Residential Bridging
(rate control; 2006Mar06)

Maintained by David V James

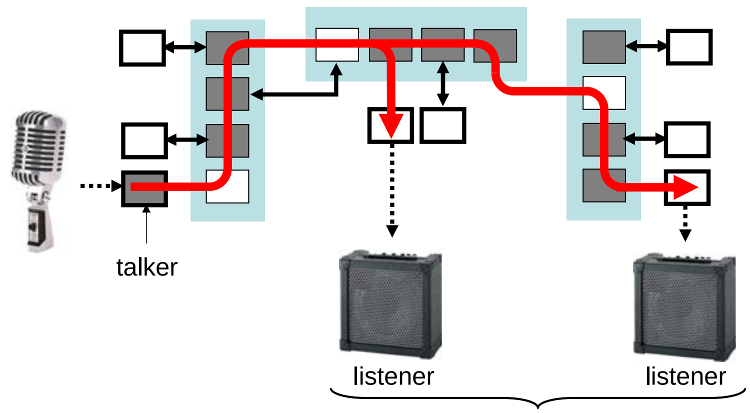
Credit is due to many others, whose reviews and comments drove this content.

Categories of work

- – Service discovery (out of scope)
  - Identify/control “talkers” and their available “plugs”
- – Subscription (802.1 centric)
  - Establish conversation between talker and listener(s)
  - Reject unless: *linkBandwidth* < *linkCapacity*
- – Clock synchronization
  - Synchronous reception, forwarding, and presentation
- – Pacing
  - Talkers must not be well behaved
  - Bridges should “sustain” such behaviors
- – Formats
  - Frame formats and content (stream IDs, time stamps)
  - Time aware service interfaces

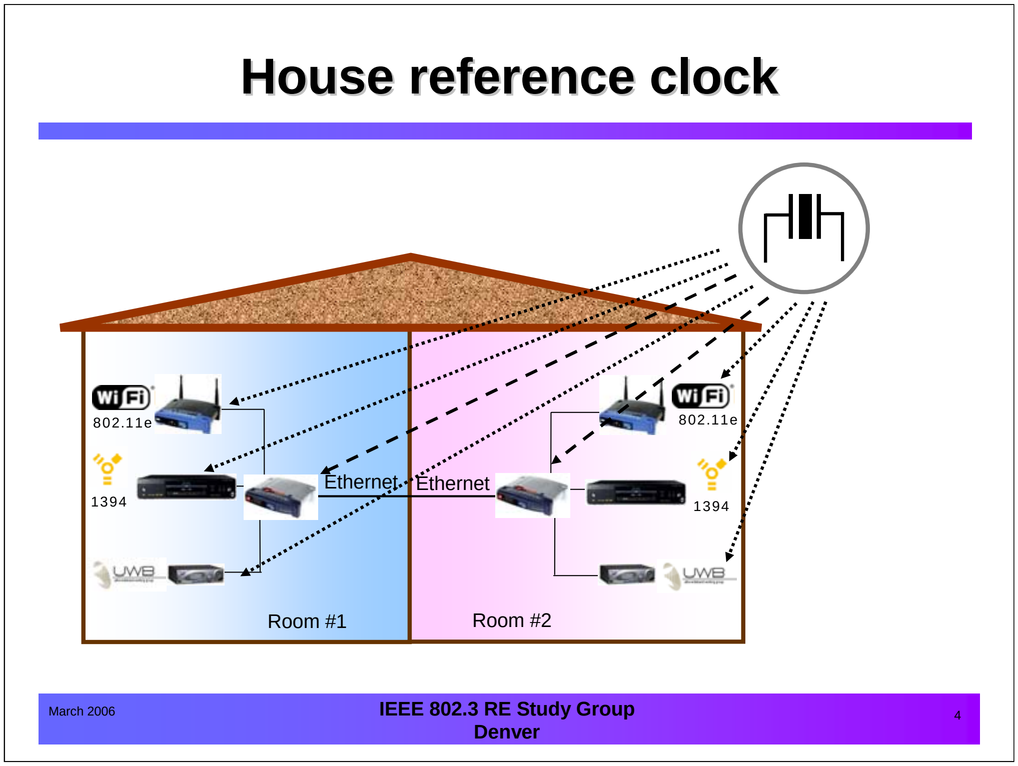
A rough breakdown of the work is described herein.

Vocabulary terms (1)



Other options are solicited

prefer not to use source/destination, due to a variety of handshakes (that sometimes go in opposite directions to the isochronous traffic)

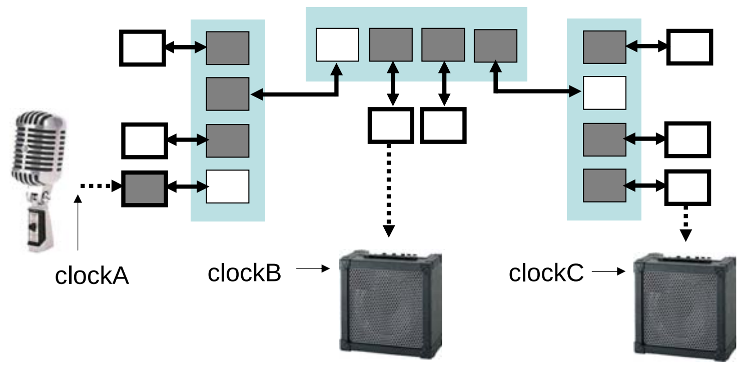


In support of synchronous transfers, all RE devices are assumed to have the same impression of time.

For this presentation, assume an 8kHz cycle time, although a decision on this value has not been finalized.

Requirement: 8kHz cycle frequencies are locked and the “same

Synchronized reception/presentation



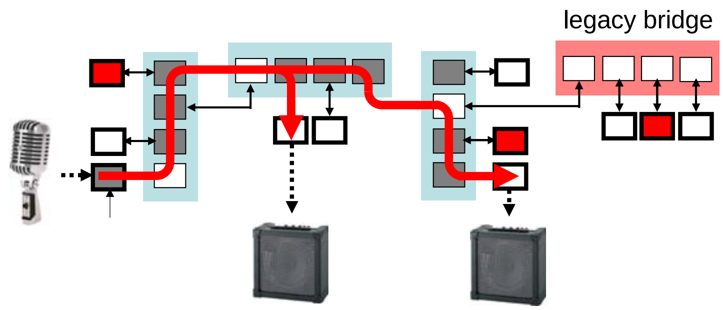
No long-term drift: clockA, clockB, clockC
Clock jitter: sub nanosecond (after PLL)

Bridge reclocking has a relatively modest clock-sync accuracy requirement, where microsecond deviations could be acceptable.

Source-data and presentation-data clocking requirements are more severe.

1) Frequency drift is unacceptable, since dropped/replicated values are audible.
2) Presentation time jitter is sub nanosecond, based on slew rates and D/A accuracies.

Ethernet compatibility (yes!)



Legend:
legacy bridge
legacy endpoint

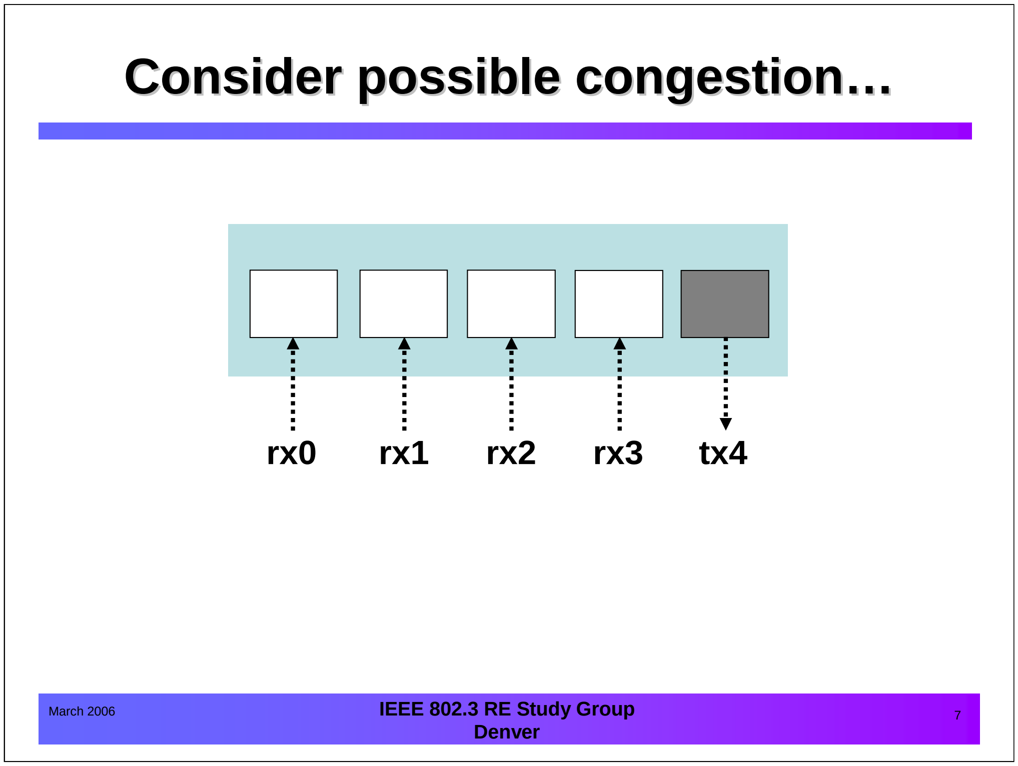
Is this Ethernet compatible? YES!

ALL non-RE traffic is unaffected (except for delays, due to loading)

ALL control traffic uses regular Ethernet frames

However:

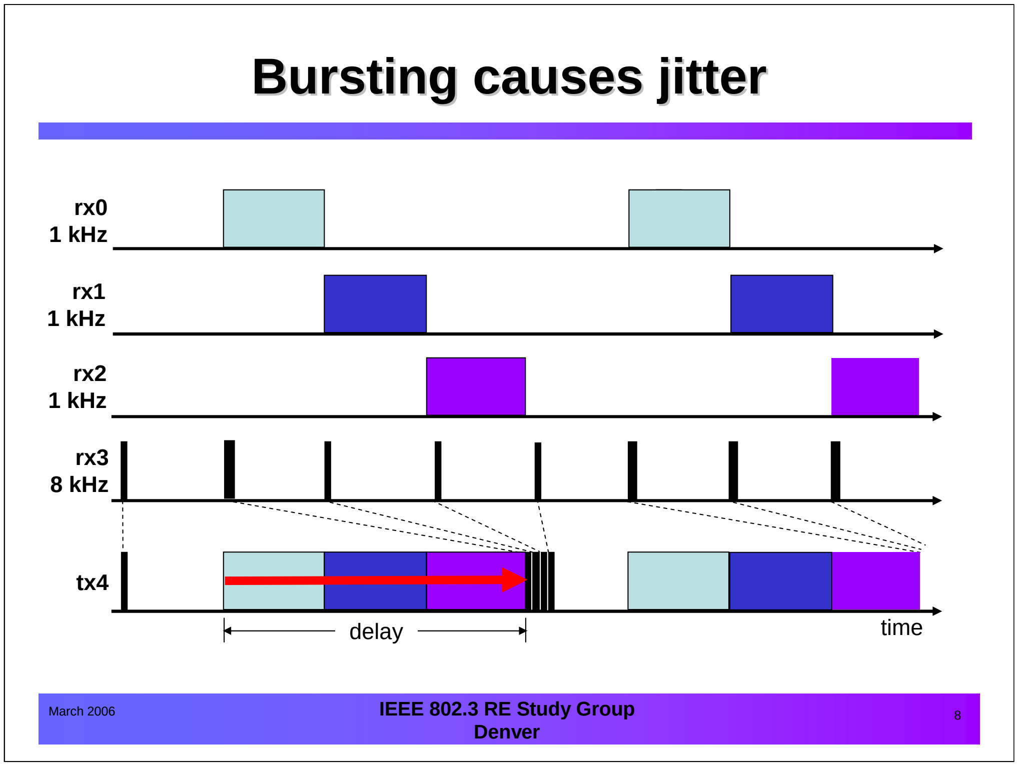
- RE traffic does not pass through legacy bridges
- RE traffic is ignored by legacy stations



Consider a bridge, with four inputs streams (coming from different talkers) and one a common listening bridge or endstation.

Of course, the sum of {rx0,…,rx3} must be less than the capacity of tx4.

Even if average rates are so limited, bursting and bunching could (if allowed to accumulate) violate this cumulative bandwidth constraint for short periods of time.



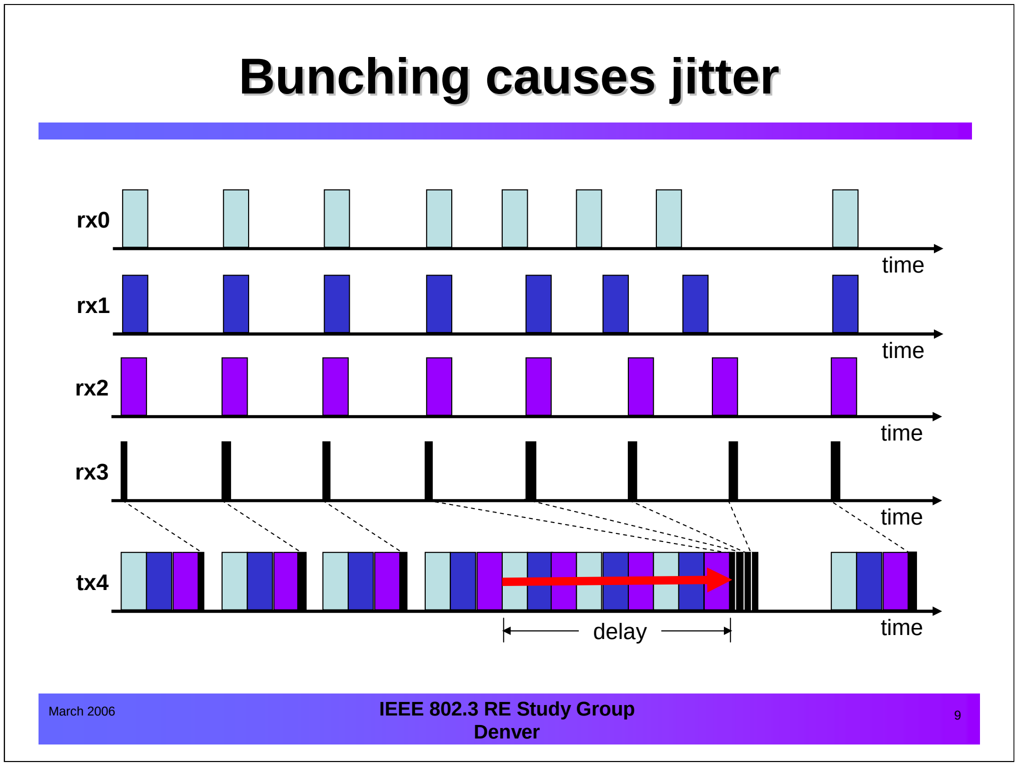
We assume transfers will be cycle-clock based, with an 8 kHz clock.

To illustrate why this is critical, suppose that one allowed distinct cycle-clock rates, such as concurrent 1 kHz and 8 kHz transmissions.

While the long term bandwidths are less than the link capacity, the short-term bandwidths exceed the link capacity.

And, from the perspective of the 8 kHz transfers, the 1 kHz transfers appear to be transient line-rate transmissions.

Excessive 8 kHz delays would then occur during the unfortunate transients.



To solve bursting, force sources send at 8 kHz, using only frame lengths (not transmit-frame periodicity) to adjust their transmission rates.

Is this sufficient to bound the latency delays?

No, there is still the problem of bunching, caused to “early” arrivals.

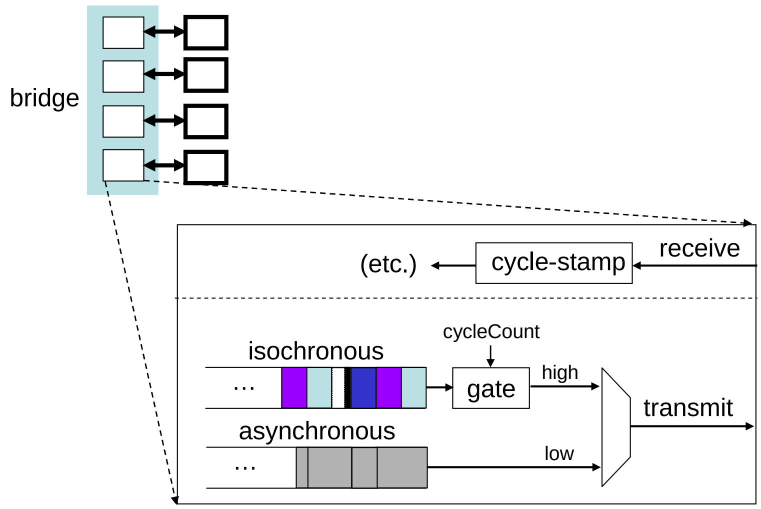
Consider, for example, a bridge that receives some isochronous data “early”.

The cumulative effect of multiple early transmissions is similar to bursting: a low-rate transfer can be blocked for multiple cycles.

The basic problem is the cumulative effect of bursting and bunching, which depends on the number of bridges, the number of bridge ports, and the interconnection topology.

The cheapest solution is to stop the cumulative effects, so that an end-station and retransmitting bridge port have isochronous timing behaviors.

Bridge re-clocking contains jitter



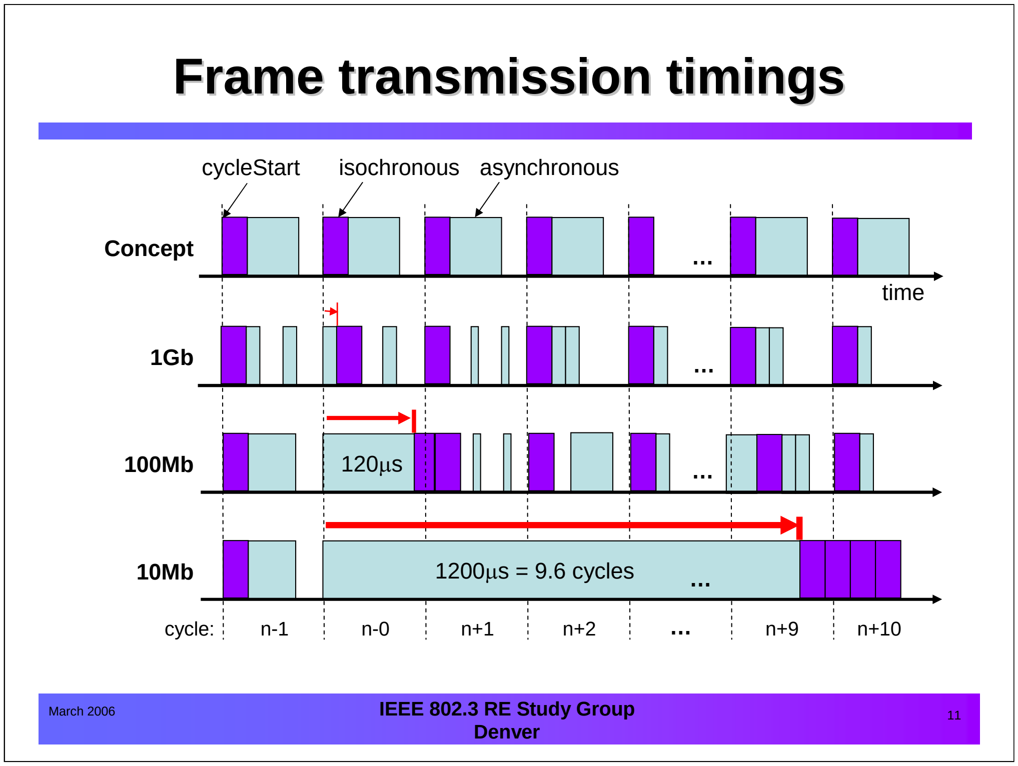
To ensure the same end-station/bridge behaviors, the same transmission model is assumed.

Isochronous data is not immediately transmitted at highest priority.
 * The isochronous data is staged in the transmit queue,
   before the target transmission cycle.
 * When the cycleCount is reached,
   the isochronous data is transmitted at highest priority.

This _does_not_ ensure the minimal transmission latency, since frames are often delayed “unnecessarily”, to the start of the “next” cycle.

Depending on link bandwidths, max packet sizes, etc., “next” could be more than one cycle (four isochronous cycles is simple and more than sufficient).

This _does_ ensure a bounded min/max transmission latency, and that latency remains unaffected by the number of bridge traversals, port counts, etc..



The isochronous concept is to:

1) Partition time into 125us intervals, called cycles
2) At the beginning of each cycle, transmit all active isochronous traffic.
3) After the isochronous traffic, transmit asynchronous traffic till the next cycle.

On a 1 Gb/s link, concept and reality are closely matched.
The maximum 1500 bytes frame extends only slightly into the next cycle.
Thus, the isochronous limit (~75%) keeps isochronous traffic within its cycle.

On a 100Mb/s link, visible differences between concept and reality are larger.
The maximum 1500 bytes frame extends far into the next cycle.
Thus, some isochronous traffic can be forced into the next cycle.

On a 10Mb/s link, dramatic differences between concept and reality are visible.
The maximum 1500 bytes frame extends through multiple cycles.
Thus, some isochronous traffic can be delayed for 9.6 cycles.

The 9.6 cycle delay associated with 100Mb/s links is architecturally acceptable, and could be factored into bridge designs. However, bridges/endstations with these much larger overrun-avoidance buffers, would introduce longer delays.

This could (perhaps) be marginally tolerable, but only if all ReBridge-to-ReBridge links were constrained to support 100Mb/s or higher rates.

Traffic classes

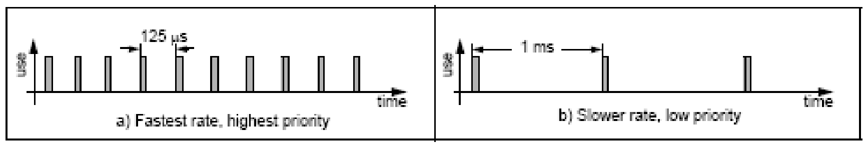


Figure 9.1—Rate-based priorities

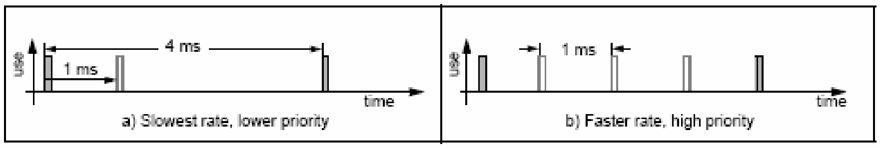
There are two rate-based priorities:

8kHz for the latency-sensitive audio

1kHz for the less latency-sensitive video

Preferred classB traffic is also supported, but has has lower precedence.

Transmitter rate controls



A slower transmission rate can get reduced latency.

This comes at the cost of "half-faking" the higher bandwidth requirement.

- The total classA1 traffic (including the ghost images) must be less than 75%.
- The total classA0+classA1 traffic (w/o ghost images) must be less than 75%.

Transmitter rate controls

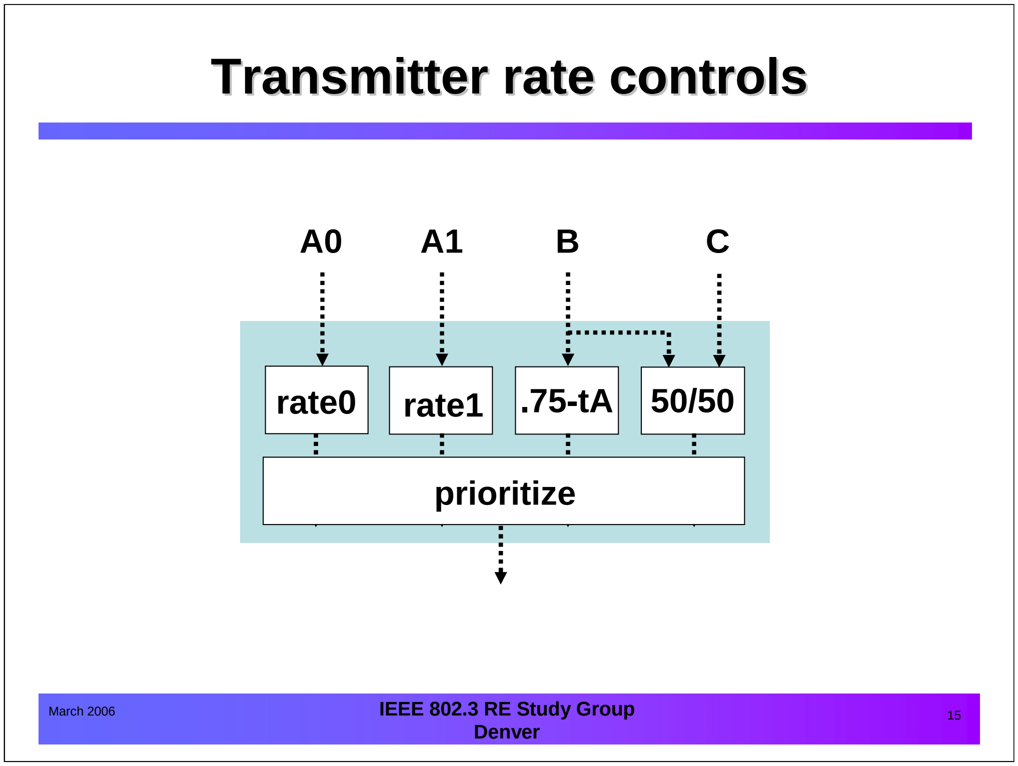
Table 9.1—Tagged priority values

Code	Interval (ms)	Name	Description
0	n/a	CLASS_C	Best effort, with minimal guaranteed BW
1	n/a	CLASS_B	Preferred, with minimal guaranteed BW
2-3	—	—	Used for other purposes
4	1.0	CLASS_A1	Guaranteed BW over short interval
5	0.125	CLASS_A0	Guaranteed BW over shorter interval
6-7	—	—	Network management

The classA0/A1 are guaranteed bandwidth and latency.

The classB is prioritized.

The classC is guaranteed a residual.



Transmission rate-limiting protocols.

classA0: limited by rate0 shaper

classA1: limited by rate1 shaper

classB: cumulative classA0/A1/B is limited to 75%, classA0/A1 has precedence.

classC: guaranteed 50% of what is left over.

Transmitter rate controls

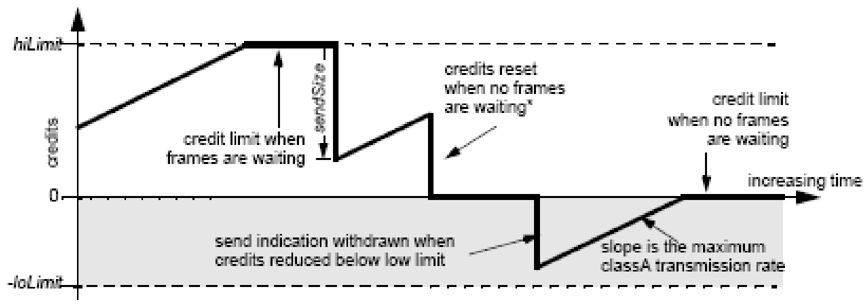


Figure 9.3—Credit-based shapers

A shaper limits the traffic over time.

The credits never accumulate when nothing is ready to be sent.

Transmitter rate controls

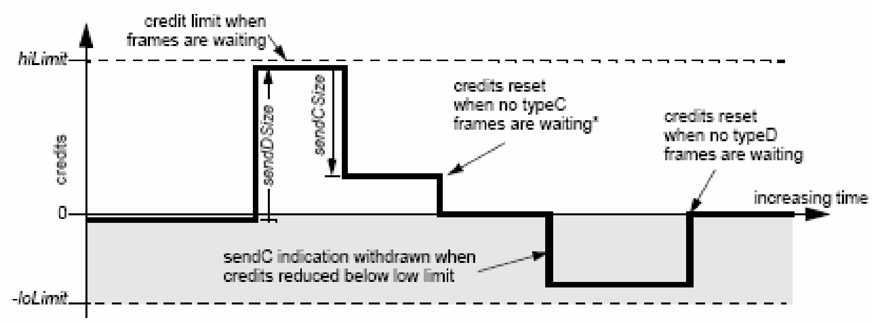


Figure 9.4—Pacer credit adjustments over time

In the case of classB/classC, the goal is equal bandwidths.

To achieve this goal, time is not involved.

Credits are adjusted positive/negative based on classB/classC transmissions.

Synchronized time-of-day clocks

Questions?

Opportunity for questions.