100BASE-Cu

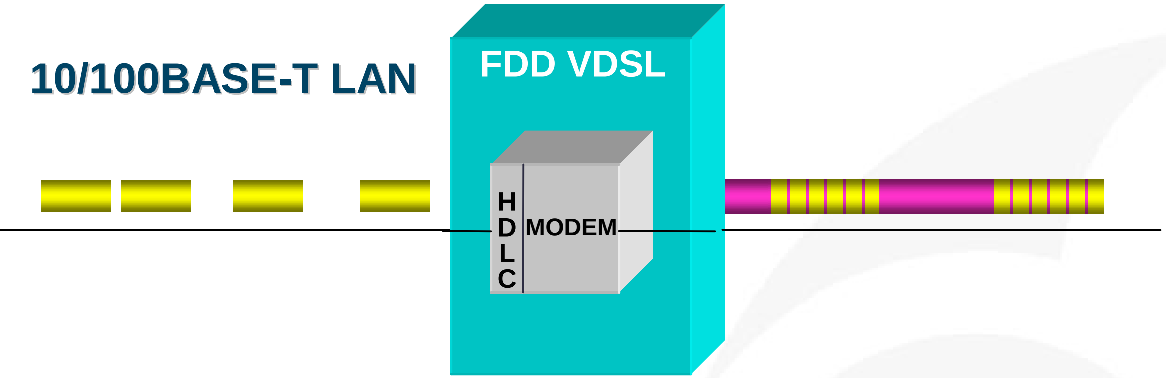
Dual Mode

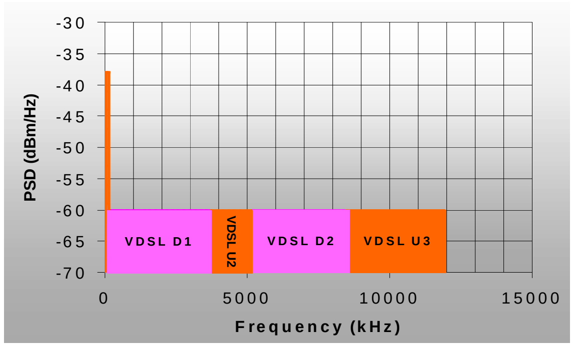
Layer Model

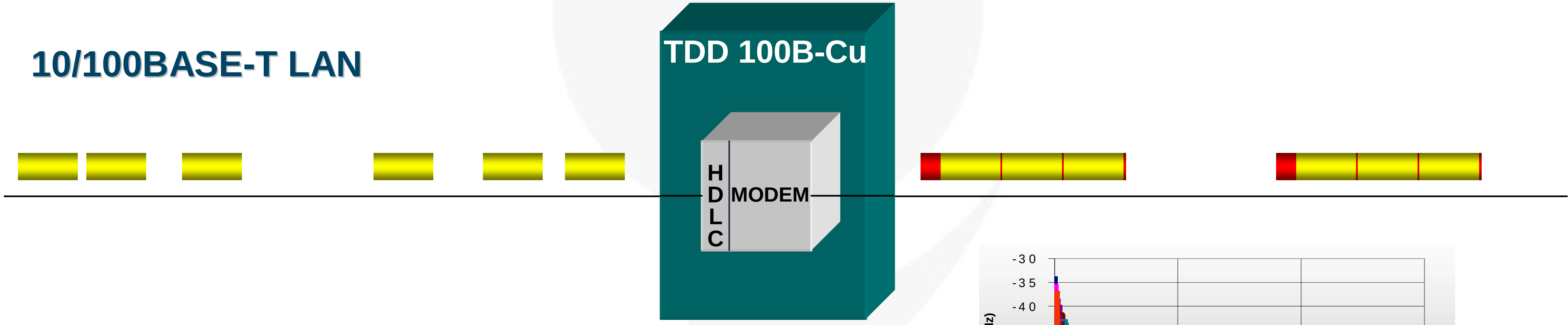
Patrick Stanley, Elastic Networks, pstanley@elastic.com (678)297-3103

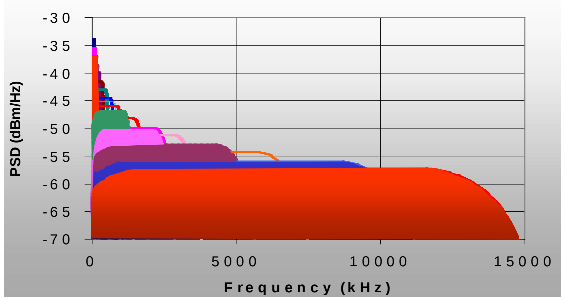
Co-Sponsors of This Presentation

100BaseCu *Dual - Mode* Component Technologies

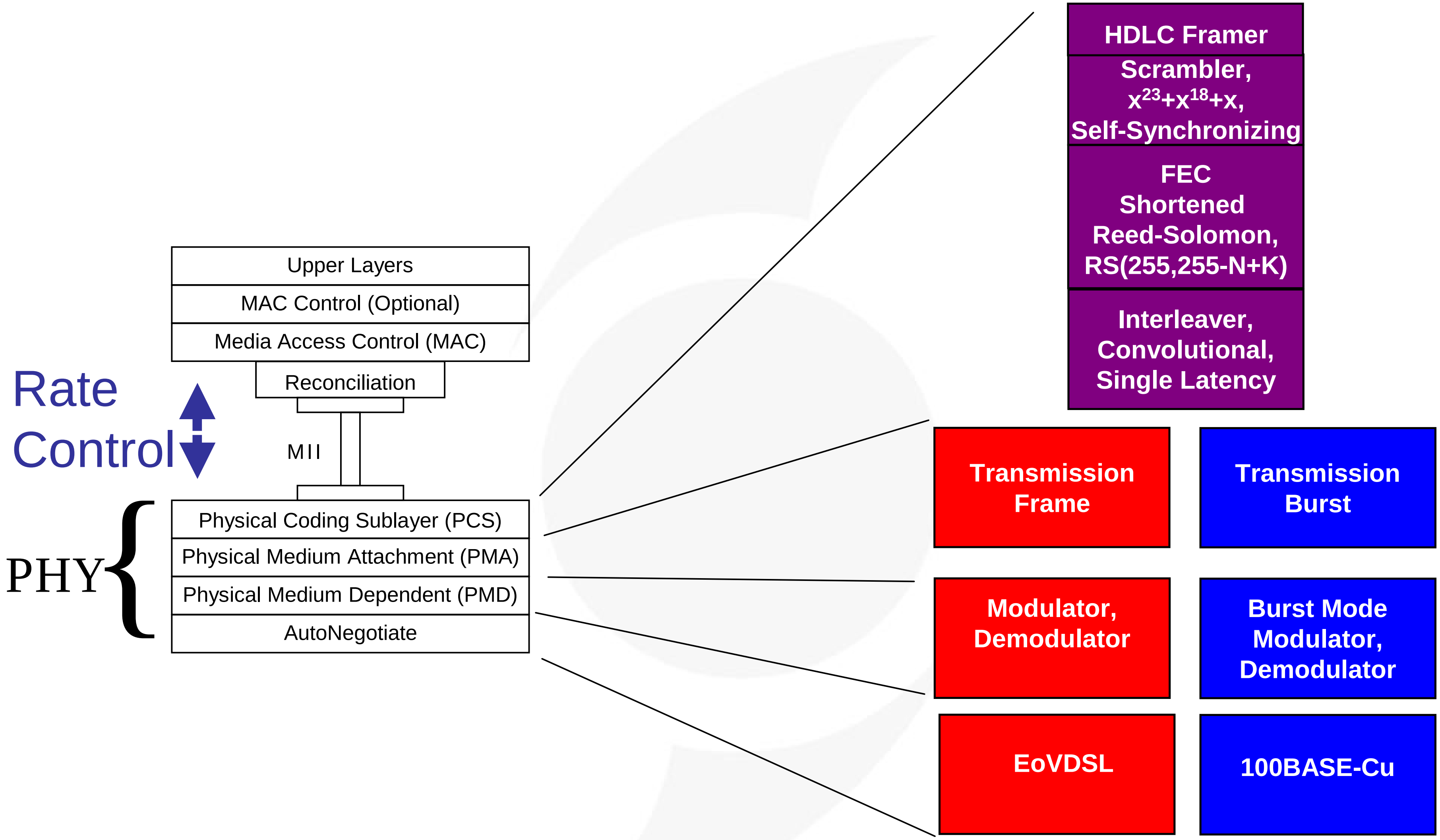






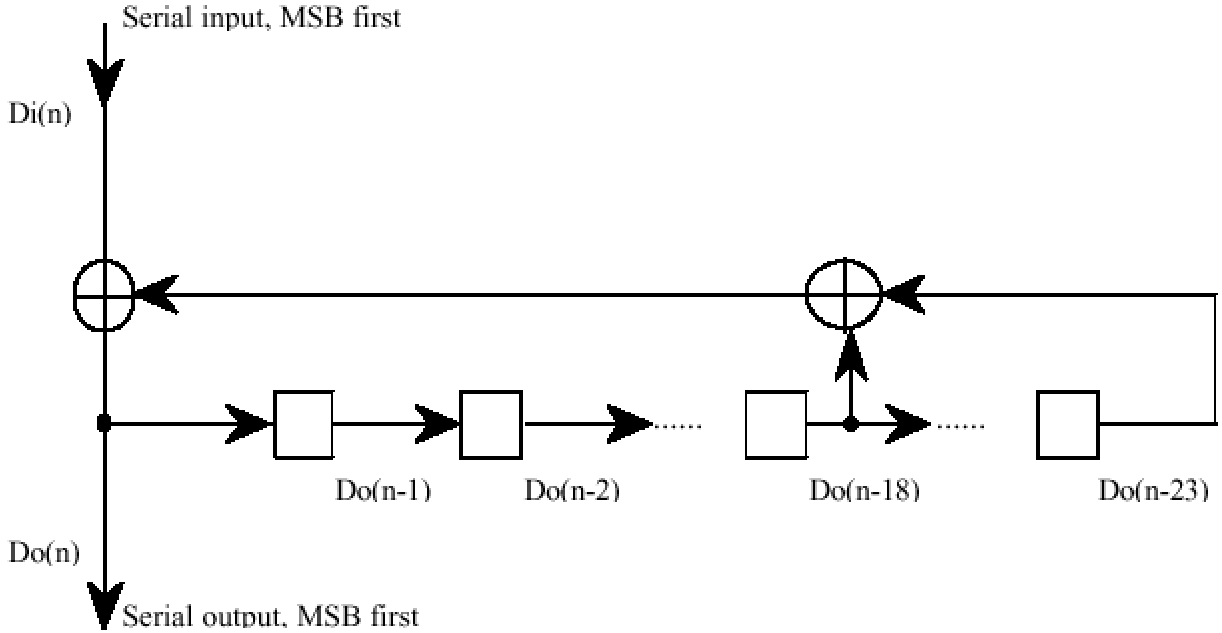


100BASE-Cu Dual Mode Layer Model



Scrambler, $x^{23}+x^{18}+x$

Self Synchronizing



FEC

Shortened Reed Solomon RS(255,255-N+K)

Codeword Length, N = 200 Octets, Data Length, K = 184 Octets => 16 Redundancy Octets = Can correct a a maximum of 8 Octets per block

Convolutional Interleaver, Single Latency

S – incoming codeword length

I – interleaver block length, octets

D – interleaving depth, octets

M – interleaving depth index

Parameter	Value	Notes
Block Length (I)	I = S/8, S/4 or S/2, octets	S = 200
Depth(D)	D = M x I + 1, octets	M = 0 - 64, programmable
Erasure Correction (E)	E = floor[t x I/S] x (M x I + 1), octets	t = 8 (RS error correction ability)
End-to-End Delay (DL)	DL = M x I x (I-1), octets	
Interleaver Memory Size	MEM = M x I x (I-1)/2, octets	

D-1 octets separate any 2 sequential octets in RS codeword

E sequential octets correctable, so noise pulse of duration Ex8/R, where R = bit rate, is protected against.

Examples at 100Mbps:

- Minimum protection: M = 0 disables interleaver, 0.64uSec protection
- Maximum protection: M = 64, I=S/2, 2mSec protection.
  - Costs: 50mSec Latency, 512KB of memory

100BASE-Cu Dual Mode “AutoNegotiate”

Proposed Operational Modes (others may be suggested)

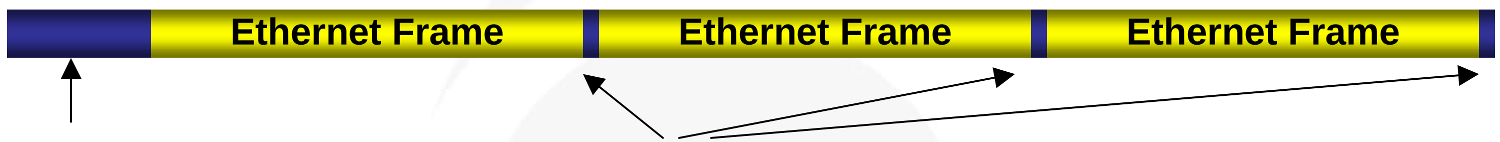
- Auto-Sense
  - Operates in Dynamic TDD mode, with Spectrum Manager listening for coupling with 988 plan
  - When coupled with 998 plan, switches to EoVDSL-like FDD, with Spectrum Manager listening for opportunity to switch back to dynamic TDD
  - Additional provisionable option can limit Upstream frequency range to favor asymmetric service, heavier on downstream
- Provisionable – Service Provider can provision to EoVDSL like FDD mode or TDD only mode

100BASE-Cu Transmission Burst

HDLC Frames concatenated into burst

100BASE-Cu Burst:

31 1518 Byte Frames per Burst Max



Preamble: 135 Bytes

Header, CRC, Flag: 7 Bytes

Preamble contains BPSK for timing recovery

HDLC inserts ‘0’ bits

“Polling” packets, which report buffer fill, piggy back onto user data bursts

Dynamic Symmetry Mechanism

- **FIFO builds burst from incoming packets in each direction**
- **If user traffic demand is symmetric, then both directions use full bursts – symmetric bandwidth on loop (if provisioned for symmetry service)**
- **If user traffic demand is heavily in one direction, that direction has full bursts, and return path has minimum bursts containing ACKs and status information**

100BASE-Cu Power On Training (POT)

- First Speed Trained is 62.5kbaud, QPSK
- This 125kbps speed is available on all loops
- Idle modems "fall back" to this speed if no user data is present
- After training POT speed, loop loss is measured. The next speed trained is ½ the maximum speed projected for measured loop loss
- Once these 2 speeds are trained, user data can start flowing
- Remaining possible speeds (based on loop loss) are trained in the background through recurrent training, which injects training bursts between data bursts

Loop Condition Metrics

- During transmissions, FEC bit and block errors monitored for current operating speed
- Between data bursts, SNR is measured at current, and other available speeds

Speed Change Description

If < up_traff_threshold packets are in the transmit FIFO, then transmission continues at POT speed

If up_traff_threshold is tripped, then speed change is initiated to highest available speed, if measured SNR at that speed looks promising

Data transfer continues at maximum available speed

If < down_traffic_threshold packets are in the transmit FIFO for fall_back_delay consecutive seconds, then a speed change is initiated to the POT speed

If FEC bit and block errors > BER_max, then a speed change is initiated to the highest lower speed whose SNR > SNR_min

If SNR of current operating speed < SNR_min, then a speed change is initiated to the highest lower speed whose SNR > SNR_min

If current speed has been in use for jump_up_delay seconds, and a higher speed has SNR > SNR_min, then jump up to the higher speed

Speed Change State Diagram