Weighted Fairness

Necdet Uzun and Mete Yilmaz

AuroraNetics, Inc.

May 15, 2001

Contents

- Objectives
- What is fairness?
- Why do we need weighted fairness?
- Fast converging weighted fairness algorithm
- Simulation results
- Conclusion

Objectives

AuroraNetics, Inc.

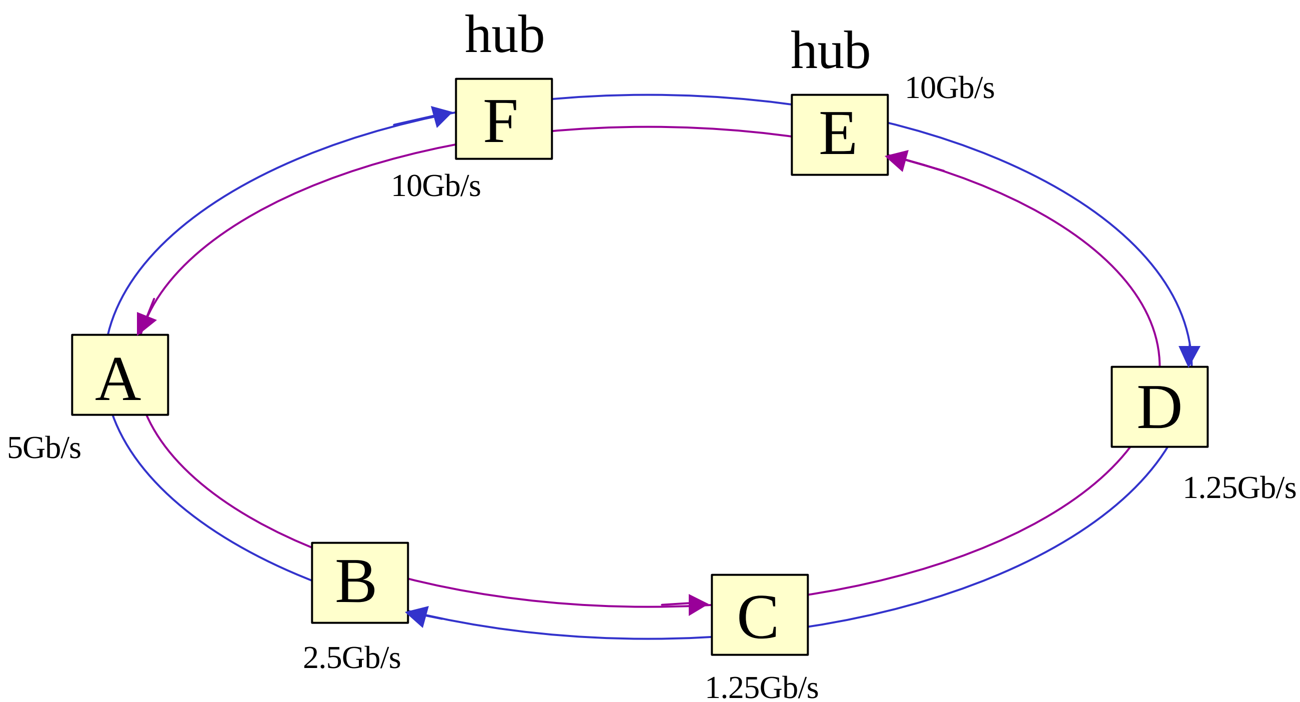
- Bandwidth for high priority is provisioned
  - Same allocated bandwidth must be available to the high priority traffic during a single failure (single/dual fiber cut)
  - A total of less than 50% of line rate may be allocated for high priority
- Bandwidth for low priority is dynamically re-allocated based on node congestions
  - Each node gets a proportion of available ring bandwidth
  - Active nodes capture all bandwidth available
  - Available bandwidth to low priority during a failure may be substantially degraded – best effort
- Fairness algorithm must allow for maximum ring throughput

Fairness algorithm

AuroraNetics, Inc.

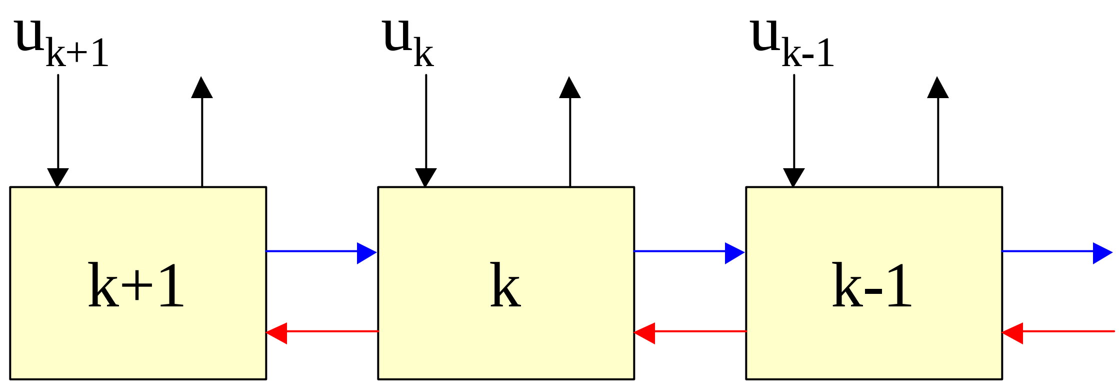
- All nodes are (un)equal
  - Every node is allowed to source proportional amount of LP traffic based on provisioned weights
  - Nodes behave identically
- Local fairness
  - All neighboring nodes contending for the same available bandwidth participate in fairness calculation (fairness domain)
  - Multiple dynamic fairness domains may be present
- Scalable
  - Distributed fairness algorithm
  - Node based fairness (does not keep track of micro-flows)
- Provides maximum possible throughput
- Does not adversely affect the delay/jitter bound performance of high priority traffic

Why weighted fairness?



Congestion management and fairness parameters

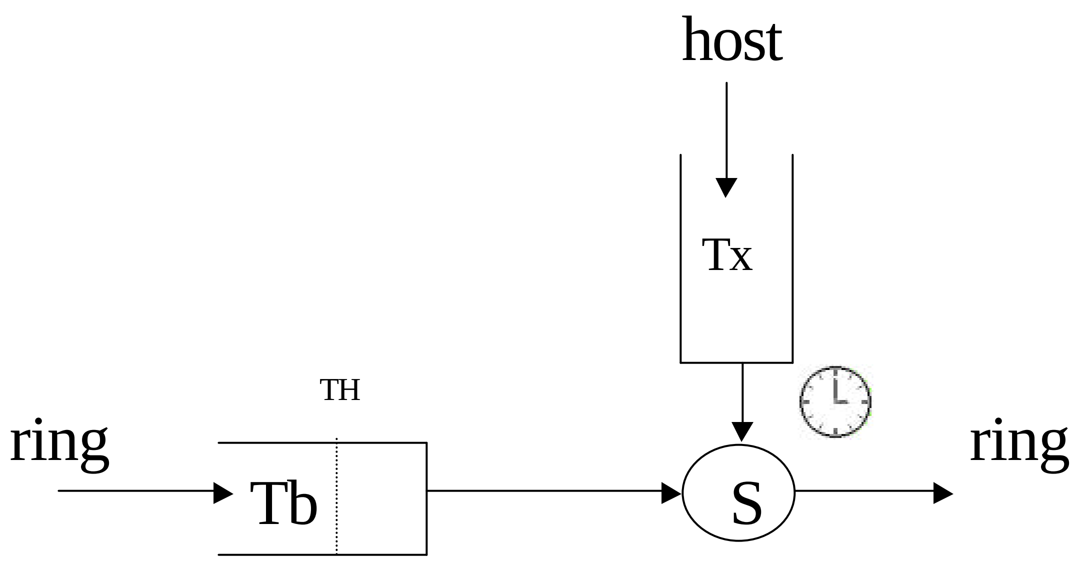
AuroraNetics, Inc.



u_k : actual usage (sourced traffic rate) of node k
a_k : allowed usage (sourced traffic rate) of node k
f_k : actual forwarded traffic through k from node k+1
w_{max_k} : max provisioned usage rate factor for node k
w_{max_k-1} : max provisioned usage rate factor for node k-1
u : usage value received from downstream node (k-1)

Node model

- Transit traffic (that is subject to weighted fairness) is stored in a FIFO queue Tb
- Transmit traffic (that is subject to weighted fairness) is stored in a FIFO queue Tx



Weighted fairness algorithm

AuroraNetics, Inc.

- Node k will scale *received usage* (from k-1) according to the ratio of provisioned weights of k and k-1. This value is k's *allowed usage.*
- If k is congested and its *actual usage* value
  - is larger than or equal to the *allowed usage,* it will forward the scaled *received usage* value upstream.
  - is less than the *allowed usage* or downstream node is not congested, then k will pass its *actual usage* value upstream.
- If k is not congested, and scaled *received usage*
  - is less than its scaled *forward rate*, k will pass the scaled *received usage* value upstream.
- If node k's *allowed usage* is greater than k's *actual usage,* it is allowed to add transmit traffic until k's *actual usage* reaches to k's *allowed usage*.

Weighted fairness algorithm

When a control packet from node k-1 is received with a usage value u, node k does following:

INIT: $a_k := a_{max\_k}$; $u_k := 0$

IF (node k congested) AND (u = NOT null)

  $a_k \Leftarrow (w_{max\_k} / w_{max\_k-1})u$

  $u \Leftarrow \min \{u_k, (w_{max\_k} / w_{max\_k-1})u$

ELSE IF (node k congested) AND (u = null)

  $a_k \Leftarrow (w_{max\_k} / w_{max\_k-1})u_k$

  $u \Leftarrow u_k$

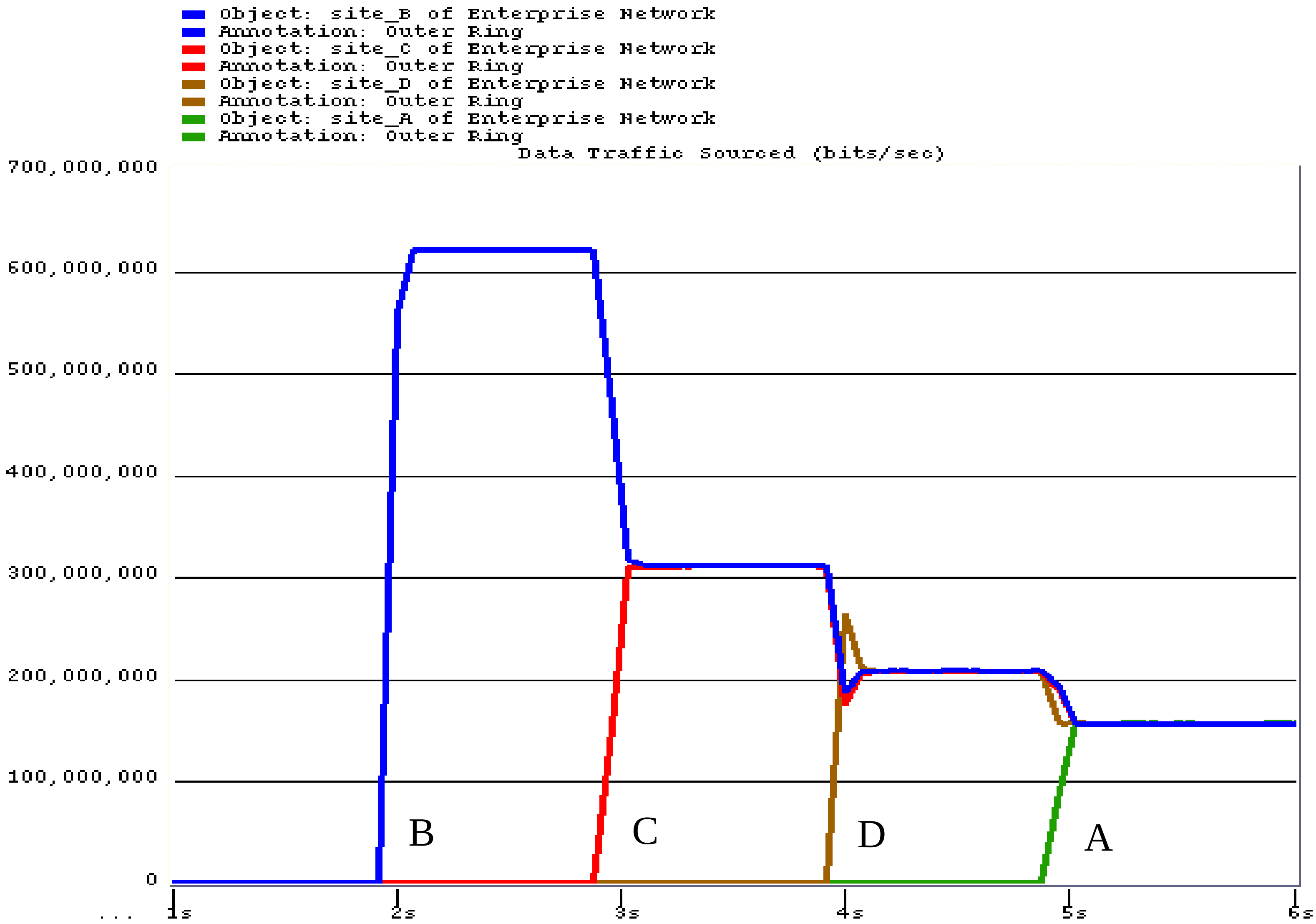
ELSE – node k not congested

  $a_k \Leftarrow u$

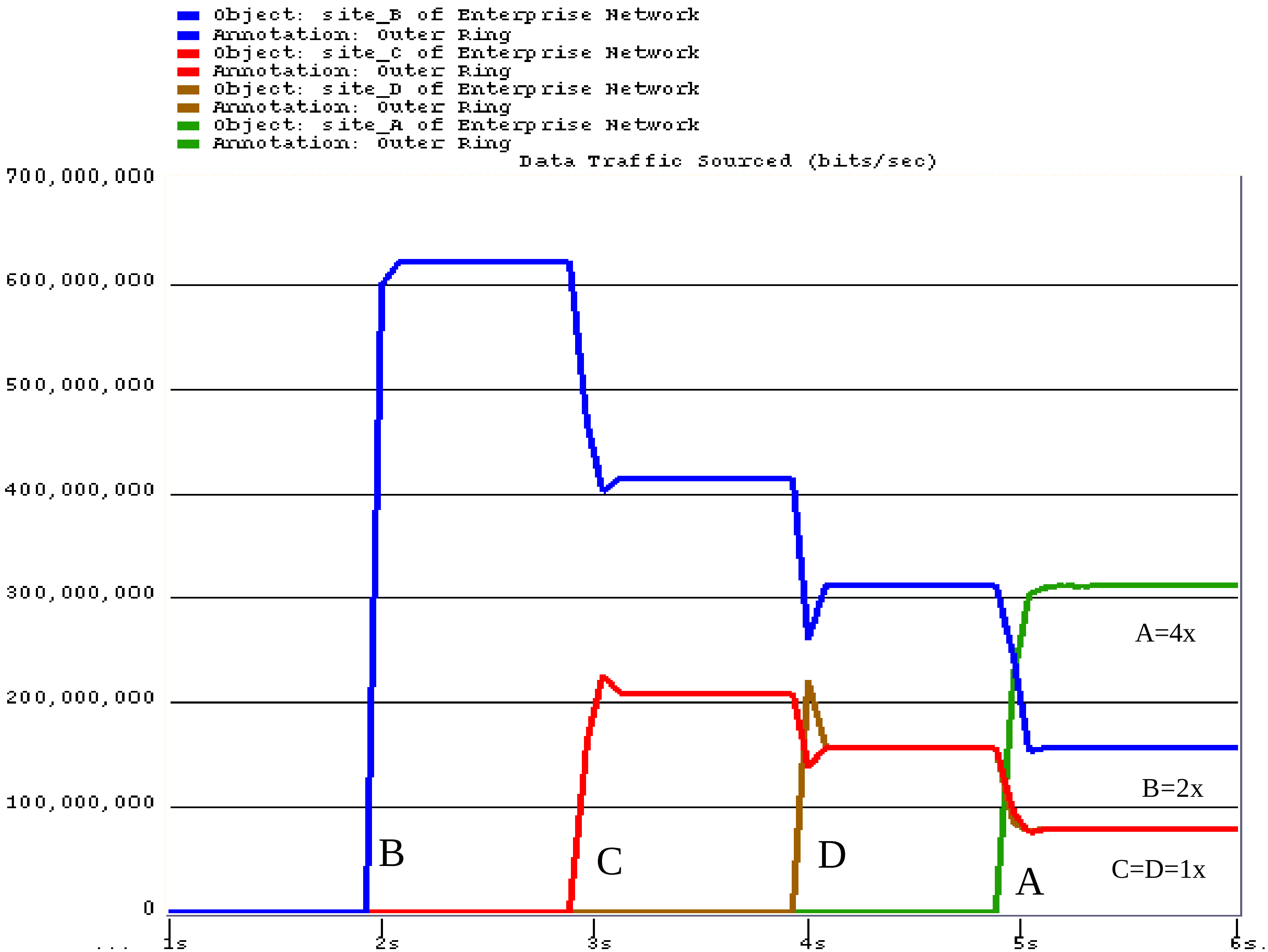
  IF $f_k < (w_{max\_m} / w_{max\_k-1})\ u$ $u \Leftarrow null$

  ELSE $u \Leftarrow (w_{max\_k} / w_{max\_k-1})u$

Weighted fairness (OC-12, equal weights)



Weighted fairness (weights 4, 2, 1, 1)



Summary

- The weighted fairness algorithm provides:
  - Fairness
  - Weighted fairness
  - Local fairness
  - Fast convergence
  - High throughput
  - Scalability