4D 4-State 3dB/6dB Transparent Trellis Code for 1000BASE-T

Sailesh K. Rao

Level One Communications, Inc.

Outline

- Goals
- Current 4D and 24D coding schemes
- The 3dB/6dB transparent 4D 4-state Trellis Code
- Equivalent 16D 2-state Trellis Code interpretation
- Achieving Finite Decoding Delay
- Latency Budgets
- Simulation Results
- Conclusions

Goals

- **Allow for conventional DFE/DFSE implementations.**
- **Use finite-length decoding**
  - facilitates use of ISI Cancellers that require dual decoders
- **Based on 4D encoding**
  - facilitates reuse of PCS principles from 100BASE-T2
- **3dB/6dB Transparency**
  - Achieve 3dB nominal coding gain with symbol decoding
  - Achieve 6dB nominal coding gain with sequence estimation.
  - facilitates use of "good" decisions for fast feedback loops such as timing recovery.

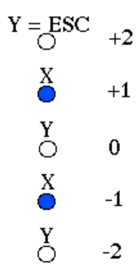
Current 4D and 24D coding schemes

- **4D 8-state Trellis Code**
  - allows for conventional DFE/DFSE implementations
  - requires long decoding delays especially when used without Decision Feedback Sequence Estimation
  - based on 4D code subsets
  - No Transparency
  - Meets Goal#1 and Goal#3
- **24D Code**
  - difficult to do DFE/DFSE implementations for Baseband line codes
  - allows for finite length (6 BT) decoding
  - based on 8D code subsets
  - No Transparency
  - Meets Goal#2

Transparent 4D 4-state Trellis Code

- **Meeting Goals:**
  - Allows for conventional DFE/DFSE implementations
  - Achieves 8BT decoding delay (4BT in Block mode) with full 6dB nominal coding gain.
  - Based on 4D code subsets
  - Achieves 3dB coding gain with symbol decoding and 6dB coding gain with sequence estimation.
  - Meets Goal#1, Goal#2, Goal#3 and Goal#4.
- **Reduces complexity of Per-survivor processing by a factor of > 4 over 4D 8-state trellis code.**
  - reduces power consumption by ~250mW.
- **Provides Error Detection capability for symbol-by-symbol decoded receivers.**

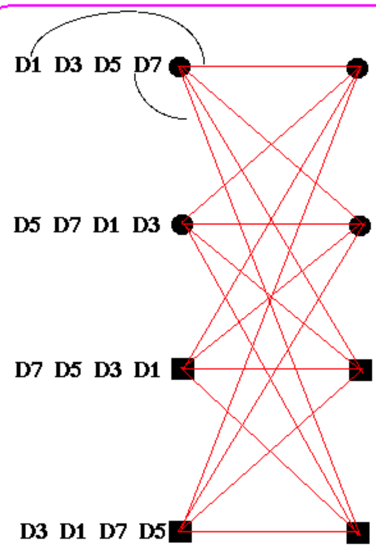
The 4D ODD subsets



Subset	X-Primary Code (Pair A first)	Y-Primary Code (Pair A first)
D1	XXXY	YYYX
D3	XXYX	YYXY
D5	XYXX	YXYY
D7	YXXX	XYYY

- **All four ODD subsets are pair permutations of each other.**
- **Each ODD subset has 24+54 = 78 code points**
- **Each ODD subset has either**
  - one X point and 3 Y points, or
  - one Y point and 3 X points
- **Each ODD subset is uniquely determined by its "unlike" pair.**

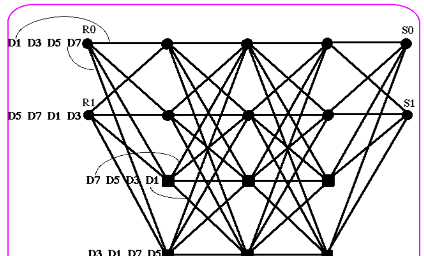
The Trellis Diagram - Single Stage



- Permanent Nodes (shown as circles)
- Nodes that are discarded once every 4 stages (shown as squares)

- **The 4-state trellis collapses to two states once every 4 stages.**

The Trellis Diagram - Four Stages



- **This four-stage segment is repeated over the data packet duration.**

The Data Encoding Method

- **Encoding is done in blocks of 4 Bytes.**
  - 7-bits select subset pattern for the 4-stage trellis
  - 25-bits select a point within the 4 subsets.
- **Designate one point in each subset as Delimiter point and code 25 bits using the other 77 points on 4 subsets.**
  - No. of points needed = 2^{25} = 33,554,432
  - No. of points available = 77^4 = 35,153,041
- **Ensures 3dB coding gain with symbol-by-symbol decoding since only ODD subsets are used.**

Partitioning an ODD subset

- **Each ODD subset with 78 points can be partitioned as:**
  - S-ESC Code points (Points with zero or one ESC symbols) = 64 points
    - Points with No ESC symbol = 32 points
    - Points with 1 ESC symbol = 32 points
  - M-ESC Code points (Points with two or more ESC symbols) = 13 points
    - Points with 2 ESC symbols = 12 points
    - ESC, ESC, ESC, -1, a 3-ESC point = 1 point
  - Delimiter Code point
    - ESC, ESC, ESC, +1, a 3-ESC point = 1 point
- **Delimiter Code point is not used for data transmission.**

Mapping 25bits to Symbols

- **Case 1: BIT[24] = 0: (Only S-ESC points)**
  - Code BIT[23:0] using S-ESC code points on the 4 subsets.
- **Case 2a: BIT[24] = 1, AND BIT[23:20] <= 12: (One M-ESC subset)**
  - Use BIT[23:20] to code one of 13 M-ESC point in M-ESC subset.
  - Use BIT[19:18] to determine position of M-ESC subset.
  - Use BIT[17:0] to code S-ESC points in other three S-ESC subsets.
- **Case 2b: BIT[24] = 1, AND BIT[23:20] >= 13: (Two M-ESC subsets)**
  - Use BIT[23:19] to determine one of 6 combinations of 2 M-ESC subset positions.
  - Use BIT[18:12] to code one of 128 points from 12X12 2-ESC constellation of the two M-ESC subsets.
  - Use BIT[11:0] to pick S-ESC points in the other two S-ESC subsets.

Mapping 7bits to Subsets

- **Use BIT[31:25] and the previous state bit, PS, to determine subset pattern and New state bit, NS.**
- **NS = BIT[31]**
- **PS is set to 0 during Idle and at End of Packet.**

	Index, 2	Index, 1	Index,0	Trellis State
Subset_0	PS+BIT[26]	BIT[25]	1	BIT[26:25]
Subset_1	BIT[28]+BIT[26]+BIT[25]	BIT[27]+BIT[26]	1	BIT[28:27]
Subset_2	BIT[30]+BIT[28]+BIT[27]	BIT[29]+BIT[28]	1	BIT[30:29]
Subset_3	BIT[30]+BIT[29]	BIT[31]+BIT[30]	1	0 BIT[31]

Note: the recovered BIT[31:25] at the receiver can be read off from the states of the best surviving path.

- **This mapping corresponds exactly to the 4-stage trellis on Page 8.**

Error Detection in Symbol-by-symbol Decoding

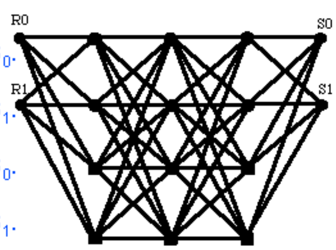
- Assume SUB{0:3}[2:0] are the received ODD subsets in symbol-by-symbol decoding for the current block.
- Let PS be the previous state bit (PS=0 during idle).
- Then, BITS[31:25] are recovered as follows:

Recovered Bits	MS Bit/Error?	LS Bit
BIT[26:25]	PS+SUB0[2]	SUB0[1]
BIT[28:27]	BIT[26]+BIT[25]+SUB1[2]	BIT[26]+SUB1[1]
BIT[30:29]	BIT[28]+BIT[27]+SUB2[2]	BIT[28]+SUB2[1]
{Error? BIT[31]}	BIT[30]+BIT[29]+SUB3[2]	BIT[30]+SUB3[1]

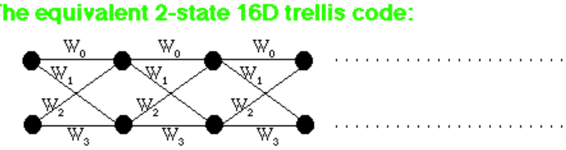
- The New State Bit, NS = BIT[31].

Equivalent 2-state 16D Trellis code

- Define 16D subsets as follows
  - W_0 comprises all paths from R_0 to S_0.
  - W_1 comprises all paths from R_0 to S_1.
  - W_2 comprises all paths from R_1 to S_0.
  - W_3 comprises all paths from R_1 to S_1.



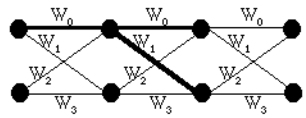
- MSED between W_i and W_j for i != j is 2.
- MSED between points in W_i is 4 for all i.
- The equivalent 2-state 16D trellis code:



The finite decoding delay follows.

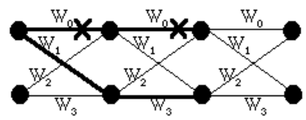
Finite Decoding delay

- Case 1: Both paths have merged at previous node



  - Previous 16D symbol has been decoded

- Case 2: Both paths have not merged at previous node



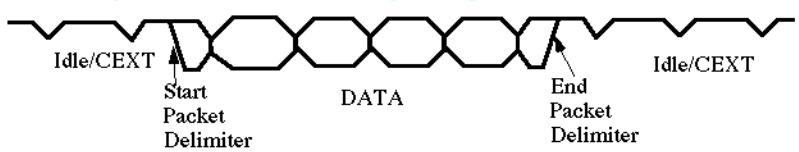
The worse path is discarded since it is distance of >=4 away from winning path.

  - Both 16D symbols have been decoded!!

Data Packet Encapsulation Options

- 4-byte Block Trellis Encoding starts at Start Packet Delimiter.
  - Terminated to state 0 at End Packet Delimiter
  - Idle uses only D1 subset code points
  - Data uses all ODD subset code points

- 4-byte Block Trellis Encoding throughout Idle/Data



Latency Budgets

- **ISI Canceller Solution (Rockwell presentation from Maui)**
  - Encoding delay stays the same (?)
  - Decoding delay for F1 is 8BT.
  - Decoding delay for F2/F3 is 4BT assuming block processing.
  - Overall latency stays the same as with 24D code.
- **Conventional DFE Solution**
  - Encoding delay increases by 4BT.
  - Decoding delay decreases by 4BT.
  - Overall latency stays the same with respect to current 4D code.

Simulation Results

- **Evaluation with Ideal DFE and AWGN**

Noise Level (dBs)	SER@Input (4D Symbols)	SER@Output (Conventional 4D 8-state Trellis Code)	SER@Output (Transparent Code)
-7.0dB	0.093	0.001	0.0013
-8.0dB	0.043	5E-5	8E-5
-9.0dB	0.016	~1E-6	~1E-6

- **Evaluation with Full Per-Survivor Processing and AWGN**

Noise Level (dBs)	SER@Input (4D Symbols)	SER@Output (Conventional 4D 8-state Trellis Code)	SER@Output (Transparent Code)
-7.0dB	0.093	0.012	0.004
-8.0dB	0.043	8E-4	4E-4
-9.0dB	0.016	1E-5	5E-6

Conclusions

- 4D 4-state trellis code combines the best properties of conventional 4D 8-state trellis code and 24D block code (for passband systems).
  - 3dB coding gain with symbol by symbol decoding
  - 6dB coding gain with sequence estimation
  - 8BT decoding delay for full 6dB gain
  - 4BT decoding delay in block mode operation
- Reduces complexity of per-survivor processing due to
  - reduced number of states (4 vs. 8).
  - automatic pruning of survivor paths to 2 once every 4 stages.
  - reduces power consumption of implementation.
- Built-in Error Detection capability for symbol by symbol decoded receivers.