Proposal to IEEE P802.3ar CMTF

This proposal is intended to serve as the basis for creating IEEE P802.3ar/D1.0.

This is ***merely a proposal*** and has ***no official status*** within IEEE 802.3 or IEEE P802.3ar.

This proposal specifies optional mechanisms to limit the rate of transmitted data within the MAC.

This proposal is based on the P802.3ar CMTF motion #1 from the July 2005 meeting in San Francisco, which stated:

> "Adopt changes to Clause 4, Annex 4A & Clause 30 described in barrass_1_0505.pdf as a baseline proposal for 802.3ar/D1.0. The changes to Clause 4 will be made after the changes to Annex 4A have been solidified in 802.3ar TF review."
>
> Move: P. Thaler Second: M. Squire
>
> 802.3ar TF vote: Y: 16 N: 1 A: 9 (31 room count)
>
> Requires ≥ 75%
>
> Passes
>
> 802.3 voting members present: Y: 14 N: 1 A: 2
>
> Passes

This proposal will be reviewed at the next P802.3ar TF meeting to be held during the IEEE 802 Plenary meeting in Vancouver, the week of November 13-18th. It is anticipated that the CMTF will produce D1.0 after the November meeting and commence a TF ballot.

Kevin Daines

Chair, IEEE P802.3ar CMTF

Proposed revisions to IEEE 802.3-200x, Clause 4

Editors' Notes: To be removed prior to final publication.

This proposal is based on IEEE P802.3as/D2.1. The proposed editing instructions define how to merge the material contained here into the base document set to form the new comprehensive standard as created by the addition of P802.3ar.

This proposal does not cover the changes necessary to Clause 4. The changes to Clause 4 will be made after the changes to Annex 4A have been solidified. This is expected to occur no later than D2.0.

Revision History:
Draft 0.9, October 2005 Proposal for Task Force review in Vancouver.

4. Media Access Control

Proposed revisions to IEEE 802.3-200x, Annex 4A

Editors' Notes: To be removed prior to final publication.

This proposed revision is based on IEEE P802.3as/D2.1. The editing instructions define how to merge the material contained here into the base document set to form the new comprehensive standard as created by the addition of P802.3ar.

Revision History:
Draft 0.9, October 2005 Proposal for Task Force review in Vancouver.

4A. Simplified full duplex media access control

4A.2.3.2.2 Interframe spacing

Change subclause 4A.2.3.2.2 as follows:

As defined in 4A.2.3.2.1, the rules for deferring ensure a minimum interframe spacing of interFrameSpacing bit times. This is intended to provide interframe recovery time to aid in frame delineation on the physical medium.

Note that interFrameSpacing is the minimum value of the interframe spacing. ~~If necessary for implementation reasons, a transmitting sublayer may use a larger value with a resulting decrease in its throughput. The larger value is determined by the parameters of the implementation, see 4A.4.~~ A larger value for interframe spacing is used for controlling the nominal data rate of the MAC sublayer (with frame granularity). While in this optional mode of operation, the MAC sublayer waits a further time after the interFrameSpacing according to the variable txAdditionalInterFrameSpacing. Furthermore the MAC sublayer waits until the optional timer txFrameRateControlTimer reaches zero before transmitting a new frame. For more details, see 4A.2.7 and 4A.2.8.

4.A.2.7.2 Transmit state variables

Insert the following new variables in subclause 4A.2.7.2:

txAdditionalInterFrameSpacing: 1..?; {calculated per frame or fixed depending on rate control method}
txFrameRateControlTimer: 0..?; {timer to enforce the minimum time between successive frame starts}
txRateLimitFrameOverheadEnable: Boolean; {Indicates the desired rate control mode}
txRateLimitPayloadRateEnable: Boolean; {Indicates the desired rate control mode}
txRateLimitFrameRateEnable: Boolean; {Indicates the desired rate control mode}
txRateLimitEnable: = txRateLimitFrameOverheadEnable
or txRateLimitPayloadRateEnable
or txRateLimitFrameRateEnable;
txAdditionalFrameOverhead = ...; {In bytes, minimum amount that is added to each packet, when txRateLimitFrameOverheadEnable is enabled}
txIfsStretchRatio = ...; {In bits, determines the number of bits in a frame that require one octet of interFrameSpacing extension, when txRateLimitPayloadRateEnable is enabled}
txFrameRateControlStart = ...; {In bits, the value loaded into the txFrameRateControlTimer at the start of each frame, when txRateLimitFrameRateEnable is enabled}
txIfsStretchCount: 0..txIfsStretchRatio; {In bits, a running counter of the number of bits during a frame's transmission that are to be considered for the minimum

interFrameSpacing extension}
txIfsStretchSize: 0..(((maxBasicFrameSize + qTagPrefixSize) x 8 + headerSize + interFrameSpacing + txIfsStretchRatio – 1) div txIfsStretchRatio);

4A.2.7.3 Receive state variables

Insert the following at the end of subclause 4A.2.7.3:

If the receiving system has limitations that prevent it from receiving frames at the maximum rate possible for the attached medium then it may define the following variables:

```
var
   rxRateLimitFrameOverheadEnable: Boolean; {Indicates the desired rate control mode}
   rxAdditionalFrameOverhead = ...; {In bytes, desired minimum amount that is added to each packet,
                                     when rxRateLimitFrameOverheadEnable is true}
   rxRateLimitPayloadRateEnable: Boolean; {Indicates the desired rate control mode}
   rxIfsStretchRatio = ...; {In bits, determines the desired number of bits in a frame that require one octet of
                             interFrameSpacing extension, when rxRateLimitPayloadRateEnable is enabled}
   rxRateLimitFrameRateEnable: Boolean; {Indicates the desired rate control mode}
   rxFrameRateControlStart = ...; {In bits, the desired value of the link partner's
                                   txFrameRateControlTimer at the start of each frame}
```

The values of these variables reflect the values that the receiver would wish to be programmed into the link partner's transmit rate limiter (see 4A.2.7 and 4A.2.8).

4A.2.8 Frame transmission

Delete existing process Deference in 4A.2.8:

```
process Deference;
begin
   cycle {Main loop}
      while (not transmitting and not (carrierSenseMode and carrierSense)) do nothing; {Wait for the start
                                                                                        of transmission or congestion}
      deferring := true; {Inhibit future transmissions}
      while (transmitting or (carrierSenseMode and carrierSense)) do nothing; {Wait for the end of
                                                                               transmission and congestion}
      if deferenceMode then Wait(interFrameSpacing); {Time out entire interframe gap if enabled}
      deferring := false {Don't inhibit transmission}
   end {Main loop}
end; {Deference}
```

Insert new process Deference in 4A.2.8:

```
process Deference;
begin
   cycle {Main loop}
      while (not transmitting and not (carrierSenseMode and carrierSense)) do nothing; {Wait for the start
                                                                                        of transmission or congestion}
      deferring := true; {Inhibit future transmissions}
      while (transmitting or (carrierSenseMode and carrierSense)) do nothing; {Wait for the end of
                                                                               transmission and congestion}
```

```
        if deferenceMode then
        begin
            Wait(interFrameSpacing + txAdditionalInterFrameSpacing);
            if (not frameWaiting
                or (txIfsStretchSize < txAdditionalInterFrameSpacing)
                or (txFrameRateControlTimer != 0)) then txIfsStretchCount := 0;
            while (txFrameRateControlTimer != 0) do nothing;
        end
        deferring := false {Don't inhibit transmission}
    end {Main loop}
end; {Deference}
```

Insert new paragraph after process Deference in 4A.2.8:

If 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 txInterFrameSpacing. The additional number of bit times is reflected by the variable txAdditionalInterFrameSpacing. If the resulting frame plus interframe spacing is less than the minimum period allowed by the frame rate control then the process waits for txFrameRateControlTimer to count down. If variable txIfsStretchCount determines the interframe spacing, txIfsStretchCount is less than txIfsStretchRatio 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 txIfsStretchSize, and saves txIfsStretchCount for the next frame's transmission. Otherwise txIfsStretchCount is set to zero.

Insert following into top of {Inner loop} of process BitTransmitter in 4A.2.8:

```
            if txRateLimitEnable then {Calculate the counter values}
                begin
                    if (txRateLimitPayloadRateEnable) then
                        begin
                            txIfsStretchSize := (txIfsStretchCount + headerSize + frameSize +
                                                interFrameSpacing) div txIfsStretchRatio; {payload rate limit}
                            txIfsStretchCount := (txIfsStretchCount + headerSize + frameSize +
                                                    interFrameSpacing) mod txIfsStretchRatio;
                                                    {Remainder to carry over into the next frame's transmission}
                        end
                    else txIfsStretchSize := 0;
                    if (txRateLimitFrameOverheadEnable and
                            (txAdditionalFrameOverhead > txIfsStretchSize))
                        then txAdditionalInterFrameSpacing := txAdditionalFrameOverhead x 8;
                        else txAdditionalInterFrameSpacing := txIfsStretchSize x 8;
                end
```

Insert following into top of procedure StartTransmit in 4A.2.8:

```
    if txRateLimitFrameRateEnable then txFrameRateControlTimer := txFrameRateControlStart;
                                                                    {Load frame rate counter}
```

Insert process FrameRateControlTimer in 4A.2.8:

FrameRateControlTimer is a process that does nothing unless the txFrameRateControlTimer variable is non-zero. When txFrameRateControlTimer is non-zero, FrameRateControlTimer waits txFrameRateControlTimer bit times:

```
process FrameRateControlTimer;
begin
   cycle
      while txFrameRateControlTimer = 0 do nothing; {Wait for a transmit to start and load timer}
      Wait(txFrameRateControlTimer)
   end
end; {FrameRateControlTimer}
```

Proposed revisions to IEEE 802.3-200x, Clause 30

Editors' Notes: To be removed prior to final publication.

This proposed revision is based on IEEE P802.3as/D2.1. The editing instructions define how to merge the material contained here into the base document set to form the new comprehensive standard as created by the addition of P802.3ar.

Revision History:
Draft 0.9, October 2005 Proposal for Task Force review in Vancouver

30. Management

30.2.5 Capabilities

Insert the following attributes into Table 30-1, below aRateControlStatus, as follows:

				DTE								Repeater				MAU					
				Basic Package (Mandatory)	Mandatory Package (Mandatory)	Recommended Package (Optional)	Optional Package (Optional)	Array Package (Optional)	Excessive Deferral Package (Optional)	Multiple PHY Package (Optional)	PHY Error Monitor Capability (Optional)	Basic Control Capability (Mandatory)	Performance Monitor Capability (Optional)	Address Tracking Capability (Optional)	100/1000 Mb/s Monitor Capability (Optional)	1000 Mb/s Burst Monitor Capability (Optional)	Basic Package (Mandatory)	MAU Control Package (Optional)	Media Loss Tracking Package (Conditional)	Broadband DTE MAU Package (Conditional)	MII Capability (Conditional)	PHY Error Monitor Capability (Optional)	Auto-Negotiation Package (Mandatory)
	aTxRateLimitFrameOver-headStatus	ATTRIBUTE	GET-SET		X[2]	X																
	aTxAdditionalFrameOver-head	ATTRIBUTE	GET-SET		X[2]	X																
	aTxRateLimitPayloadRat-eStatus	ATTRIBUTE	GET-SET		X[2]	X																
	aTxIfsStretchRatio	ATTRIBUTE	GET-SET		X[2]	X																
	aTxRateLimitFrameRat-eStatus	ATTRIBUTE	GET-SET		X[2]	X																
	aTxFrameRateControlStart	ATTRIBUTE	GET-SET		X[2]	X																
	aRxRateLimitFrameOver-headStatus	ATTRIBUTE	GET			X																
	aRxAdditionalFrameOver-head	ATTRIBUTE	GET			X																
	aRxRateLimitPayloadRat-eStatus	ATTRIBUTE	GET			X																
	aRxIfsStretchRatio	ATTRIBUTE	GET			X																
	aRxRateLimitFrameRat-eStatus	ATTRIBUTE	GET			X																
	aRxFrameRateControlStart	ATTRIBUTE	GET			X																

Insert note after NOTE 1 in Table 30-1 as follows:

NOTE 2: The aTxRateLimitFrameOverheadStatus, aTxAdditionalFrameOverhead, aTxRateLimitPayloadRateStatus, aTxIfsStretchRatio, aTxRateLimitFrameRateStatus and aTxFrameRateControlStart attributes are Mandatory in systems that implement rate control, except for systems use a fixed value of txIfsStretchRatio of 104 bits (see 4.4.2). They are Recommended for other systems.

Insert subclause 30.3.1.1.37 as follows

30.3.1.1.37 aTxRateLimitFrameOverheadStatus

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

transmit rate limit frame overhead off	Transmit rate limit mode disabled
transmit rate limit frame overhead on	Transmit rate limit mode enabled
unknown	Transmit rate limit mode unknown

BEHAVIOUR DEFINED AS:

A GET operation returns the current Transmit Rate Limit Frame Overhead mode of operation of the MAC sublayer.

A SET operation changes the mode of operation of the MAC sublayer to the indicated value. A SET operation shall have no effect on a device whose mode cannot be changed through management or that can only operate in a single mode.

This attribute maps to the variable txRateLimitFrameOverheadEnable (see 4A.2.7.2)

Insert subclause 30.3.1.1.38 as follows

30.3.1.1.38 aTxAdditionalFrameOverhead

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

A GET operation returns the current Transmit Additional Frame Overhead of the MAC sublayer.

A SET operation changes the mode of operation of the MAC sublayer to the indicated value. A SET operation shall have no effect on a device whose mode cannot be changed through management or that can only operate in a single mode.

This attribute maps to the variable txAdditionalFrameOverhead (see 4A.2.7.2)

Insert subclause 30.3.1.1.39 as follows

30.3.1.1.39 aTxRateLimitPayloadRateStatus

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

transmit rate limit payload rate off	Transmit rate limit mode disabled
transmit rate limit payload rate on	Transmit rate limit mode enabled
unknown	Transmit rate limit mode unknown

BEHAVIOUR DEFINED AS:

A GET operation returns the current Transmit Rate Limit Payload Rate mode of operation of the MAC sublayer.

A SET operation changes the mode of operation of the MAC sublayer to the indicated value. A SET operation shall have no effect on a device whose mode cannot be changed through management or that can only operate in a single mode.

This attribute maps to the variable txRateLimitPacketPayloadRate (see 4A.2.7.2)

Insert subclause 30.3.1.1.40 as follows

30.3.1.1.40 aTxIfsStretchRatio

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
A GET operation returns the current Transmit Inter Frame Spacing Stretch Ratio of the MAC sublayer.

A SET operation changes the mode of operation of the MAC sublayer to the indicated value. A SET operation shall have no effect on a device whose mode cannot be changed through management or that can only operate in a single mode.

This attribute maps to the variable txIfsStretchRatio (see 4A.2.7.2)

Insert subclause 30.3.1.1.41 as follows

30.3.1.1.41 aTxRateLimitFrameRateStatus

ATTRIBUTE

APPROPRIATE SYNTAX:
An ENUMERATED VALUE that has one of the following entries:

transmit rate limit frame rate off	Transmit rate limit mode disabled
transmit rate limit frame rate on	Transmit rate limit mode enabled
unknown	Transmit rate limit mode unknown

BEHAVIOUR DEFINED AS:
A GET operation returns the current Transmit Rate Limit Frame Rate mode of operation of the MAC sublayer.

A SET operation changes the mode of operation of the MAC sublayer to the indicated value. A SET operation shall have no effect on a device whose mode cannot be changed through management or that can only operate in a single mode.

This attribute maps to the variable txRateLimitPacketFrameRate (see 4A.2.7.2)

Insert subclause 30.3.1.1.42 as follows

30.3.1.1.42 aTxFrameRateControlStart

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
A GET operation returns the current Transmit Frame Rate Control Start of the MAC sublayer.

A SET operation changes the mode of operation of the MAC sublayer to the indicated value. A SET operation shall have no effect on a device whose mode cannot be changed through management or that can only operate in a single mode.

This attribute maps to the variable txFrameRateControlStart (see 4A.2.7.2)

Insert subclause 30.3.1.1.43 as follows

30.3.1.1.43 aRxRateLimitPacketOverheadStatus

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

receive rate limit packet overhead off	Receive rate limit mode disabled
receive rate limit packet overhead on	Receive rate limit mode enabled
unknown	Receive rate limit mode unknown

BEHAVIOUR DEFINED AS:

A GET operation returns the current Transmit Rate Limit Packet Overhead desired of the remote MAC sublayer.

This attribute maps to the variable rxRateLimitPacketOverheadEnable (see 4A.2.7.3)

Insert subclause 30.3.1.1.44 as follows

30.3.1.1.44 aRxAdditionalPacketOverhead

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

A GET operation returns the current Transmit Additional Packet Overhead desired of the remote MAC sublayer.

This attribute maps to the variable rxAdditionalPacketOverhead (see 4A.2.7.3)

Insert subclause 30.3.1.1.45 as follows

30.3.1.1.45 aRxRateLimitPayloadRateStatus

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

receive rate limit payload rate off	Receive rate limit mode disabled
receive rate limit payload rate on	Receive rate limit mode enabled
unknown	Receive rate limit mode unknown

BEHAVIOUR DEFINED AS:

A GET operation returns the current Transmit Rate Limit Payload Rate desired of the remote MAC sublayer.

This attribute maps to the variable rxRateLimitPayloadRateEnable (see 4A.2.7.3)

Insert subclause 30.3.1.1.46 as follows

30.3.1.1.46 aRxIfsStretchRatio

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

A GET operation returns the current Transmit Inter Frame Spacing Stretch Ratio desired of the remote MAC sublayer.

This attribute maps to the variable rxIfsStretchRatio (see 4A.2.7.3)

Insert subclause 30.3.1.1.47 as follows

30.3.1.1.47 aRxRateLimitFrameRateStatus

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

receive rate limit frame rate off	Receive rate limit mode disabled
receive rate limit frame rate on	Receive rate limit mode enabled
unknown	Receive rate limit mode unknown

BEHAVIOUR DEFINED AS:

A GET operation returns the current Transmit Rate Limit Frame Rate mode desired of the remote MAC sublayer.

This attribute maps to the variable rxRateLimitFrameRateEnable (see 4A.2.7.3)

Insert subclause 30.3.1.1.48 as follows

30.3.1.1.48 aRxFrameRateControlStart

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

A GET operation returns the current Transmit Frame Rate Control Start desired of the remote MAC sublayer.

This attribute maps to the variable rxFrameRateControlStart (see 4A.2.7.3)