
- Avoid estimation of rates of sources, or determining if a source is bottlenecked at this link
- Use only per-VC or per-port measurements and not per-flow or per-source
- Do not use CCR values from BRM cells
  CCR from FRM cells can be used





Summary

- ERICA+ modified for pt-mpt works ok
- Additional delay due to FRM wait and BRM consolidation $\Rightarrow$ slower transient response than pt-pt
- Two new algorithms 6 and 7 speed up the transient response, while eliminating consolidation noise and controlling overhead
- Four Different Fairness Definitions: source, flow, VC/Source, VC/flow
- Source-based fairness can be achieved even though sources can not be distinguished in an mpt-pt VC

References

- All our contributions and papers are available on-line at http://www.cis.ohio-state.edu/~jain/
  See Recent Hot Papers for tutorials.

- Sonia Fahmy, et al, "Fairness Definition and Flow Control for ATM Multipoint Connections," Submitted to the International Conference on Network Protocols (ICNP), May 1998, http://www.cis.ohio-state.edu/~jain/papers/mpt2pt.htm

References (Cont)

- Sonia Fahmy, et al, "Design and Evaluation of Feedback Consolidation for ABR Point-to-Multipoint Connections in ATM Networks," Submitted to the Journal of Computer Communications, 1998, http://www.cis.ohio-state.edu/~jain/papers/pt2mpt.htm
- Sonia Fahmy, et al, "Fairness for ABR multipoint-to-point connections," Submitted to SPIE 98, April 1998, http://www.cis.ohio-state.edu/~jain/papers/spie98.htm

References (Cont)

- Sonia Fahmy, et al, "Feedback Consolidation Algorithms for ABR Point-to-Multipoint Connections in ATM Networks," Proceedings of IEEE INFOCOM 1998, March 1998, http://www.cis.ohio-state.edu/~jain/papers/cnsldt.htm
- S. Fahmy, et al, "Performance analysis of ABR point-to-multipoint connections for bursty and nonbursty traffic with and without VBR background," ATM Forum/97-0422, April 1997, http://www.cis.ohio-state.edu/~jain/atmf/a97-0422.htm

References (Cont)

- S. Fahmy, et al, "Feedback consolidation algorithms for ABR point-to-multipoint Connections," ATM Forum/97-0615, July 1997, http://www.cis.ohio-state.edu/~jain/atmf/a97-0615.htm
- S. Fahmy, et al, "Fairness for ABR multipoint-to-point connections," ATM Forum/97-0832, Sep 1997, http://www.cis.ohio-state.edu/~jain/atmf/a97-0832.htm
- S. Fahmy et al, "A switch algorithm for ABR multipoint-to-point connections," ATM Forum/97-1085, December 1997, http://www.cis.ohio-state.edu/~jain/atmf/a97-1085.htm

Thank You!

This work was partly sponsored by Rome Laboratory/C3BC under Contract #F30602-96-C-0156