Option to bypass error marking
(supporting comment #205)

Adee Ran, Intel

Oren Sela, Mellanox

IEEE P802.3bj 100 Gb/s Backplane and Copper Cable

January 2013, Phoenix

Supporters

- Andre Szczepanek, Inphi
- Stephen Bates, PMC-Sierra

Reminder: D1.2

91.6.1 FEC_bypass_correction_enable

When this variable is set to one the Reed-Solomon decoder performs error detection without error correction (see 91.5.3.3). When this variable is set to zero, the decoder also performs error correction. This variable is mapped to the bit defined in 45.2.1.93a (1.200.0).

91.6.2 FEC_error_indication_enable

This variable is set to one to enable indication of decoding errors to the PCS sublayer (see 91.5.3.3) when this feature is supported. When set to zero, the error indication function is disabled. This variable is mapped to the bit defined in 45.2.1.93a (1.200.1).

91.6.3 FEC_bypass_correction_ability

The Reed-Solomon decoder may have the option to perform error detection without error correction (see 91.5.3.3) to reduce the delay contributed by the RS-FEC sublayer. This variable is set to one to indicate that the decoder has the ability to bypass error correction. The variable is set to zero if this ability is not supported. This variable is mapped to the bit defined in 45.2.1.93b (1.201.0).

91.6.4 FEC_error_indication_ability

The RS-FEC sublayer may have the option to indicate decoding errors to the PCS sublayer by intentionally corrupting 66-bit block synchronization headers as defined in 91.5.3.3. This variable is set to one to indicate that the RS-FEC sublayer has the ability to indicate decoding errors to the PCS sublayer. The variable is set to zero if this ability is not supported. This variable is mapped to the bit defined in 45.2.1.93b (1.201.1).

D1.2 allowed 4 modes

Mode	Correctable Errors	Uncorrectable Errors	Latency	MTTFPA
A (?)	Correct	Mark	Baseline+ ~140 ns	Sufficient
Bypass	Pass through	Pass through	Baseline	Too short
Correct	Correct	Pass through	Baseline+ ~90 ns	Depends on UCR
Detect	Mark	Mark	Baseline+ ~50 ns	Sufficient

(Following and extending the nomenclature suggested by Zhongfeng Wang)

Latency column assumes hardware-saving implementation (no parallelization of final error marking logic).

Only 3 possible modes in D1.3

Mode	Correctable Errors	Uncorrectable Errors	Latency	MTTFPA
A (?)	Correct	Mark	Baseline+ ~140 ns	Sufficient
Bypass	Pass through	Pass through	Baseline	Too short
~~**Correct**~~	~~Correct~~	~~Pass through~~	~~Baseline+ ~90 ns~~	~~Depends on UCR~~
Detect	Mark	Mark	Baseline+ ~50 ns	Sufficient

- Mode C (correct errors without marking uncorrectable errors) is illegal in D1.3. Error correction is thus possible only in mode A.
- Mode A is safer than mode C in terms of MTTFPA… but mode B, which is the most unsafe, is still allowed!
- If the reason for disallowing mode C is the reduced MTTFPA, then mode B can't be allowed either

Hardware / latency tradeoff

- A parallelized implementation may reduce latency for mode A (and also for mode C) but at a hardware cost:
  - Error marking can be implemented with negligible latency (mode A ≈ mode C) with a 200% increase in gate count of RS-FEC decoder, compared to minimal implementation (Intel and Mellanox estimates).
  - Minimal RS-FEC decoder is at least 5% of the gate count of a full-blown NRZ PHY (estimates vary)
  - Therefore, a minimum-latency implementation of mode A would add at least 10% to the PHY gate count.
- Choice is done by PHY vendor at design time and affect all links!

Analysis of correction without marking (mode C)

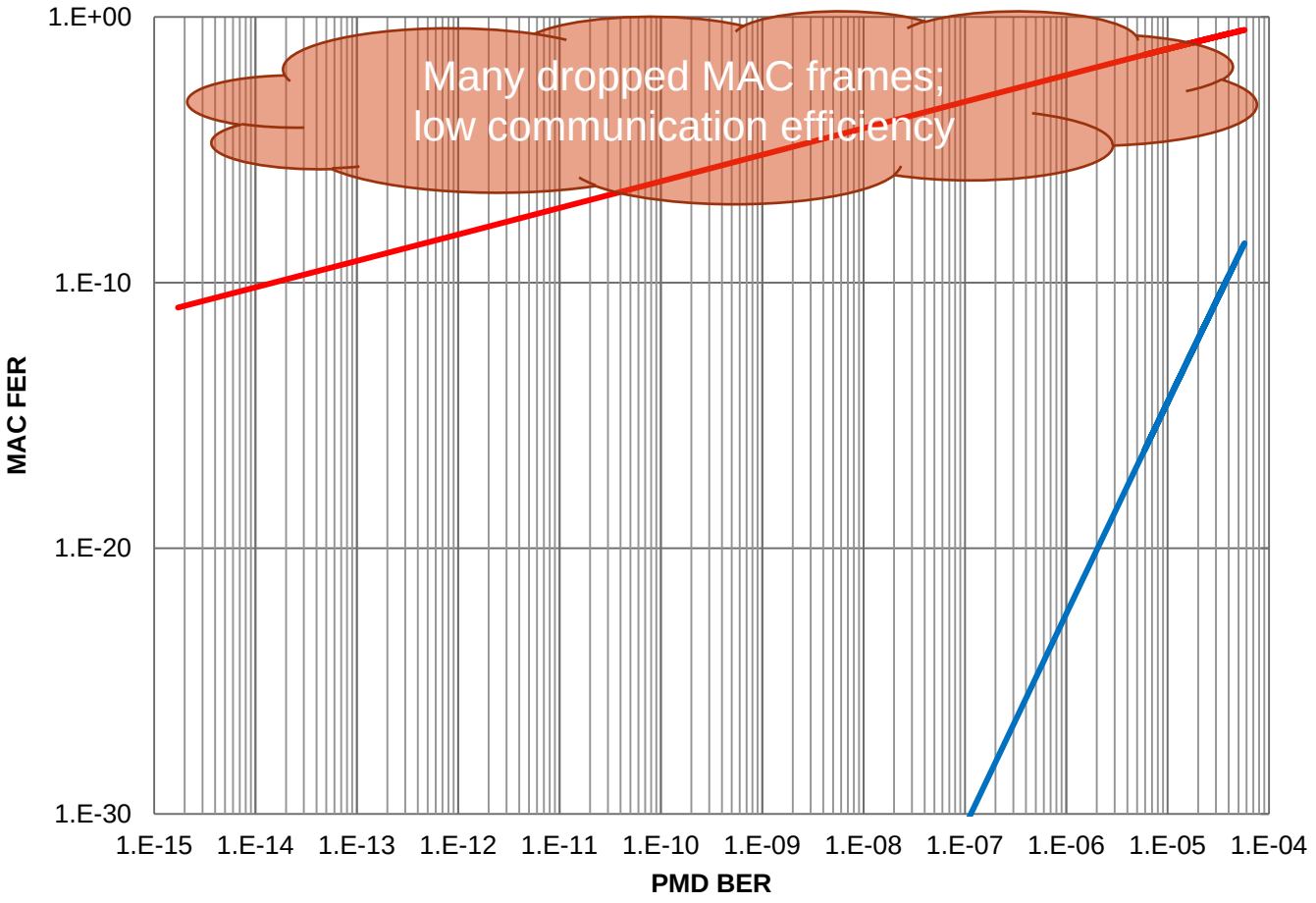
- With PMD BER of 4e-5, FER target of 1.7e-10 is obtained, but MTTFPA≈5K years
- However, to improve MTTFPA 10-fold, PMD BER should only be improved slightly, to ~3e-5
  - This relates to ~0.15 dB SNR improvement (AWGN assumption)
  - Each additional 0.15 dB improves x10
- Improving PMD BER to 6.1e-6 brings MTTFPA to ~AOU (13G years)
  - This requires only ~0.89 dB SNR improvement
  - Also creates a virtually error-free link (FER<1e-16)

Calculations assume CR4/KR4 RS-FEC, with highly correlated errors (DFE C=1).
Using C=0.1 results in ~2x higher MTTFPA .

Comparison to other modes

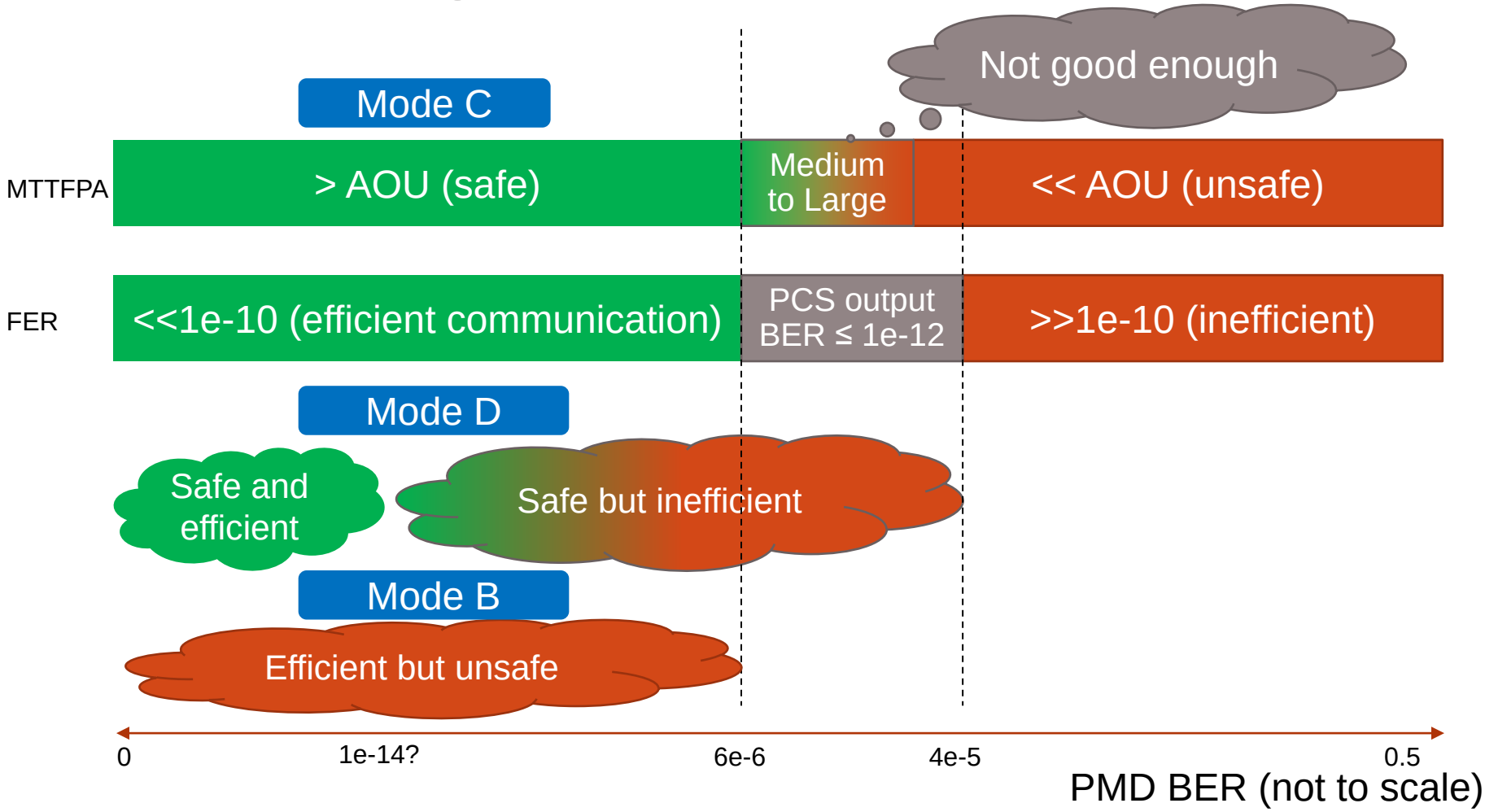
- If **mode B** (bypass FEC) is used instead with the same PMD BER, MTTFPA is ~minutes
  - Just lowering PCS BER to <1e-12 requires ~5.25 dB SNR improvement! And even that was shown to have MTTFPA << AOU
  - This PCS BER will only make FER ≈ 2e-9, far from being error free
- If **mode D** (Detect and mark errors) is used instead with the same PMD BER, MTTFPA is probably completely safe
  - But FER=0.24 ➔ a useless system…
  - An order of magnitude improvement in FER costs ~1.1 dB SNR improvement
  - Getting to FER<1.7e-10 requires PMD BER to drop to 3e-14, or 5.6 dB SNR improvement!
- Safe and efficient operation in modes B and D may be feasible only in non-typical cases (e.g. “engineered links”)

MAC FER vs. PMD BER



Note:
Mode B FER is somewhat better than mode D, but this mode is unsafe for most of the BER values in this chart.

Work regions

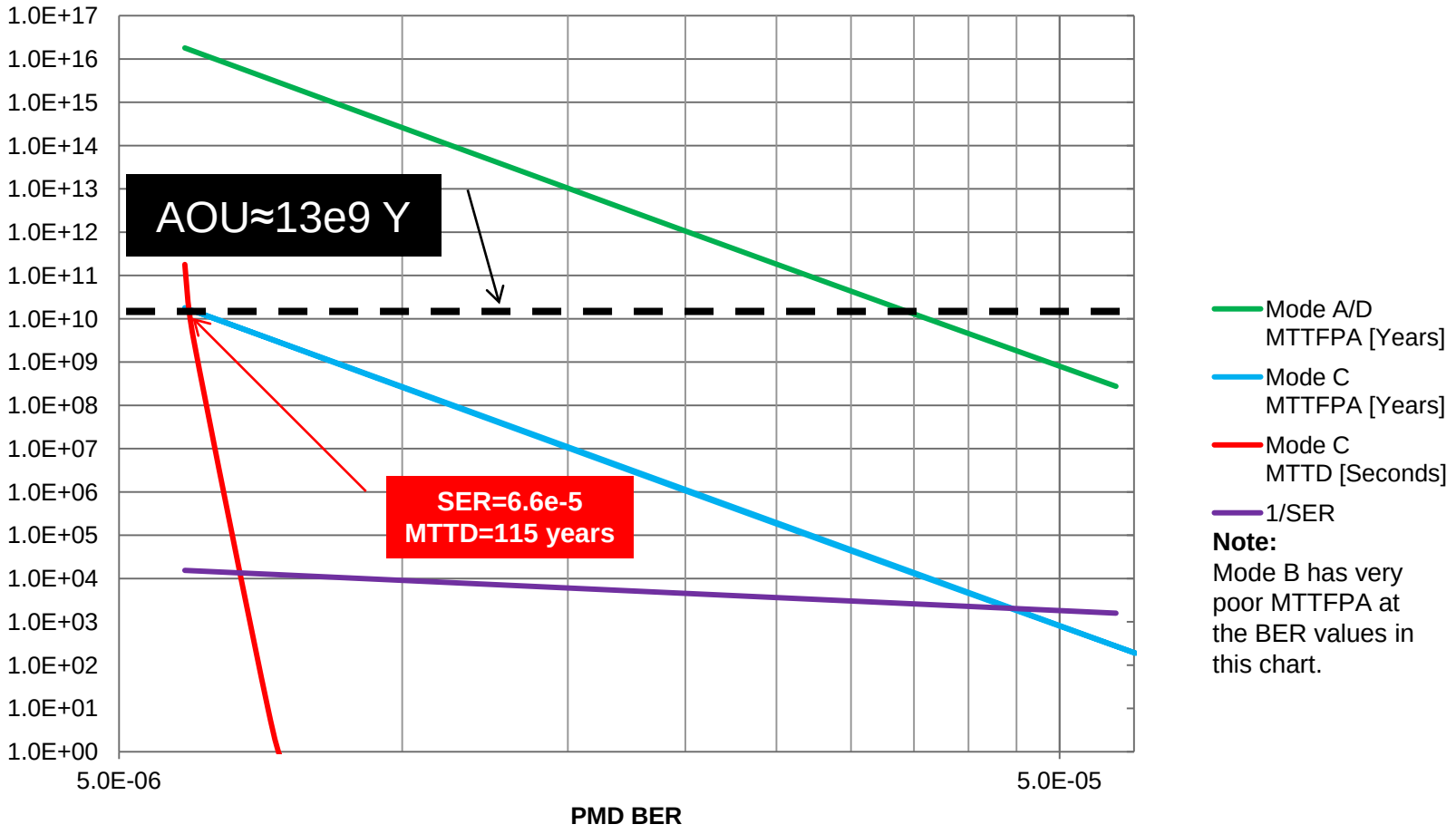


Solution for “Gray area” in mode C

- There is a severe degradation of MTTFPA in a small PMD BER region which still meets MAC FER requirements. In fact, we need the MTTPFA to hold up even if the FER is below spec.
- **Suggested remedy:**
  - Use an estimated symbol error rate (SER) to predict MTTFPA in mode C.
  - Count symbol errors in a counter; reset the counter after every 2^{13} codewords (~419 microseconds). (in addition to the counters 91.6.7)
  - **In mode C only, if the error counter exceeds a threshold K – start corrupting sync headers for a period of 60 to 75 ms**. That would cause the BER monitor to quickly assert hi_ber; AN would restart the link, and prevent the risk of false packet acceptance.
  - Suggested value: K=417*, which creates a sharp distinction between good and bad links:
    - For a link with MTTFPA of 15e9 years (SER=6.6e-5), mean time to disconnect (MTTD) is >100 years
    - For a link with MTTFPA of 2.5e9 years (SER=8.3e-5), MTTD is ~0.4 second
    - See backup for calculations
- With this solution, choice between mode A and mode C can be made per case – no global hardware or latency cost!

* Threshold is calculated using number of symbols per codeword and the maximum allowed SER for CR4/KR4 RS-FEC. If implemented with KP4, a different threshold would be required.

MTTFPA and MTTD vs. PMD BER



Summary

- **Mode C** is the safest and most efficient of the “optional modes”; channels with better than minimum performance (PMD BER slightly lower) can achieve an error-free and MTTFPA-safe link. Link health can be monitored in the RS-FEC sublayer and be used to drop unsafe links via hi_ber.
- **Mode A** provides MTTFPA safety in bad SNR cases (even when BER objective is not met), at the cost of additional latency.
- In **Mode D**, MTTFPA safety is guaranteed and latency is reduced; but BER/FER requirements may be difficult to meet.
- **In Mode B**, meeting the BER/FER objective requires extremely good SNR, which may not be feasible on most systems; even then, MTTFPA is likely too small.

Mode C should be allowed. Mode B should not.

Proposal

- **Allow error marking to be optionally bypassed** (capability declared and controlled similar to FEC_correction_bypass)
  - This is mode C, optional to implement
- **Supply a safeguard for MTTFPA in this mode**
  - If more than 417 symbol errors occur in 2^{13} codewords, start corrupting all sync headers (➔ hi_ber assertion) for a period of 60 to 75 ms
    - PHYs with AN will drop link and restart within that period
    - PHYs without AN will re-lock and restore normal operation after that period – to enable link recovery (can re-use break_link_timer if implemented)
    - New counters, timer, and state machine definitions are required
  - Note that the current hi_ber trigger is BER monitor based on sync headers, which is not enough to guarantee long MTTFPA with RS-FEC
- **Prevent bypassing both correction and marking (mode B)**
  - If both options are supported and enabled, normative behavior is mode D
  - If products implement mode B, it would be in a non-standard way, not supported by IEEE, and users choose it at their risk.

Comments submitted on D1.3

- Comment #205 by Adee Ran – the proposal in this presentation enhances the originally suggested remedy
- Comment #18 also addressed
- Comments #110 and #241 effectively addressed, if additional hi_ber condition is implemented

Backup

MTTD calculation

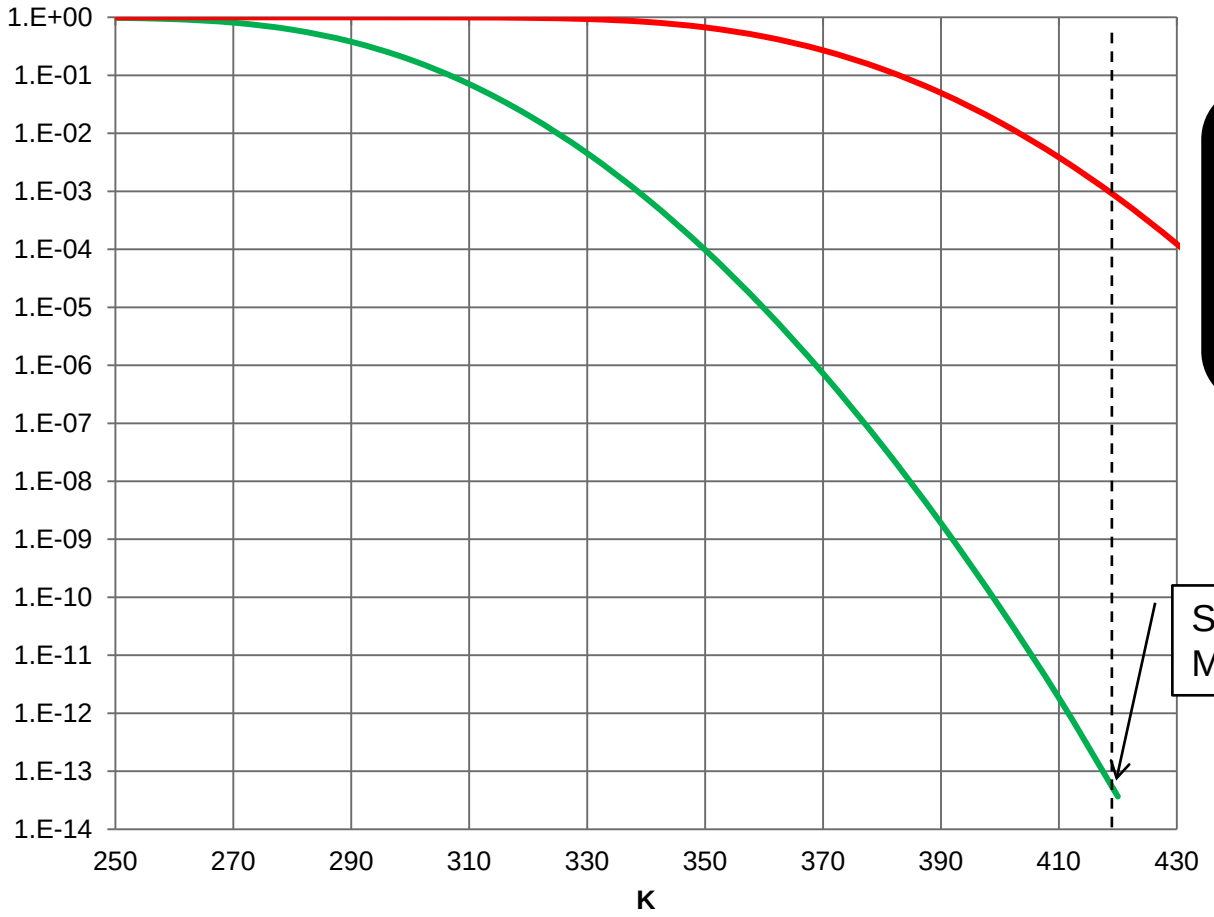
- To get MTTFPA of ~AOU, we need SER<6.6e-5
- The number of symbol errors in 2^{13} codewords (a period T≈419 microseconds) has a Binomial distribution ~B(n,SER) with $n=528*2^{13}$; probability of having exactly k errors in this period is

$$P(x=k)=\binom{n}{k}SER^{k}(1-SER)^{n-k}$$

- The cumulative distribution (probability of having *up to K* errors) can be calculated using analytical methods (Excel function BINOMDIST is one implementation). It is very sensitive to SER as shown in the next slide.
- MTTD is obtained from

$$MTTD(K)=\frac{T}{1-P(x\leq K)}$$

Cumulative Binomial distribution sensitivity to SER



Graph shows the probability of having more than K symbol errors in 2^{13} FEC codewords (~0.4 ms), for two values of SER.

Prob(x>K | SER=6.6e-5)

Prob(x>K | SER=8.3e-5)

Suggested threshold yields MTTD>100 years

Link restart through Autoneg

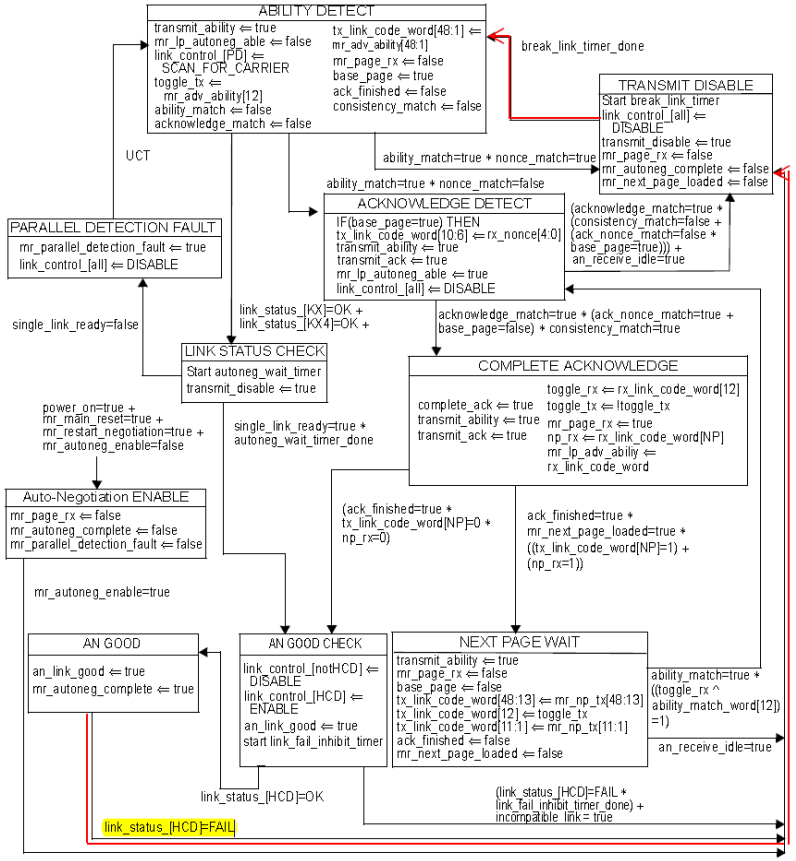


Figure 73–11—Arbitration state diagram