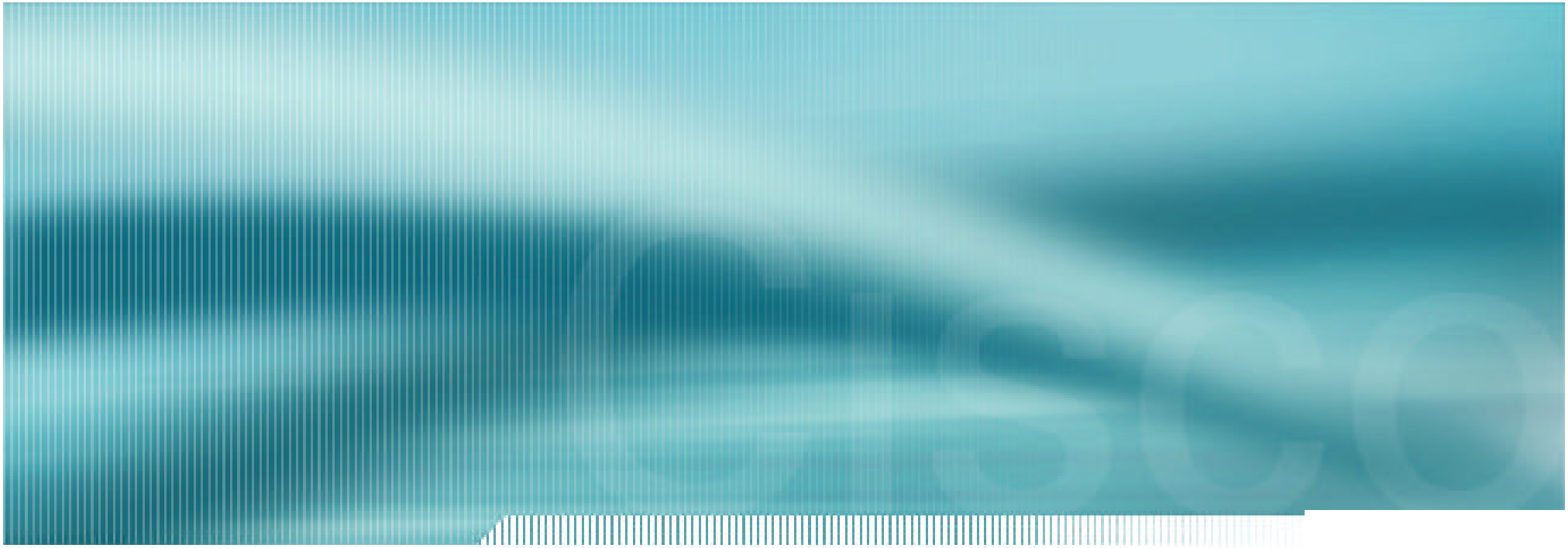


Rate control for Ethernet congestion management

Hugh Barrass (Cisco Systems)

Agenda

- 3 types of rate control
- Changes to Clause 4A and Clause 30
- Interaction with CRS and deference
- Remote rate control request
- Conclusions and proposals

3 types of rate control

Rate control will fix a link at a reduced rate

- **Bits transmitted at the same rate; packet rates reduced**
- **There are 3 types of rate control to define**

a) Constant (per packet) overhead

- **Effectively increases min IPG**

b) Limited (payload) bit rate

- **Dependant on packet length (as in 802.3ae)**

c) Limited packet rate

- **Counts the number of SFD's per second**

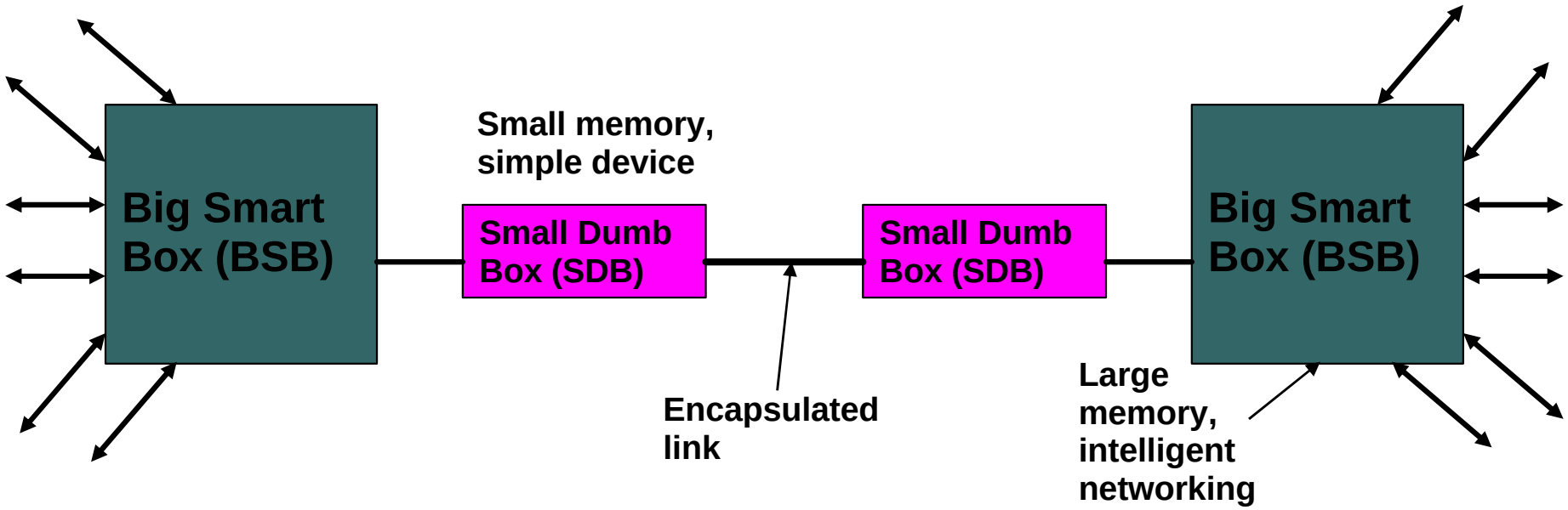
Constant packet overhead

Explored in detail in daines_cmsg_1_0409.pdf (thanks Kevin)

- **Becomes a significant problem for inline MACsec implementations**
- **… or other "dongle" encapsulator applications**

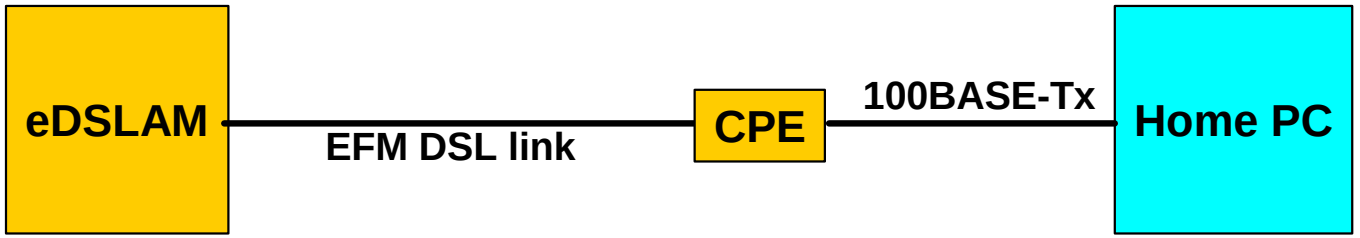
Inline encapsulators (dongles) must be economic devices

- **Small buffers, limited smarts (maybe line powered)**
- **Network performance across constricted link sucks!**



Limited payload bit rate

- Example in barrass_1_0704.pdf for high speed NIC
  - Ethernet link rate exceeds NIC bus rate, creating constriction
  - Limited intelligence & buffer in NIC – arbitrary packet drop
- Also applies for .3ah (EFM-DSL) CPE devices
  - Very simple CPE, with limited buffering,
  - Bridges between (e.g.) 100Mb LAN & 30Mb WAN links
- Can be used as a friendlier way of enforcing SLA
  - Link is limited to customer bit rate instead of policing & packet drop
  - Better overall network performance (if customer makes use of it)



Limited packet rate

- **No specific demand for this as yet, but…**
  - **… applications are easy to imagine**
- **Device with limited lookup engine rate**
  - **(e.g. cheap 10G)**
- **Interrupt driven or microcoded NIC**
  - **Must service each packet before proceeding with next**
  - **DMA allows high bit rate for large packets**

Agenda

- 3 types of rate control
- Changes to Clause 4A and Clause 30
- Interaction with CRS and deference
- Remote rate control request
- Conclusions and proposals

Changes to Clause 4A

- Changes are only considered for full duplex
  - … therefore propose Clause 4A
- 4A.2.3 Frame transmission model
  - Includes 4A.2.3.2.2 interframe spacing
- Propose to use similar method to 802.3ae
  - Look in Clause 4 for details
  - Redefine interFrameSpacing into 2 parts
  - Add specification for 2nd part of IFS
  - Needs additional state & counter for frame rate method

Change to 4A.2.3.2.2 Interframe spacing

"A larger value for interframe spacing is used for controlling the nominal data rate of the MAC sublayer (with packet granularity). While in this optional mode of operation, the MAC sublayer waits a further time after the interFrameSpacing according to the variable additionalInterFrameSpacing. Furthermore the MAC sublayer waits until the optional timer frameRateControlTimer reaches zero before transmitting a new frame. For more details, see 4A.2.7 and 4A.2.8"

These changes would allow the definition of 802.3ae WAN timing to be incorporated into the Annex 4A reduced MAC

- **Useful for rate controlled applications using WAN PHY**

Change to 4A.2.7 Transmit state variables

New variables (1)

additionalInterFrameSpacing: 1..?; {calculated per frame or fixed depending on rate control method}

frameRateControlTimer: 0..?; {down counter to enforce the minimum time between successive frame starts; always counts down each bit time until it reaches zero}

rateLimitPacketOverheadEnable: Boolean; {Indicates the desired rate control mode}

rateLimitPayloadRateEnable: Boolean; {Indicates the desired rate control mode}

rateLimitFrameRateEnable: Boolean; {Indicates the desired rate control mode}

rateLimitEnable: = rateLimitPacketOverheadEnable or rateLimitPayloadRateEnable or rateLimitFrameRateEnable

Change to 4A.2.7 Transmit state variables

New variables (2)

additionalPacketOverhead = ...; {In bytes, minimum amount that is added to each packet, when rateLimitPacketOverheadEnable is enabled}

ifsStretchRatio = ...; {In bits, determines the number of bits in a frame that require one octet of interFrameSpacing extension, when rateLimitPayloadRateEnable is enabled}

frameRateControlStart = ...; {In bits, the value loaded into the frameRateControlTimer at the start of each frame, when rateLimitPayloadRateEnable is enabled}

ifsStretchCount: 0..ifsStretchRatio; {In bits, a running counter that counts the number of bits during a frame's transmission that are to be considered for the minimum interFrameSpacing extension, while operating in ifsStretchMode}

ifsStretchSize: 0..(((maxUntaggedFrameSize + qTagPrefixSize) x 8 + headerSize + interFrameSpacing + ifsStretchRatio – 1) div ifsStretchRatio);

Change to 4A.2.8 Frame transmission

Process Deference;

Change line:

“if deferenceMode then Wait(interFrameSpacing);” becomes

“if deferenceMode then

begin

wait (interFrameSpacing + additionalInterFrameSpacing);”

if (not frameWaiting or (ifsStretchSize < additionalInterFrameSpacing) or (frameRateControlTimer != 0)) then ifsStretchCount := 0;

while (frameRateControlTimer != 0) do nothing;

end”

Add text after process definition:

If the any rate limiting is enabled, the Deference process continues to enforce interframe spacing for an additional number of bit times, after the completion of timing the interFrameSpacing. The additional number of bit times is reflected by the variable additionalInterFrameSpacing. If the resulting frame plus interframe spacing is less than the minimum period allowed by the frame rate control then the process waits for frameRateControlTimer to count down. If variable ifsStretchCount determines the interframe spacing, ifsStretchCount is less than ifsStretchRatio and the next frame is ready for transmission (variable frameWaiting is true), the Deference process enforces interframe spacing only for the integer number of octets, as indicated by ifsStretchSize, and saves ifsStretchCount for the next frame’s transmission. Otherwise ifsStretchCount is set to zero.

Change to 4A.2.8 Frame transmission

Process BitTransmitter;

After line:

“begin {Inner Loop}” add

“if rateLimitEnable then {Calculate the counter values}

begin

if (rateLimitPayloadRateEnable) ifsStretchSize := (ifsStretchCount + headerSize + frameSize + interFrameSpacing) div ifsStretchRatio; {payload rate limit}

else ifsStretchSize := 0;

ifsStretchCount := (ifsStretchCount + headerSize + frameSize + interFrameSpacing) mod ifsStretchRatio; {Remainder to carry over into the next frame’s transmission}

if (rateLimitPacketOverheadEnable and (additionalPacketOverhead > ifsStretchSize)) additionalInterFrameSpacing := additionalPacketOverhead x 8;

else additionalInterFrameSpacing := ifsStretchSize x 8;

end”

Change to 4A.2.8 Frame transmission

Process StartTransmit;

After line:

"begin" add

"if rateLimitFrameRateEnable then
frameRateControlTimer := frameRateControlStart
{Load frame rate counter}"

Add text after process definition:

The frameRateControlTimer counts down each bit time until it reaches zero. If rateLimitFrameRateEnable is false the counter will never load and will remain zero.

Management

- Clause 30 definitions are needed for the following variables:
  - rateLimitPacketOverheadEnable, rateLimitPayloadRateEnable, rateLimitFrameRateEnable {enables for the rate limiter features}
  - additionalPacketOverhead {for fixed packet overhead}
  - ifsStretchRatio {for fixed payload rate limit}
  - frameRateControlStart {for frame rate limit}
- Additionally, objects are needed for local receive capabilities
- Analogous to the 6 objects for transmit rate limits
  - 3 boolean objects to indicate a limitation
  - 3 parameters for the 3 types of rate limit

Agenda

- 3 types of rate control
- Changes to Clause 4A and Clause 30
- Interaction with CRS and deference
- Remote rate control request
- Conclusions and proposals

Interaction with CRS and deference

- The definition in the previous section follows the same format used in Clause 4 for carrier extension and IFS stretch (802.3ae)
  - The IFS is stretched regardless of state of CRS mode …
  - … but only if deference mode is enabled
  - Except that the frame rate timer continues in any case
- Need to consider the interaction of rate control with other mechanisms
  - External frame spacing (EPON)
  - Ethernet over DSL (CRS mode)
  - Etc…

Interaction with (non) deference

- The EPON (or other) external scheduler determines when a frame is available to send
  - This would normally be in bursts (but not necessarily)
  - Using rate limiter would enable reduction in buffering requirements
  - Steady state case should be handled through scheduler
  - Scheduler may not (easily) be able to handle frame rate limit
- Propose to leave as defined
  - Frame rate limiter will operate
  - Other rate limiters will not

Interaction with CRS mode

- **The PHY is controlling payload data rate using CRS**
  - **Additional IFS stretch due to rate limiter appears redundant**
  - **Furthermore EFM copper PHY will compress and eliminate IFS in any case**
  - **However, there may be a case for MAC rate limiting in addition to PHY rate limiting (inline security tagger with E-DSL)**
- **Propose to leave as defined**
  - **If rate limiter produces net b/w more than PHY, CRS will determine rate**
  - **If rate limiter produces net b/w less than PHY, CRS will not be asserted (& rate limiter functions as normal)**
  - **Combination space operates at sub-optimal efficiency (sometimes packets are delayed unnecessarily)**

Agenda

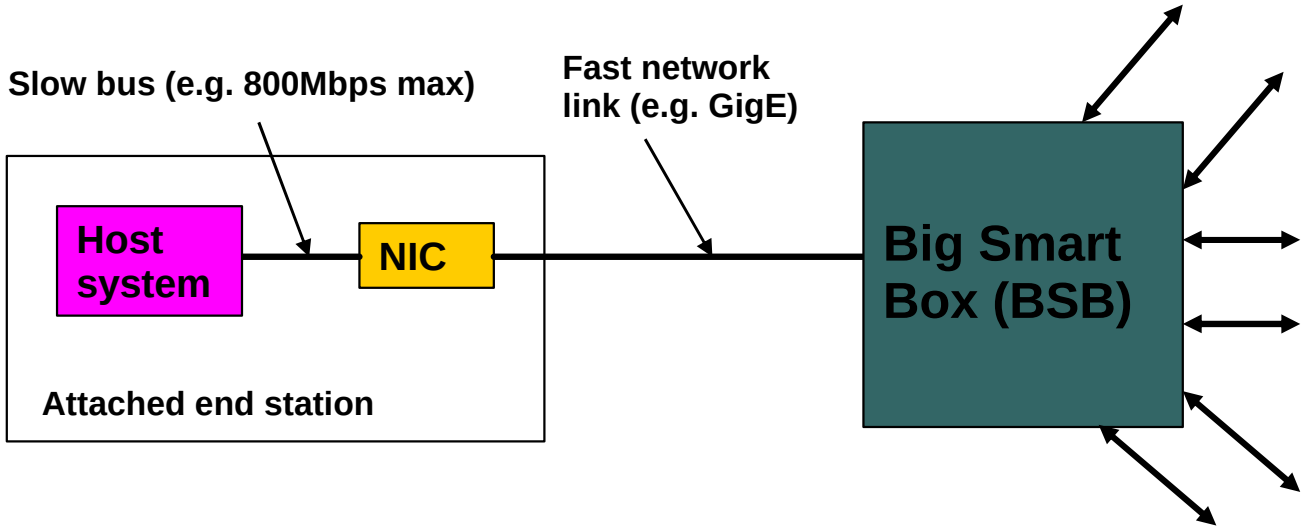
- 3 types of rate control
- Changes to Clause 4A and Clause 30
- Interaction with CRS and deference
- Remote rate control request
- Conclusions and proposals

Remote rate control request

A case can be made for defining a remote mechanism

- A device can tell its link partner to limit the Tx rate
- In addition to the MIB method

Example configuration (justification for rate control)



Network management could set egress rate control on BSB

- But end station may be moved arbitrarily
- Much more convenient for end station to signal its requirement

Request definition

- **Rate control is pseudo static**
  - **No real-time requirement**
  - **Two suggestions (so far)**
- **Slow protocol frame**
  - **Similar to .3ah OAM**
  - **Defined entirely within 802.3 (OAM layer?)**
- **Piggy-back on LLDP**
  - **Discovered device parameter includes rate limit**
  - **Would need modification to 802.1**
- **Proposal is still open**
  - **Rx MIB attributes defined, await detailed proposal for mechanism**

Agenda

- 3 types of rate control
- Changes to Clause 4A and Clause 30
- Interaction with CRS and deference
- Remote rate control request
- Conclusions and proposals

Summary

- Changes to Clause 4A & 30
- 3 types of mechanisms for f-d MACs
- MIB attributes for rate control
- Still to do:
  - Address remote request (if required)
  - Changes to Clause 4/4A & Clause 2 (31) to clean up MAC client interface

Proposals

- **Adopt changes to Annex 4A & Clause 30 described in this presentation**
- **Review further interaction with (non) deference and CRS mode**
- **Review further necessity and definition for remote rate control request**
- **Study required changes for MAC client interface**