Analysis of B3 vs. MID

Revision 1.0

David Bagby
Bob O'Hara
Dave Roberts

Advanced Micro Devices
One AMD Place
Sunnyvale CA, 94088

Common Ground

- There was a lot of concensus reached around B3 functionality
- Alteration of NID to unique address
- FC field definitions
- Uniform 32-bit CRC
- Textual ESSID representation

Disagreement Surrounds WDS Support

- All believe that WDS is important
- Two solutions: 94/248 vs. MID
- Argument could be characteriszed as: stability vs. efficiency

B3 Summary Description

- B3 uses directed addressing for all frames in a transmission "dialog"
- RTS carries return address for CTS, Data carries return address for ACK

Issue with B3

- Problem results in wireless distribution system because 4 addresses are needed to be carried in a data frame: immediate transmitter, immediate receiver, original source, and final destination
- B3 only specified that 3 addresses be carried in a data frame
- Result was that second AP in forwarding sequence could not send ACK for Data frame to first AP

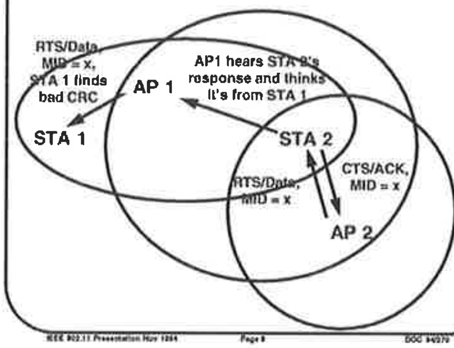
MID Summary Description

- MID attempts to remedy situation by making CTS and ACK non-directed
- A token (MID) is used to match a CTS with an RTS and an ACK with Data
- Transmitter "randomly" picks a MID value from 11-bit space (2048 choices)
- MID is included with RTS and Data and returned with CTS and ACK
- MID uses the bits B3 used for sequence number, plus a couple more, and so is also used for MPDU duplicate detection

Problem with MID

- The problem with the MID is the same problem MPDUID originally had
- Two stations in close proximity can choose the same MID value and frames in a dialog can become miscorrelated
- This can lead to errored frames not being retried and dropped

Example of Miscorrelation



MID Answers to Miscorrellation Problem

- It's unlikely
- It's only a problem if the frames are the same size
- Therefore, it's not a problem

"It's Unlikely..."

- We have an 11-bit token
- MID proponents claim random number means stations are highly unlikely to choose the same token
  - 1 in 2048 chance of choosing any particular value
- MID proponents claim that even if they do, it probably won't happen twice
  - i.e., probability is 1/(2048)^2 = very small

A Look At "Random" Numbers

- "Random" numbers are generated using an algorithm operating on a seed value
  - Linear Congruential
  - LFSR
- They are "random" at a macroscopic scale (spectral test, etc.) but not at a microscopic scale
- Same seed value always generates same output
- Seed value for next choice is the output of previous choice
  - $R_n = F(R_{n-1})$
- PRNG results in a sequence of numbers

Example PRNG

- A 3-bit linear congruential
  - F(n) = (5n + 3) mod 8
  - RN(n) = F(RN(n-1))
- This produces the sequence:
  - 0, 3, 2, 5, 4, 7, 6, 1, 0, 3, ...
- The last RNG generated is used as the seed to generate the next random number, also known as the state of the generator
- Generators that cycle through all possible states before repeating are called maximal length generators
  - We want this property for a MID for duplicate detection

PRNG/MID Implications

- Given a particular state value, the PRNG "picks" the next value in sequence with P=1. It picks all other numbers from set with P=0
- Note that for the purposes of correlating frames in a dialog, we don't care what the MID value is. 13 is just as good a 1238.
- Note also that a simple counter has the same property as above. A PRNG is just a strange counter
- Since we don't care what the value of the PRNG output is, a counter is just as good as a PRN.
- ***The MID might as well be a simple counter!***

Stations Can Become "In Sync"

- Even if stations start out (at powerup) with different positions in the "random" sequence, traffic patterns can cause them to reach the same position
- Once they reach the same position, the probablly gets very high that they will collide
  - One station's position will "crossover" the other station's
- This becomes much more likely the more stations you have
- If outbound traffic patterns are similar, then stations will tend to stay in sync for a while
  - It may take them a long to reach this point, but it will also take them a long time to get out of it

Introducing More Randomness

- Use a larger seed
  - E.g., use low order bits of larger seed as PRN
  - Breaks maximal length property
- Generating a new MID for Data/ACK than that used for RTS/CTS doesn't help
  - You'll generate the same value!
  - You'll use duplicate detection tags more quickly, leading problems
- In general, doing anything breaks duplicate detection needs

Randomness Conclusions

- "Random" numbers aren't as random as you might think
- They are really no better than a sequence number
- Once stations become sync'd, the probability is quite high that collisions will occur
  - Probability of stations becoming sync'd increases greatly with increase in number of stations
- Trying to change this breaks duplicate detection properties
  - Maximal lenght property no longer exists which increases the probability of a duplicate detection failure

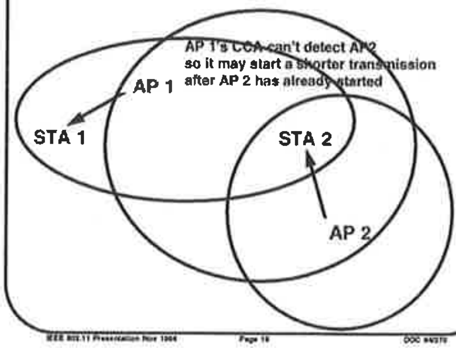
"It's Only A Problem With Frames Of Equal Size..."

- False: It's a problem with all frames that <u>end</u> at the same time
- AND, frames of equal size are actually very likely

Problem Window

- The problem occurs when a transmitter waits for a reply and sees what it thinks is a valid response
- This occurs if a false ACK or CTS is returned within a SIFS after the transmitter stops
- Different start times are acceptable as long as stop times are within a SIFS of each other

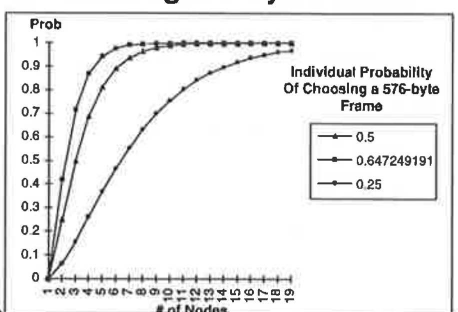
CCA Doesn't Necessarily Help



Probability of STAs Choosing Same Size Is High

- Frame sizes are not randomly distributed
  - I measured the sizes of 1236 TCP/IP frames
  - There were 25 different sizes
- Of 1236 frames, 800 were 576 bytes (65%)
- Assuming random choosing (false), then with 5 nodes the probability that any two choose 576-byte frames is 94%
  - See following graph
- Reality is that frame sizes are not "chosen" randomly, which makes things worse
  - Station doing file transfer outbound will send all large frames (the data)
  - Station doing file transfer inbound will send all short frames (network layer acknowledgements)

Probability of Two Nodes Choosing 576-Byte Frames



MID Conclusions

- "Random" numbers aren't random
  - PRN is no better than sequence # for MID purposes
- MID collision is more probable than first glance might suggest
  - Stations can get in sync and stay in sync for quite a while
  - Problem can occur many times in succession
- Frame sizes cluster and probability that two stations choose same frame size is high with only moderate number of nodes
- All probabilities increase with larger number of nodes (and increase is non-linear, as in case of two stations choosing same size frame)
- This IS a problem!

Arguments Against 94/248

- Efficiency
- Simplicity

Efficiency

- Some say MID is more efficient than B3
- Using MID results in dramatic reduction in frame header overhead

Frames	B3	MID	Delta
RTS/CTS/Data/ACK	60	48	-20%
Data/ACK	34	30	-11.8%

- Yes, but this is a meaningless measurement!

Throughput Delta

- The only meaningful measurement is change in throughput
  - Users don't care how big the frame headers are!
- AMD created a spreadsheet to calculate throughput based on total dialog exchange parameters (PHY preamble, SIFS/DIFS time, realistic payload sizes, etc).
  - We even figured in the decrease in throughput caused by retrys caused by increase in frame error rate caused by more bits being transmitted
  - This spreadsheet was distributed to the email reflector

Throughput Delta Results

Payload size	RTS/CTS	MID/B3 Diff
585	Y	1.59%
585	N	0.59%
39	Y	5.75%*
39	N	2.92%

- 585-byte payload represents 576-byte IP or IPX packet with 9-byte LLC/SNAP
- 39-byte payload represents 30-byte minimum size IPX packet with 9-byte LLC/SNAP

* Don't do this! But we already knew that.

Further Throughput Results

- Largest deltas occur when payload is small
  - As expected...
- But throughput is bad at this point anyway...
  - Headers and IFS times dominate in any case
- Expected throughput for B3 at 39-byte payload w/o RTS/CTS is only 284.5 Kbps
- Using a MID rather than B3 increases throughput to only 293 Kbps
- Using a MID isn't going to increase user satisfaction any...

Simplicity

- MID Proponents claim that MID results in easier state machine design
- This is true
  - Receiver copies MID unconditionally to the CTS or ACK
- But this is putting the cart before the horse
- The MID scheme doesn't work reliably!

Overall Conclusions

- The MID is unreliable
- Its proponents admit this but claim that the probability of failure is low
- BUT they haven't put forth the detailed analysis to show this!
- We've shown that the probability can be quite high under reasonable loading conditions and station count
- Although B3 directed frames result in a small decrease in throughput (<3%), B3 doesn't rely on probabalistic arguement to show that it works