Calculating the Delay Added by Qav Stream Queue

This paper is a result of discussions at the July 2009 plenary meeting. At that meeting, we were trying to work out a formula for the worst-case delay that a bridge needs to add to the Talker Advertise message. We came up with a scenario that seemed to us to be the worst case, but it has not been proved so. It assumes 802.3 media, not enough has been decided about how 802.11 media will handle streams to allow those formulas to be derived. In addition, this work has only been done for class A traffic and further work is needed to extend it to class B traffic.

Within 802.3 media there is a difference in how the calculations are performed due to the fact that at 1 Gb and higher speeds the maximum interference on the media has a duration that is less than 25% of the 125 µs measurement interval for class A (75% is the default maximum bandwidth that can be reserved for streams).

Fast Ethernet (100 Mb) Media

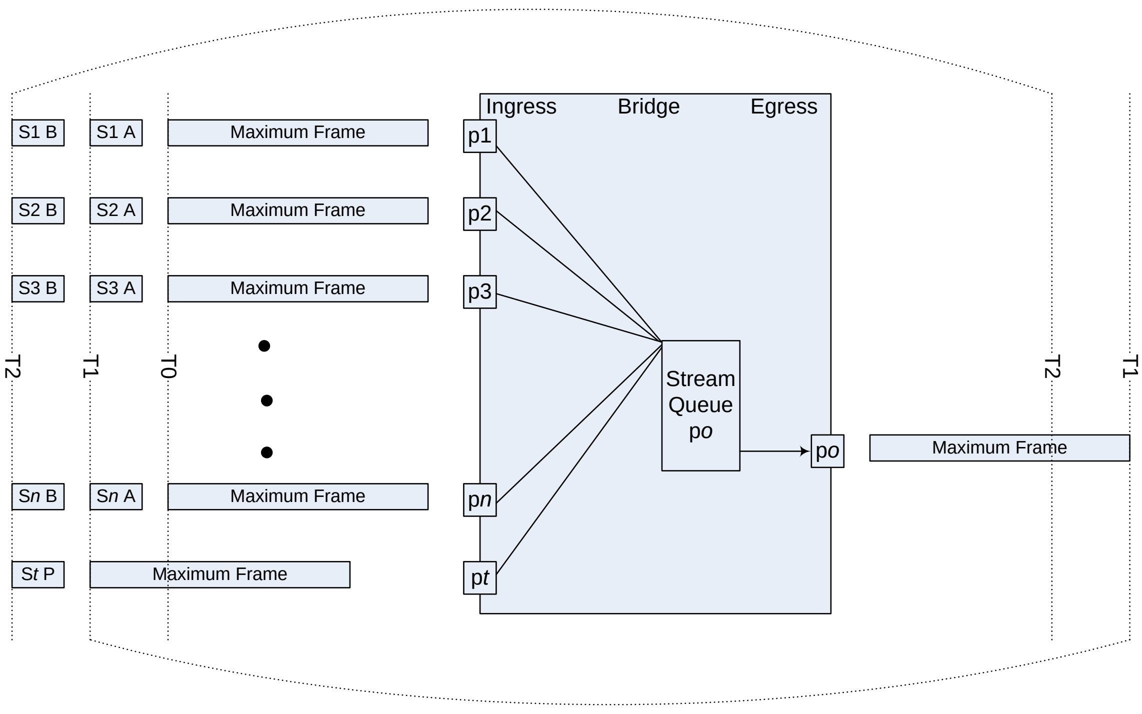

Figure 1

Please refer to Figure 1. We want to calculate the delay for a target stream packet (S*t* P) received on the target port (p*t*) through the Stream Queue and onto the media for the output port (p*o*). Other streams, stream 1 (S1) on port 1 (p1) through stream *n* (S*n*) on port *n* (p*n*), are also mapped to the output port (p*o*). For the worst case, we assume that the default maximum bandwidth is reserved for Class A (75%) on port p*o*.

At time T0, the input ports p1 through p*n* each receive the last octet of a maximum length frame (1500 application octets plus overhead added by all the protocols that the bridge supports and the physical layer overhead of 20 octets).

At time T1, the stream queue for p*o* is empty so a maximum length frame is selected for transmit just as stream frames are received on ports p1 through p*n* and a maximum length frame is received on our target port p*t*. The streams frames S1 A through S*n* A are now in the stream queue for port p*o* and the queue begins to acquire credits.

At time T2, additional stream frames are received on ports p1 through p*n* which are placed in the stream queue for port p*o*. In addition, a stream frame is received on the target port p*t* (frame S*t* P) is received and placed in the stream queue for port p*o*. This is the point for measuring the delay imposed by the stream queue on frame S*t* P. The stream queue contains the following frames: S1 A through S*n* A, S1 B through S*n* B, and lastly S*t* P. The output port has transmitted a number of octets of the maximum length frame equivalent to the size of frame S1 B (with overhead).

The delay from time T2 until the last octet of frame S*t* P is transmitted on port p*o*, qDelay, is given by these formulas:

$$\text{qDelay} = \text{qDelayOctets} \bullet \textit{octetTransmitTime}$$

$$\text{qDelayOctets} = \textit{maxInterferenceOctets} + 2 \bullet (\text{maxReservedOctets} - \textit{StPoctets}) - \text{ceil}((\text{maxReservedOctets} - \textit{StPoctets}) \div \text{N}) + \textit{StPoctets}$$

$$\text{maxReservedOctets} = \text{floor}(\textit{classAmeasurmenInterval} \div \textit{octetTransmitTime} \bullet \textit{deltaBandwidth(A)})$$

$$\text{N} = \min(\textit{fanInLimit}, \text{floor}((\text{maxReservedOctets} - \textit{StPoctets}) \div \textit{minFrameOctets}))$$

Where:

octetTransmitTime is the time to transmit one octet on port p*o* (0.08 μs).

maxInterferenceOctets, for Fast Ethernet, this is the maximum length frame supported by the bridge on port p*o* with physical layer overhead. Another source of interference is the Energy Efficient Ethernet recovery time; but, on Fast Ethernet, this is less than the maximum frame length and is mutually exclusive.

StPoctets is the number of octets in the target stream frame, S*t* P, with physical layer overhead.

classAmeasurmentInterval is the measurement interval for SR Class A (see 34.6.1 of 802.1Qav).

deltaBandwidth(A) is the maximum proportion of transmit bandwidth that can be reserved for SR Class A traffic (see 34.3 of 802.1Qav). Typically, this is 75% (for our purposes we assume the number has been converted to a decimal fraction, i.e. 0.75).

fanInLimit is the number of ports on the bridge that can support stream reception, not counting ports p*t* and p*o*. It would be possible to add features to SRP to set a *FanInLimit* lower than this if SRP guarantees the number.

minFrameOctets is the minimum frame size on Ethernet of 64 octets plus 20 octets of physical layer overhead.

The functions ceil and floor are standard mathematical functions which give the integer portion of their parameter, ceil returns the smallest integer not less than its parameter whereas floor returns the largest integer not greater than its parameter.

Gigabit Ethernet and Faster Media

The discussions at the July 2009 Plenary meeting did not include Ethernet speeds faster than 100 Mbs. These are my own thoughts and are in need of a full review.

At speeds of 1 Gbs and above the maximum interference size is the recovery time for exiting low power idle (16.5 µs at 1 Gbs, less at higher speeds). Since at least 31.25 µs is reserved for non-stream traffic every 125 µs, it would, at first glance, seem that the worst case queue delay would be when all the rest of the reserved stream bandwidth for a period were to be received in the instant prior to the target frame and the link were in low power idle. This assumption yields:

$$\text{qDelay} = (\text{maxReservedOctets} + \textit{maxInterferenceOctets}) \bullet \textit{octetTransmitTime}$$

However, this formula also assumes that all the input ports on the bridge are running at the same speed. If the input ports in Figure 1 are running at 100 Mbs then the bunching effect of interfering traffic can cause a burst of stream traffic of maxBurstSize (see 801.1Qav Annex L.1). During this burst the port is receiving stream data faster than its nominal data rate. If this happens on all the ports providing interfering streams then the output queue will become backed up because its shaper limits its own output rate.

Referring again to Figure 1, this time with ports p1 through p*n* operating at 100 Mbs and port p*o* operating at 1 Gbs or faster. At time T0 the burst of stream data begins on ports p1 through p*n*. At time T2 the last frame of the burst is received on each of ports p1 through p*n* followed by the last frame of the target stream on port p*t* (we assume no interfering traffic on port p*t* so that frames are received every *classAmeasurmentInterval*). Assume N is the p*o* operating rate divided by 100 Mbs (i.e. 10 for 1Gbs, 100 for 10 Gbs). Unless explicitly subscripted the values are for port p*o*.

$\text{streamOctets}_n = \text{ceil}((\text{maxReservedOctets} – \textit{StPoctets}) \div \text{N})$

Note: to be completely accurate the floor function should be used for values of *n* greater than (maxReservedOctets – *StPoctets*) modulo N. However, for our purposes we will just assume all values of *n* use the above value.

$\text{idleSlope}_n = \text{streamOctets}_n \bullet 8 \div \textit{classAmeasurmentInterval}$

$\text{sendSlope}_n = \text{idleSlope}_n - \textit{portTransmitRate}_n$

$\text{loCredit}_n = \text{streamOctets}_n \bullet 8 \bullet \text{sendSlope}_n \div \textit{portTransmitRate}_n$

Note: both sendSlope_n and loCredit_n are negative.

$\text{hiCredit}_n = \text{maxInterferenceOctets}_n \bullet 8 \bullet \text{idleSlope}_n \div \text{portTransmitRate}_n$

$\text{maxBurstSize}_n = \textit{portTransmitsRate}_n \bullet (\text{loCredit}_n - \text{hiCredit}_n) \div \text{sendSlope}_n$

$\text{maxBurstTime}_n = \text{maxBurstSize}_n \div \textit{portTransmitRate}_n$
$= (\text{loCredit}_n – \text{hiCredit}_n) \div \text{sendSlope}_n$

totalBitsQueued = N • maxBurstSize$_n$ + floor(maxBurstTime$_n$ ÷ *classAmeasurmentInterval*) • *StPoctets* • 8

idleSlope = maxReservedOctets • 8 ÷ *classAmeasurementInterval* ≈ *portTransmitRate* • *deltaBandwidth(A)*

qDelay = (totalBitsQueued ÷ idleSlope) – maxBurstSize$_n$ + *maxInterferenceTime*

Where:

maxInterferenceTime is the worst case Energy Efficient Ethernet recovery time for the output port. If EEE is disabled then this would be the time to transmit the maximum length frame supported by the output port.

portTransmitRate is the operating speed of the port in bits per second.

Note: work still to be done to account for bridges where N is limited due to a *fanInLimit* enforced by SRP or due to the number of ports implemented on the bridge.