Radiation Tolerance and Information Theory

K. Andrews, D. Divsalar, S. Dolinar, F. Pollara

Jet Propulsion Laboratory
California Institute of Technology, Pasadena CA, USA

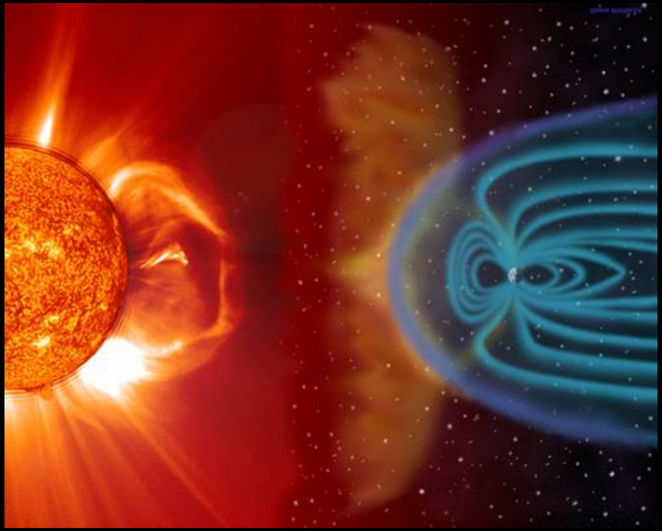
Background

- Several future space missions will operate in medium to high radiation environments
  - They will require technologies or strategies to extend their capabilities and obtain reliable operation in these extreme environments
- The current state-of-the-practice of radiation hardening technology is reminiscent of the state of communication technology before Claude Shannon illustrated the principles of information theory and coding for communication
  - It was then believed that the only way to increase communication range, rate, or reliability was to increase the transmitter power or antenna gain
  - Rudimentary repetition codes had been suggested but they didn't provide an effective overall improvement
- Shannon's idea was instead to operate the system in a regime where it makes many errors and eliminate them through the efficient introduction of controlled redundancy
  - The existence and practical implementation of codes that accomplish this has made an enormous impact on communication systems
  - Our approach to radiation hardening is similar: to devise efficient introduction of redundancy to protect storage and computation devices
- Coding theory provides powerful methods to make storage devices more reliable to both transient errors, caused by single event upsets, and to permanent damage due to massive radiation effects

Europa-Orbiter
- Only 150 Mbits of rad-hard memory are planned
- Severe implications on mission design: transmit data as soon as possible, with high peak data rate
- Improved rad-tolerant methods can revolutionize mission design

Reference: C. E. Shannon. "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, 1948

Faults Affecting Computing Systems



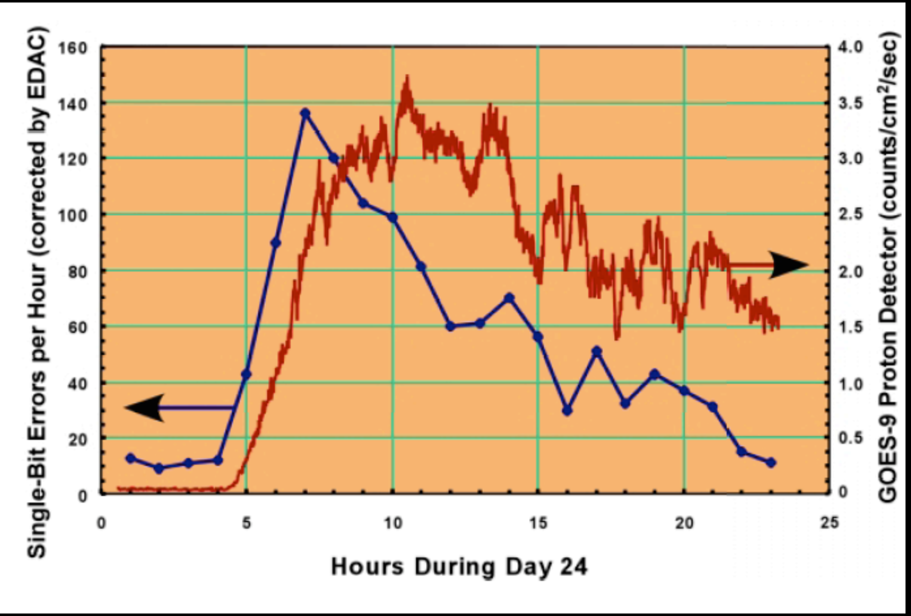
- **Future high-density electronic systems will be even more susceptible to computation and communication noise**
  - **High levels of VLSI integration are generally achieved through a reduction of cell size and of the charge representing data bits. Cells become more sensitive to environmental disturbances leading to higher rates of faults**
- **The sources of noise are not limited to radiation, but also include thermal perturbations, electromagnetic interference and quantum mechanical effects**

- **Two general classes of faults:**
  - **Permanent faults – manufacturing defects, catastrophic radiation effects (Total Dose)**
    - The total dose effects represent cumulative ionization damages
  - **Transient faults – noise,** radiation (Single Event Effects [*SEE*])
    - SEEs are caused by single high-energy ion passing through a device. They include:
    - Single Event Upsets (*SEU*) and Single Event Latchups (*SEL*)
    - SEUs cause soft errors; SELs can be destructive under certain conditions
    - The total dose effects can be reduced by using suitable shields. *SEE* susceptibility is not significantly affected by shielding

- **The effect of faults on computing systems can be analyzed using information theoretic concepts**
  - **Bounds analogous to those found by Shannon for communication systems can be derived**
  - **Error correcting codes can be used to reduce the effect of transient faults**

Faults Affecting Computing Systems

Cassini SSR Errors During Solar Flare



Single-Event Upsets are Real

- First Observed in Bipolar Flip-Flops in 1979
  - Original work treated with skepticism
  - SEU emerged as one of the major issues for application of microelectronics in space
- JPL missions have struggled with SEU problems
  - Galileo's microprocessor was initially susceptible to SEUs at moderate rate
  - Design had to be changed to make device usable
- SEU effects have become worse as devices have evolved
  - Lower "critical charge" because of small device dimensions
  - Large numbers of transistors per chip and overall complexity

SEE Effects in Operational Spacecraft

- Multiple-Bit Errors in Cassini Solid-State Recorder (SSR) occurred even though extensive testing was done during design phase

Methods to Combat Faults

- **von Neumann addressed the issue of computation in the presence of noisy gates in 1952 and developed a technique called multiplexing**
  - **In general, these methods are extremely expensive computationally and not efficient**
    - Traditional fault-tolerant *computational circuits* are designed by using modular hardware redundancy, by replicating the original circuit ***N*** times, and calculating the desired function multiple times in parallel.
    - The outputs of all replicas are then compared, and the final result is chosen using a majority rule.
      [With the usual assumption that the voting mechanism is fault-free]
- **The redundancy factor is the *"ratio of the circuit sizes of the redundant and non-redundant designs"***

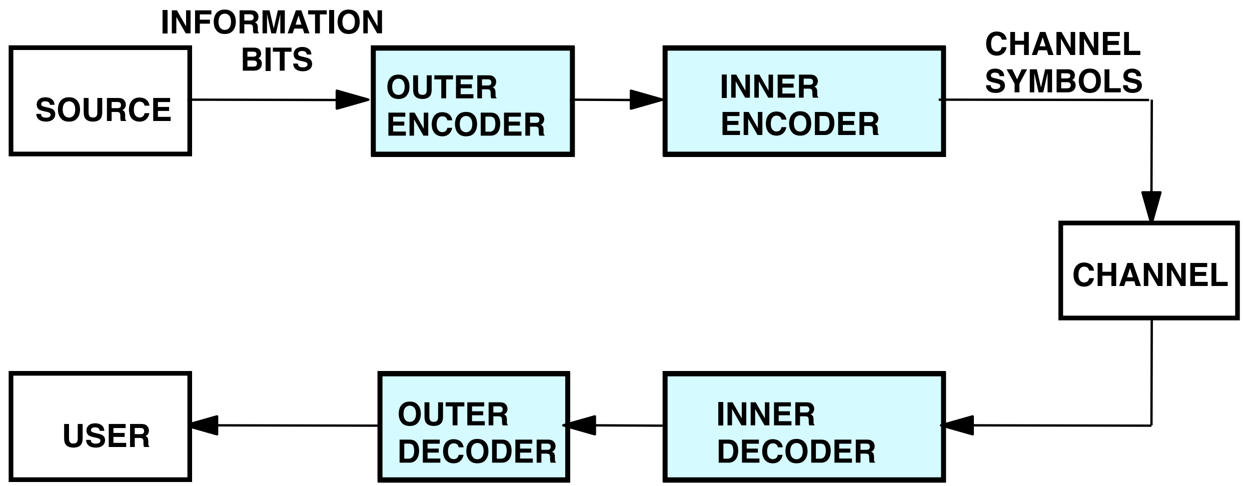
- **Error correcting/detecting codes for memories have been in use for many years and simple redundancy schemes for entire storage device failure are also well known**
  - Error-detection-and-correction (EDAC) algorithms
  - Used in solid-state recorders on many spacecraft
  - Different levels of correction, depending on algorithm
    - Single and double bit detection, with single-bit correction
    - Double bit detection and correction

Example - Hamming Codes
"SECDED" = Single Error Correction Double error Detection
(39, 32) = 32 data bits + 7 parity
"DECTED" = Double Error Correction Triple Error Detection
(79, 64) = 64 data bits + 15 parity

- **Our proposed methods are based on much more powerful error correcting codes**
  - **Their decoding complexity is manageable for very large blocks**
  - **The decoding algorithm can function even if the decoder suffers SEUs**
  - **Optimal scrubbing strategies are developed**

Reference: J. Von Neumann, "Probabilistic logics and synthesis of reliable organisms from unreliable components," Automata Studies, pp. 43-98, 1956

Channel Coding



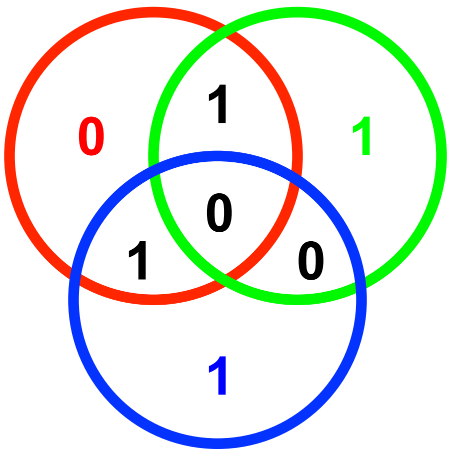
COMMUNICATION SYSTEM MODEL WITH CONCATENATED CODING

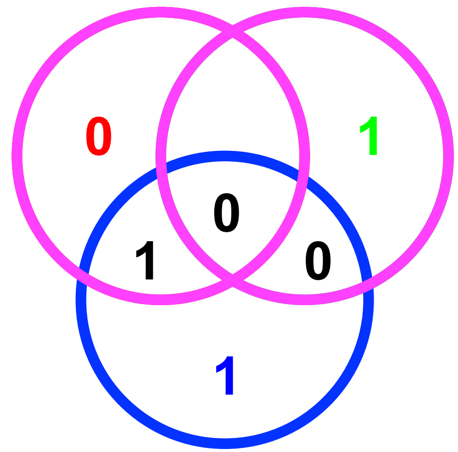
Coding
Gain
2.0 dB   PIONEER 9 (1968) - FIRST DEEP-SPACE ENCODED COMMUNICATION SYSTEM
2.2 dB   MARINERS-VIKING (1969-71)
3.5 dB   VOYAGER (1977)
7.5 dB   VOYAGER (1980) - FIRST CONCATENATED CODING SYSTEM
8.4 dB   CASSINI & PATHFINDER (Very high complexity decoder)

Further improvements seemed beyond reach due to enormous increase in complexity, but …

>9.0 dB   TURBO and LDPC CODES

Example of Error-Correcting Coding: The (7,4) Hamming Code





4 **INFORMATION BITS** ARE PLACED IN THE INTERSECTIONS OF THE VENN DIAGRAM

EACH CIRCLE IS FILLED WITH A "PARITY BIT" TO FORM A 7-BIT **CODEWORD**

SINGLE ERRORS CAN BE CORRECTED BY FINDING THE CIRCLES WITH AN ODD NUMBER OF 1'S AND COMPLEMENTING THE BIT IN THEIR INTERSECTION

Error correction coding reduces the required transmitter power for a fixed data rate (or increases the data rate for a fixed transmitter power) for a desired reliability (residual bit error rate)

This is possible because the redundancy introduced by coding is more than offset by its ability to correct a certain amount of errors

The (7,4) Hamming code can correct only one error in a block of seven symbols!

Much more powerful codes have been used in space missions

Main coding metrics:
- Performance
  - Power efficiency
  - Bandwidth efficiency
- Complexity (decoding speed)
- Latency

History

1947: Hamming Codes

1948: Shannon lays the mathematical foundations of coding

1955: Convolutional Codes

1960: Reed Solomon Codes

[1962: LDPC Codes first described]

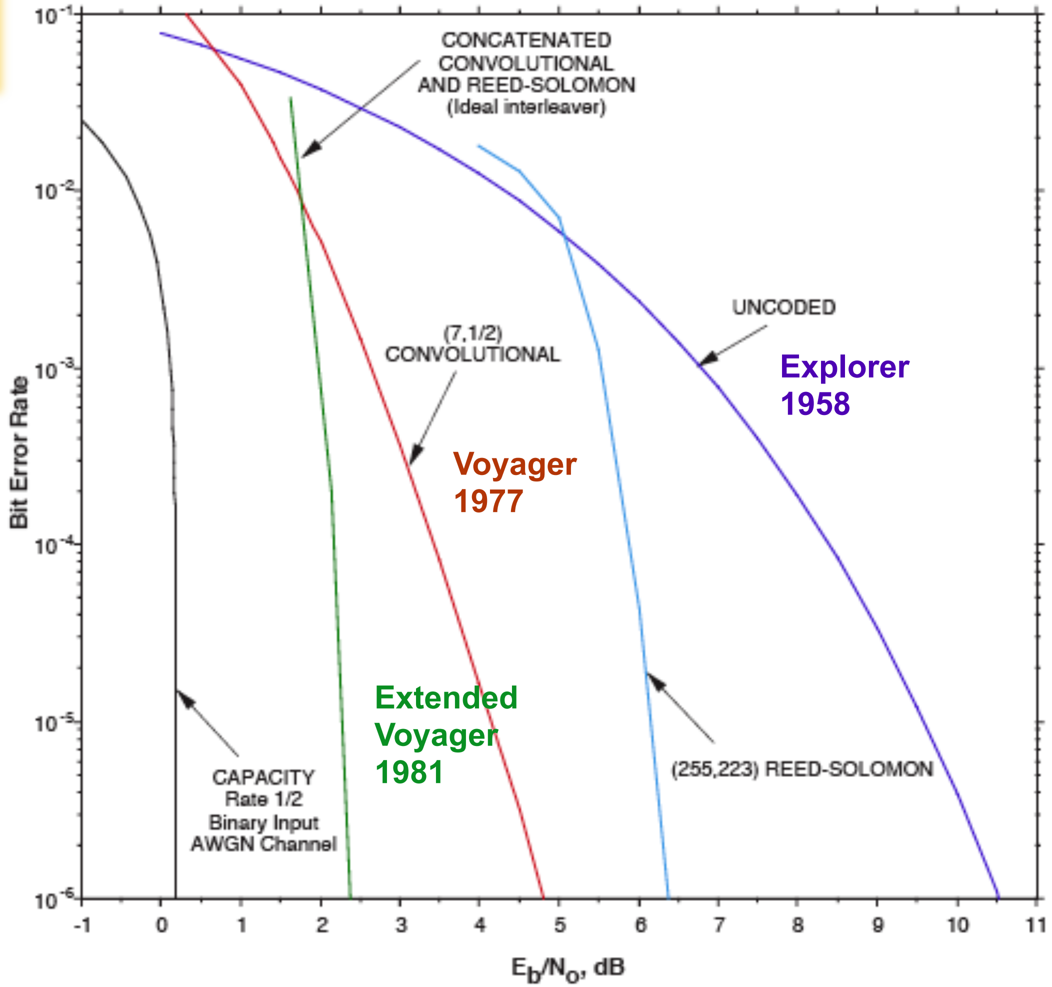
1987: CC/RS concatenated code standardized

The coding revolution:

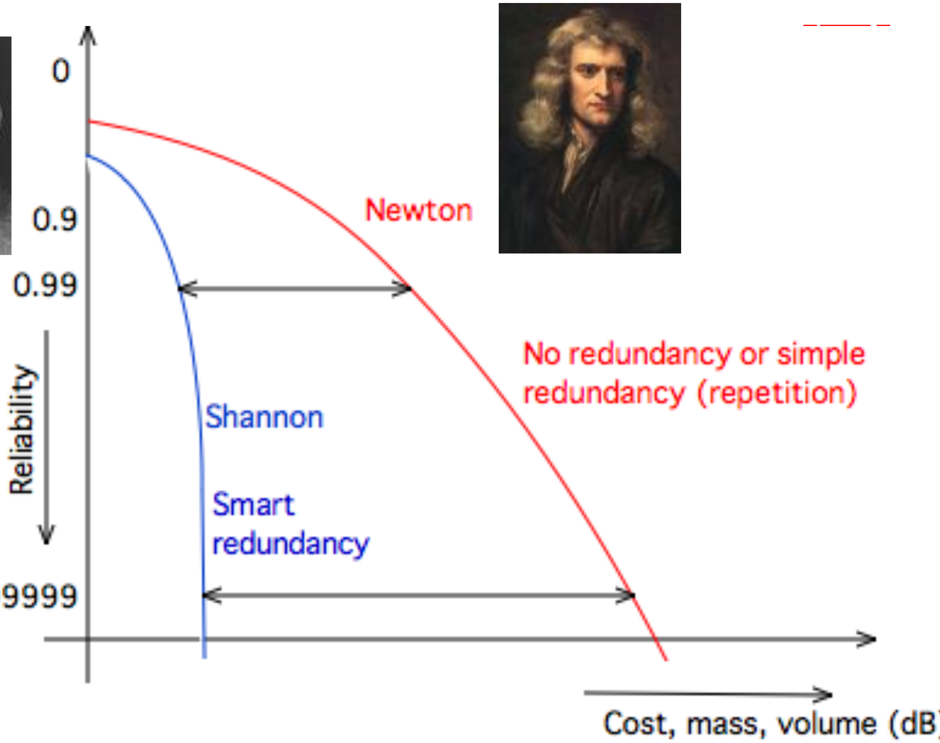
1993: Turbo codes

1996: LDPC Codes rediscovered

Power Efficiency



Shannon vs. Newton



- Shannon proved that there are optimal methods to combat noise by properly encoding the signal
- The rate of information transmission R across the channel is expressed as the mutual information between the transmitted and received signals
- R = H(X) - H(X|Y), can be interpreted as the amount of information sent less the uncertainty of what was sent
- Similarly, R = H(Y) - H(Y|X), measures the amount received less the part which is due to noise
- The capacity C of a noisy communication system is defined as the optimal or maximum possible rate of transmission over the channel

- A communication system reaches capacity when its information source is "matched" to its channel
  - Optimization of the rate of transmission consists of minimizing the lost information due to noise while maximizing the information contained within the source
  - Alternatively, optimization of the rate of transmission consists of minimizing the interference while maximizing the information contained within the received signal
- Shannon did not provide constructive methods to achieve capacity, but proved that “random codes” can achieve it
  - For 50 years, researchers have been looking for codes that are “good” but have some structure, to facilitate decoding
  - Now near-capacity codes are known, which require only moderate decoding complexity
- The fault-tolerance of a computing system can also be evaluated using of Shannon's theory

More Powerful Methods to Combat Faults

- Reliable information storage (and computation) is possible if the probability of memory failure can be made arbitrarily small
- This can be accomplished by using error-correcting codes

- Our proposed methods extend the current state-of-the-art and practice in several directions:
  - Powerful error/erasure correcting codes are used. This is possible because the complexity of these new codes is manageable
  - Error correcting codes are not used just during normal read and write operations, but they are repeatedly used to "scrub" the memory from any errors present
  - Permanently compromised memory cells are treated as "erasures" or kept in a suitably compressed list of "bad" cells not to be used again
  - Entire memory block failures are dealt with improved RAID-like systems based on a redundancy scheme provided by powerful error correcting codes
- The overall result is vastly improved reliability at the expense of a very modest increase in device size and processing power, without having to resort to conventional, expensive radiation hardening techniques

Optimal Use of Error-Correcting Codes and Scrubbing

"All models are wrong; some models are useful"
G. Box

A Simple Model for the Analysis of Optimal Strategies to Reduce SEUs Effects

- **Suppose we have *k* bits of data that must be preserved error-free for a long time with high probability**

- **To do so, we have three items at your disposal:**

1 - Memory cells

- **A memory cell can store one bit, but it is unreliable because of SEUs. SEUs have the effect of complementing bits. SEUs are modeled by a Poisson probability distribution with event rate λ**
- **Obviously, we will need some number *n* of memory cells, n > k**
- **Each memory cell costs c_m "deniers"**

2 - Encoder

- **An encoder for an (*n, k*) code. The encoder is free**

3 - Decoder

- **A decoder for an (*n, k*) code. The decoder's answer is correct with high probability, but it costs $c_s = k\,10^{e/n}$ deniers per use, where *e* is the number of symbol errors actually corrected**

- **A solution to this problem is to encode the *k* bits into *n* symbols, and store these in a size-*n* memory. After time *t* has elapsed, we "scrub" the memory by reading the contents, decoding them, re-encoding them, and re-writing the memory. This is repeated every *t* time units**

- **Question: What is the optimal strategy? That is, how much memory should one use, and how often should one scrub, to minimize the cost?**

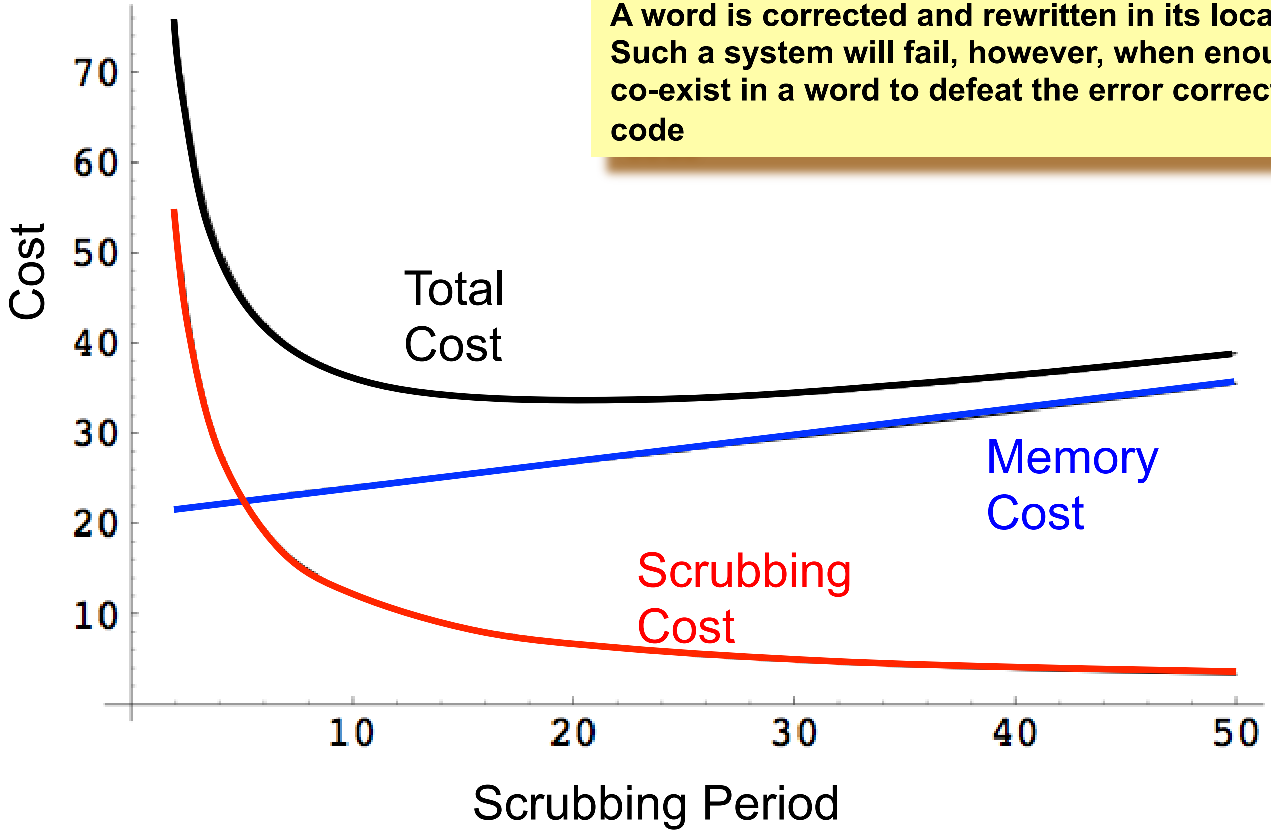
A Simple Model for the Analysis of Optimal Strategies to Reduce SEUs Effects

- **Let X_i, $i=1,\ldots,n$ be i.i.d Poisson random variables with Poisson Parameter λt**
  - **then $Z=\sum_{i=1}^{n} X_i$ also has Poisson probability with parameter $n\lambda t$, which represent the average number of errors in t seconds for n memory cells**
- **Assume the error correction capability of the (n,k) code is e. For example, for MDS codes $e = (n-k)/2$**
- **For bounded distance type decoding we would like that the probability of $Z > e$ be very small, say P_e**
- **If $n\lambda t$ is large enough then we can approximate Poisson with Gaussian distribution with mean $n\lambda t$ and variance $n\lambda t$**
- **With such approximation we can set $n\lambda t + q\sqrt{n\lambda t} = e$ where q can be selected to achieve a certain probability of error P_e**
- **For example: $q=3 \rightarrow P_e=10^{-3}$; $q=5 \rightarrow P_e=3\text{x}10^{-7}$; $q=6 \rightarrow P_e=10^{-9}$**

Total cost = scrubbing cost + memory cost or $\boxed{c = c_s/t + n\, c_m}$

Optimal Use of Error-Correcting Codes and Scrubbing

A word is corrected and rewritten in its location with period *t*. Such a system will fail, however, when enough multiple errors co-exist in a word to defeat the error correction capability of the code

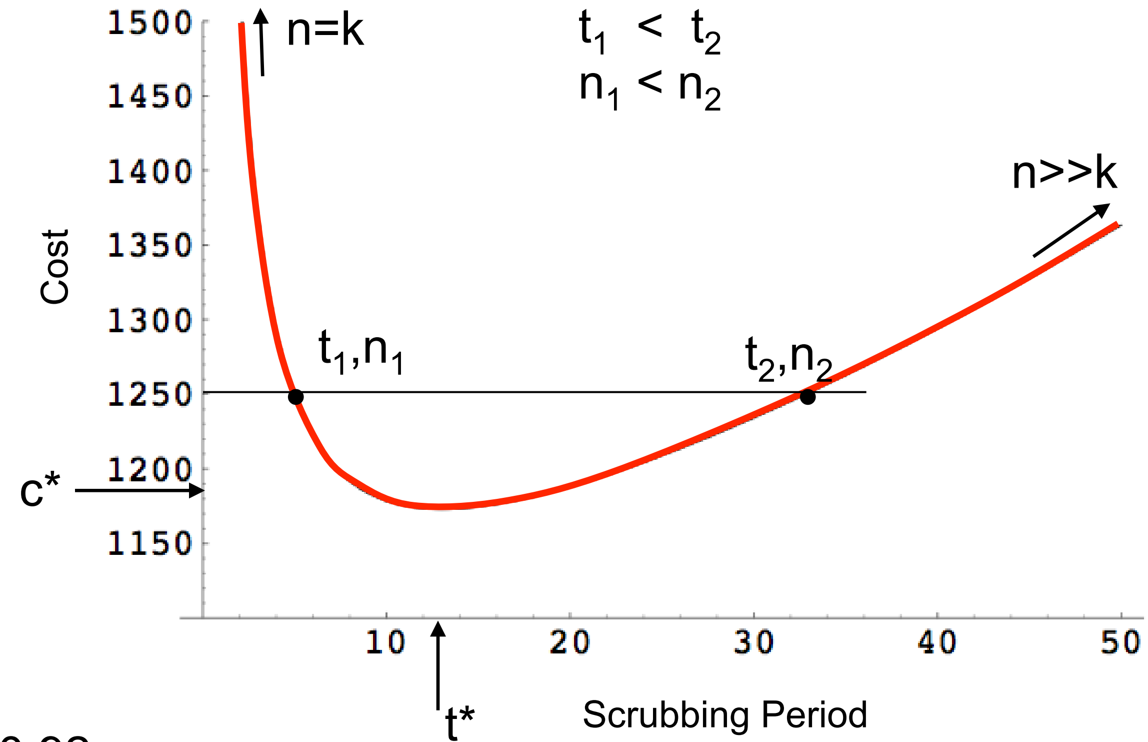


The scrubbing interval can be deterministic or probabilistic

- **Deterministic scrubbing is a mechanism dedicated to cycling through the memory system, reading every word and checking for its correctness**
- **Probabilistic scrubbing depends on the fact that a word is read and checked whenever it is needed [Not addressed here]**

Optimal Use of Error-Correcting Codes and Scrubbing

$\lambda = 0.002$
$k = 1000$
$C_m = 1$
$3\,\sigma \rightarrow P_e = 10^{-3}$

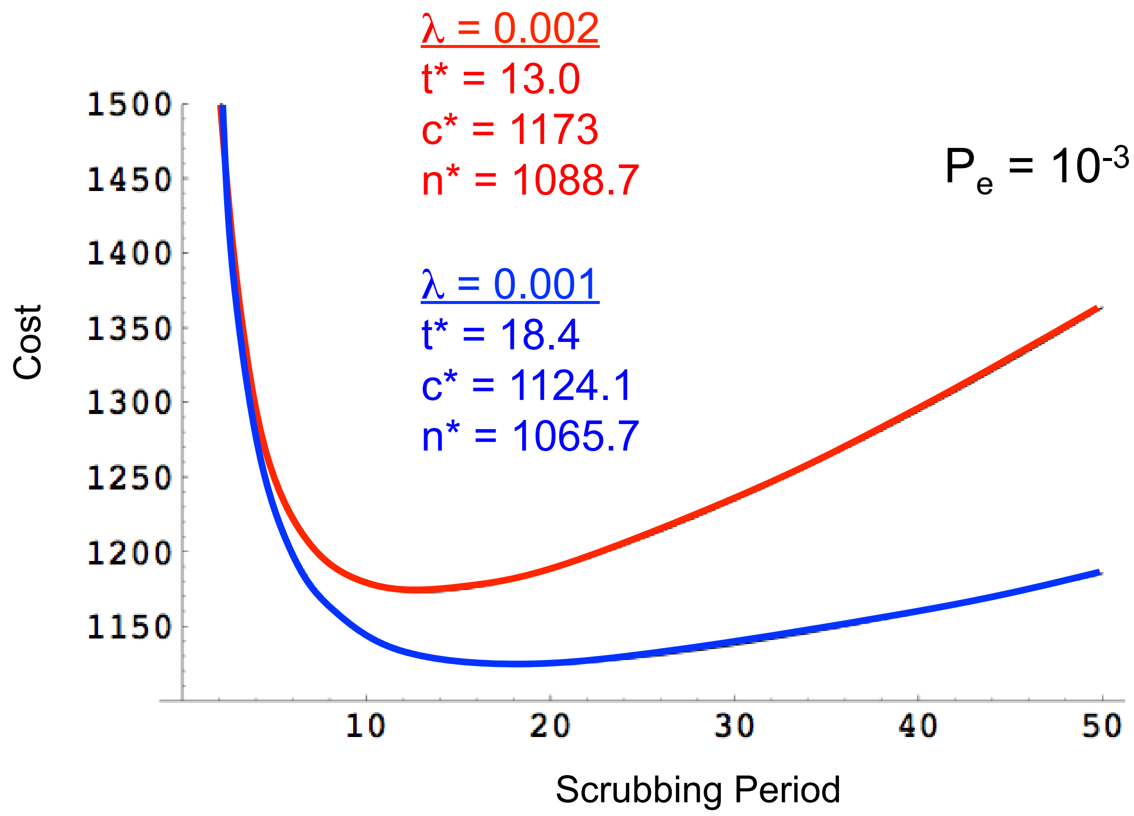


Optimal solution:

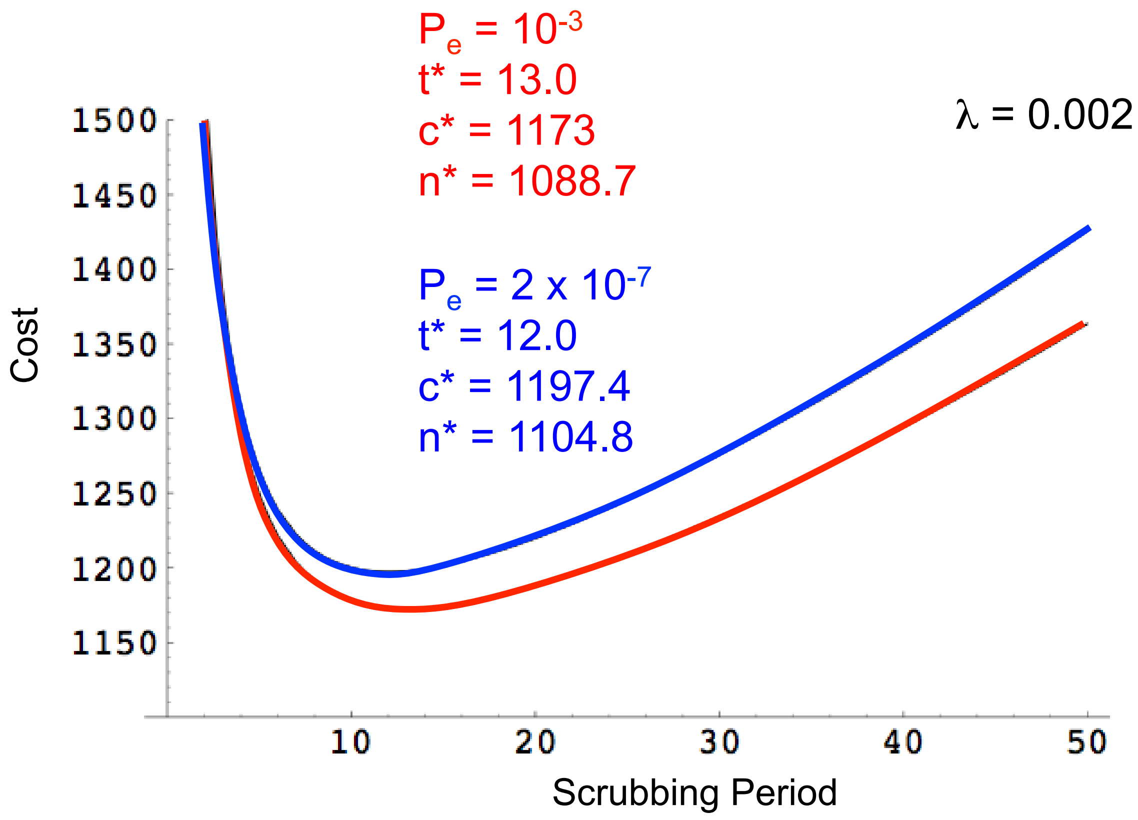
$t^* = 13.0$
$c^* = 1173$
$n^* = 1088.7$

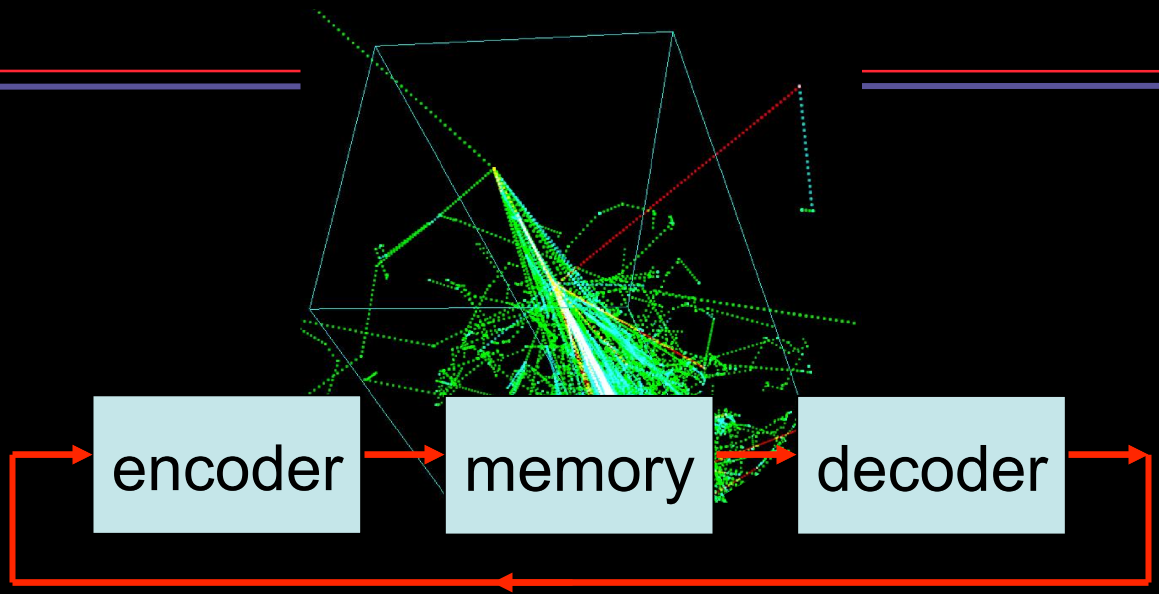
Code Rate = $k/n = 0.92$

Optimal solution for different SEU rates



Optimal solution for different target P_e





- **Memory protection can be extended to cases where errors happen also in the decoder**
- **LDPC codes are decoded by "belief propagation" iterative methods, which can tolerate errors in the decoding process**

Other Work to Improve Memory/Computation Reliability

- We mentioned already the work of von Neuman on "multiplexing" (1952)
- Pippenger (1988) found the size and depth requirements necessary for reliable computation using unreliable components
- Taylor (1965), Realized that von Neumann multiplexing is a special type of error correction code (repetition code), which is not efficient
  - Showed that faulty memory systems have nonzero storage/computational capacity
  - Shortly after the appearance of low-density parity-check(LDPC) codes (invented by Gallager), he observed that this class of codes is particularly well suited for decoding with a network of unreliable gates
  - The intuition was that faults in computation are tolerated due to the quantization of results from the iterative, consensus-building belief-propagation decoding algorithm
- Kuznetsov (1971) introduced many refinements, forming what is now called the Taylor-Kuznetsov (TK) scheme.
- Hadjicostis generalized the TK scheme to fault tolerant linear finite-state machines
- Spielman considered general models of computation (including distributed computation), and extended von Neumann multiplexing to Reed-Solomon codes

[Note the connection with "Distributed computing over unreliable links" -> link failures can be modeled as gate failures]

- Vasic (2007) proved a key result on the fraction of errors that a memory can tolerate and still be reliable, using "expander codes". He also proved new bounds on the performance of the TK scheme, and showed that protecting computations can be accomplished with the same methods used for protecting memories (under some assumptions)

Other thoughts: Sociological implications – democracy as a network of unreliable individuals !?

Conclusion

- We thought very hard on optimal scrubbing schemes, using powerful error correcting codes
- We believe that these concepts have the potential to revolutionize space instrument capabilities in harsh environments, by increasing the size and reliability of affordable storage and allowing much simpler and lower cost methods for transmission of the scientific data

Acknowledgements

- The authors would like to thank Prof. Bane Vasic of the University of Arizona for insightful discussions on information theoretic methods for radiation tolerance, and for providing and explaining his recent results
- We are also grateful to Dr. Bruce Moision (JPL) for discussing his work on reliable computations, and for explaining the merits of LDPC codes in fault protection applications