Weighted Fairness Algorithm

and 3 Priority Support

Necdet Uzun, Pinar Yilmaz - Cisco Systems

Gunes Aybay – Riverstone Networks

Gal Mor – Corrigent Systems

David Cheon - Sun Microsystems

Sateesh Kumar – Redwave Networks

David Meyer – Mindspeed Technologies

Yong Kim - Broadcom

Agenda

- Weighted Fairness
  - Pseudo-code
- 3 Priority Support
  - Pseudo-code
- Implementation Details
- Conclusion

Weighted Fairness

- Each node has an assigned weight:
  - WEIGHT
- Advertise usage value scaled by weight
- Scale received usage value by weight

Weighted Fairness – Pseudo-code

- Every decay interval:
  - nlp_my_usage = lp_my_usage / WEIGHT;
  - if (rcvd_usage != NULL_RCVD_INFO)
    allowed_usage = (rcvd_usage*WEIGHT);
    else
    allowed_usage += (MAX_LRATE - allowed_usage) / (LP_ALLOW);
- When sending a usage message:
  - if (congested){
    if (nlp_my_usage < rcvd_usage)
    rev_usage = nlp_my_usage;
    else
    rev_usage = rcvd_usage;
    } else if ((rcvd_usage != NULL_RCVD_INFO) &&
    (lp_forward_rate > (allowed_usage/WEIGHT))
    rev_usage = rcvd_usage;
    else
    rev_usage = NULL_RCVD_INFO

3 Priority Support

- Provide 3 priority classes in the ring
- High Priority
  - Guaranteed bandwidth (provisioned)
  - Bounded delay and bounded jitter
- Medium Priority
  - Committed bandwidth (provisioned), best effort for excess traffic
  - Bounded delay and (loosely) bounded jitter
- Low Priority
  - No guarantees
  - Best effort for bandwidth, delay and jitter

Node Model

- Mapped from Diffserv/MPLS/802.1Q (3 bits)

  If Priority < “Med Prio Threshold” then

  Low Priority

  Else if Priority < “High Prio Threshold” then

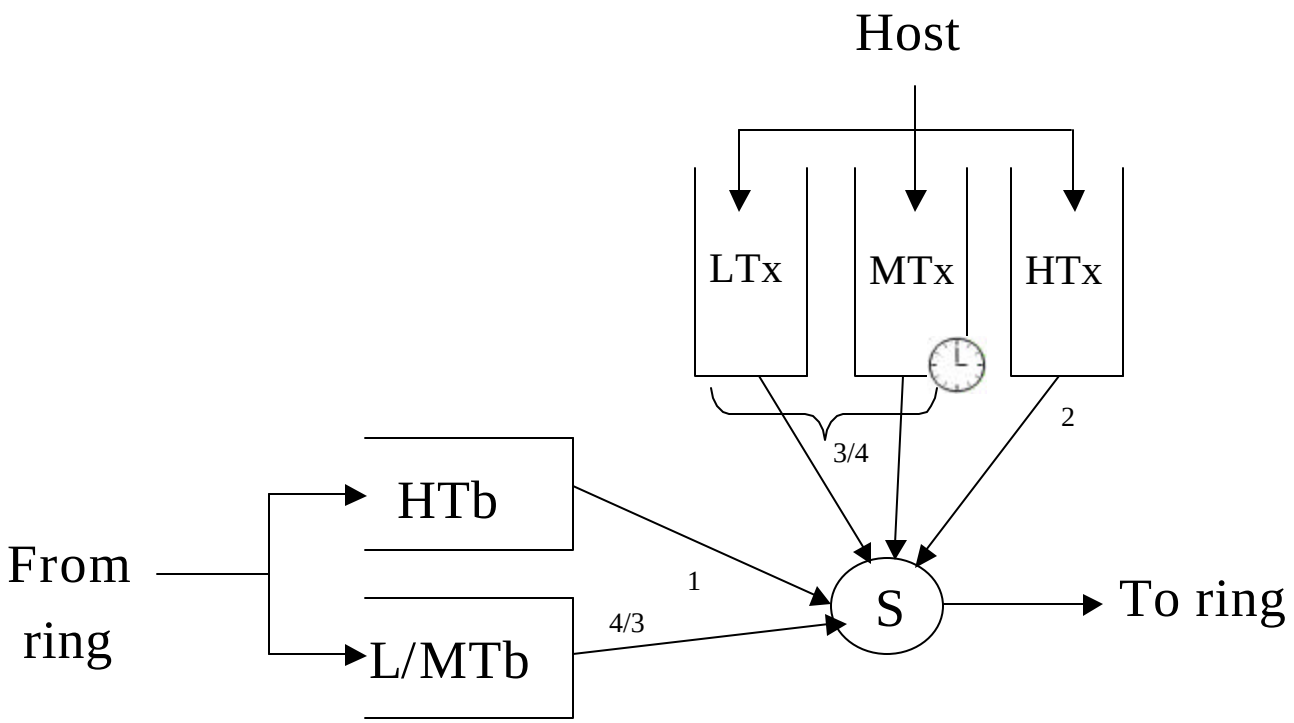
  Med Priority

  Else High Priority

- Committed Access Rate (CAR) for MP
  - MP Traffic exceeding CAR is subject to fairness algorithm control in the transmit path

Node Model (cont)

- Two transit buffers
- Three transmit buffers
  - Token bucket counter for Medium Priority



3 Priority Support – Pseudo-code

1. High Transit Buffer
2. If (Low Transit Buffer > High Threshold) then
   Low Transit Buffer
3. High Transmit Buffer
4. If (Medium Priority Token Available) then
   Medium Transmit Buffer
   Decrement Medium Priority Token count
5. If (My Usage OK)
   If (Medium Priority Pkt Waiting) then
      Medium Transmit Buffer
      Increment My Usage
   else if (Low Priority Pkt Waiting) then
      Low Transmit Buffer
      Increment My Usage
6. If (Low Priority Pkt Waiting) then
   Low Transit Buffer
   Increment Forward Rate

3 Priority Support – Pseudo-code

```
if  ( hpTbBuf.inUse()) {
    pkt = hpTbBuf.dequeue(bytes);

} else if ( lpTbBuf.GTHighThresh() ) {
    pkt = lpTbBuf.dequeue(bytes);
    increaseFwdRate(bytes);

} else if (hpTxBuf.inUse()) {
   pkt = hpTxBuf.dequeue(bytes);

} else if (mpTxBuf.inUse() && mpTxBuf.isTokenAvailable()) {
    pkt = mpTxBuf.dequeue(bytes);
    mpTxBuf.decrementToken(bytes);

} else if (mpTxBuf.inUse() && isMyUsgOk()){
    pkt = mpTxBuf.dequeue(bytes);
    increaseMyUsage(bytes);

} else if ((lpTxBuf.inUse()) && isMyUsgOk() ) {
    pkt = lpTxBuf.dequeue(bytes);
    increaseMyUsage(bytes);

} else if (lpTbBuf.inUse()){
    pkt = lpTbBuf.dequeue(bytes);
    increaseFwdRate(bytes);
}
```

3 Priority Support – Pseudo-code

- My Usage is OK (myUsgOK) if:
  - it is less than current allowed usage
  - it is not greater than forward rate if low transit buffer is not empty
  - low transit buffer is not greater than low threshold
  - it is less than maximum allowed usage
- myUsgOk = (my_usage < allow_usage)
  && !((my_usage > fwd_rate) && lpTbBuf.inUse())
  && !(lpTbBuf.usgGTLowThresh())
  && (my_usage < max_usage)
- My Usage counts:
  - Low Priority transmit traffic
  - Medium Priority transmit traffic above CAR
- Forward Rate counts:
  - Low Priority transit traffic
  - Medium Priority transit traffic (all or only excess?)

Implementation Details

- Weights statically assigned by management software
  - May become dynamic with a separate protocol to handle distribution and assignment of weights
- No changes in transit path to support 3 Priorities
- Need token bucket counter for medium priority on the transmit path
- Assumes MAC client will not allow medium priority to starve low priority on the transmit path (medium priority always goes before low priority as long as there are packets) otherwise weighted RR may be used?

Conclusions

- 3 Priority classes on the transmit and two on the transit provides enough performance
- Weighted fairness is necessary to provide bandwidth differentiations among nodes
- VDQs together with multi-chock point detection may enhance the ring performance.