98-0154: Determining the Number of Active ABR Sources in Switch Algorithms

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

Depa y

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

- ERICA
- New algorithm
- Examples
- Proof
- Simulation Results

Original ERICA

End of measurement interval:

- Target ABR Capacity
  = Target Utilization × Available Bandwidth
- Load Factor z = ABR Input Rate/Target ABR Capacity
- FairShare
  = Target ABR Capacity/Number of Active VCs
- VC's Share = Current Cell Rate/Load Factor z

BRM to be sent:

- ER Calculated = Max (FairShare, VC's Share)

Number of Active VCs

- FairShare
  = Target Capacity/Number of Active VCs
- Number of Active VCs: Number of VCs that sent one or more cells in the last ΔT interval
  $\Rightarrow$ A VC that sends 1 cell is counted as an active VC
  A VC that sends 1000 cells is also counted as an active VC
- Activity of a VC is a discrete variable: 0 or 1

Effective Number of VCs

- Idea: Activity can be a continuous variable.
  $\Rightarrow$ A VC can have activity level anywhere between 0 and 1
- Effective Number of VCs
  $= \Sigma_i$ Activity of i^{th} VC
- FairShare = Target Capacity/Effective Number of VCs
- <u>Example:</u> 3 sources with activity of 0.5, 0.75, 1

  Available capacity = 149 Mbps

  Target Utilization = 0.9

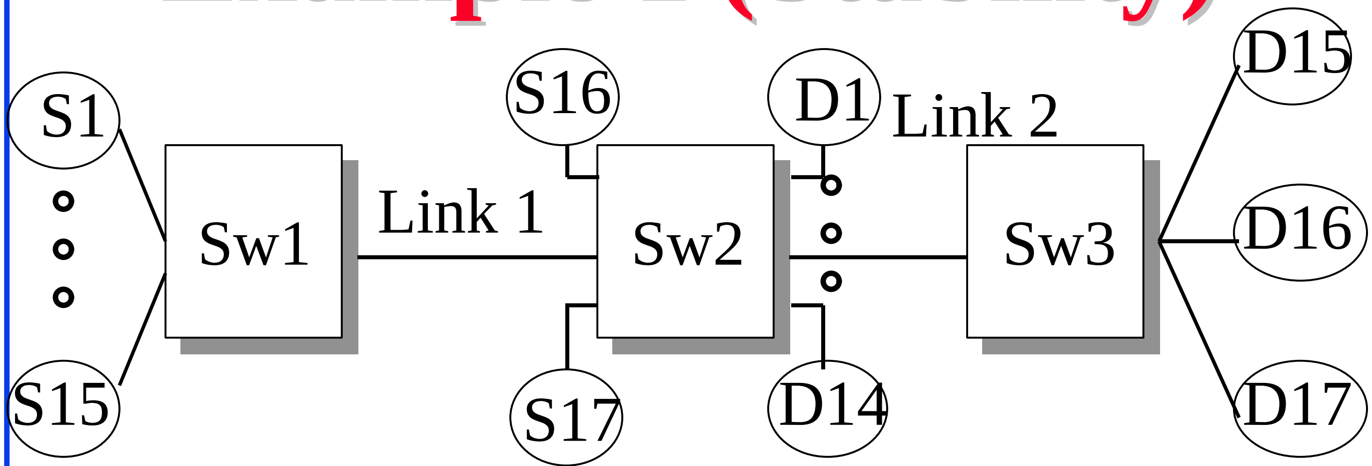
  FairShare = $0.9 \times 149/(0.5+0.75+1)$ = 59.6 Mbps

Determining Activity Level

- Activity level = Min (1, Source rate/FairShare)
  $\Rightarrow$ VCs operating $\geq$ FairShare are each counted as 1; VCs operating < FairShare only contribute a fraction
- Effective number of VCs = Σ_i Activity level of VC i
- FairShare =

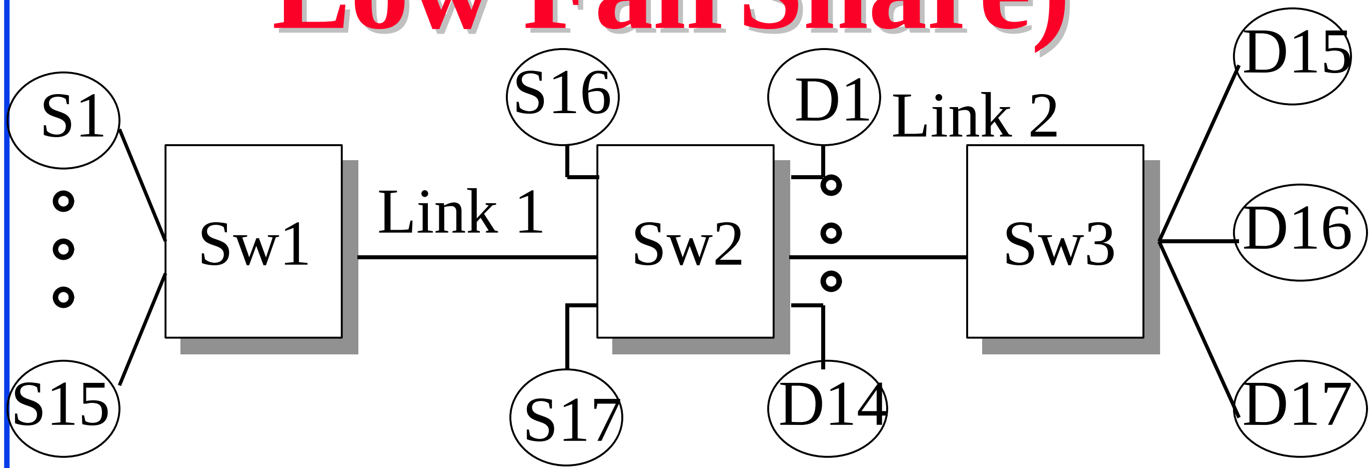
  Target ABR Capacity/Effective Number of VCs
- Definitions are recursive
- However, starting with any arbitrary value of FairShare, the procedure converges quickly

Example 1 (Stability)



- Target capacity for Link 1 and Link 2= 150 Mbps
- For Sw2, (S15, S16, S17) = (10, 70, 70)
- *Iteration 1:* FairShare = 70 Mbps
  - Activity = (10/70, 70/70, 70/70) = (1/7, 1, 1)
  - Effective # of VCs = 1 + 1 + 1/7 = 15/7
- *Iteration 2:* FairShare = Target capacity/Effective Number of VCs = 150/2.14 ≈ 70 Mbps

Example 2 (Rising from a Low FairShare)



- Rates = (10, 50, 90)
- Assume FairShare = 50
- *Iteration 1:*
    - Activity = (10/50, 50/50, 1) = (0.2, 1, 1)
    - Effective # of VCs = 0.2 + 1 + 1 = 2.2
- *Iteration 2:* FairShare = 150/2.2 ≈ 70 Mbps

Example 3 (Dropping from a High FairShare)

- Same configuration, rates = (10, 50, 90), FairShare = 75 Mbps
- *Iteration 1:*
  - Activity = (10/75, 50/75, 1) = (0.13, 0.67, 1)
  - Effective # of VCs = 0.13 + 0.67 + 1 = 1.8
- *Iteration 2:* FairShare = 150/1.8 = 83 Mbps
- Assume sources send at new rates, except for S15
- Activity = (10/83, 83/83, 83/83) = (0.12, 1, 1)
- Effective # of VCs = 0.12 + 1 + 1 = 2.12
- FairShare = 150/2.12 ≈ 70 Mbps

Proof

- *Claim:* This procedure leads to max-min fairness in all cases
- *Proof*: Two Steps
  1. This is equivalent to MIT scheme
  2. MIT scheme leads to max-min fairness [Charny95]

Derivation of Step 1

- MIT Scheme: FairShare = [ABR Capacity $- \Sigma_{i = 1 \text{ to } Nu} R_{ui}]/N_o$ where:

  R_{ui} = Rate of i^{th} underloading source ($1 \leq i \leq N_u$)

  N_u = # of underloading VCs, N_o = # of overloading VCs
- FairShare $* N_o$ = ABR Capacity $- \Sigma_{i = 1 \text{ to } Nu} R_{ui}$
- FairShare $* N_o + \Sigma_{i = 1 \text{ to } Nu} R_{ui}$ = ABR Capacity
- FairShare $* [N_o + \Sigma_{i = 1 \text{ to } Nu} R_{ui}$/FairShare] = ABR Capacity
- FairShare = ABR Capacity/N_{eff}, where:

  $N_{eff} = N_o + \Sigma_{i = 1 \text{ to } Nu} R_{ui}$/FairShare

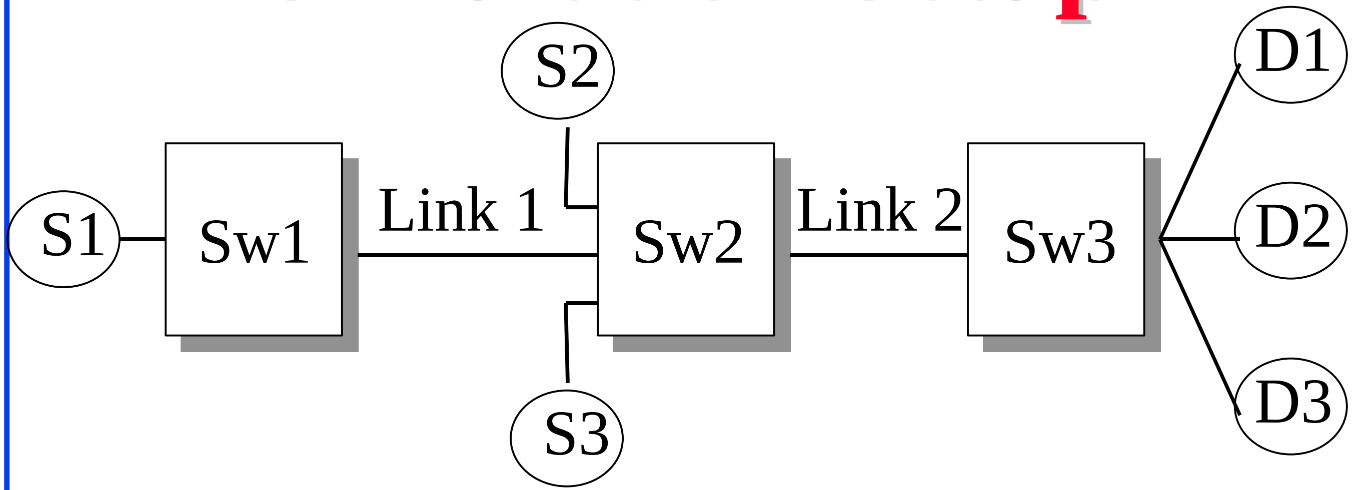
Benefits

- Simulation results show that:
  - Method works even with short measurement intervals and low rate sources
  - Max-min fairness is achieved even without the previous fairness solution:

MaxAllocPrevious = maximum allocation in the previous interval, initialized to FairShare

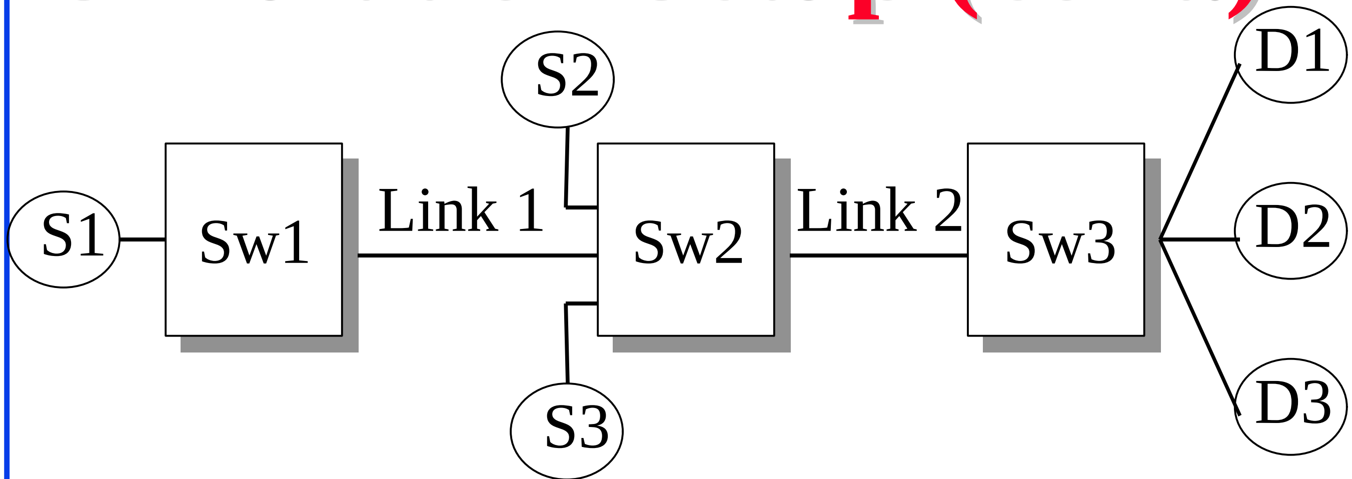
IF (load factor z > 1 + δ)
    THEN
        ER = Max (CCR/z, FairShare)
    ELSE
        ER = Max (CCR/z, *MaxAllocPrevious*)

Simulation Setup



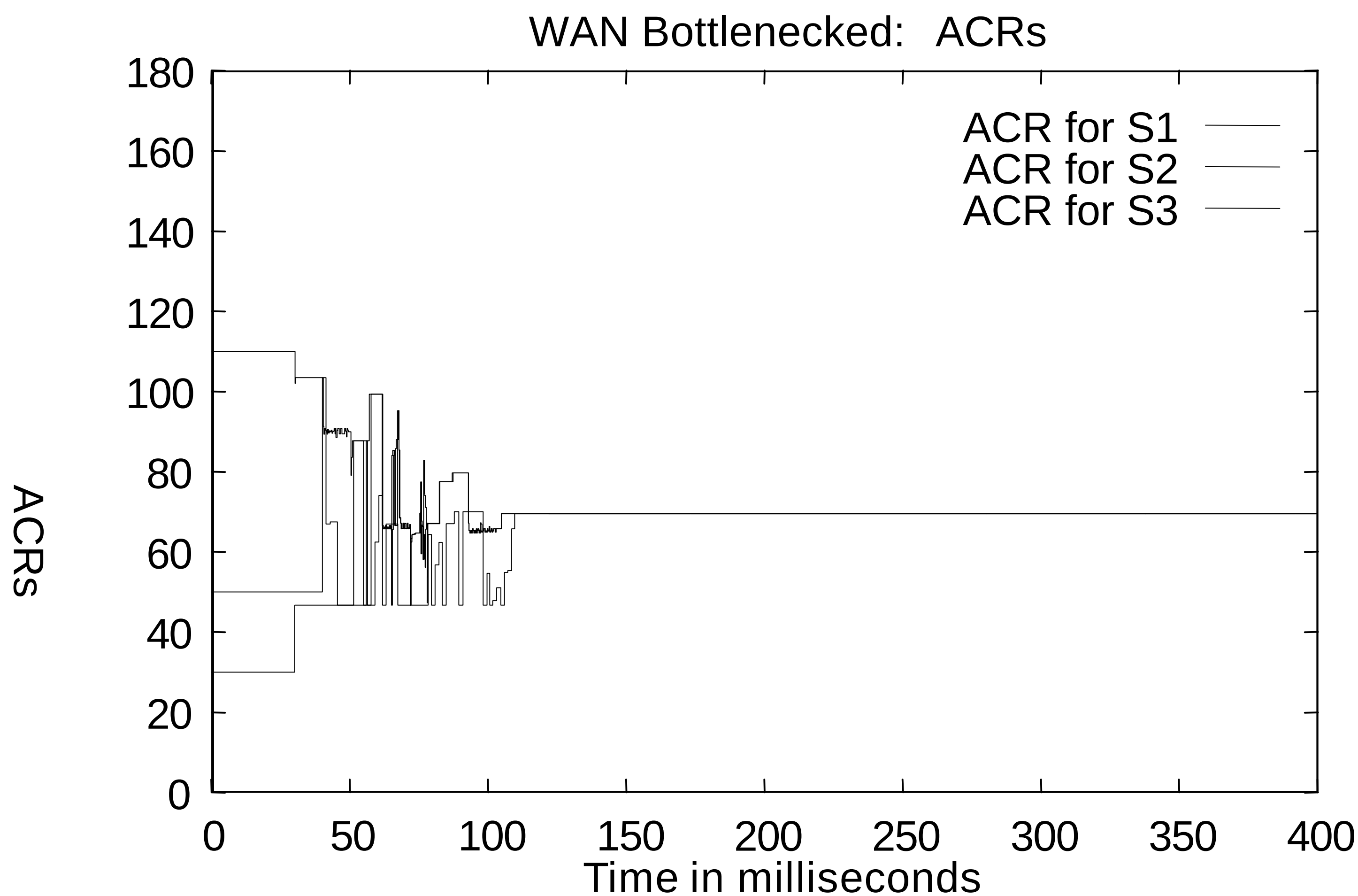
- ∀ links: bandwidth = 155.52 Mbps, length = 1000 km
- All VCs are bidirectional
- S1 is bottlenecked at 10 Mbps, ICR for S2 = 30 Mbps, for S3 = 110 Mbps, S1+S2+S3=150 Mbps
- Tests if S2 and S3 reach same ACR, using bandwidth left over by S1

Simulation Setup (cont.)



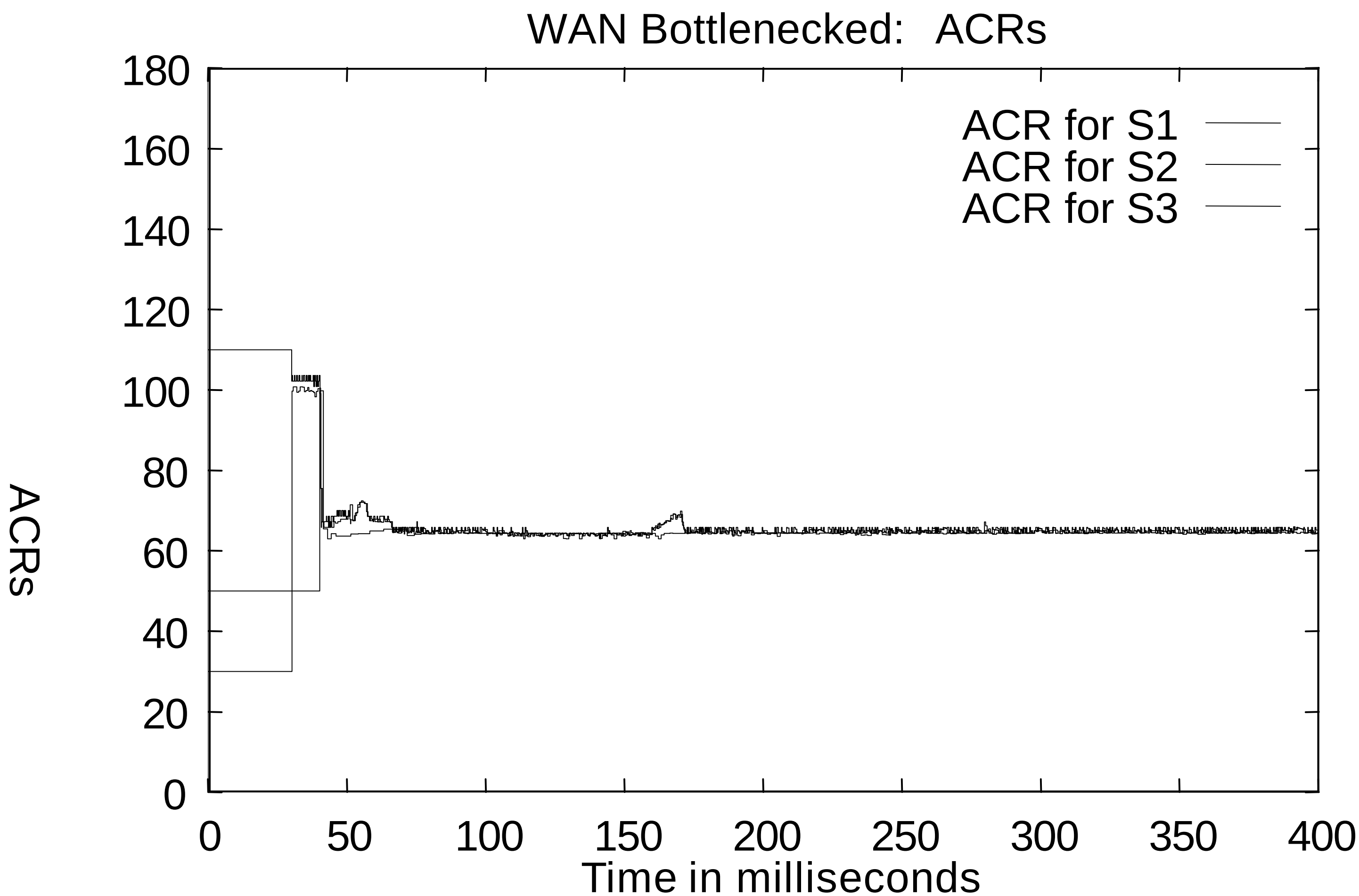
- Same as configuration used in examples, except that S1 VC is bottlenecked at *S1 itself* (not Link 1), to show effect of source bottlenecks
- RIF = 1, TBE = large
- Switch target utilization parameter = 90%
- Switch interval = min (time (100 cells), 1 ms)

Results: ERICA



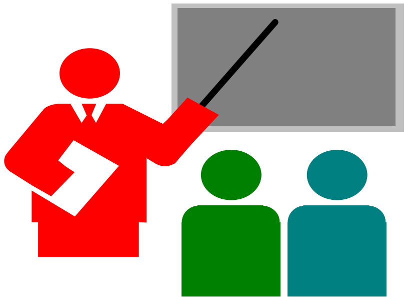
ERICA with *MaxAllocPrevious* solution attains fairness

Results: New Method



New method also attains fairness. Note faster convergence

Conclusions



- New method distinguishes underloading and overloading connections to compute activity levels, effective # of active connections, and fair share.
- Method is provably max-min fair, and maintains the *fast* transient response, queuing delay control, and simplicity of ERICA. It overcomes the need for the ERICA fairness steps and is less sensitive to measurement interval length.