Computing Fair Rates in RPR

Anoop Ghanwani
(anoop@lanterncom.com)

IEEE 802.17 Plenary Meeting
Vancouver, BC, Canada
July 2002

Introduction

- The current draft defines two fairness modes
  - Conservative mode
  - Aggressive mode
- This presentation focuses on performance issues with the aggressive mode
  - Conservative mode was not well-specified
  - A simulator was not available for testing
- In order to better address performance concerns the standard should allow more flexibility for computing the fair rate

Fair Rate Computation

- Aggressive mode
  - Advertise add_rate when congested
  - Advertise NULL when not congested
- Conservative mode
  - Always advertise a locally computed fair rate
    - Ramp-up using a predefined function when not congested
    - Ramp-down when congested

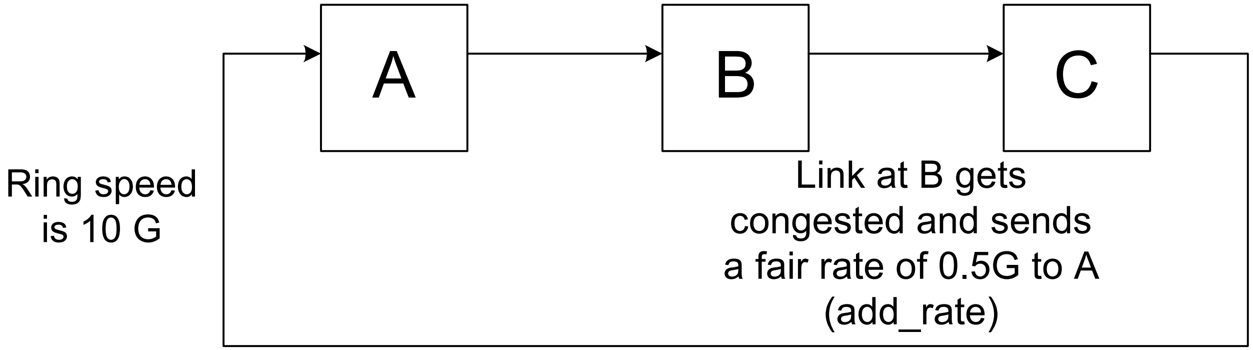
Limitations of Using the add_rate As An Estimate of the Fair Rate

- The add rate is a guess of the fair rate
- That guess can sometimes be very bad
- A very small add_rate at a congested node can cause oscillations
  - Size of oscillations is the difference between the add_rate and the actual rate available
  - Results in poor utilization

An Example Where the add_rate is a Bad Estimate of the Fair Rate

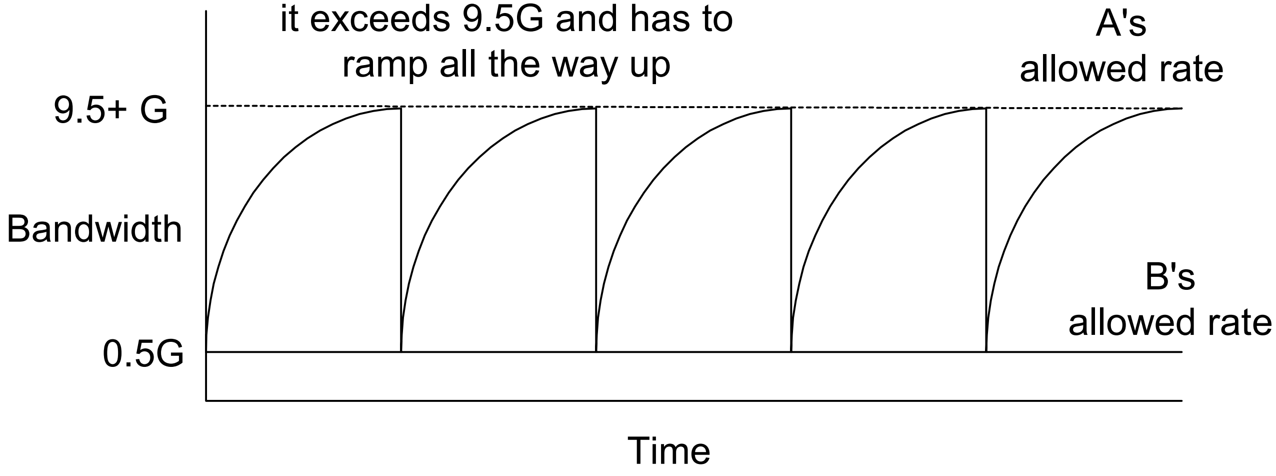
A is a greedy source sending traffic to C (tries to send as much traffic as it can)

B is a non-greedy source and sends a steady stream at 0.5G to C

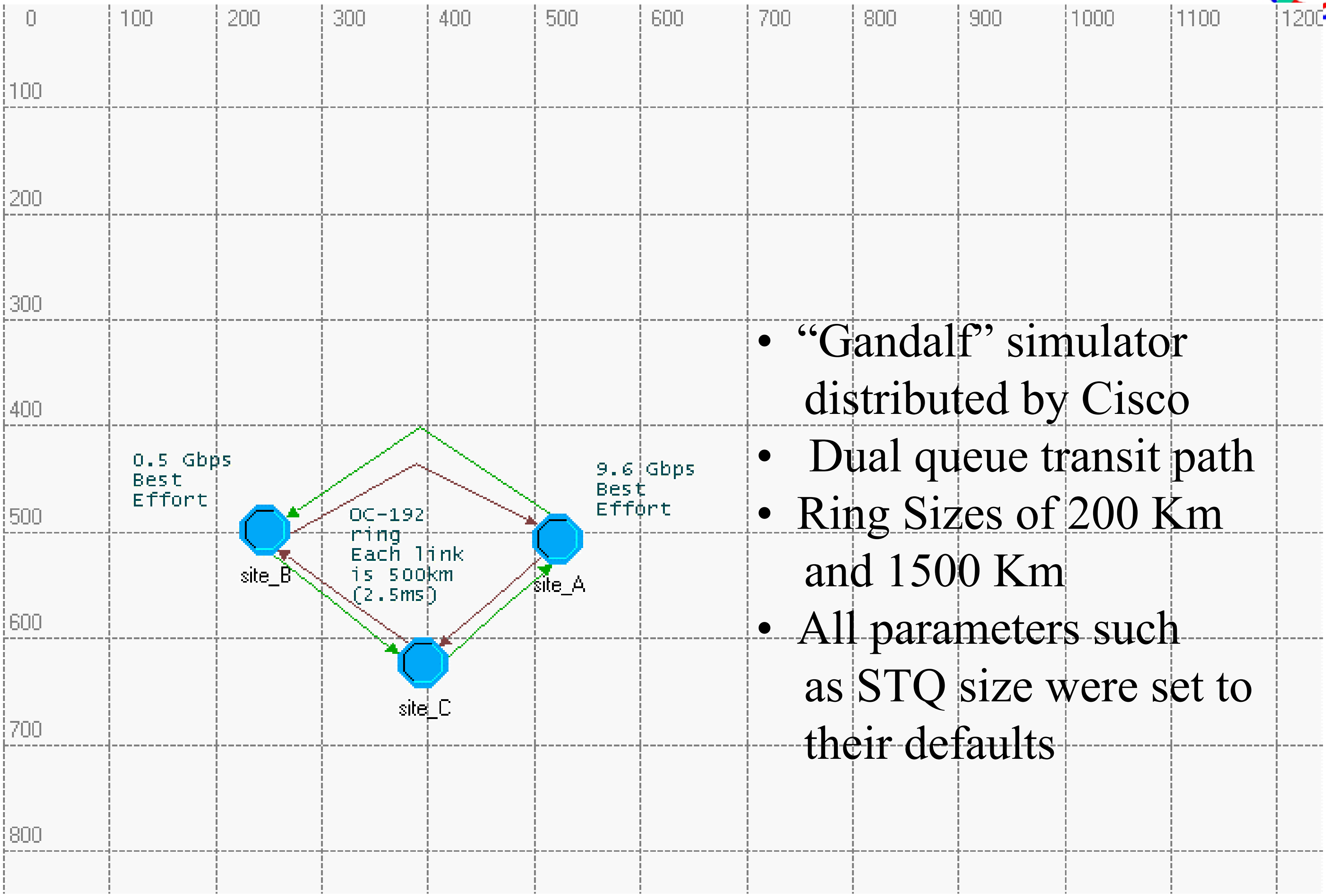


Link at B gets congested and sends a fair rate of 0.5G to A (add_rate)

A is shut off by B every time it exceeds 9.5G and has to ramp all the way up

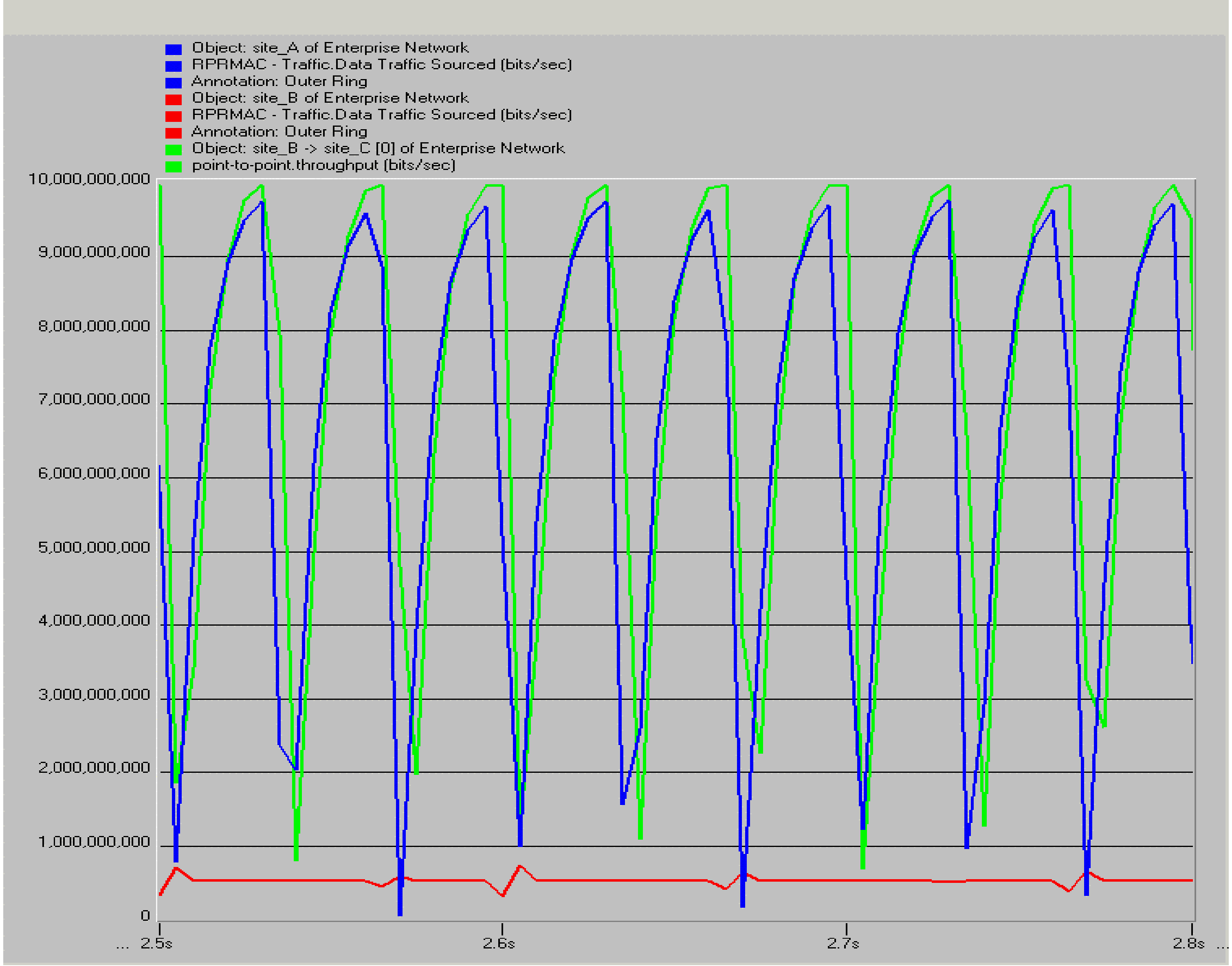


Simulation Setup



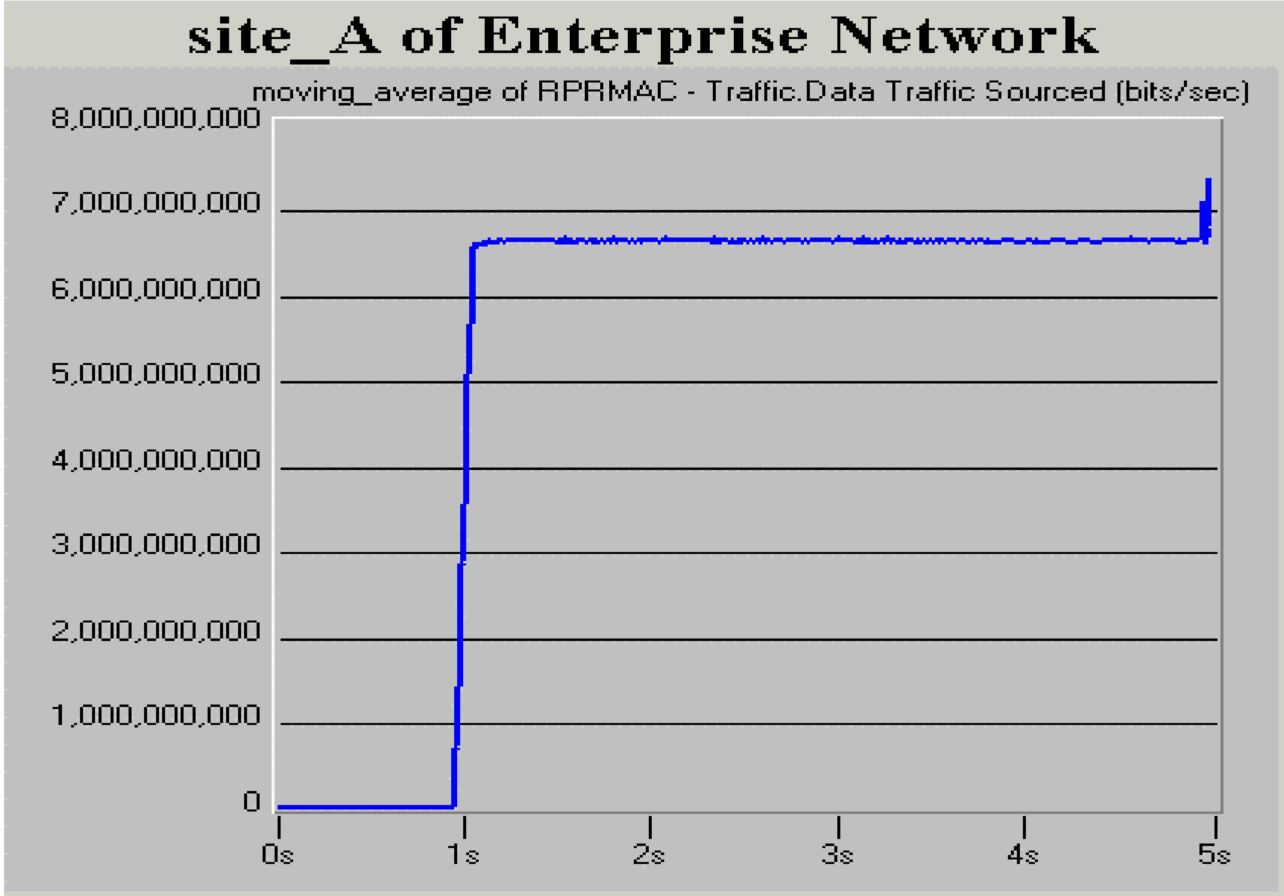
- "Gandalf" simulator distributed by Cisco
- Dual queue transit path
- Ring Sizes of 200 Km and 1500 Km
- All parameters such as STQ size were set to their defaults

Instantaneous Output Rate On Each Link



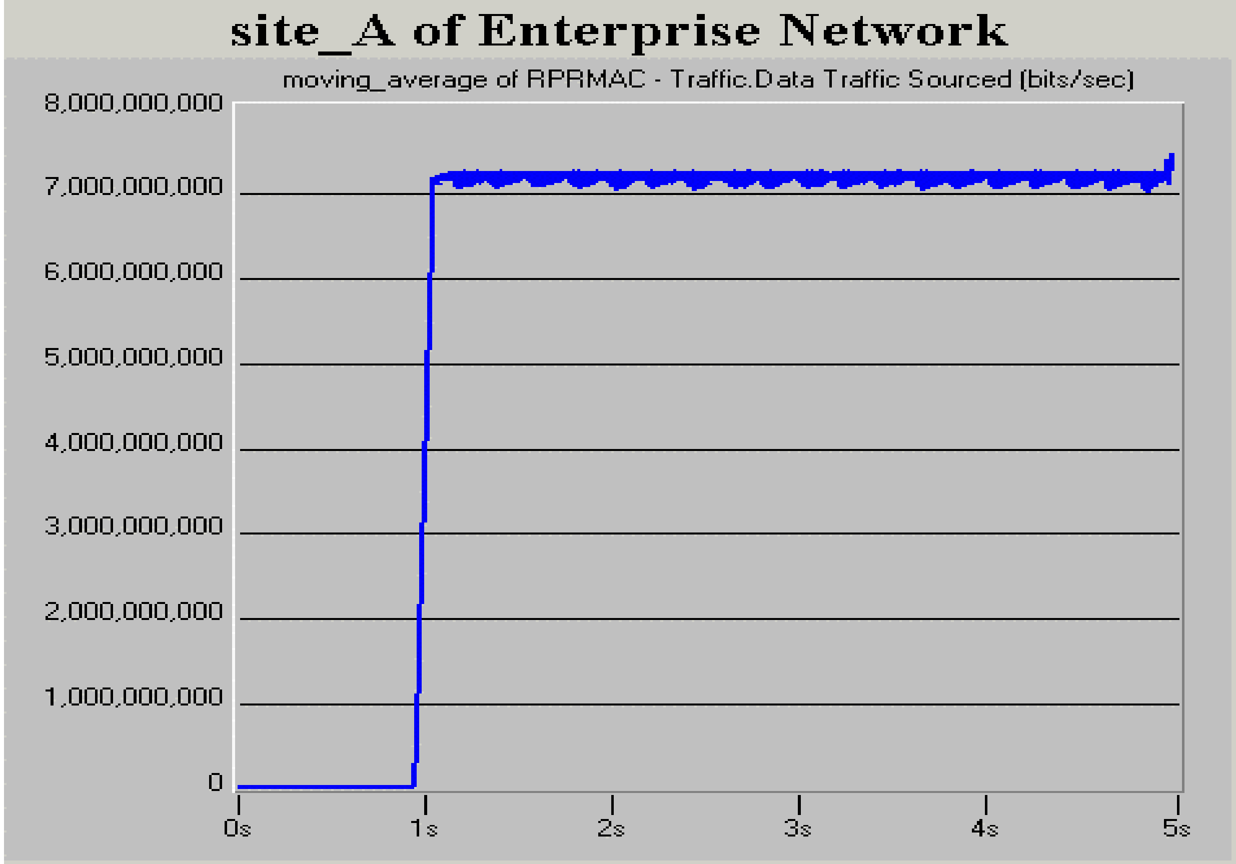
Moving Average of A's Rate (1500 Km Ring)

- Ideally should have been ~9.5G
- Instead it's ~6.7G – 30% loss of throughput due to persistent large oscillations!



Moving Average of A’s Rate (200 Km Ring)

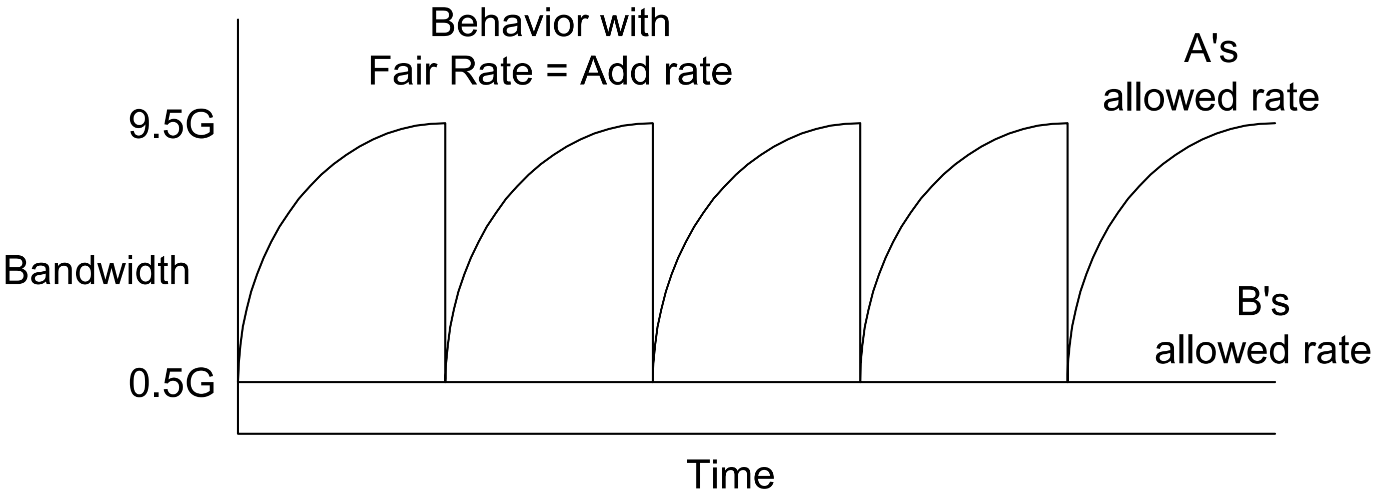
- Ideally should have been ~9.5G
- Instead it’s ~7.2G (slightly better than for 1500 Km since the ring is smaller and feedback is faster)

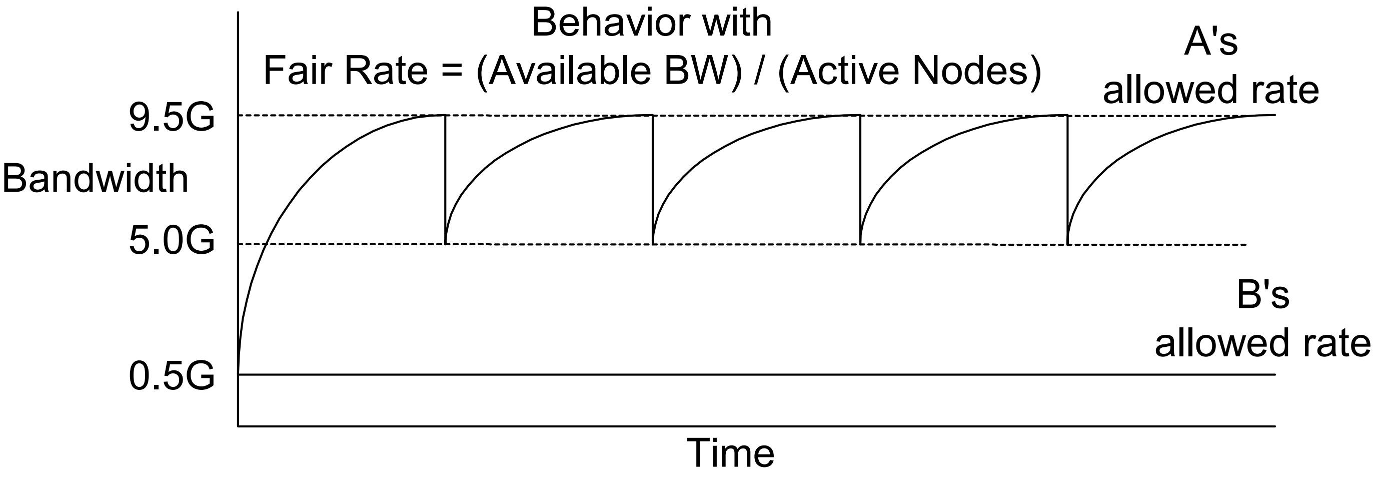


A Proposed Fix

- A very simple fix is as follows:
  - During each decay interval, count the number of active sources
  - Compute the fair rate as the available bandwidth divided by the number of active sources (instead of using the add_rate)
  - This will ensure that big differences in the add rates don't end up affecting performance as badly
  - In the example, this means the advertised fair rate would be 5 Gbps
- This is just one possibility – there are other methods for estimating the fair rate
  - It is actually possible to compute a fair rate that is close to the 9.5 Gbps value in the example

Expected Behavior With the Fix





Ways to Make the Fairness Algorithm Flexible

- Remove the notion of fairness modes and leave the standard completely flexible by defining only the basic constructs required for interoperability
  - Define the syntax/semantics of the fairness messages
  - Define what a station does with a fairness message when it receives it
  - Leave out the details of fair rate computation

OR

- Define additional modes as long as they can be shown to interoperate with some degree of performance
  - Each mode has its own way of determining the fair rate
  - The details of each are specified in the standard
  - An implementer must choose at least one of these

Implications of a Flexible Fair Rate Calculation in the Standard

- Allows differentiation among vendors
- Allows a carrier to select equipment that is best optimized for their needs
- Continues to allow interoperability
  - Possible implications with respect to performance of multi-vendor rings
  - Typically, a ring will perform only as well as the least capable node for a given scenario

Minimum Requirements on Rate Advertisement for Interoperability

- When congested, a station must send a non-NULL value
  - The value is not specified
  - Alternatively, allow multiple modes
- When uncongested, a station may send a NULL value
  - But it doesn’t *have to*
  - May send a fair rate if it is capable of calculating one
    - This is what the conservative mode does anyway
    - Allow for other methods

Behavior On the Receipt of a Fairness Message

- If the message has a non-NULL value, then set the allowed rate to that value adjusted by the local station weight
- If the message has a NULL value, increment the current allowed rate

Conclusions

- This presentation highlighted a performance limitation of the aggressive mode of the fairness algorithm
  - Causes poor network utilization
- We already have two modes in the standard
  - Aggressive and Conservative
- We should allow more flexibility to address performance concerns
  - Define additional modes; or
  - Remove the notion of modes and define only the basic constructs required to achieve interoperability