A Near-ideal Fairness Scheme & Support for Real Time Services

Vasan Karighattam

Intel Corporation

Email: vasan.karighattam@intel.com

Agenda

- Objective
- Current issues
- A new metric
- Ring access algorithm
- Transit path design
- Delay bound & fairness
- Conclusion

Question

Can we design the RPR so that we have 100% network utilization, near-ideal throughput fairness and GPS like delay-bounds per flow for real time services?

Problems With Current Schemes

- Short-term throughput unfairness for real-time flows
- Longer term throughput unfairness due to some transit buffer designs
- Consequently packet-packet delays are not optimum
- Real-time services can only be a small fraction of line rate (20%?)
- Fairness schemes are only for BE traffic

Nature of Real-time Services

- Client traffic is characterized by a token bucket filter (r,b) where r is the rate and b is the depth
- The node should provide isolation of flows, so a flow can only have a limited negative effect on other flows
- Data traffic is a bursty statistical generation process => any r that produces a reasonably small (b/r) for a reasonable delay bound is much greater than the average data rate
- A client can improve their delay bound by increasing the rate r and let the burst pass through more quickly
- Consequently, network utilization due to real-time guaranteed traffic is quite low – 20-30%

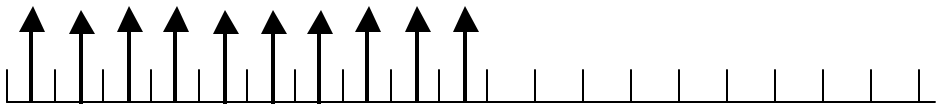
Real Time Services

- Ring access algorithms should be effective in order to improve delay bound and increase real time services

➡ Current schemes state that ring has the highest priority

a. If they are work-conserving, delays for downstream nodes will be higher
b. If non work-conserving, reduces delays some what, but not deterministic (not provable due to the ad hoc nature of some schemes)

Short-term Throughput Unfairness



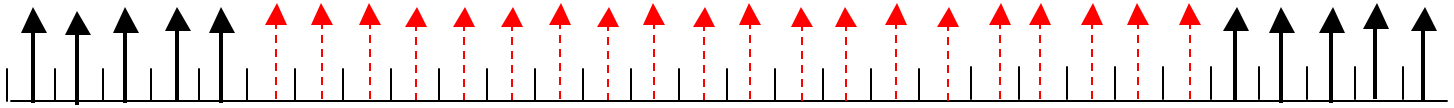
Node 1:
Connection Rate = 0.5

Node 2:
Connection Rate = 0.05

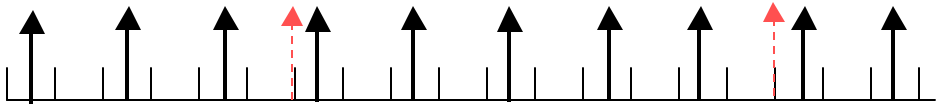
.....

.....

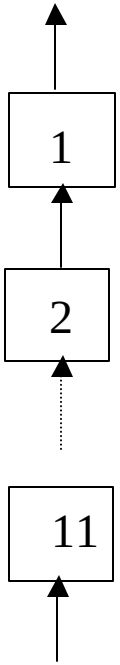
Node 11:
Connection Rate = 0.05



If Ring has highest priority

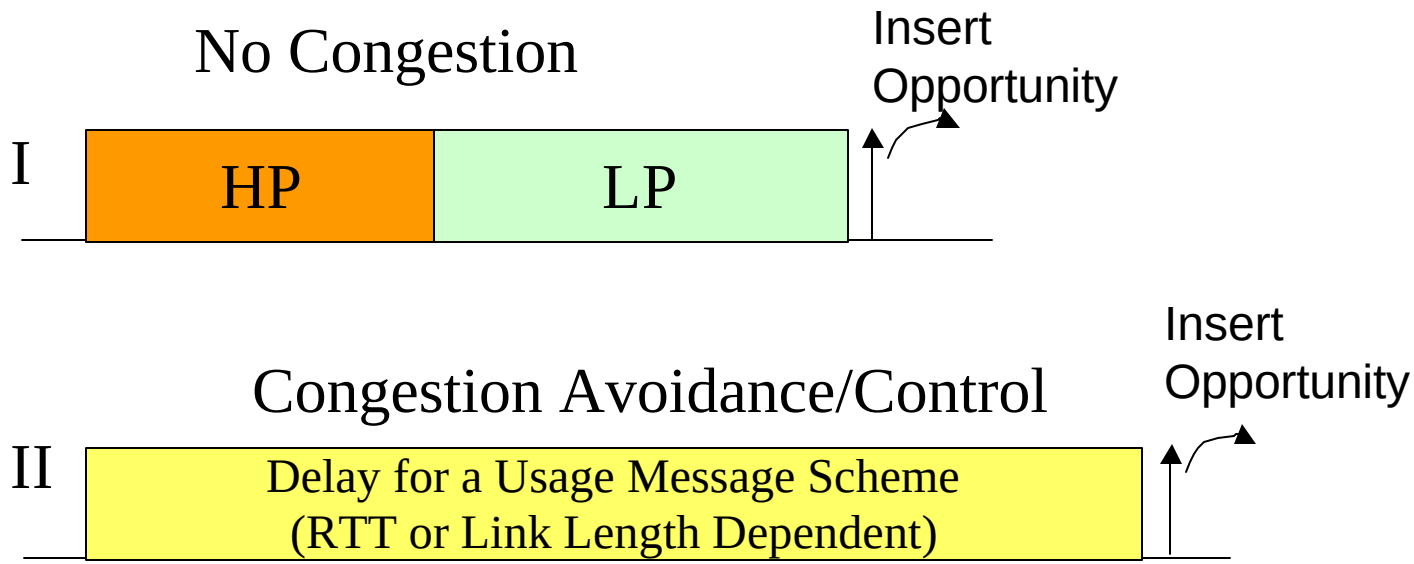


Virtual Deadlines (Tags)



Longer-term Unfairness

- Single transit buffer could cause significant delays even with reduced network utilization
- Even with multiple transit buffers, if LP traffic can override HP traffic under congestion, delays will still be too high



Ring Worst Case Fairness Index

A Ring Access Scheme is said to be worst-case fair[1] for flow f, if for any packet p in f, the following holds:

$$D(p) < \frac{Q(p)}{r_f} + C_f$$

The normalized Ring Worst case Fair Index (RWFI) is defined as

$$C = \max_f \frac{r_f C_f}{B}$$

D(p) delay of a packet p is the real time that elapses between the arrival time of p and the time p is completely transmitted.

[1]*Bennett Zhang, "WF²Q: Worst-case fair weighted fair Queuing," in Proc. IEEE INFOCOM '96, San Francisco, CA, Mar. 1996*

Ring Worst Case Fairness Index

Q(*p*)	Size in bits of the queue for flow *f* in front of and including *p* at the time of *p*'s arrival.
C_f	Constant independent of other flows sharing the ring
r_f	Rate guaranteed for flow *f*
B	Total ring bandwidth.

- Ideal value of RWFI = 0 for the hypothetical model
- **RWFI for our scheme is a constant (see slide #20)**
- For many existing schemes, RWFI will increase linearly with number of active nodes upstream and number of active flows!
- **All ring scheduling / fairness schemes should provide the value of RWFI for their scheme for comparison**

Ring Access Algorithm

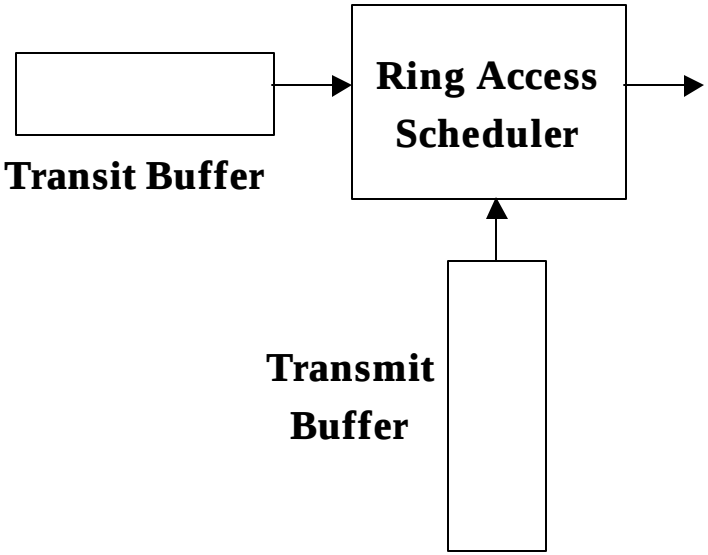
Assumptions

- Each packet is stamped with a tag or deadline (coarsened to reduce size), based on its virtual clock
- Tags are stored in a priority queue using a scheme such as the Leap Forward Virtual Clock Scheme (see backup)
- Packets also carry the packet-packet delay or rate information
- Within each node, packets are transmitted in the non-decreasing order of tags in the transmit queue

- This is a work conserving scheme

Ring Access Algorithm

Ring Access scheme is fair for all classes of traffic

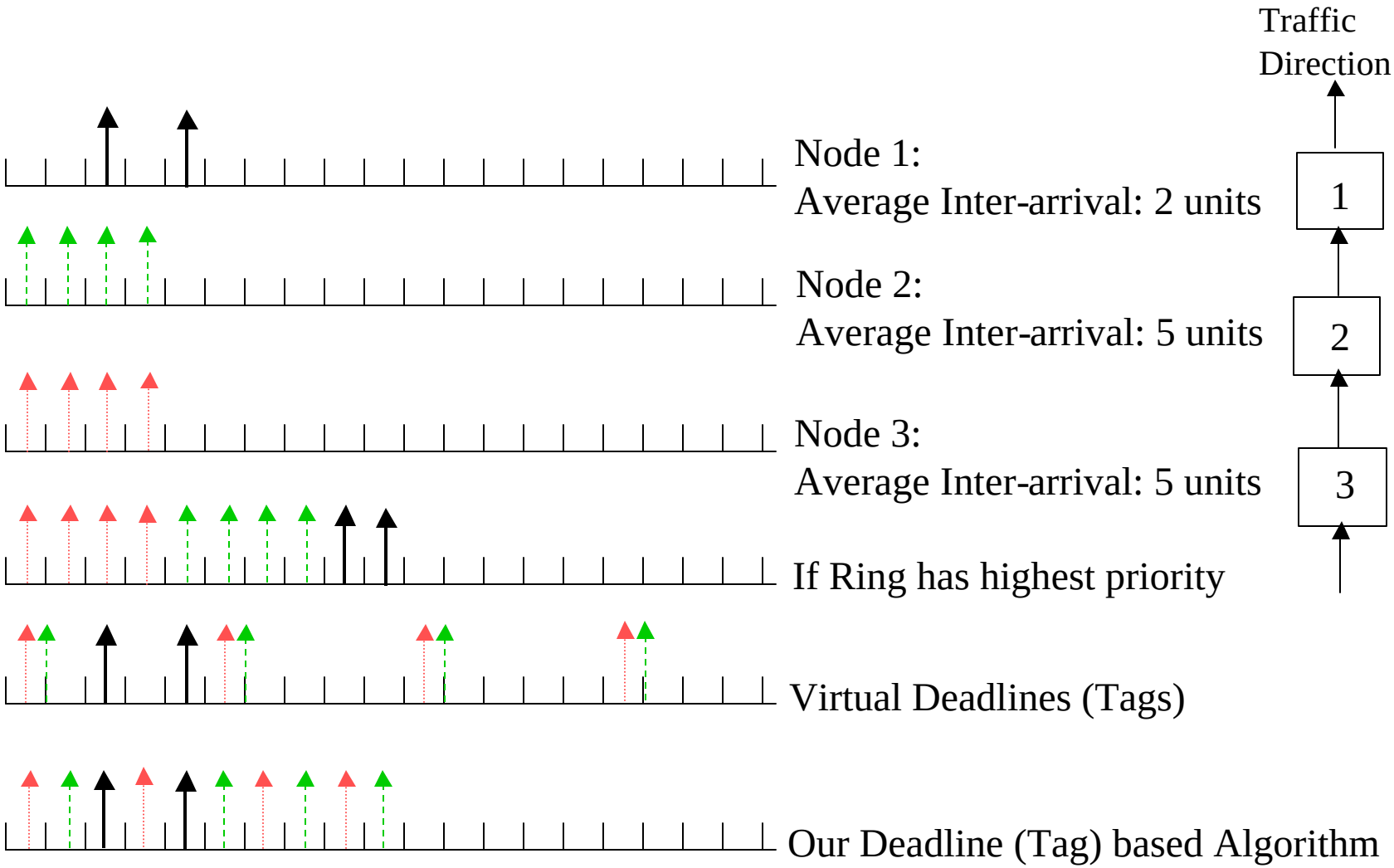


Algorithm

- Compare priorities. Highest priority (transit or transmit) goes first.

2. Compare the tag at the head of transmit queue and transit buffer. The smallest tag goes first. The transit packets may be Quarantined temporarily.
3. Break ties with packet-packet delay (average rate) info. Highest rate goes first.
4. If rates are equal, transit packet gets priority. Approximate WFQ / GPS order is achieved. Go to step 1.

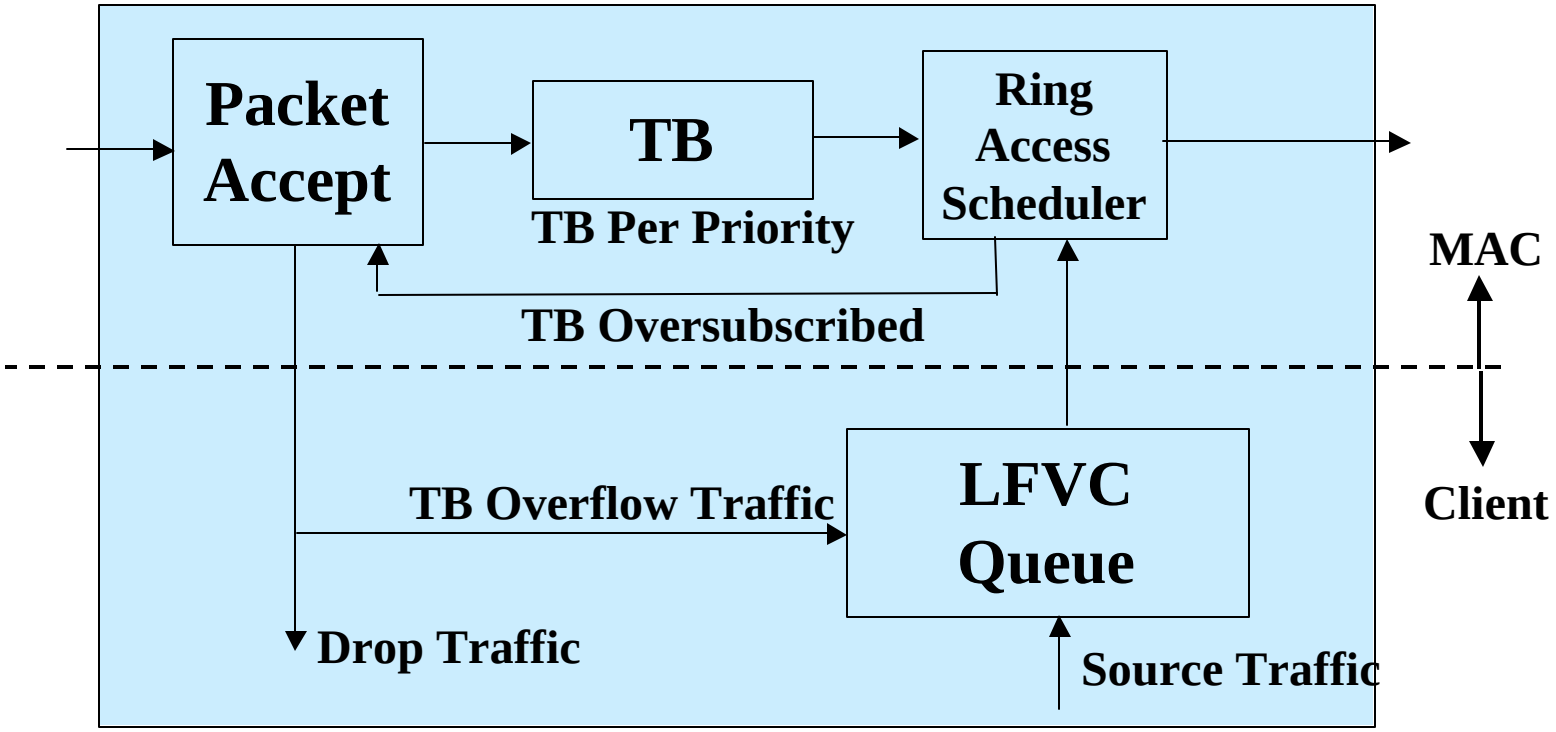
Example - 1


Traffic Direction
1
2
3
Node 1:
Average Inter-arrival: 2 units
Node 2:
Average Inter-arrival: 5 units
Node 3:
Average Inter-arrival: 5 units
If Ring has highest priority
Virtual Deadlines (Tags)
Our Deadline (Tag) based Algorithm

Tolerant Real Time Service

- Voice and Video applications that buffer data until the playback point
- Moderate (Bounded) delay, higher network utilization and degraded protection
- Ring Access Scheduler described can provide the minimum playback points
- Short bursts of each flow should be let through to provide “Controlled Sharing”
- Client’s scheduling should support this FIFO type of behavior as well (see backup)

Transit Path Design



Transit Path Design

- A sliding window of future tags in their order of transmission is maintained
- TB overflow implies rest of the packets have tags that are much higher (and therefore, can be delayed)
- Overflow packets are received and re-queued using existing tags
- Work conserving nature, coupled with this access algorithm provides minimum delay

Delay Bounds & Fairness

Since we are extending the LFVC scheme (described in the backup) on to the ring,
We can use the same equations
Assume the sources are leaky-bucket constrained (s, r)
Source Node – Destination Node Delay

$$\frac{s}{r} + (N-1)\frac{L_{\max}}{r}$$

N is the number of nodes between src and dest. L_{max} is the maximum size packet for this flow
Ring Worst Case Fairness Index (RWFI) is $\frac{L_{\max}}{B}$
Where B is the link bandwidth.
Note: This expression is approximate. The exact equation is available

Proposal

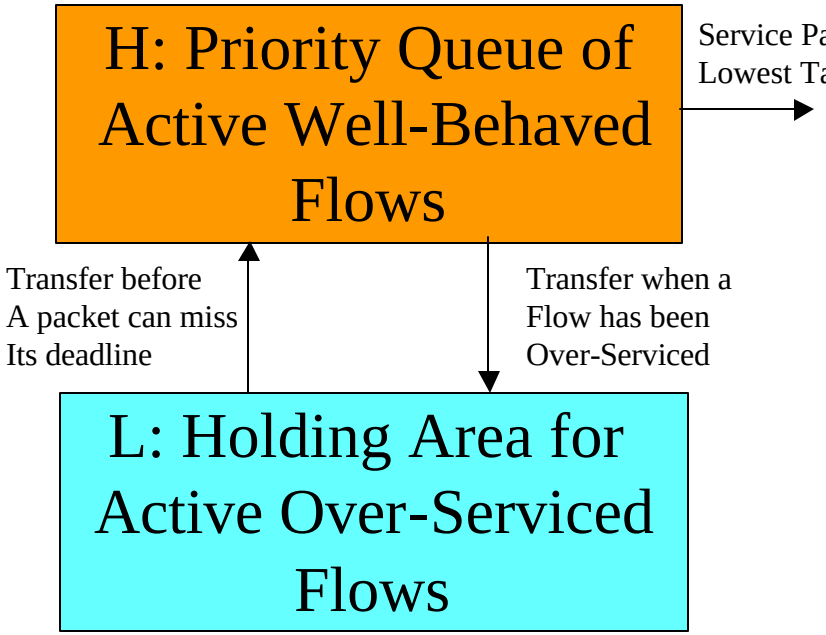
- Header needs to carry the coarsened tag
- Header needs to carry packet-packet delay or rate info
- Ring access scheduler state machine as proposed
- Transit path design as proposed

Conclusion

- We have described a scheme that can provide 100% network utilization, very tight delay bounds and optimal worst case fairness index (both are equivalent to WF^2Q)

BACKUP

Digression – Client Implementation



Leap Forward Virtual Clock (LFVC)[1]

1. Implement this structure for each class of service supported
2. Transfer criterion for each class Is varied to balance delay and throughput
   Guaranteed Service – Low delay / jitter, low network utilization, fully Protected
   Tolerant Real-Time Service – Moderate (Bounded) delay, high network utilization, degraded Protection
   Best Effort Service – Delay is Unbounded, high network utilization, degraded Protection

[1] *S.Suri, G. Varghese, G.Chandramenon. Leap Forward Virtual Clock. In Proceedings of Infocom 97, Kobe, Japan, Apr. 1997*

LFVC Scheme in the Client

Per Class of Service.

- Packets are tagged with their deadline value based on a discrete virtual clock.
- Packets are placed in a priority queue H (keyed off of t_f, the tag).
- Packets from a flow that has been over-serviced, are moved once to a low priority queue, L. Packets are always transmitted in their non-decreasing order of tags, from H.
- Packets are moved back once from L to H before they can miss their deadline.

Client Implementation

- LFVC style scheme for all classes of service
- A Tag Coarsening scheme reduces the complexity to O(log log N)
- Delay and Throughput requirements can be balanced for different services
- Increases utilization of real-time services on the network – increase in revenue

Tolerant Real Time Service

- Transfer from H-L is relaxed to simulate FIFO like behavior with an upper bound on delay
- The criterion is approximately $T_f > K\Delta_f$

  where Δ_f is the time required to transmit the largest size packet at the flow's rate, and K is the relaxation factor (think of it as how many maximal length packets are transmitted from this flow, f before the transfer to L)
- The traffic in this class is managed through admission control

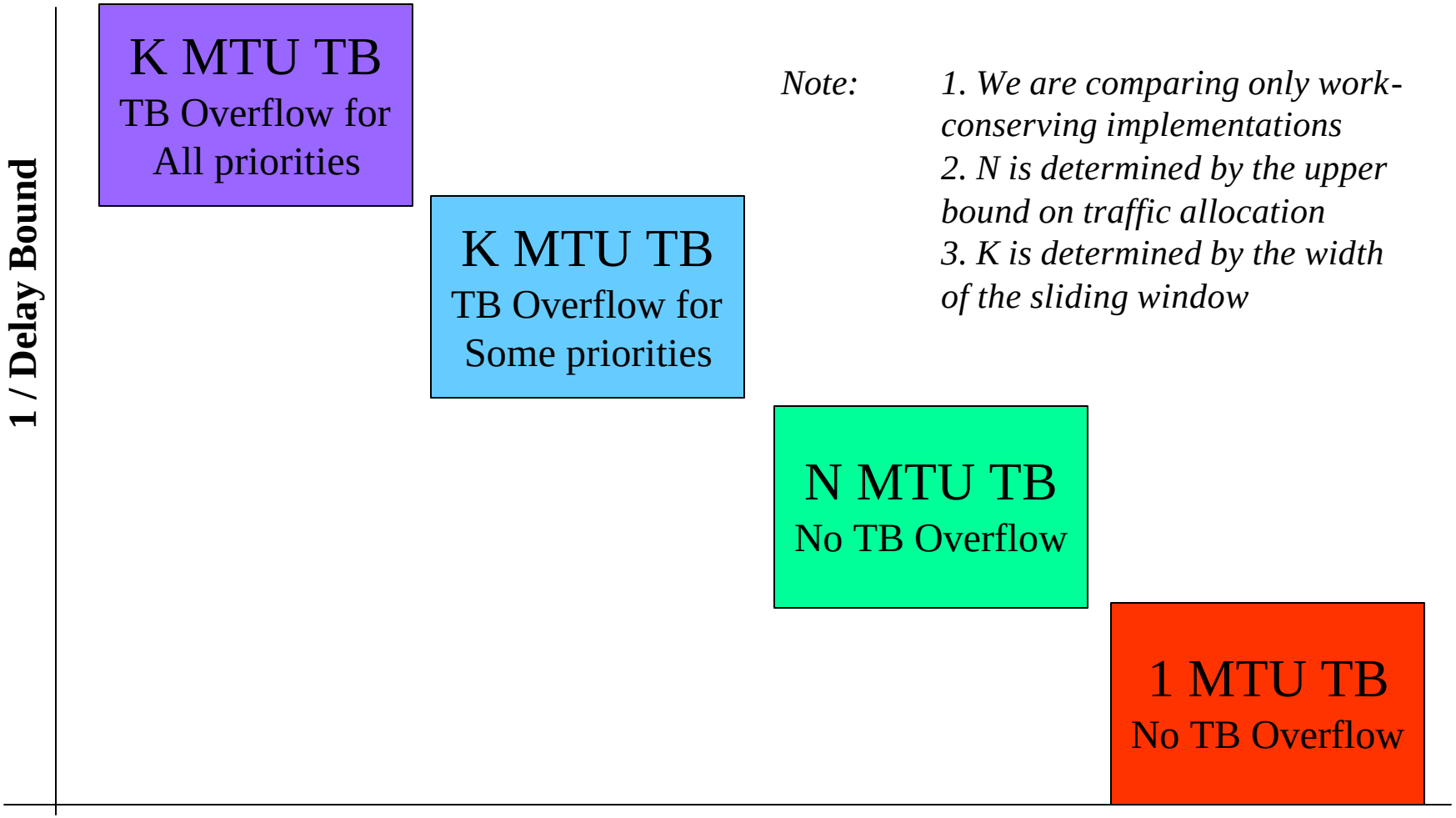
Best Effort Service

- It can be thought of as the trivial case of the LFVC scheme where there is no L buffer
- Multiple flows can still be maintained if necessary and bandwidth and fairness can still be provided at a gross level
- Congestion control only applies to this class

Service Provider's View

- The new and improved guaranteed service will be the premium service with potential to generate more revenue than currently – very tight delay bounds
- The tolerant / adaptive real time service with absolute upper bound on delay (and degraded protection), should have more customers (mainly because of acceptable service quality and monetary incentives) – more revenue due to volume
- Best effort as always

A Class of Implementations



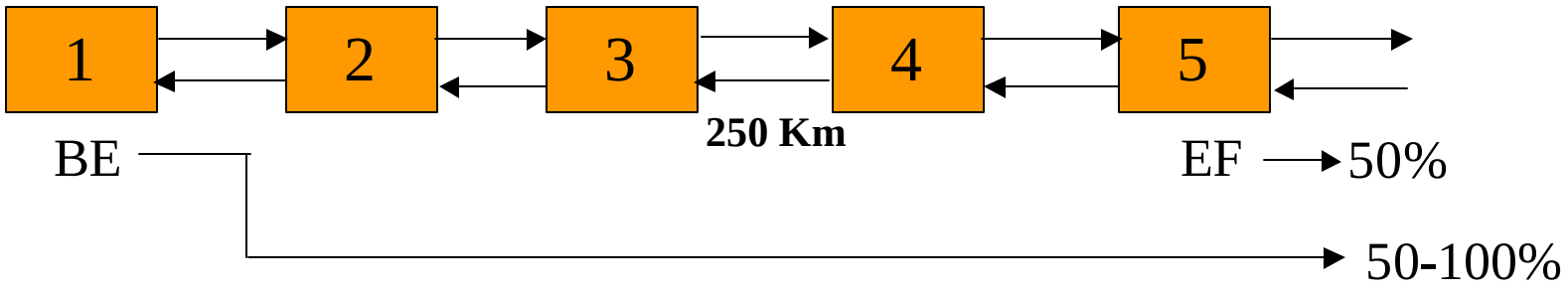
Note:

1. We are comparing only work-conserving implementations
2. N is determined by the upper bound on traffic allocation
3. K is determined by the width of the sliding window

Congestion Control

Assumptions:

1. Ring size is only limited by ETE delay of guaranteed services
2. Average node-node distance = 250 Km
   Higher distances will still work with increased queuing delays for the rest of BE traffic.
   Look at step 5 in the Algorithm.
3. Worst Case – 50% EF, 50% BE Traffic
4. Buffer size calculations for a 10G Ring



Congestion Control

Algorithm

1. Monitor Input and Output rates of Transit Buffer(Best Effort).
2. If the difference is 50%, send out an ECN message to the neighbor upstream to reduce BE traffic by 50%
3. The neighbor immediately starts buffering up to 50%. The congested node should see BE traffic drop by 50% after a maximum ring link time – 1.25ms for 250 Km (1-2K Ring size)
4. If some or all of the BE traffic is from an upstream neighbor, a new message is passed upstream with the new BW reduction request, and so on. The message stops when an upstream node detects no BE traffic from upstream. Each node reduces its own BE traffic in proportion to the amount it is currently sourcing beyond its allocation.
5. If the buffer overflows (the only reason is link > 250 Km), Receive the rest of the packets and pass it up to the traffic manager for re-queuing – Fail Safe

Transit Buffer Sizing

BE Transit Buffer:

Option 1

A 250 Km link will equate to 780 KB for 50% BE traffic. Twice that = 1560KB to account for the round trip time.

Note that Transit Buffer does not have to be at high water mark for ECN message. It is sent out as soon as the difference in rates Reaches 50%.

Including an additional 500 us for LPF rate estimation,

Transit Buffer for BE is sized at 2MB for each direction

Option 2

Since we have to build a fail-safe mechanism any way, we opt not to include any transit buffer for BE traffic. Instead, all the BE traffic is received and re-queued by the traffic manager.

Option 2 seems to be more attractive than Option 1! The only justification for option 1 is some reduced queuing delay when there is no congestion.

Transit Buffer Sizing

EF Transit Buffer (Guaranteed Service)

With this Algorithm, 1 or 2 MTUs is no longer sufficient. We recommend about 8-10 MTU. This will handle the short-term or longer-term throughput unfairness discussed above.

AF Transit Buffer (Bounded Delay)

This class, if supported, will be controlled through admission control mechanisms. Supporting short bursts would require about 8-10 MTU.