JUNE 4, 2019

Legaltech© news ONLINE

An ALM Publication

Nervous System: The Great Fixer of Information Coding Errors

In this month's look at the history of cybersecurity, David Kalat looks back at how one man's frustration at losing time led to one of the great breakthroughs in information theory.

BY DAVID KALAT, BRG

With the aggressive pace of technological change and the onslaught of news regarding data breaches, cyber-attacks, and technological threats to privacy and security, it is easy to assume these are fundamentally new threats. The pace of technological change is slower than it feels, and many seemingly new categories of threats have actually been with us longer than we remember. ***Nervous System*** *is a monthly blog that approaches issues of data privacy and cybersecurity from the context of history—to look to the past for clues about how to interpret the present and prepare for the future.*

Telecommunications involves transmitting chunks of binary data over sometimes great distances through all types of interference and barriers. Meanwhile, computer storage systems seek to archive these chunks of binary data on hard drives or solid-state devices that get dropped, shaken, overheated and generally mishandled. Either way, the integrity of the binary data is at risk.

For any individual bit of data, how can the user know if a "1" is really a "1" or if it was flipped from a "0" by interference? As it happens, this fundamental challenge was met in the earliest days of computer science—and solved by a frustrated young man tired of wasting his weekends.

Richard Hamming was an American mathematician who had programmed IBM calculating machines for the Manhattan Project. After the war, he was employed at Bell Labs as a general purpose "computer janitor." He distinguished himself with the invention of what came to be known as Hamming Error Correcting Codes (along with a slew of subsequent discoveries and patents, all of which bear the "Hamming" name).

In 1947, Hamming was running analysis on the Model V relay computer to prepare it for delivery to the US Army's Aberdeen Proving Ground facility in Maryland. He was, as he put it, "low man on the totem pole" and was allotted processing time on this machine only over the weekend. He could initiate his program at 5:00 p.m. Friday and return at 8:00 a.m. Monday to collect the output. When Hamming

returned to the computer on one Monday morning, he was deflated to find the machine had produced nothing whatsoever.

The Model V relay computers suffered around two to three relay failures a day, meaning one failure per two or three million operations. To guard against undetected errors, the engineers had rigged a primitive form of error detection. If the computer could not read a section of data after three tries, it would abort the process. In this case, some glitch in Hamming's input data rendered a section hard to read and caused the computer to give up.

Hamming waited until Friday to try again, only to find yet another blank reel of tape the following Monday. He had now wasted two entire weeks and realized continued faults could likely cost him weeks to come. In exasperation, he decided if the machine was smart enough to discover an error, he would make it smart enough to fix the error, too.

As a first attempted solution, Hamming replicated each bit of the source input three times to create a buffer of redundancy. The intended meaning of each triplet would be inferred by the concept of majority rules. In order to send "1," this approach would transmit "111." The triplet is a "codeword," meaning this particular code considers "111" to be a valid codeword meaning "1" and "000" a codeword meaning "0."

No other combinations are considered valid in this particular system. Anything else received—such as "001" or "101"—would immediately indicate that corruption had occurred. In this system, error correction occurs by identifying the closest valid codeword for any corrupted word: "001" is corrected back to "000," "101" is corrected to "1," and so on.

It may have been effective, but Hamming was bothered by the overhead costs. This method required three times as much data storage and transmission capacity as the original data. Hamming found his "Eureka!" moment while commuting between the Murray Hill location of Bell Labs and New York. "New Jersey isn't worth looking at," he figured, and set his mind to thinking through better alternatives.

Using his triplet codewords, he had added two bits of error checking to every bit of information in order to rely entirely on redundancy to catch and fix problems. In his new scenario, he found a way to add bits that also added context. Now, every codeword contained a clue about its neighbors in the chain—checking for error was no longer just about checking the "majority rules," but also whether that result agreed with the conclusion implied by the other information around it. In this way, instead of needing three bits of code to convey a single bit of information, he had a way to use just seven bits of code for four bits of information.

Devising error-correcting codes involves balancing three competing principles. First, keeping codewords short is important to keep the error-correcting system from overwhelming the actual information being transmitted or stored. This is in tension with the need to keep a significant distance between valid codewords so that errors are more easily detected. Last, a limited number of codewords are short while also maintaining a useful distance. A code that balances these competing principles without waste is said to be *perfect*. By this standard, Hamming's 7,4 code is *perfect*.

Following some haggling over patents, Hamming published his landmark 1950 paper, "Error Detecting and Error Correcting Codes." Bell Labs quickly adapted his idea into its telecommunications systems, and versions are still used roughly 70 years later to provide error correcting in mass storage devices and situations in which retransmitting corrupted data would be either impossible or prohibitively costly.

Yet while the 7,4 code is perfect in terms of information theory, its design allows the correction of only a single error in every seven bits of data. For especially unreliable and noisy communications channels, more robust error-checking systems have been devised. Their creation and development, however, derive from the pioneering work Hamming performed in the late 1940s, when he let his inspired frustration take the wheel.

David Kalat *is Director, Global Investigations + Strategic Intelligence at Berkeley Research Group. David is a computer forensic investigator and e-discovery project manager.* *Disclaimer for commentary: The views and opinions expressed in this article are those of the author and do not necessarily reflect the opinions, position, or policy of Berkeley Research Group, LLC or its other employees and affiliates.*