Contents

[Credit Based Shaper (CBS) algorithm: 1](#_Toc166507204)

[1. Credit value 1](#_Toc166507205)

[2. Forwarding rule in CBS algorithm: - 1](#_Toc166507206)

# Credit Based Shaper (CBS) algorithm:

## Credit value

The credit value is part of the credit-based sharper algorithm used by the MTL Queue Scheduler.

The credit value is accumulated every transmit clock cycle, that is, 40 ns for 100 Mbps and 8 ns for 1000

Mbps. The credit to be added or subtracted per cycle can be fractional based on the required idleSlope and sendSlope values.

idleSlope:

The rate of change of credit [bps]. when the value of credit is increasing. (i.e., while transmitting is FALSE).

sendSlope:

The rate of change of credit, in bits per second, when the value of credit is decreasing (i.e., while transmitting is TRUE.

hiCredit:

The maximum value that can be accumulated in the credit parameter.

loCredit:

The minimum value that can be accumulated in the credit parameter.

## Forwarding rule in CBS algorithm: -

[ Reference: - chrome-extension://oemmndcbldboiebfnladdacbdfmadadm/https://www.duo.uio.no/bitstream/handle/10852/87168/5/Analysis\_of\_bandwidth\_reservation\_strategy\_for\_AVB\_classes\_in\_Ethernet\_TSN\_switch.pdf ]

1. If the transmission line is free, the scheduler transmits a frame of the highest priority class that satisfies the conditions: a) its queue is not empty; and b) it has a non-negative credit.

2. The credit of an AVB class is reduced linearly with rate send slope when the class transmits.

3. The credit of an AVB class increases linearly with rate idle slope when the following conditions hold simultaneously for that class: a) its queue is not empty; and b) other AVB or BE classes are transmitting.

4. Whenever an AVB class has a positive credit and its queue becomes empty, the credit is set to zero; this is called a credit reset.

5. If the credit is negative and the queue becomes empty, the credit increases with rate idle slope until the zero value.

## Credit-Based Shaper Calculations

The credit-based shaper algorithm has a single externally determined parameter, idleSlope, that determines the maximum fraction of the portTransmitRate that is available to the queue associated with a traffic class (bandwidthFraction), as shown in equation below: -

bandwidthFraction = idleSlope / portTransmitRate

As an example, this means that if I have a portTransmitRate of 100 Mb/s and I have assigned

SR class A to queue 7 and SR class B to queue 6, then I can assign queue 7 (class A) a

bandwidth of 50 Mb/s and class B (queue 6) a bandwidth of 25 Mb/s by assigning queue 7 a

bandwidthFraction of 0.5 and queue 6 a bandwidthFraction 0.25.

This also means that the idleSlope for class A (queue 7) is now 50 Mb/s, and the idleSlope for class B is 25 Mb/s.

idleSlope is “the rate of change of credit, in bits per second, when the value of credit is

increasing.”

sendSlope is “the rate of change of credit, in bits per second, when the value of credit is

decreasing.” And is given by:

sendSlope = (idleSlope – portTransmitRate)

A diagram of a slope

Description automatically generated

There is also an upper limit for the credit value, hiCredit, and a lower value, loCredit, as

shown in figure. Two equations for calculating these values, as shown below:

loCredit = maxFrameSize x (sendSlope/portTransmitRate)

hiCredit = maxInterferenceSize x (idleSlope/portTransmitRate)

hiCredit: -

Calculating the hiCredit value is straightforward as the maximum interference size

(maxInterferenceSize) is simply the size of one maximum sized Ethernet frame, which is

2000 octets, meaning 2000 x 8 bits = 16000 bits. In the case of a 100 Mb/s Ethernet

connection and a traffic class (queue) with 50% bandwidth reservation, hiCredit will then be

50% of 2000 octets: 16000 bits x (50 Mb/s /100 Mb/s) = 16000 x 0.5 = 8000 bits.

hiCredit = 8000 bits.

loCredit: -

In order to calculate loCredit I need to find the maxFrameSize, which is the maximum

number of bits that can be transmitted in a frame for a stream (traffic class). This value is

defined through the SRP protocol, which uses a traffic specification (TSpec) for each stream.

In addition to defining the maximum number of bits per frame, the TSpec also defines a

maximum number of frames which can be transmitted (maxIntervalFrames). Both the

maxFrameSize and the maxIntervalFrames are measured over a class measurement interval

which applies at the source (talker) of the stream. For class A this class measurement interval

is 125 μs and for class B it is 250 μs.

So, going back to the example above with a link speed of 100 Mb/s and a traffic class with

50% bandwidth reservation. If I am using traffic class A, then maxFrameSize will be 50% of

the maximum number of bits I can transmit during 125 μs over a 100 Mb/s link. Which is:

100 000 000 b/s x 0,000125 s x 0.5 = 6250 bits. However, this number needs to be an integral number of octets, as 6250 is not, the number is rounded down to 6248.

Having found the maxFrameSize I now need to find the sendSlope. Knowing the idleSlope is

equal to the reserved bandwidth of the class (queue), 50 Mb/s, then sendSlope = 50 Mb/s -

100 Mb/s = - 50 Mb/s, which is the rate of change of credit per second when transmitting a

frame across the output port.

So, we have

maxFrameSize = 6248 bits

sendSlope = (50 Mb/s - 100 Mb/s) = - 50 Mb/s

As we know that loCredit = maxFrameSize x (sendSlope/portTransmitRate)

So, loCredit = 6248 \* (-50/100) = -3124 bits

= > Using the example above, if the queue assigned to class A forwards a frame of size 842 B

across the output port of the switch, then the credit cost of this will be:

time it takes to transmit the frame across the output port of the switch x sendSlope.

The time it takes to transmit a 842 B frame over a 100 Mb/s link is:

842\*8 bit / 100 000 000 b/s = 0,00006736 s.

The cost will then be 0,00006736 s x -50 000 000 b/s = -3368 bit. If we assume the credit was

at zero before the frame forward, then afterwards it will be -3368. It will then take

3368/idleSlope seconds before the credit returns to zero.