P802.1Qat Delay and Bandwidth Parameterization

Parameters for delay and bandwidth capacity calculations for IEEE P802.1Qat SRP

Version 4

Norman Finn

Cisco Systems

Introduction

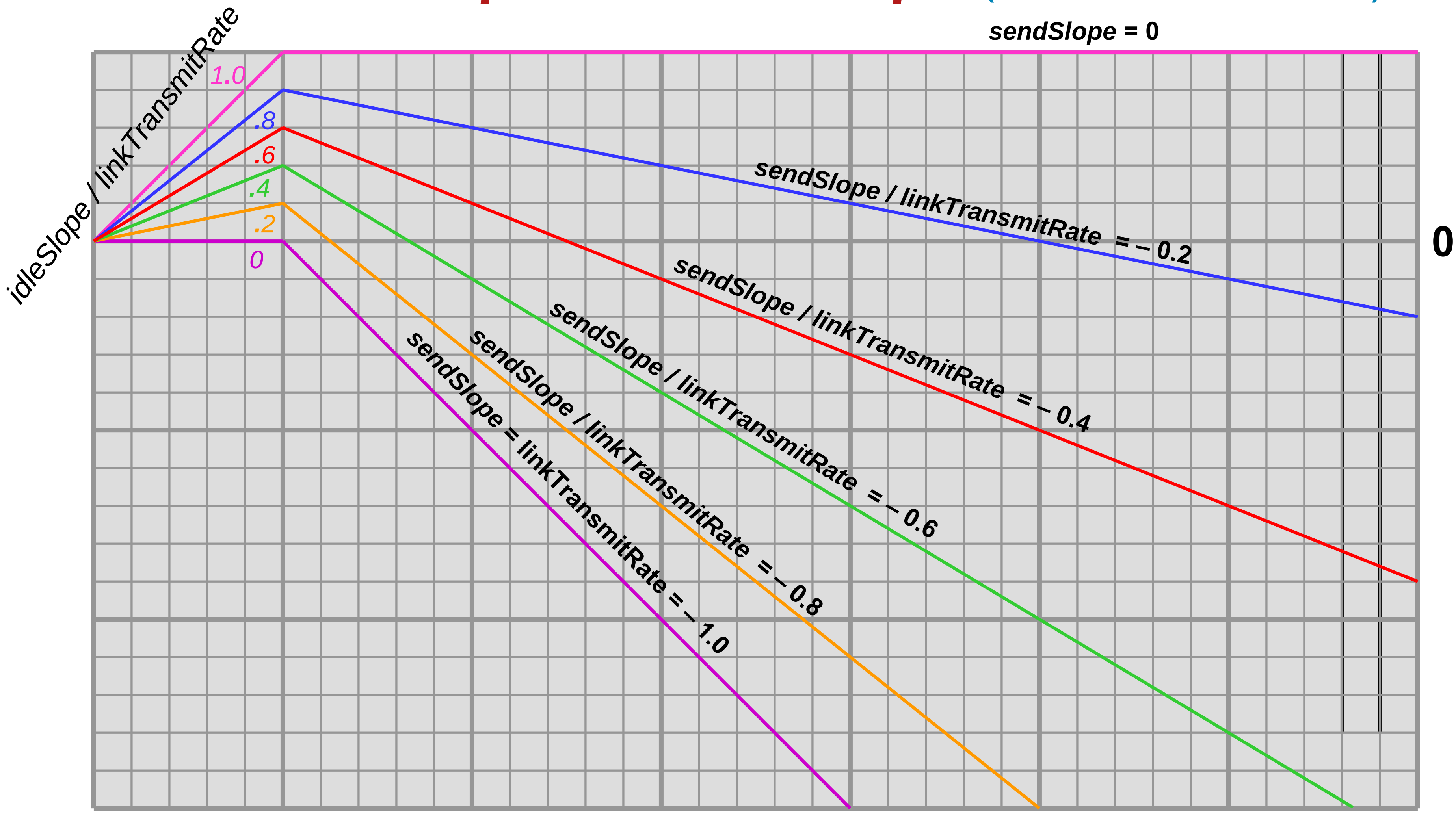
Introduction

- The current revision of the assumptions document says:
  - Maximum Interference Amount per Hop
    - Class A: 1 Max size frame + Sum of the Maximum size of the Class A frames on each of its other ports – Ref 5
    - Class B: 1 Max size frame + 1 Max size Class A burst (based on max Class A BW allocation) + Amount of other Class B frames on each of its other ports
- This presentation will attempt to define what "Max size Class A burst" means, and extend the concept to any number of Classes.
- This will lead us to the appropriate management parameters to use to characterize the per-Class and Per-Port limitations on bandwidth reservations.

IEEE P802.1Qav Draft 2.0

- Variables appearing in *italics* are from IEEE P802.1Qav Draft 2.0, e.g. *idleSlope*.
- Subscripts may be added indicating a per-Class value. For example, $idleSlope_X$ would be the total data rate for reservation for Class X on a given output port.

Credits: *idleSlope* vs. *sendSlope* (P802.1Qav/D2.0)



- *idleSlope / linkTransmitRate* and *sendSlope / linkTransmitRate* for various data rates

Disclaimer

- I would be surprised if this whole presentation is not in a textbook, already.
- But, I have not read that textbook.
- In the meantime, here is the information.
- If someone can provide a reference to the textbook, the Task Group, including me, would be grateful.

Latency Calculations

Worst-case latency contributions

- The worst case latency for a single hop from Bridge to Bridge, measured from arrival of the last bit at Port *n* of Bridge A to the arrival of the last bit at Port *m* of Bridge B, can be broken out into the following components:
  - Input queuing delay. (There are no input queues in the 802.1 architecture, but if present, the implementation must account for them.)
  - Interference delay. (The subject of this presentation.)
  - Frame transmission delay. (One maximum frame time at output line rate for non-cut-through architecture.)
  - LAN propagation delay. (Depends on length of output wire, measured by P802.1AS.)
  - Store-and-forward delay. (Includes all forwarding delays, assuming that the input and output queues are empty.)

Store and forward delay

- Store and forward delay includes all delay causes other than those enumerated in the previous slide. This would include, for example:
  - Time needed to pass from the input port to the output port, assuming empty queues.
  - The difference, if any, in the delay incurred by a frame that bypasses an empty queue, vs. that incurred by a frame that must be enqued.
  - Time added by the lengthening of the frame due to additional frame headers such as Q-tags or Sec-tags (may be negative).
  - Time needed to encrypt an 802.1AE frame.

Interference delay

- The interference delay for frame X can be broken out into the following components:
  - Queuing delay: Caused by the frame that was selected for transmission an arbitrarily small time before frame X arrived (became eligible for transmission selection), plus the delay caused by queued-up frames from all 802.1Qat frames with higher priority than frame X's class (i.e., the "max burst size" for SR Classes with higher priority than X).
  - Fan-in delay: Caused by other frames in the same class as frame X that arrive at more-or-less the same time from different input Ports.
  - Permanent delay: Frames that reside in a buffer for a long time, relative to the output queuing delay, because of the history of activity in the network.

Queuing delay

Queuing delay

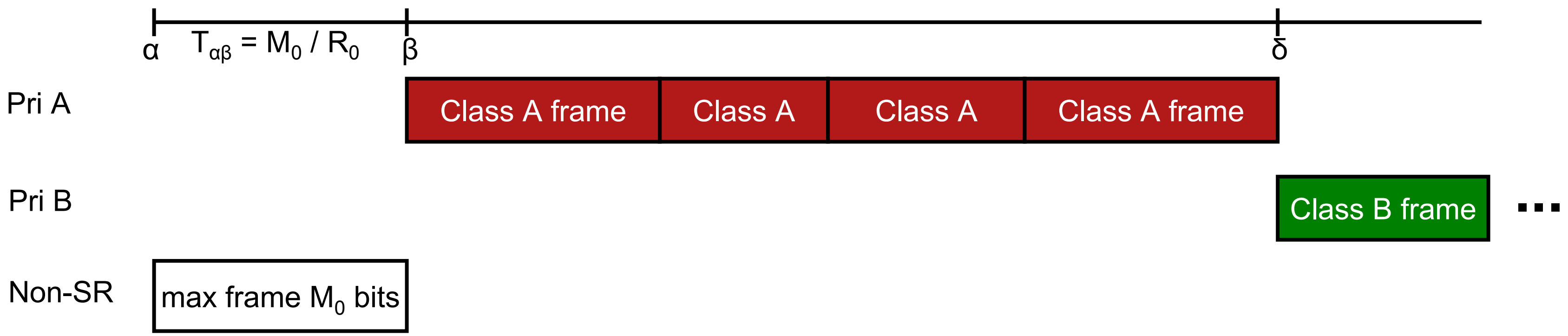
- The queuing delay for frame X can be broken out into the following components:
  - The frame that was selected for transmission an arbitrarily small time before frame X arrived (became eligible for transmission selection).

    This is well understood – it is *maxInterferenceSize* / (Line rate).
  - The delay caused by queued-up frames from all 802.1Qat frames with higher priority than frame X's class (i.e., the "max burst size" for Class X).

Max-size Class A burst

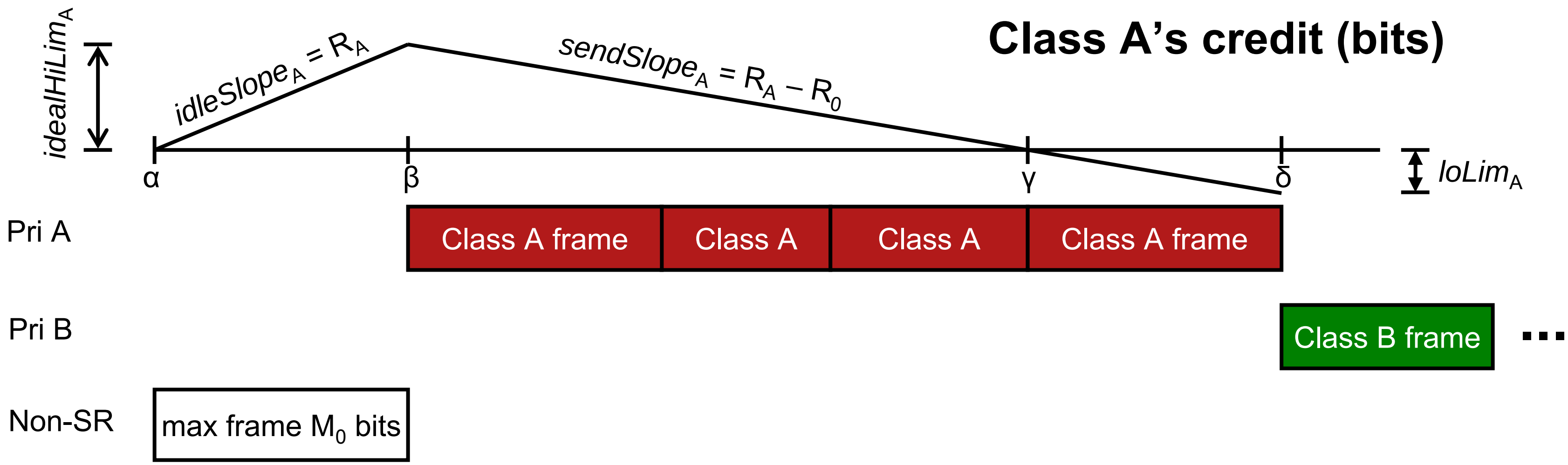
- Suppose that the queue for Class A is full, and has accumulated the maximum amount of credit.
  - Because Class A frames have priority over all other traffic (even BPDUs), the maximum credit for Class A is merely the credit accumulated during the “one max frame transmit time” required to transmit a lower-priority frame.
- Until the that credit is gone, Class B (C, D, ...) frames cannot be transmitted.
  - If Class A were permitted to use 100% of the LAN bandwidth, then the Class A queue would never catch up, because it would use credit as fast as it was gained.
  - If Class A were permitted to use 99% of the LAN bandwidth, then that max accumulated credit would be drained at 1% of the LAN bandwidth, until it is gone.

Class A queue latency



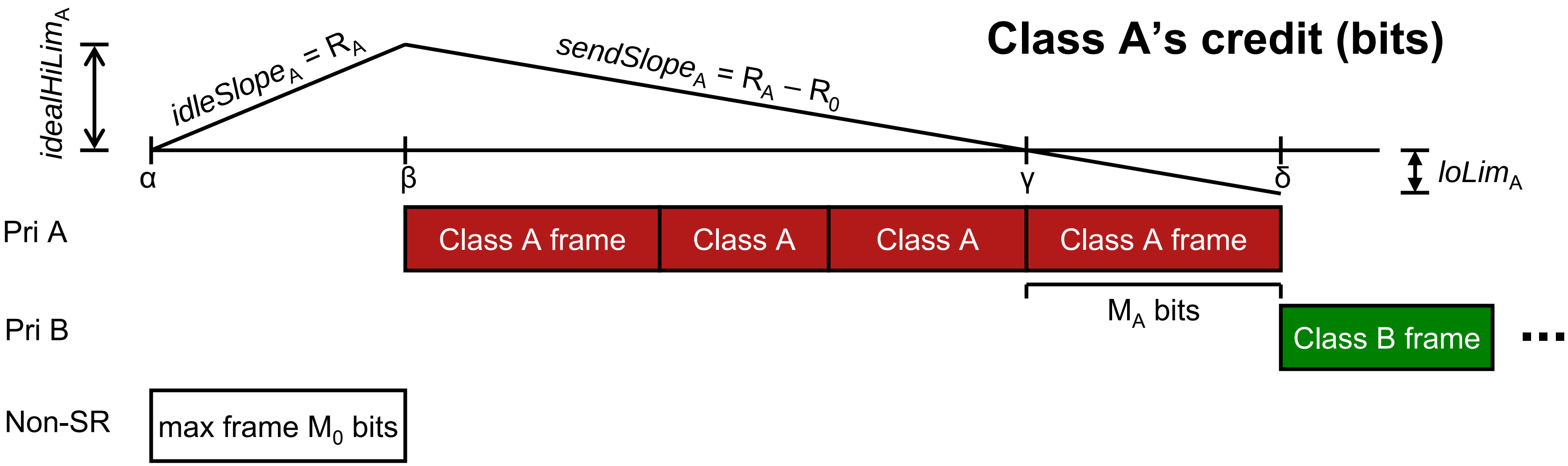
- At point α, the Class A and Class B queues are empty (else, they would be sending), so a maximum length (M_0 bits) non-SR frame starts. An instant later, Class A and B frames arrive.
- Let R_0 = the LAN data rate (*linkTransmitRate*). Class A sends at time β; its queue latency $T_{\alpha\beta} = M_0 / R_0$.
- Class B starts sending at time δ.

Max-size Class A burst



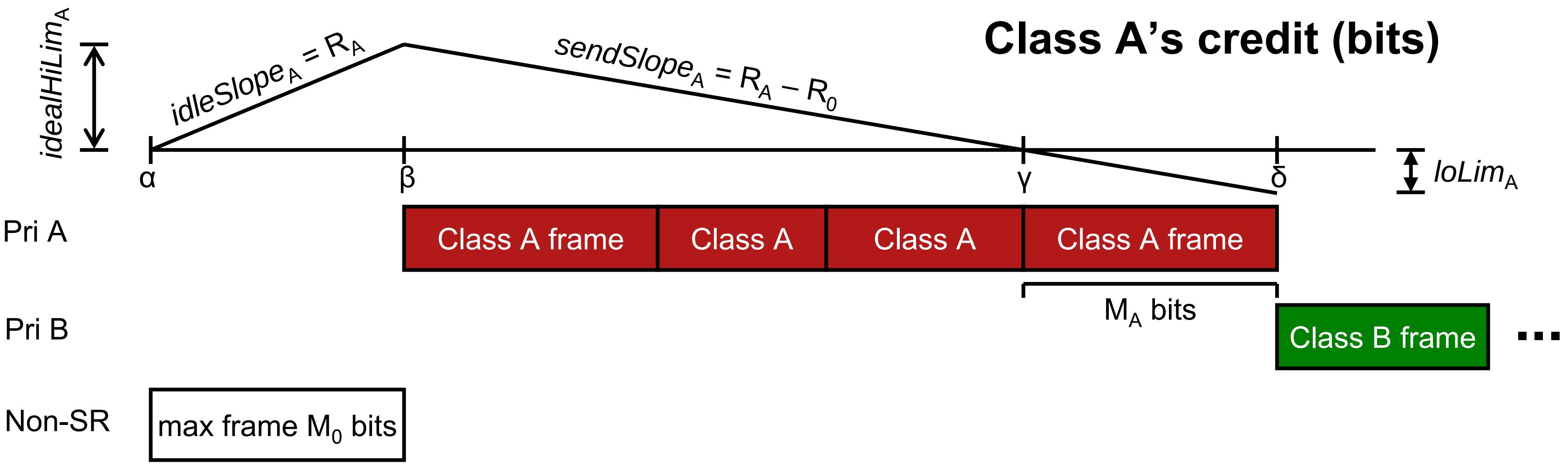
- Let R_A be Class A's reserved data rate, R_B for Class B, etc.
- Class A accumulates up to *idealHiLim*$_A = R_A \bullet M_0 / R_0$ credits at the rate *idleSlope*$_A = R_A$ during the max frame transmission.
- This credit is drained at the rate *sendSlope*$_A = (R_A – R_0)$, which is negative, down to 0 at point γ.

Max-size Class A burst



- Since a frame can be transmitted when credits = 0 at point γ, Class A's credits continue to drain to the value *loLim*$_A$ = $(R_A – R_0) \bullet M_A / R_0$, as one more maximum-length frame (M_A bits in time M_A / R_0) is sent.
- Class B can start sending at point δ.

Max-size Class A burst



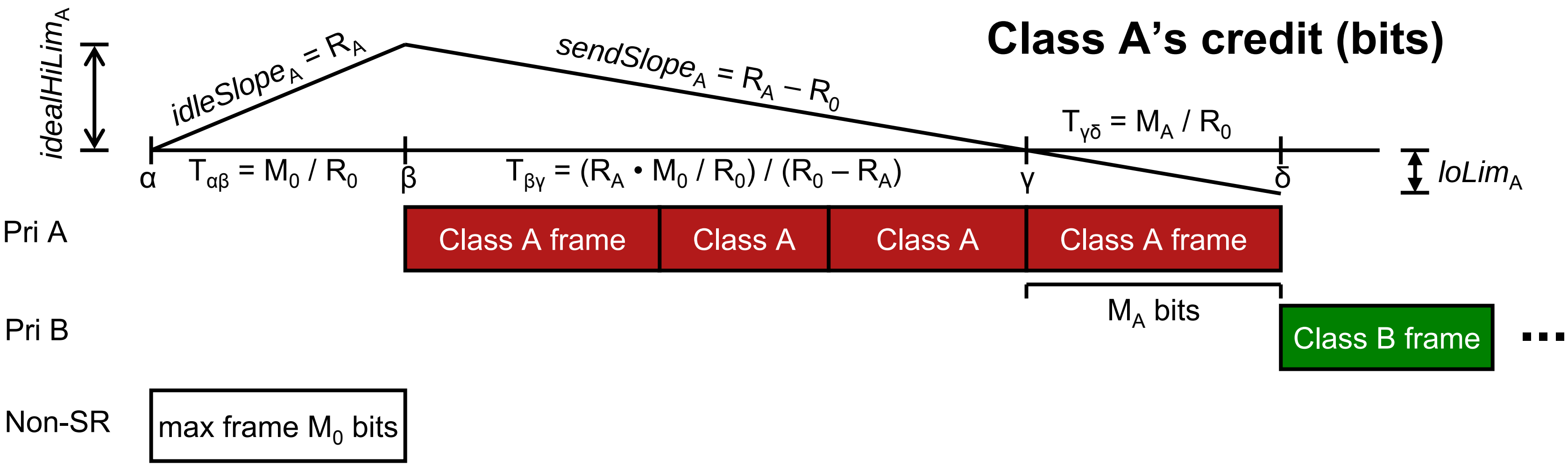
- max Class A burst size

  $= -\,(\mathit{idealHiLim}_A - \mathit{loLim}_A)\,/\,\mathit{sendSlope}_A$

  $= -\,(R_A \bullet M_0 / R_0 - (R_A - R_0) \bullet M_A / R_0) / (R_A - R_0)$

  $= (R_A \bullet M_0 / R_0) / (R_0 - R_A) + M_A / R_0.$

Class B queue latency

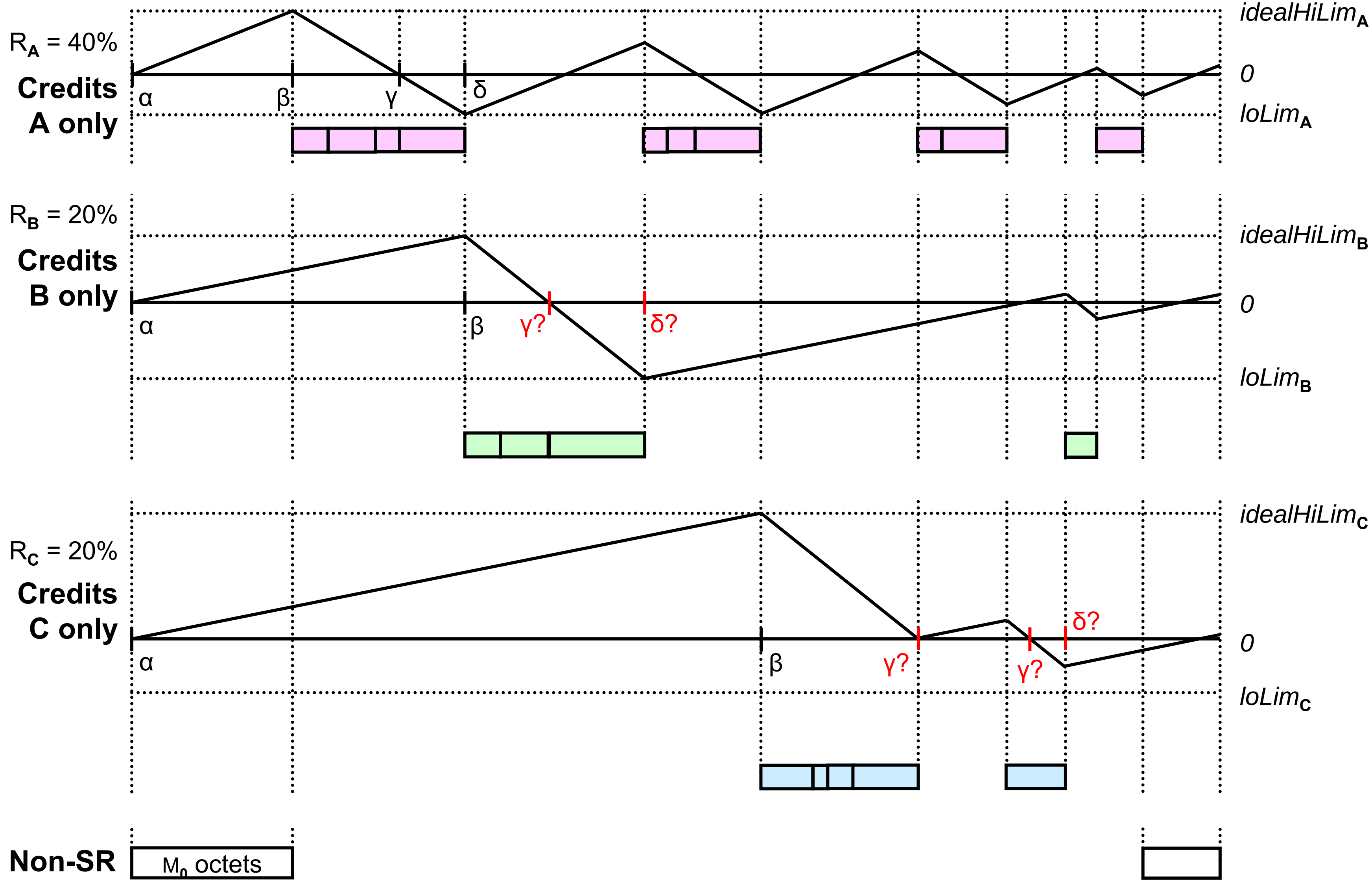


- Class B's queue latency is $T_{\alpha\delta} = T_{\alpha\beta} + T_{\beta\gamma} + T_{\gamma\delta} = M_0 / R_0 + \mathit{idealHiLim}_A / \mathit{sendSlope}_A + M_A / R_0 = M_0 / R_0 + (R_A \bullet M_0 / R_0) / (R_0 - R_A) + M_A / R_0$.
- This reduces to $T_{\alpha\delta} = M_0 / (R_0 - R_A) + M_A / R_0$.

What about Class B, C, ... ?

- In the worst case, when the non-SR frame starts transmitting (time α on the following diagram), all other classes' data arrives an instant later (by fan-in).
- The question becomes, when is the first Class X frame sent?
- A three SR Class example is on the following page.
- At time α, all SR Classes have 0 credit. (If they have any frames to transmit, they go ahead of the interfering frame; if they do not, then they are forced to 0 credit.)

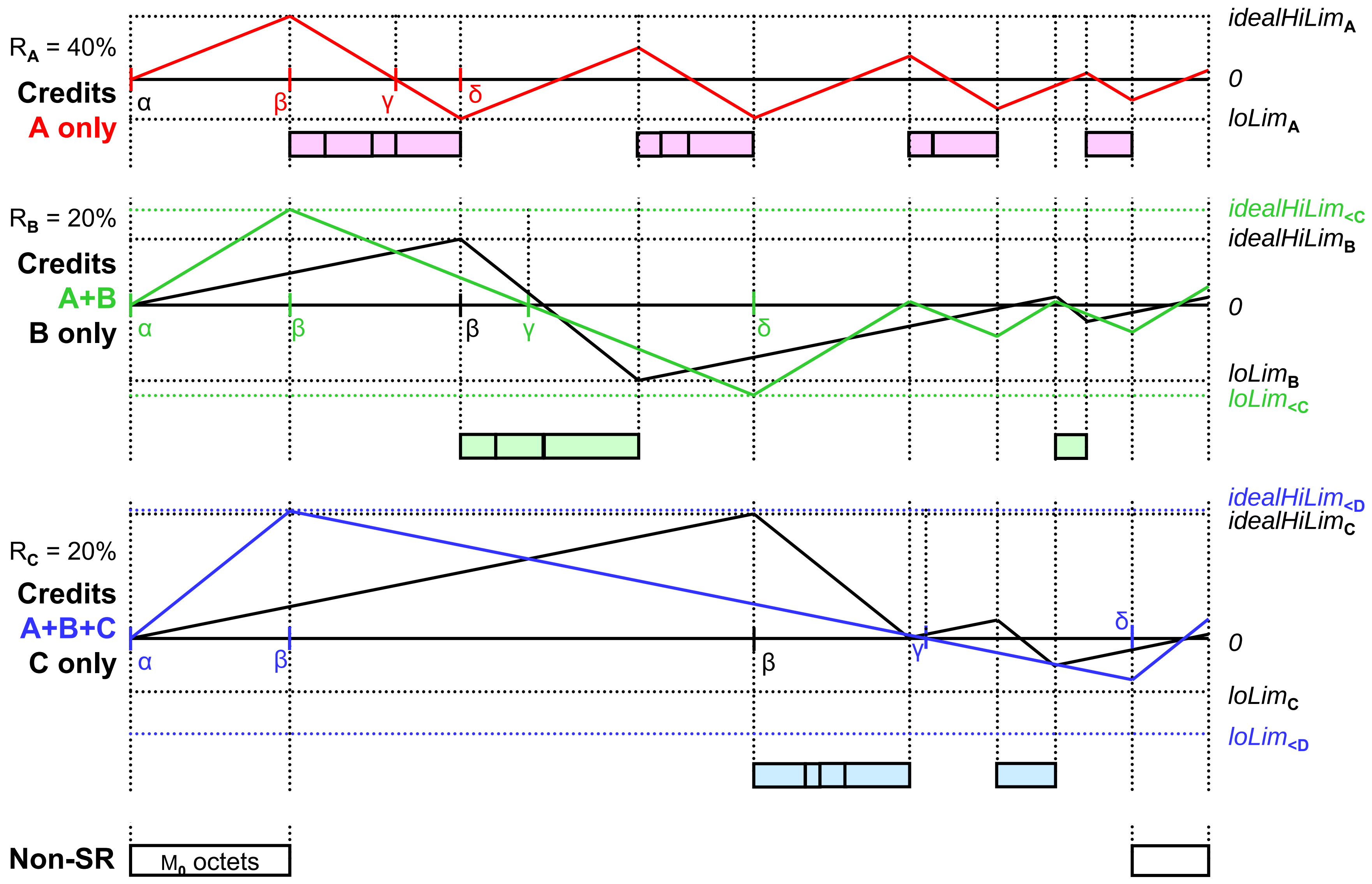
Three SR Classes



What about Class B, C, ... ?

- Calculating Class B's point β in terms of Class A is easy – Class B starts transmitting at Class A's point δ.
- Calculating Class C's point β is tougher:
  - Some combination of Class A and Class B frames are transmitted after Class A and Class B reach 0 credits;
  - The possibilities for frames transmitted from Class B's point γ to point δ is difficult to predict, and point γ is uncertain;
  - The definition of Class B's point δ is unclear; is it when Class B finishes transmitting, or when both Class A and Class B have negative credits?
- When calculating Class C's point β, the trick is to use the sum of Class A's credits plus Class B's credits.

Reference diagram



Combining Classes' credits

- By looking at Classes A, B, ... to X–1 together, as a single Class, points γ and δ are again defined.
- Let's use "<X" as a subscript for the sum of all Classes with higher priority (lower letters) than Class X.
- The credit acquisition rate $idleSlope_{<X}$ for Classes A through X–1 is the combined data rates of all Classes higher in priority than X, so:
  $idleSlope_{<X} = \Sigma_{k<X} R_k$.
- This is just the sum of the Classes' $idleSlope_k$ values.

Combining Classes' credits

- Note in the diagram, however, that the combined classes accumulate credits only until the single interfering non-SR interfering frame stops transmitting, and then the credits start decreasing linearly.
- So, the upper limit for the combined credits is not the sum of the individual Class's credits; it is the number of bits divided by the slope, or:

  $idealHiLim_{<X} = (\Sigma_{k<X} R_k) \cdot M_0 / R_0$.

Combining Classes' credits

- Similarly, $\textit{sendSlope}_{<X}$ is – ($\textit{linkTransmitRate}$ – $\textit{idleSlope}_{<X}$), so:
  $\textit{sendSlope}_{<X} = -(R_0 - \Sigma_{k<X} R_k)$.
- These combined rates hold true until some Class's buffer empties and its credits are forced to 0. In the worst-case scenarios we are examining, this does not happen.

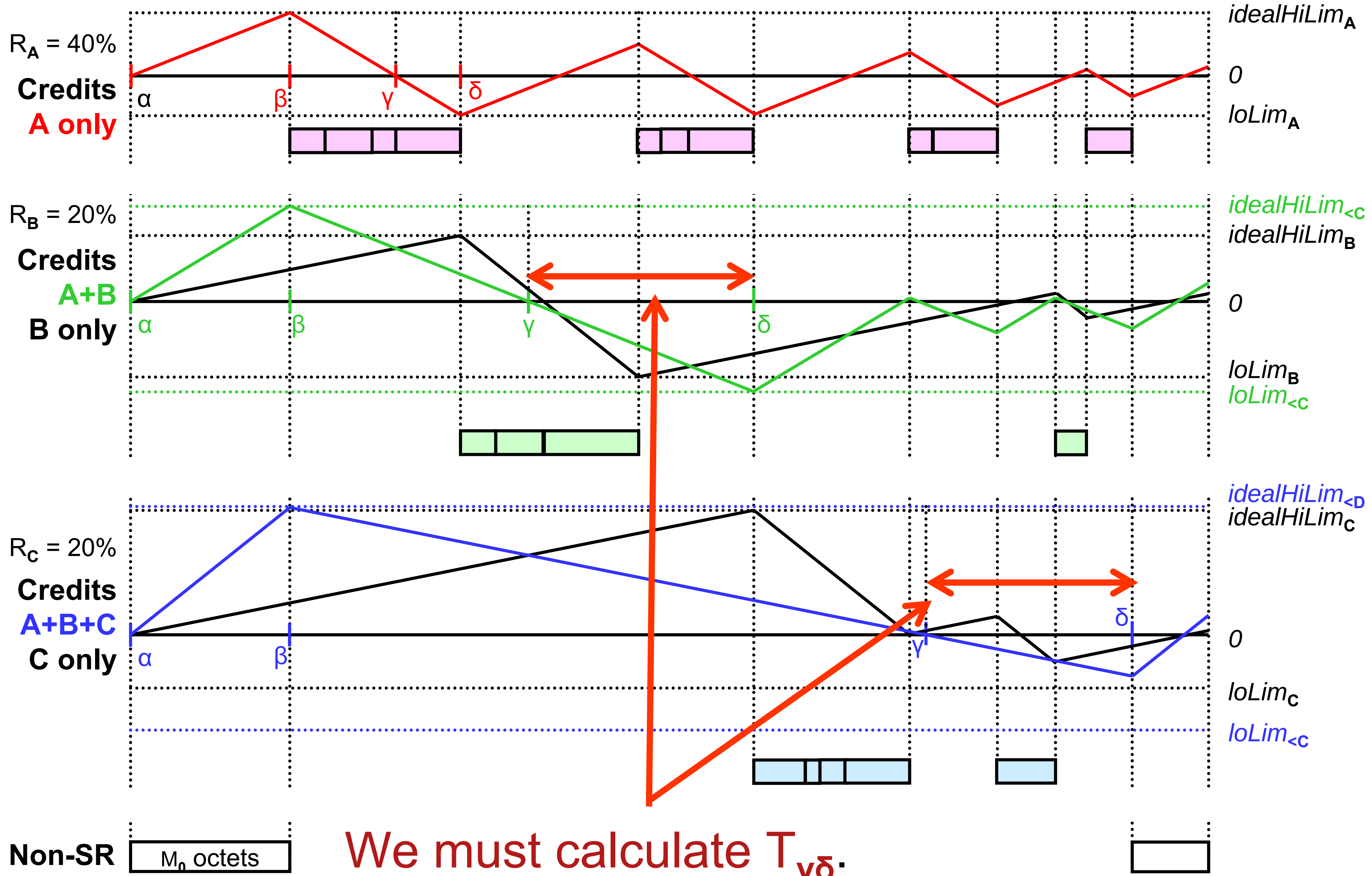
What about Class X?

- Defining $W_{<X} = -\ sendSlope_X = R_0 - \Sigma_{k<X} R_k$, we have:
  $idleSlope_{<X} = -\ W_{<X}$
  and
  $idealHiLim_{<X} = \Sigma_{k<X} R_k \bullet M_0/R_0 = (R_0 - W_{<X}) \bullet M_0 / R_0$.
- For all combined Classes, $T_{\alpha\beta}$ is the same as $T_{\alpha\beta}$ for Class A, the time for the original non-SR interfering frame to transmit. $T_{\alpha\beta} = M_0 / R_0$.
- The combined Classes A through X–1 drain from $idealHiLim_{<X}$ to 0 in time $idealHiLim_{<X} / W_{<X}$, so:
  $T_{\beta\gamma} = ((R_0 - W_{<X}) \bullet M_0 / R_0) / W_{<X}$.

What is $loLim_{<X}$?

- The total delay for Class X is $T_{\alpha\delta} = T_{\alpha\beta} + T_{\beta\gamma} + T_{\gamma\delta} = -(idealHiLim_{<X} - loLim_{<X}) / idleSlope_{<X}$. We have $idealHiLim_{<X}$ and $idleSlope_{<X}$. What is $loLim_{<X}$?
- At point γ, in the case of Class B waiting for Class A, Class A's credit reached 0. In the worst case, this happened just as a maximum-length Class A frame started transmission, leading to the credit reaching $loLim_A = (R_A - R_0) \bullet M_A / R_0$.
- For the combined Classes A through X–1, the total credit reaching 0 could happen when some Classes' credits are above 0 and some below. This makes it more difficult to determine $loLim_{<X}$, the lower limit for the combined Classes' credits.

From 0 to $loLim_{<X}$



We must calculate $T_{\gamma\delta}$.

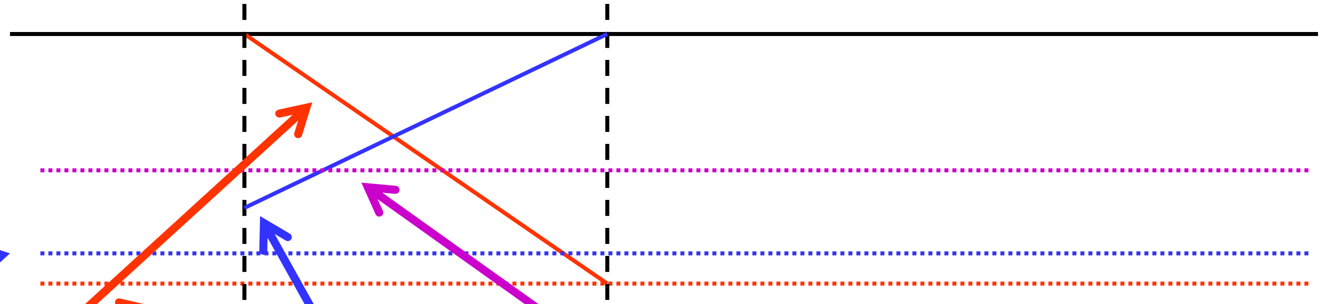
What about Class X? Computing *loLim*$_{<X}$

- If we simply take the sum: $\Sigma_{k<X}$ lolim$_{k}$, we overestimate the worst case. This is because, in order for Class Q to reach its *loLim*$_{Q}$, it must start at 0 and transmit a maximum length M_Q frame. While Class Q's frame is transmitting, all the other Classes' credits are rising, so they cannot be at *loLim* credits when Class Q finishes..
- It is also not simply the time needed to transmit one copy of each Class's max-length frame; some classes can transmit more than a single last frame after the total credits = 0, even if all Classes' credits reach 0 simultaneously.

What about Class X? Computing *loLim*$_{<X}$

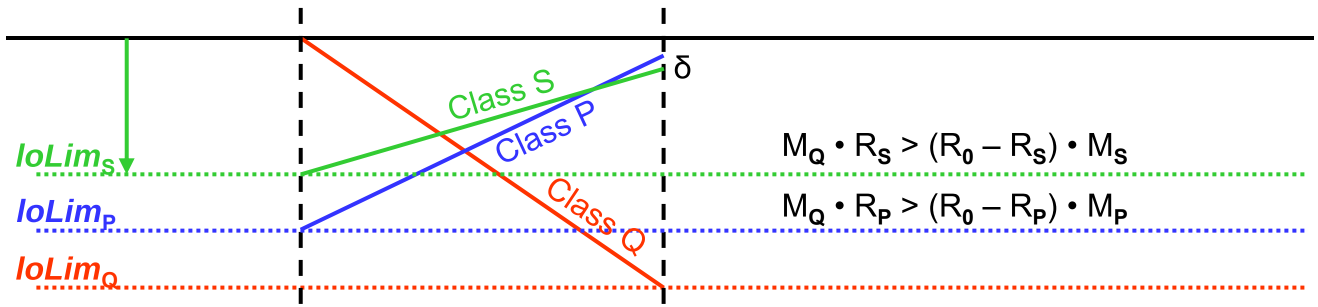
- *Some* Class must transmit the last frame.
- That last frame is a maximum length frame. If it were not, then either:
  - Extending it to the maximum length still leaves the other Classes at negative credit; or
  - Extending it to the maximum length leaves one or more other Classes with 0 or positive credit, in which case they will transmit more frames.
- Either way, if the last frame is not a maximum length frame, this is not the worst case.

What about Class X? Computing *loLim*$_{<X}$



- Class Q has a higher priority than X, and must have credits ≥ 0 to transmit its very last frame before Class X finally gets to transmit a frame. And at the point transmission of that Class Q frame, all of the other classes, e.g. Class P, must have low enough credit that they cannot climb above 0 by the end of transmission of Class Q.
- But, that required value of P's credit at the start of transmission of the Class Q frame could be more or less than *loLim*$_P$.
- Since it is impossible for a Class to drop below its *loLim*, this condition (*loLim*$_P$ > required initial value) would mean that Class Q could not transmit the last frame.

What about Class X? Computing *loLim*$_{<X}$

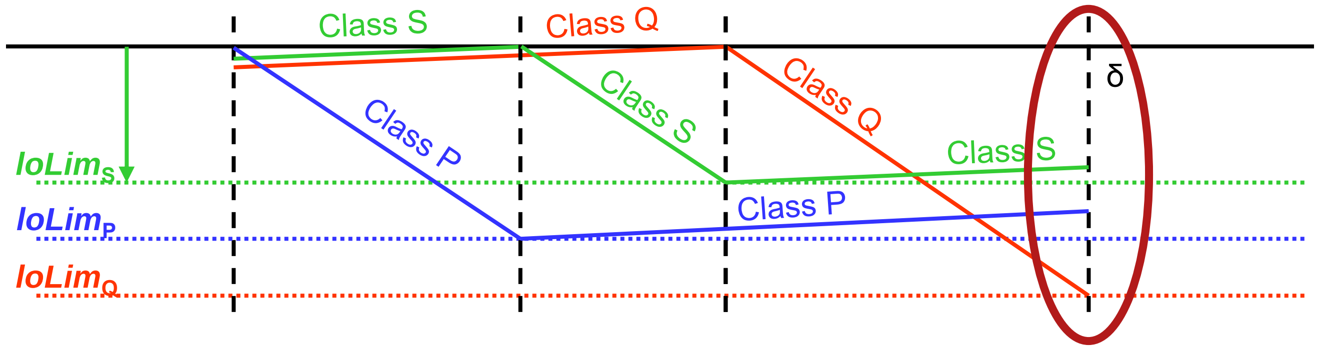


- So, the very lowest that Credit$_{<X}$ might reach is:

  $lolim_{<X} \geq lolim_Q + lolim_P + lolim_S + idleSlope_S \cdot M_Q/R_0 + idleSlope_P \cdot M_Q/R_0$

- But even this is pessimistic, because it assumes that both P and Q are at their respective *lolim* values at the same time, when only one can be at its *lolim*.

What about Class X? Computing *loLim*$_{<X}$



- But, having understood this, we can ask what happens if:

  1. Class Q is the lowest-priority class with higher priority than Class X (i.e., X = Q+1); and

  2. Every Class k (including Class Q) has reserved an infinitesimal fraction of the LAN bandwidth (i.e., $R_k << R_0$).

- Then, we can get the total *lolim*$_{<X}$ as close as we wish to $\Sigma_{k<X}$*lolim*$_k$!!

What about Class X? Computing $\mathit{loLim}_{<X}$

- So:
  $\mathit{loLim}_{<X} = \Sigma_{k<X}\ \mathit{loLim}_k = \Sigma_{k<X}\ (R_k - R_0) \bullet M_k / R_0$
- But, in this worst case, $R_k << R_0$, so:
  $\mathit{loLim}_{<X} = - \Sigma_{k<X} M_k$

Queuing delay to first Class X frame

- We how have all the pieces to compute the queuing delay to the first Class X frame:

- $W_{<X} = - R_0 - \Sigma_{k<X} R_k$
  $sendSlope_{<X} = - W_{<X}$
  $idealHiLim_{<X} = (R_0 - W_{<X}) \bullet M_0 / R_0$
  $loLim_{<X} = - \Sigma_{k<X} M_k$
  $qDelay_X = M_0 / R_0 + (idealHiLim_{<X} - loLim_{<X}) / sendSlope_{<X}$
  $= M_0 / R_0 + ((R_0 - W_{<X}) \bullet M_0 / R_0 + \Sigma_{k<X} M_k) / W_{<X}$

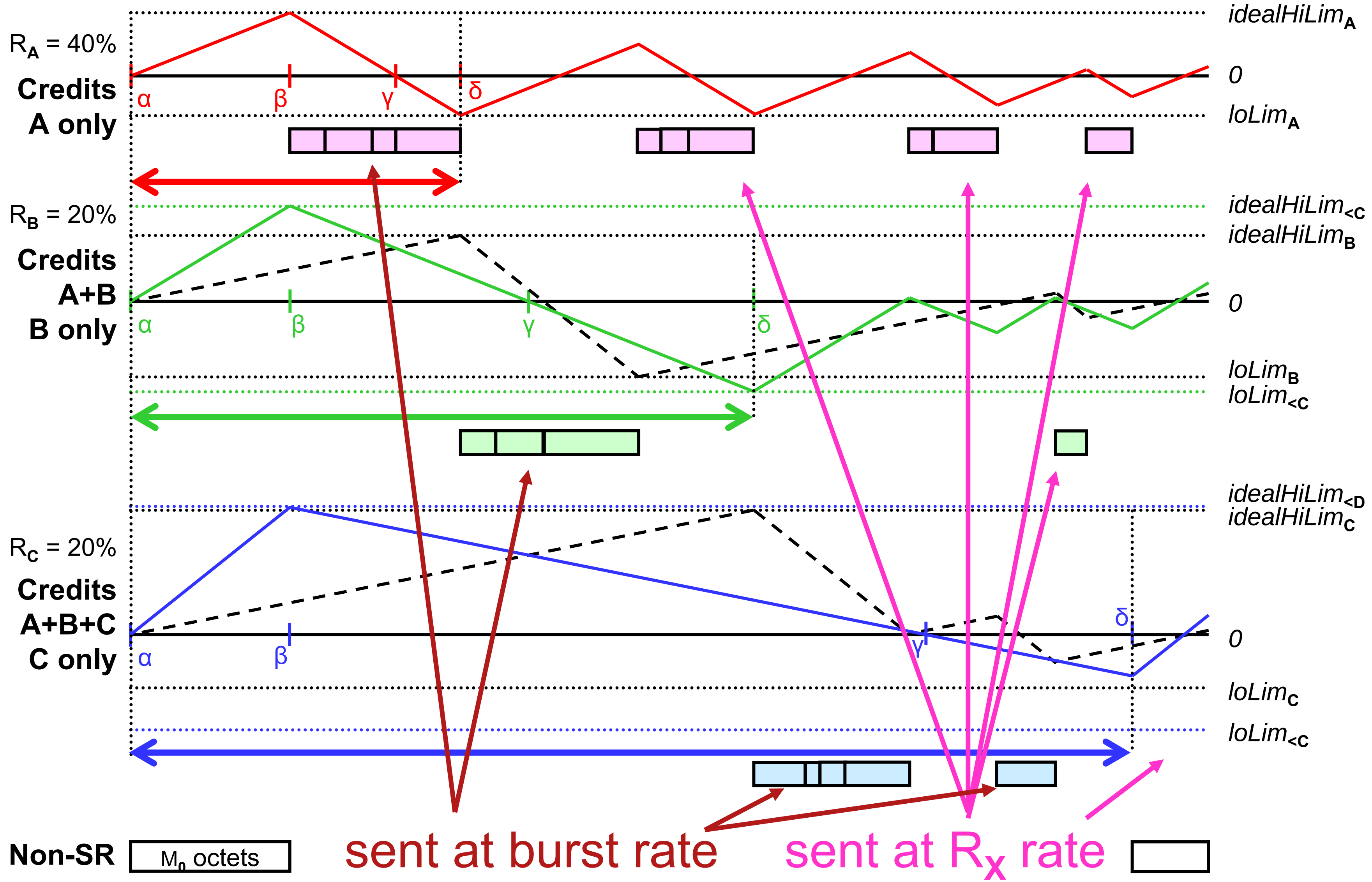
- $qDelay_X = (M_0 + \Sigma_{k<X} M_k) / W_{<X}$

Max size burst

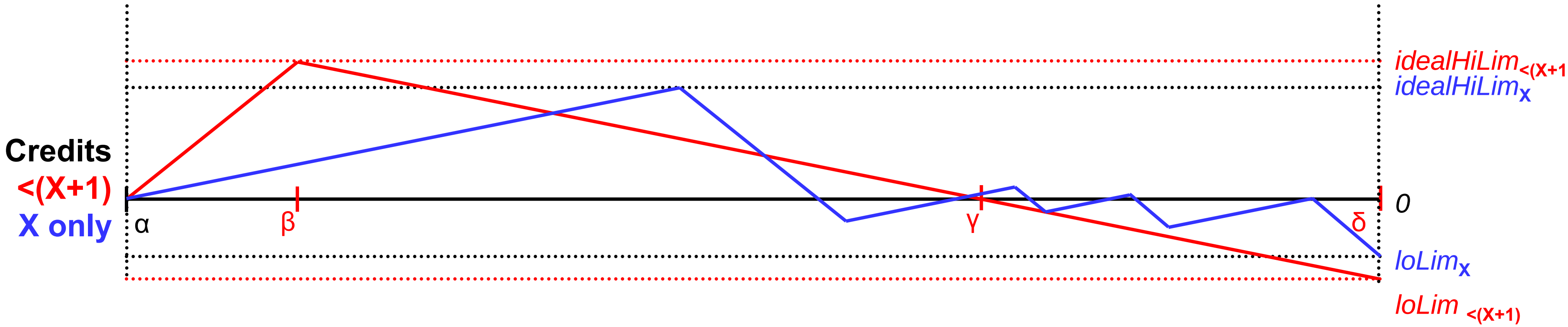
Max size burst for Class X

- The maximum sized burst that can be generated for Class X is defined as the number of bits that can be transmitted at a higher rate than the normal, reserved rate, for Class X, R_X.
- But, over the long term, the Class X frames are arriving at is never higher than R_X.
- We need to know what times and what data rate to use to calculate the max size burst.

Max size burst for Class X

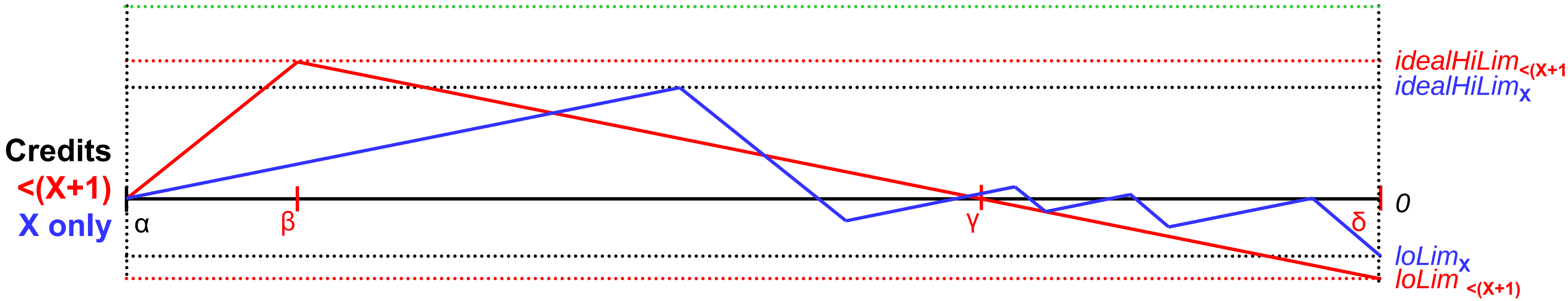


Max size burst for Class X



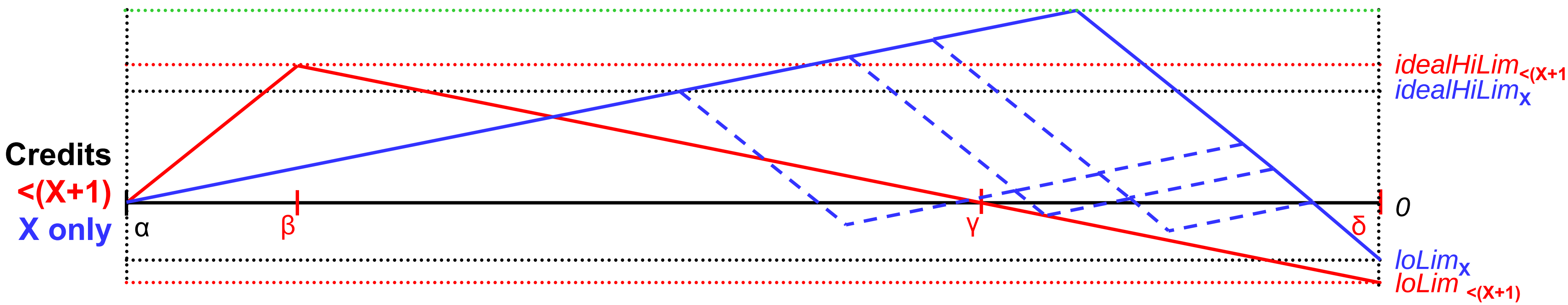
- Class X's Credit diagram during the worst-case delay for Class X+1 (e.g., Class X = Class C, Class X+1 = Class D).
- Sum of credits of Classes A through X in red.
- Class X's credits in solid blue.

Max size burst for Class X



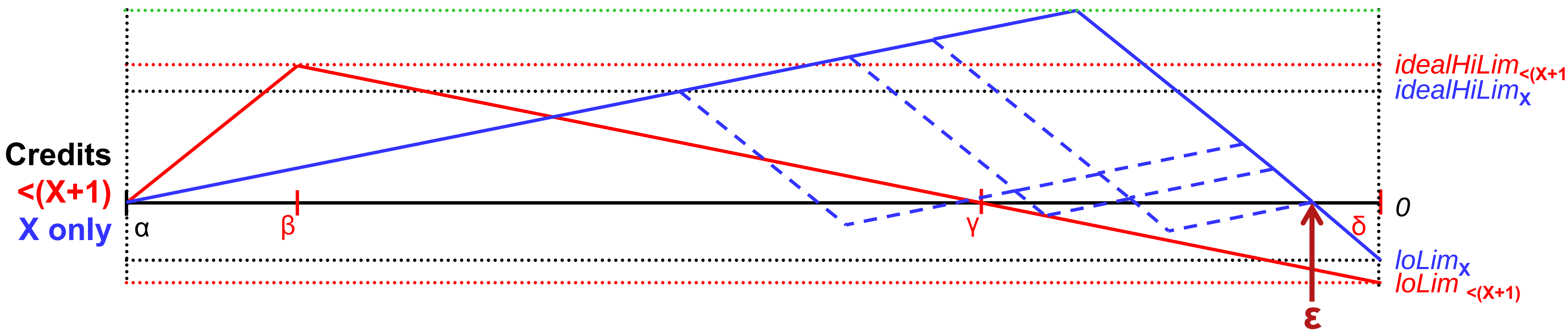
- In the worst case for $T_{\alpha\delta}$, it takes Classes A through X all the way from point α to point δ for their combined Credits to reach *loLim*$_{<(X+1)}$.
- Similarly, in the worst case for Class X, it reaches *loLim*$_X$ at the same moment (point δ).
- Almost all of the bandwidth is allocated to Class X; only a tiny bit is allocated to Classes < X, in order to trigger the worst case $T_{\alpha\delta}$.

Max size burst for Class X



- This complex shape for Class X is equivalent to simply acquiring all of the Credits first, and discharging them, afterward.

Max size burst for Class X



- We know the total time, $T_{\alpha\delta} = (M_0 + \Sigma_{k=AtoX}M_k) / W_X$.
- We know the time from point ε to point δ, $T_{\varepsilon\delta} = M_k / R_0$.
- We know that, from point α to point ε, Class X is transmitting at an average rate of almost the speed, $\Sigma_{k=AtoX}R_k$, $= R_0 - W_X$, because we have assumed for this worst case that Class X has been allocated almost all of the lower Classes' bandwidth.

Max size burst for Class X

- So, the total number of Class X bits output in the worst case Class X burst is:

$$\begin{aligned} maxBurst_X &= T_{\alpha\varepsilon} \bullet (R_0 - W_X) + M_X \\ &= (T_{\alpha\delta} - T_{\varepsilon\delta}) \bullet (R_0 - W_X) + M_X \\ &= ((M_0 + \Sigma_{k=AtoX} M_k)/W_X - M_X/R_0) \bullet (R_0 - W_X) + M_X \end{aligned}$$

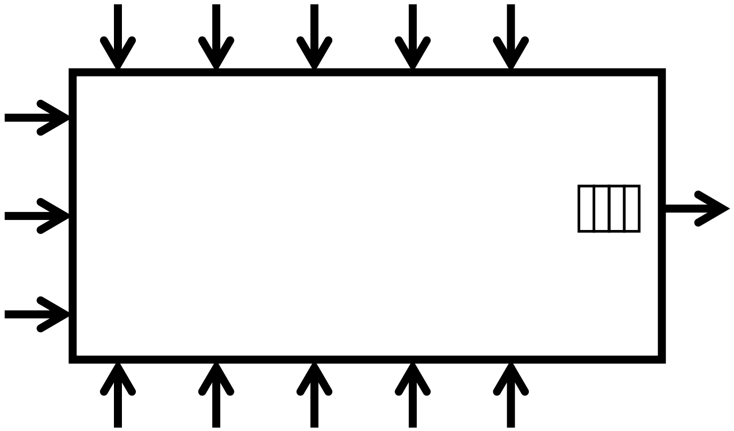
- This reduces to:

$$maxBurst_X = (M_0 + \Sigma_{k=AtoX} M_k) \bullet (R_0/W_X - 1) + M_X \bullet W_X/R_0$$

- Note that this value is highest when W_X is small, and thus $R_0/W_X - 1$ is large. The second term is small, in this case. This corresponds to the case where data rates are highest, and the least bandwidth is left over for lower-priority data.

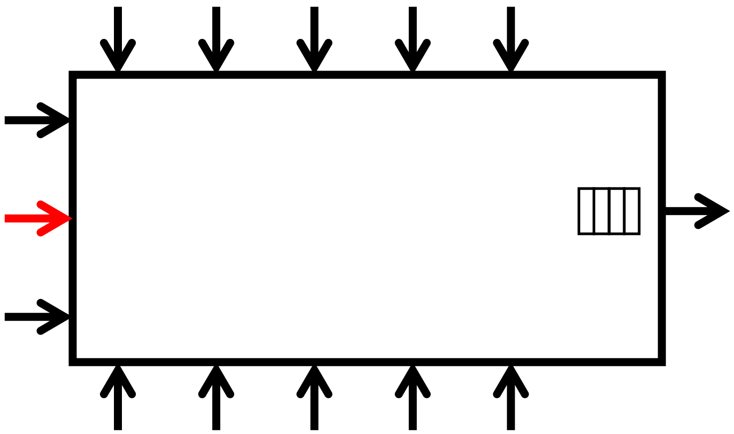
Fan-in delay

Fan-in burst



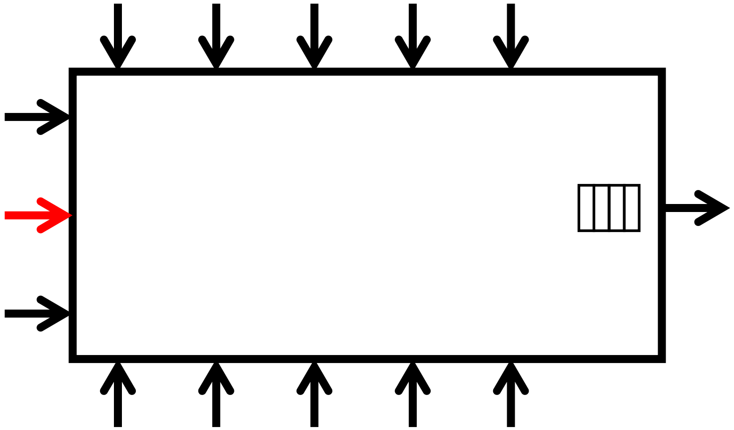
- Imagine that every input Port encounters a maximum interference event for Class X at the same time. The output Port will be starved for data.
- Then, the input Ports all receive a frame at the same moment; all are delivered to the output queue.

Fan-in burst



- Assuming that all input ports are configured the same, and all are configured the same as the output port, we can observe that not all input ports can deliver a max size burst from the transmitting device; if they did, their combined data rates would be (in the above example) 13 times the data rate of the output port, and this is not allowed by the reservation protocol.

Fan-in burst



- In fact, in the case of identical input ports, we can maximize the amount of fan-in data by having:

  One port (in red) has almost the entire bandwidth reserved for Class X reserved for active streams, so delivers almost $maxBurst_X$ bits of burst fan-in data.

  All of the other ports have only a tiny bit of bandwidth, and deliver a single frame of max length M_X.

- In this example, the result is $maxBurst_X + 12 \cdot M_X$ bits of fan-in data.

Fan-in burst size computation

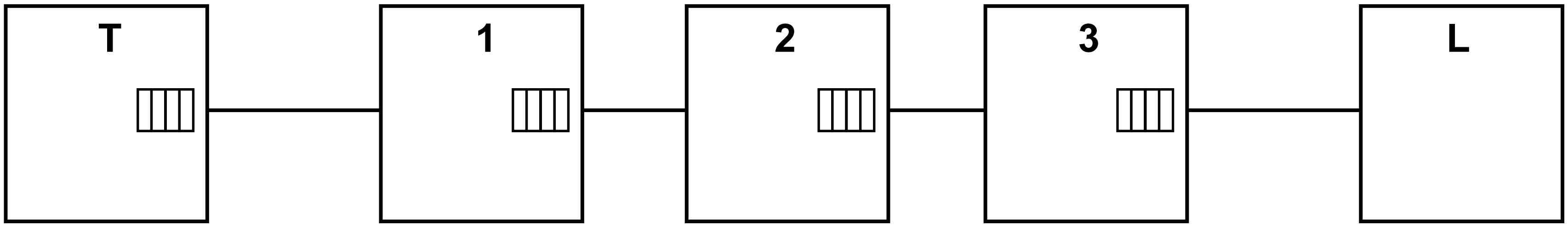
- We can give a procedure, not a formula, for determining the maximum amount of fan-in data for a given Class and output port:
  1. Determine B_O, the maximum possible bandwidth for Class X on the output port.
  2. For each possible input port i, determine the maximum possible bandwidth B_i for Class X. Let us assume that this information is obtained via LLDP from the transmitting side.
  3. For each input port i, calculate $maxBurst_{X,i}$ using $max(B_O, B_i)$ for the bandwidth. (In the maxBurst formula, use $W_X = R_0 – max(B_O, B_i)$.)
  4. Add $max(maxBurst_{X,i})$ to the total fan-in data.
  5. Set $B_O = B_O – B_i$ and repeat from step 4 until $B_O = 0$.
  6. For each remaining port, add $M_{X,i}$ to the total fan-in data.

Fan-in delay

- The fan-in delay is equal to the total fan-in burst data computed from the previous slide, output at the line rate of the output port.

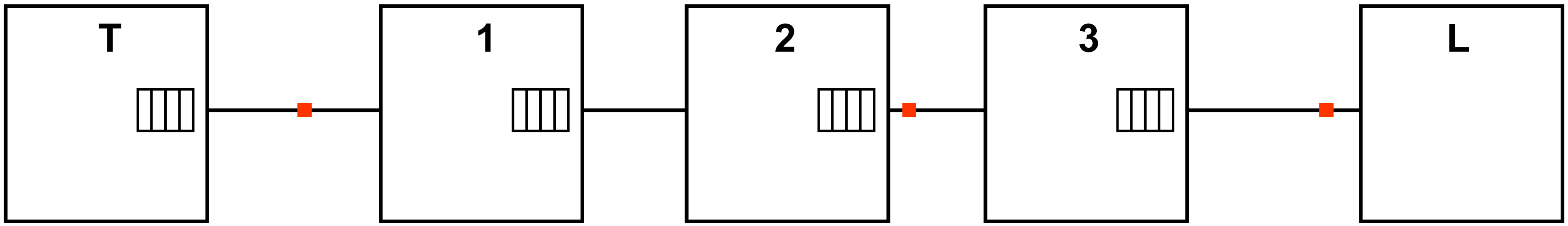
Permanent delay

“Permanent” delay



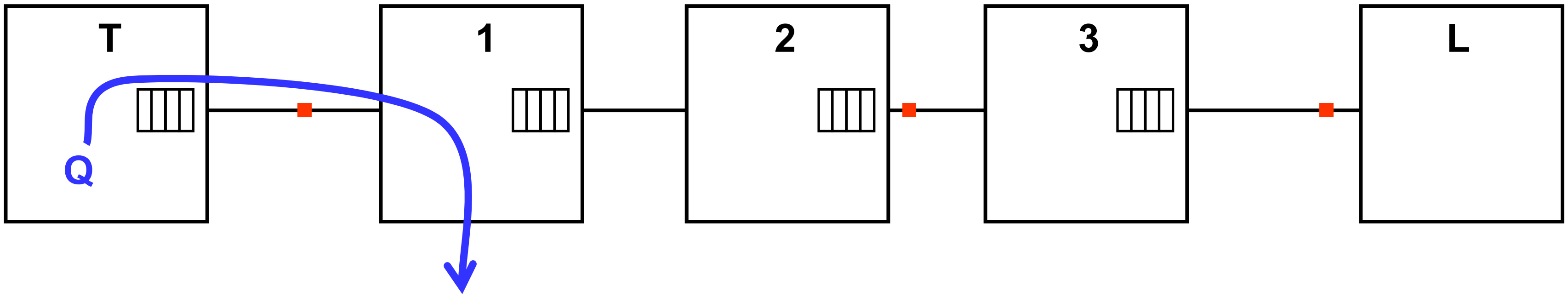
- Talker T reserves the highest possible bandwidth for a Class X stream to Listener L through Bridges 1, 2, 3.
- That is, the bandwidth registered $B = R_X$.

Building up buffer occupancy



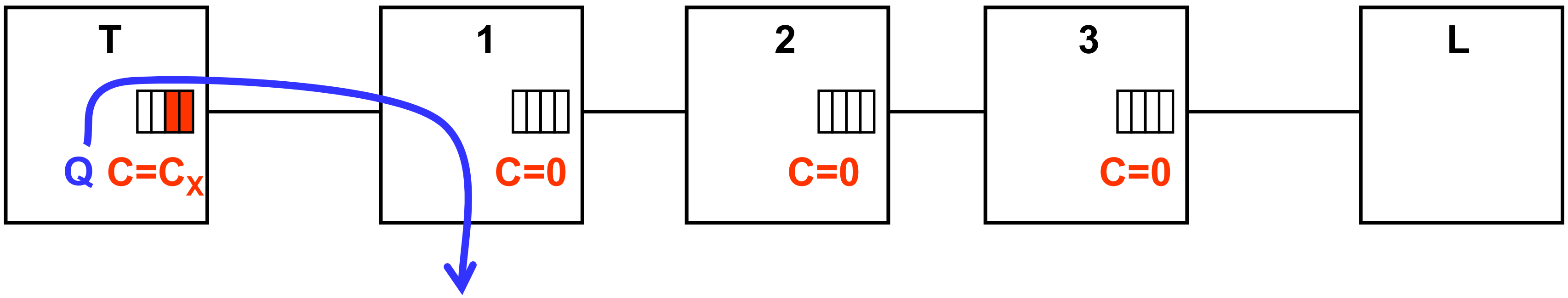
- Talker T starts transmitting a regular stream at the maximum rate.
- Let us assume there is no interfering traffic, very low minimum delay, and very short links.
- Then a typical case for frames is shown, above.

Building up buffer occupancy



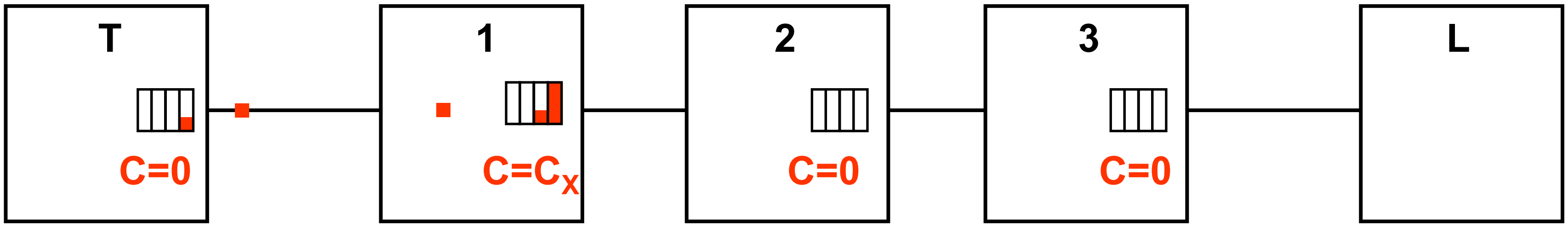
- Now, let us suppose that a second stream, Q, generates the maximum possible interference to Class X.

Building up buffer occupancy



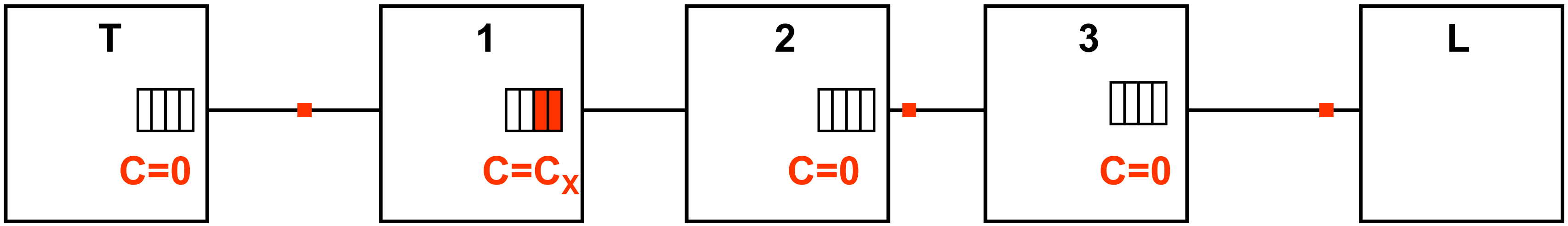
- Then, Talker T's queue fills up to the worst-case queue-delay value, C_X, as shown in the previous slides.
- Bridge 1 – 3's queues, on the other hand, are starved; they have no frames buffered, but are not building up credits, for exactly that reason.
- Credits shown as "C=n".

Building up buffer occupancy



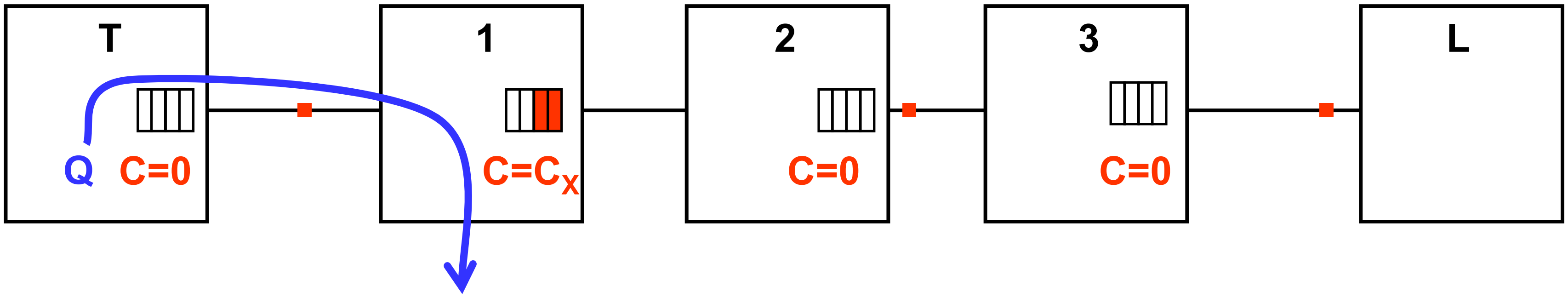
- Let's suppose that the interfering traffic stops just as Talker T is able to dump its Class X traffic.
- Bridge 1 receives the maximum burst, at line rate.
- But Bridge 1 has to buffer most of these frames, because it had no credits.

Building up buffer occupancy



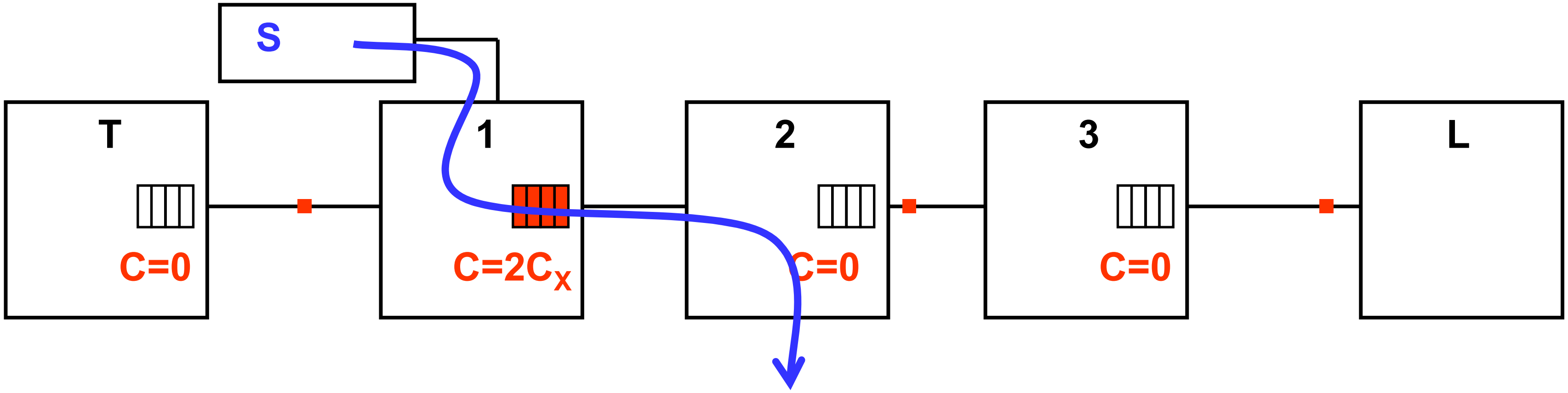
- The new steady state is just as before, with all Bridges' credits hovering around 0, except that Bridge 1 has C_X credit's worth of buffer filled up.

Building up buffer occupancy



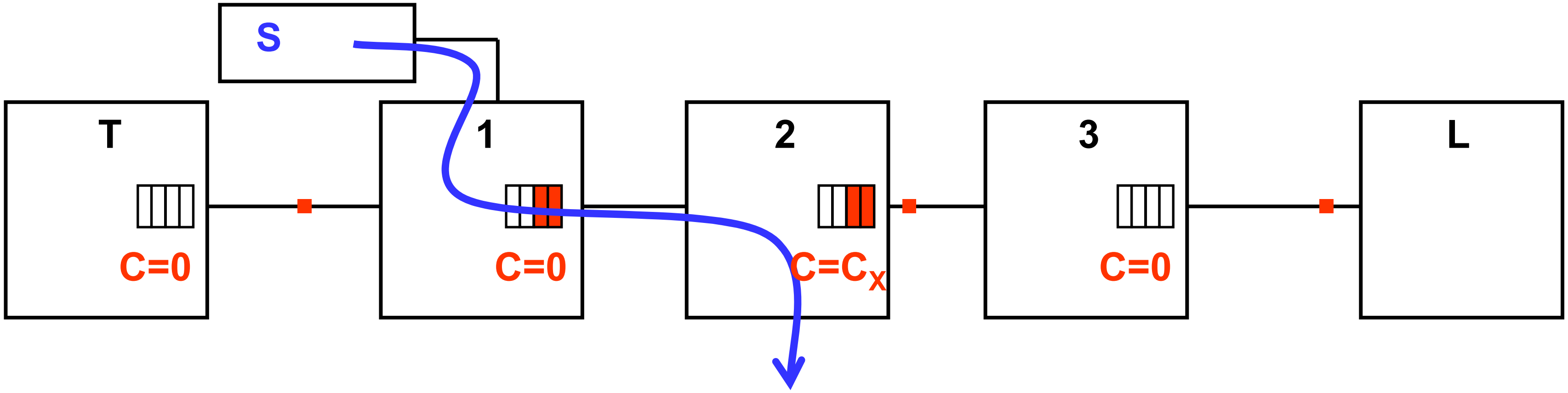
- Now, if stream Q starts up, again, there is no problem.
  - Talker T's queue fills up, again, but this time, Bridge 1 has data to send.
  - So, when Talker T dumps its queue, it will only restore Bridge 1 to its (full) steady state.

Building up buffer occupancy



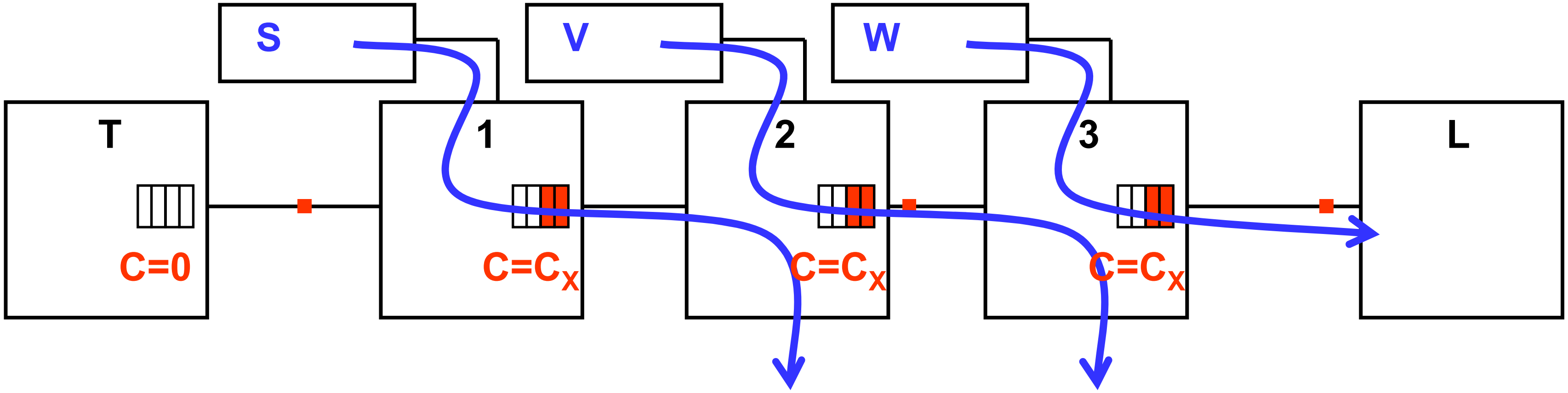
- But, if a new source S fires up, it can cause further congestion in Bridge 1.
- For a moment, Bridge 1's queue fills up even further.

Building up buffer occupancy



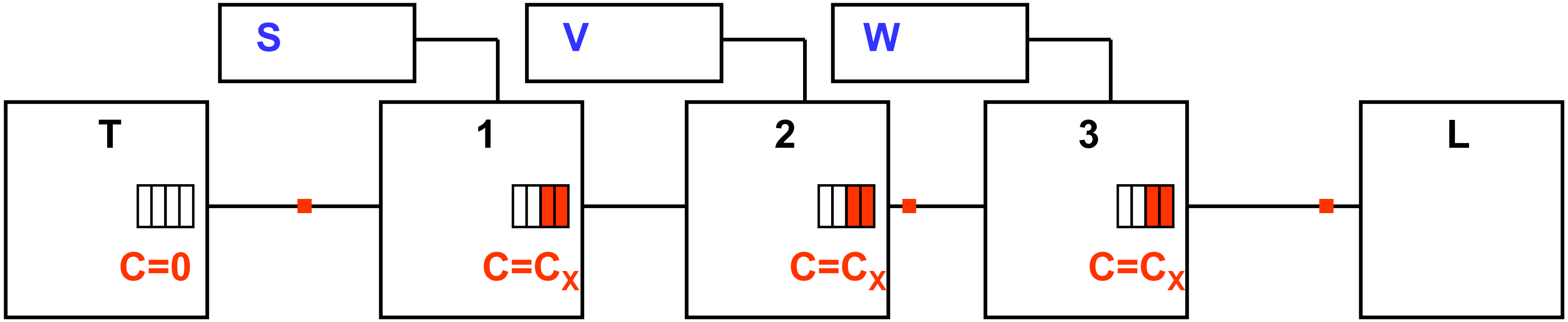
- When the steady state is reached, we have two Bridges with permanently partially-full queues.

Building up buffer occupancy



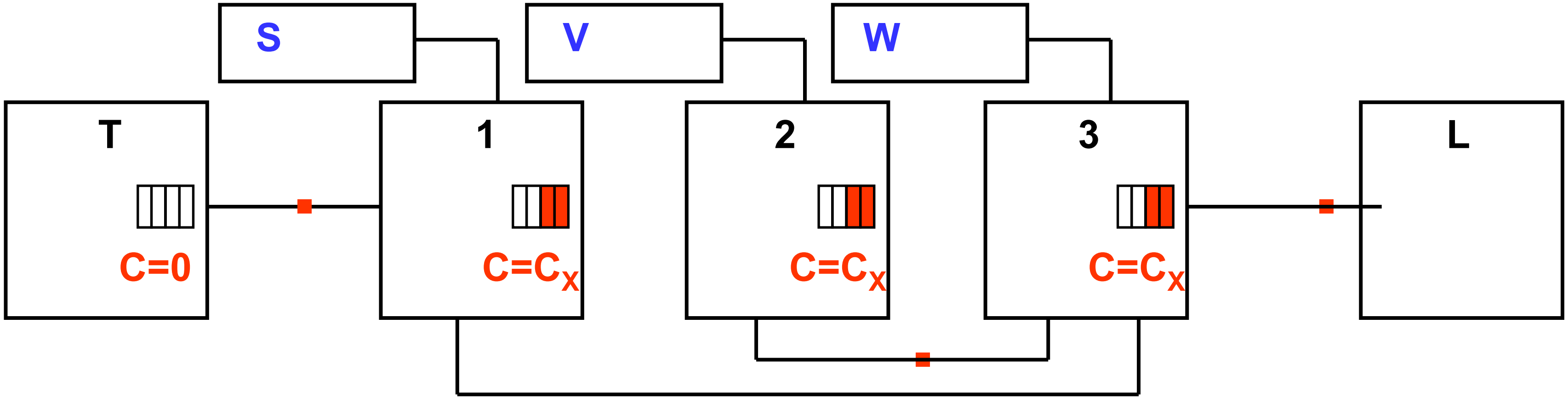
- Which leads us, of course, to the state where all Bridges' queues are partially full.
- The problem does not get any worse, because no Bridge ever starves. Starvation is required before a buffer can get permanent partial occupancy.

Building up buffer occupancy



- It may, therefore, be desirable for all Bridges to reserve a little bit more bandwidth than the actual total reserved by the Talkers, so that this “permanent” buffer build-up can drain away.
- However, the problem will not get significantly better on the time scales of the output queuing delay, so does not factor into those calculations.
- And, of course, this additional capacity increases the delay at each hop.

Building up buffer occupancy



- Since this kind of event could happen on multiple input ports at the same time, the “permanent” data equals the worst-case fan-in data.

Maximum interference delay and maximum buffer requirement

Maximum interference delay

- The worst-case interference delay for Class X on a given Port is the sum of three parts:
  1. From queuing delay: $qDelay_X = (M_0 + \Sigma_{k<X}M_k) / W_{<X}$
  2. The fan-in delay computed from the slide, "Fan-in burst size computation" times the data rate for Class X, $(R_0 – W_X)$.
  3. The "permanent" buffer contents contribution, which is equal to the fan-in contribution in step 2.

Maximum per-Class buffer requirement

- Buffer requirement consists of three parts:
  1. A contribution from the queuing delay, equal to $maxBurst_X$.
  2. The fan-in burst size computed from the slide, "Fan-in burst size computation".
  3. The "permanent" buffer contents contribution, which is equal to the fan-in contribution in step 2.
- Fortunately, 2 and 3 cannot both happen! The first time the fan-in burst happens, it becomes the permanent burst. If the fan-in burst happens again, it can only happen after an equal-sized pause in the data, which drains the permanent burst, then refills it.
- So, the worst-case per-Class buffer requirement for Class X is the sum of 1 and 2, above.

Maximum total buffer requirement

- The worst-case total buffer requirement for the SR priority levels on a given Port is the sum of two parts:
  1. The worst-case buffer requirement for the lowest-priority SR level enabled on that Port.
  2. One max-size frame $M_{i,z}$ for each Class z of higher priority than Class X for each input port i.
- That is, the SR priority levels can share space, since the worst case buffer requirement for Class X is when it is taking almost all of the bandwidth from the higher-priority Classes. The only additional requirement is for the fan-in from higher priorities on other Ports.
- Thus, the SR priority levels will do well to share their buffer space on each port.

Parameterization

Per-Port Per-Class Parameters

- All of the above results for a given output Port and Class X can be computed from six values:
  - R_0 The LAN data rate for the output port
  - M_0 The maximum non-SR frame size for the output port, from the start of the frame to the start of the next frame.
  - M_X The maximum Class X frame size for the output port, start to start.
  - W_X The data rate reserved for all SR Classes of lower priority than Class X, plus all non-SR classes, on the output port.
  - $maxBurst_{i,X}$ The max size burst for Class X from input port i.
  - $M_{i,X}$ The maximum Class X frame size, start to start, for Class X from input port i.

Per-Port Parameters

- In some environments, e.g. in an enterprise with 500-port Bridges, but only one SRP Class, the fan-in component can contribute much more to buffers size and delay than the burst component.
- Also, some Ports can have different buffer capacities, relative to their speeds.
- It would therefore be useful to define a maximum fan-in F_P for each Port P, that can be less than the physical fan-in. The fan-in limitation could cause a reservation to be rejected (or rescinded) because the number of Talker Ports sending traffic to some Listener Port P would exceed the allowable F_P for that Port.

Passing parameters in LLDP

- $M_{i,X}$ and $maxBurst_{i,X}$ are parameters that are governed by the transmitter's parameters R_0, M_0, M_X, and W_X (and F_P, if added), to the input port, not those of an output port, on the Bridge where they are needed.
- Therefore, M_X and $maxBurst_X$ for every SR Class X need to be transmitted, either in MSRP or in LLDP, and the received values used as $M_{i,X}$ and $maxBurst_{i,X}$ in the receiving Bridge.

Next steps

Next steps

1. Discuss, correct, validate, rewrite, or discard this presentation.
2. Re-examine the assumptions list in light of the results.