Error detection and correction

In mathematics, computer science, telecommunication, and information theory, **error detection and correction** has great practical importance in maintaining data (information) integrity across noisy channels and less-than-reliable storage media.

General definitions of terms

Definitions of error detection and error correction:

- Error detection is the ability to detect the presence of errors caused by noise or other impairments during transmission from the transmitter to the receiver.
- Error correction is the additional ability to reconstruct the original, error-free data.

There are two basic ways to design the channel code and protocol for an error correcting system:

- Automatic repeat-request (ARQ): The transmitter sends the data and also an error detection code, which the receiver uses to check for errors, and requests retransmission of erroneous data. In many cases, the request is implicit; the receiver sends an acknowledgement (ACK) of correctly received data, and the transmitter re-sends anything not acknowledged within a reasonable period of time.
- Forward error correction (FEC): The transmitter encodes the data with an **error-correcting code** (ECC) and sends the coded message. The receiver never sends any messages back to the transmitter. The receiver decodes what it receives into the "most likely" data. The codes are designed so that it would take an "unreasonable" amount of noise to trick the receiver into misinterpreting the data.

It is possible to combine the two, so that minor errors are corrected without retransmission, and major errors are detected and a retransmission requested. The combination is called hybrid automatic repeat-request.

Error detection schemes

In telecommunication, a *redundancy check* is extra data added to a message for the purposes of error detection.

Several schemes exist to achieve error detection, and generally they are quite simple. All error detection codes (which include all error-detection-and-correction codes) transmit more bits than were in the original data. Most codes are "systematic": the transmitter sends a fixed number of original data bits, followed by fixed number of *check bits* (usually referred to as redundancy in the literature) which are derived from the data bits by some deterministic algorithm. The receiver applies the same algorithm to the received data bits and compares its output to the received check bits; if the values do not match, an error has occurred at some point during the transmission. In a system that uses a "non-systematic" code, such as some raptor codes, data bits are transformed into at least as many code bits, and the transmitter sends only the code bits.

Repetition schemes

Main article: repetition code

Variations on this theme exist. Given a stream of data that is to be sent, the data is broken up into blocks of bits, and in sending, each block is sent some predetermined number of times. For example, if we want to send "1011", we may repeat this block three times each.

Suppose we send "1011 1011 1011", and this is received as "1010 1011 1011". As one group is not the same as the other two, we can determine that an error has occurred. This scheme is not very efficient, and can be susceptible to problems if the error occurs in exactly the same place for each group (e.g. "1010 1010 1010" in the example above will be detected as correct in this scheme).

The scheme however is extremely simple, and is in fact used in some transmissions of numbers stations.[*citation needed*]

Parity schemes

Main article: Parity bit

A *parity bit* is an *error detection* mechanism that can only detect an odd number of errors.

The stream of data is broken up into blocks of bits, and the number of 1 bits is counted. Then, a "parity bit" is set (or cleared) if the number of one bits is odd (or even). (This scheme is called even parity; odd parity can also be used.) If the tested blocks overlap, then the parity bits can be used to isolate the error, and even correct it if the error affects a single bit: this is the principle behind the Hamming code.

There is a limitation to parity schemes. A parity bit is only guaranteed to detect an odd number of bit errors (one, three, five, and so on). If an even number of bits (two, four, six and so on) are flipped, the parity bit appears to be correct, even though the data is corrupt.

Checksum

Main article: Checksum

A checksum of a message is an arithmetic sum of message code words of a certain word length, for example byte values, and their carry value. The sum is negated by means of ones-complement, and stored or transferred as an extra code word extending the message.

On the receiver side, a new checksum may be calculated from the extended message. If the new checksum is not 0, an error has been detected.

Checksum schemes include parity bits, check digits, and longitudinal redundancy check.

Cyclic redundancy checks

Main article: Cyclic redundancy check

More complex error detection (and correction) methods make use of the properties of finite fields and polynomials over such fields.

The cyclic redundancy check considers a block of data as the coefficients to a polynomial and then divides by a fixed, predetermined polynomial. The coefficients of the result of the division is taken as the redundant data bits, the CRC.

On reception, one can recompute the CRC from the payload bits and compare this with the CRC that was received. A mismatch indicates that an error occurred.

Hamming distance based checks

If we want to detect *d* bit errors in an *n* bit word we can map every *n* bit word into a bigger *n+d+1* bit word so that the minimum Hamming distance between each valid mapping is *d+1*. This way, if one receives a *n+d+1* word that doesn't match any word in the mapping (with a Hamming distance *x* <= *d+1* from any word in the mapping) it can successfully detect it as an erroneous word. Even more, *d* or fewer errors will never transform a valid word into another, because the Hamming distance between each valid word is at least *d+1,* and such errors only lead to invalid words that are detected correctly. Given a stream of *m*n* bits, we can detect *x <= d* bit errors successfully using the above method on every *n* bit word. In fact, we can detect a maximum of *m*d* errors if every *n* word is transmitted with maximum *d* errors.

Hash function

Any hash function can be used as a integrity check.

Horizontal and vertical redundancy check

Other types of redundancy check include horizontal redundancy check, vertical redundancy check and "double", "dual" or "diagonal" parity (used in RAID-DP).

Error correction

Automatic repeat request

Main article: Automatic repeat-request

Automatic Repeat-reQuest (ARQ) is an error control method for data transmission which makes use of error detection codes, acknowledgment and/or negative acknowledgement messages and timeouts to achieve reliable data transmission. An acknowledgment is a message sent by the receiver to the transmitter to indicate that it has correctly received a data frame.

Usually, when the transmitter does not receive the acknowledgment before the timeout occurs (i.e. within a reasonable amount of time after sending the data frame), it retransmits the frame until it is either correctly received or the error persists beyond a predetermined number of retransmissions.

A few types of ARQ protocols are Stop-and-wait ARQ, Go-Back-N ARQ and Selective Repeat ARQ.

Hybrid ARQ is a combination of ARQ and forward error correction.

Error-correcting code

Main article: Forward error correction

An error-correcting code (ECC) or forward error correction (FEC) code is redundant data that is added to the message on the sender side. If the number of errors is within the capability of the code being used, the receiver can use the extra information to discover the locations of the errors and correct them. Since the receiver does not have to ask the sender for retransmission of the data, a back-channel is not necessary in forward error correction, so it is suitable for simplex communication such as broadcasting. Error-correcting codes are used in computer data storage, for example CDs, DVDs and in dynamic RAM. It is also used in digital transmission, especially wireless communication, since wireless communication without FEC often would suffer from packet-error rates close to 100%, and conventional automatic repeat request error control would yield a very low goodput.

Two main categories are convolutional codes and block codes.

Convolutional codes are processed on a bit-by-bit basis, and only causes a processing delay corresponding to a few bit periods. In convolutional coding, a Viterbi decoder is typically used on the receiver side.

Block codes are processed on a block-by-block basis. Early examples of block codes are repetition codes, Hamming codes and multidimensional parity-check codes. More efficient codes, often used in modern systems, are Reed-Solomon codes, turbo codes, BCH codes, Reed-Muller codes, Binary Golay codes, and low-density parity-check codes (LDPC). For example, the code rate of the Reed Solomon block code denoted RS(204,188) processes blocks of 188 bytes of useful information at a time, and appends

204 - 188 = 16 redundant bytes to each block. It can handle 8 incorrect bytes per data block.

Shannon's theorem is an important theorem in forward error correction which describes a theoretical upper bound for the attainable spectral efficiency in bit/s/Hz of a channel coding scheme (an error-correcting scheme, typically combined width a digital modulation scheme) for a specific signal-to-noise ratio (SNR). For example, if the SNR is 0 dB, the spectral efficiency can not be higher than 1 bit/s/Hz, resulting in that the information rate in bit/s (the useful bitrate, excluding redundant ECC) can not be higher than the bandwidth in hertz.

The effectiveness of the coding scheme may also be measured in terms of code rate *k/n*, which is the ratio between *k* information bits and *n* transmitted data bits. Finally the effectiveness may be expressed as coding gain in decibel, which is the allowable reduction in signal-to-noise ratio whilst still attaining the same bit error rate and data rate as the uncoded system.

Error-correcting memory

Main article: Dynamic random access memory#Errors and error correction

Because soft errors are extremely common in the DRAM of computers used in satellites and space probes, such memory is structured as ECC memory (also called "EDAC protected memory"). Such memory controllers traditionally use a Hamming code, although some use triple modular redundancy. Even though a single cosmic ray can upset many physically neighboring bits in a DRAM, such memory systems are designed so that neighboring bits belong to different words, so that a single event upset (SEU) causes only a single error in any particular word, and so can be corrected by a single-bit error correcting code. As long as no more than a single bit in any particular word is affected by an error between accesses, such a memory system presents the illusion of an error-free memory.[1]

Applications

Applications that require low latency (such as telephone conversations) cannot use Automatic Repeat reQuest (ARQ); they must use Forward Error Correction (FEC). By the time an ARQ system discovers an error and re-transmits it, the re-sent data will arrive too late to be any good.

Applications where the transmitter immediately forgets the information as soon as it is sent (such as most television cameras) cannot use ARQ; they must use FEC because when an error occurs, the original data is no longer available. (This is also why FEC is used in data storage systems such as RAID and distributed data store).

Applications that use ARQ must have a return channel. Applications that have no return channel cannot use ARQ.

Applications that require extremely low error rates (such as digital money transfers) must use ARQ.

The Internet

In a typical TCP/IP stack, error detection is performed at multiple levels:

- Each Ethernet frame carries a CRC-32 checksum. The receiver discards frames if their checksums do not match.
- The IPv4 header contains a header checksum of the contents of the header (excluding the checksum field). Packets with checksums that don't match are discarded.
- The checksum was omitted from the IPv6 header, because most current link layer protocols have error detection.
- UDP has an optional checksum. Packets with wrong checksums are discarded.
- TCP has a checksum of the payload, TCP header (excluding the checksum field) and source- and destination addresses of the IP header. Packets found to have incorrect checksums are discarded and eventually get retransmitted when the sender receives a triple-ack or a timeout occurs.

Deep-space telecommunications

NASA has used many different error correcting codes. For missions between 1969 and 1977 the Mariner spacecraft used a Reed-Muller code. The noise these spacecraft were subject to was well approximated by a "bell-curve" (normal distribution), so the Reed-Muller codes were well suited to the situation.

The Voyager 1 & Voyager 2 spacecraft transmitted color pictures of Jupiter and Saturn in 1979 and 1980.

- Color image transmission required 3 times the amount of data, so the Golay (24,12,8) code was used.[*citation needed*][2]
- This Golay code is only 3-error correcting, but it could be transmitted at a much higher data rate.
- Voyager 2 went on to Uranus and Neptune and the code was switched to a concatenated Reed-Solomon code-Convolutional code for its substantially more powerful error correcting capabilities.
- Current DSN error correction is done with dedicated hardware.
- For some NASA deep space craft such as those in the Voyager program, Cassini-Huygens (Saturn), New Horizons (Pluto) and Deep Space 1—the use of hardware ECC may not be feasible for the full duration of the mission.

The different kinds of deep space and orbital missions that are conducted suggest that trying to find a "one size fits all" error correction system will be an ongoing problem for some time to come.

- For missions close to the earth the nature of the "noise" is different from that on a spacecraft headed towards the outer planets.
- In particular, if a transmitter on a spacecraft far from earth is operating at a low power, the problem of correcting for noise gets larger with distance from the earth.

Satellite broadcasting (DVB)

The demand for satellite transponder bandwidth continues to grow, fueled by the desire to deliver television (including new channels and High Definition TV) and IP data. Transponder availability and bandwidth constraints have limited this growth, because transponder capacity is determined by the selected modulation scheme and Forward error correction (FEC) rate.

Overview

- QPSK coupled with traditional Reed Solomon and Viterbi codes have been used for nearly 20 years for the delivery of digital satellite TV.
- Higher order modulation schemes such as 8PSK, 16QAM and 32QAM have enabled the satellite industry to increase transponder efficiency by several orders of magnitude.
- This increase in the information rate in a transponder comes at the expense of an increase in the carrier power to meet the threshold requirement for existing antennas.
- Tests conducted using the latest chipsets demonstrate that the performance achieved by using Turbo Codes may be even lower than the 0.8 dB figure assumed in early designs.

Data storage

Error detection and correction codes are often used to improve the reliability of data storage media.

A "parity track" was present on the first magnetic tape data storage in 1951. The "Optimal Rectangular Code" used in group code recording tapes not only detects but also corrects single-bit errors.

Some file formats, particularly archive formats, include a checksum (most often CRC32) to detect corruption and truncation and can employ redundancy and/or parity files to recover portions of corrupted data.

Reed Solomon codes are used in compact discs to correct errors caused by scratches.

Modern hard drives use CRC codes to detect and Reed-Solomon codes to correct minor errors in sector reads, and to recover data from sectors that have "gone bad" and store that data in the spare sectors.[3]

RAID systems use a variety of error correction techniques, to correct errors when a hard drive completely fails.

Information theory and error detection and correction

Information theory tells us that whatever the probability of error in transmission or storage, it is possible to construct error-correcting codes in which the likelihood of failure is arbitrarily low, although this requires adding increasing amounts of redundant data to the original, which might not be practical when the error probability is very high. Shannon's theorem sets an upper bound to the error correction rate that can be achieved (and thus the level of noise that can be tolerated) using a fixed amount of redundancy, but does not tell us how to construct such an optimal code.

Error-correcting codes can be divided into block codes and convolutional codes. Other block error-correcting codes, such as Reed-Solomon codes, transform a chunk of bits into a (longer) chunk of bits in such a way that errors up to some threshold in each block can be detected and corrected.

However, in practice errors often occur in bursts rather than at random. This is often compensated for by shuffling (interleaving) the bits in the message after coding. Then any burst of bit-errors is broken up into a set of scattered single-bit errors when the bits of the message are unshuffled (de-interleaved) before being decoded.

[edit] List of error-correction, error-detection methods

This list contains methods of error correction (Reed-Solomon, for example is a method) and practical techniques for error correction (like the Check digit, a practical method).

- Berger code
- Chipkill, an application of ECC techniques to volatile system memory.
- Constant-weight code
- Convolutional codes are usually decoded with iterative Viterbi decoding techniques
- Differential space–time codes, related to space–time block codes.
- Dual modular redundancy, subset of N-modular redundancy, related to triple modular redundancy
- Erasure codes are a superset of Fountain codes
- Forward error correction
- Group code
- Golay code, the Binary Golay codes are the most commonly used Golay codes
- Goppa code that is used to create the McEliece cryptosystem
- Hadamard code
- Hagelbarger code
- Hamming code
- Lexicographic code

- Longitudinal redundancy check
- Low-density parity-check code
- LT codes are near optimal rateless erasure correcting codes.
- m of n codes
- Online codes are an example of rateless erasure codes.
- Parity bit
- Raptor codes are a class of fountain codes.
- Reed-Solomon error correction
- Reed-Muller code
- Repeat-accumulate code
- Sparse graph code
- Space–time code
- Space–time trellis code
- Tornado codes are optimal Fountain codes
- Triple modular redundancy
- Turbo code
- Viterbi algorithm

Practical uses of Error Correction methods

- Concatenated error correction codes, the Compact Disc and Voyager Program spacecraft use concatenated error correction technologies
- Check digit, commonly used on UPC barcodes
- Luhn algorithm, the most commonly used base 10 checksum that can perform limited error detection but not error correction
- Luhn mod N algorithm, the above algorithm but implementable in a non base 10 form
- Verhoeff algorithm, a modular based form not related to the Luhn algorithms that can detect most forms of transposition errors in financial cryptographic applications