ANOTHER VIEW of LOW POWER IDLE / IDLE TOGGLE

Idle Toggle from a different viewpoint

Geoff Thompson
Nortel

Presentation to 802.3az
Energy Efficient Ethernet
Orlando, March, 2008

Outline

- Apology
- Purpose of PREAMBLE
- History of PREAMBLE implementation
- The great resulting architectural “misteak”
- Why this matters
- Looking at our problem with new eyes

Apology

- This is all dreadfully simple and elementary, too simple for us really.
- I think I have a point though
- We have been looking at the right problem from the wrong viewpoint.
- Please bear with me while I move the camera around to the other side

Purpose of PREAMBLE

- To condition receive circuitry from an *inactive receive state* to *valid data decoding* of a received data signal
- To cover the transition time between the two above states
  with a signal that is useful towards minimizing the time between the two.

History of PREAMBLE (in Ethernet)

- Originally (1973) Ethernet (3 Mb/s) had a quiet (no signal) idle (on mixing coax)
- At 3Meg, it needed essentially no preamble (it had 1 bit, like start bit on ASCII)
- 1980 Version (DIX) needed more preamble for cable to charge up time and time for PLL to lock.
- Those needs drove decision to current design (63 bits + start bit, re-expressed into octets)

History of PREAMBLE (implementation)

- Putting preamble into MAC description was an implementation decision, NOT an architectural one.
- In 1980 PLL was discrete (headed for custom bipolar), everything else (i.e. all the gate count) was put into a custom nMOS chip (which came to be known as the MAC chip)
- When the Ethernet spec was written the preamble description was thus included in the MAC portion of the definition (i.e. within the *Pascal* formal description).

History of PREAMBLE (implementation #2)

- None of the previous withstanding, on an architectural basis, the preamble belongs in the PHY
- It’s job is to take care of the PHY’s data recovery set-up needs.
- The preamble, thus, “should” be free to vary in order to meet the needs of new PHYs

Side Comment

- I don’t really expect us to restructure the entire 802.3 document to fix this…

BUT…

- It is useful to at least look at the problem. as though we owned the preamble and could mess with it.

(Thus that is where we are going to go.)

Looking at (say) 100BASE-TX
from preamble centric view

ASSUMPTIONS:

- Close as possible to current 100BASE-Tx
- Transmitter off 1 octet after end of data

QUESTIONS:

- How long does preamble have to be for next packet?
  - Worst case?
  - How big is the IPG before you hit worst case?
  - Is it worth having other shorter values?
  - How big does the preamble have to be for min IPG?
  - How many steps is it worth having in between?

Next Steps

- Look at higher speeds from same viewpoint
- Send preamble burst (without data payload) every once in awhile just to keep receiver conditioning current.
- Timing and duration of preamble requirements for fronting data, and for refresh would be PHY dependent.
- Ideally, we would move the fixed preamble from “the common MAC” to each PHY.
- More practically, use the preamble we have as a fixed portion and just add to it.

How this looks to MAC Client

- MAC-Client interface and service is “almost” unchanged.
- Look more like original Ethernet ½ Duplex MAC
- Presenting data to CSMA/CD MAC normally results in a “defer” (assuming traffic, collision or need to send preamble)
- Thus “our” MAC says “defer” until enough preamble has been sent.
- Defer time may be shorter if data is presented during refresh burp.
- Back-to-back packets go out with normal timing.

This is the starting point I propose:

- How long does preamble ***NEED*** to be for each speed?
  - How does that vary with interpacket time?
  - Do we need refresh preamble bursts (no data)?
  - What do the implementations need to be to satisfy the above?
  - How simple (i.e. large grained digital) can we make the implementations in the face of an analog (IPG length) requirement?

FEEDBACK ?

DISCUSSION?

Adoption?

THANK YOU !!

Geoff Thompson / Nortel
<thompson@ieee.org>