Communication Methods

What is Communication?

Communication involves sharing ideas, information, sounds or images. Communicating over a distance requires three stages: encoding, transmission and decoding. For example, 1500 years ago the Incas, a highly organized, methodical culture without any written language, used quipus to communicate numeric information. A quipu was a method of encoding numbers for record keeping that involved an intricate set of knots tied into colored cotton cords that were connected in specific ways. The knots represented numbers stored in base ten, with the absence of a knot indicating zero so that the user could distinguish, for example, between 48 and 408. Runners transmitted the information by carrying the quipu from one location to another. Whoever needed to look up the record, studied the knots and decoded the information.

Methods of encoding/decoding and transmission have advanced with technology. We now can encode/decode with binary numbers and transmit via electricity, radio waves and fiber-optic cables. All of these technologies require mathematics to make them viable.

The Challenge of Long Distance Communication

1. How does information get encoded and decoded?
2. How can we transmit the information fast enough to make it practical?
3. How can we send information without data loss or corruption?
4. How can we send information securely?

We will discuss the first three of these in detail. The answer to the fourth question involves cryptography, a deep and ancient subject with amazing recent mathematical advances whose treatment is beyond the scope of this article. Ironically it is Number Theory, the most artistic, pure, abstract and theoretical branch of mathematics that provided us with a method of encryption for sending our credit card numbers securely over the Internet. Interested readers are referred to Singh.

The Encoding of Information – Analog Versus Digital

There are two paradigms for encoding information. With analog encoding, information is changed continuously from one physical form to another. For example, we can encode sound into the vibrations of a needle, and store it as scratches on a vinyl disk. A phonograph decodes this information by reversing the process. The disadvantages of analog encoding are noise and inefficiency in the encoding process. Telegraphs, radio, television, cameras, tape recorders and telephones were all originally analog inventions. Today, there are digital versions of all of these. There is a great deal of mathematics involved with analog encoding, including Fourier transforms, but it is beyond the scope of this article. The advanced reader may refer to Lathi.

With digital encoding, we transform information into a sequence of numbers. Written language can be encoded by using a specific number for every unique symbol in the language's alphabet. A picture can be encoded by splitting up the image into tiny pieces and using numbers to represent the color, intensity and position of each piece. Sound can be encoded by splitting up the soundtrack into small pieces and using numbers to represent the frequency, intensity and quality of the sound at each tiny fraction of a second. A similar process can be used with video. CD and DVD players are themselves digital devices.

Compared to analog encoding, digital encoding always misses information, but if you take enough samples, then the missing pieces of information in between are not discernable. It is like looking outside through a screen door. If the screen is very fine, you barely notice its presence, even though it really covers a substantial percentage of your view. Of course, if you do not take enough samples then you may end up with fuzzy pictures, choppy recordings, or missing information.

Binary Numbers and Digital Encoding

It is easiest physically to build transmission devices that send and receive only two different types of signals. Smoke signals and Morse code are early examples of this. Today, electricity can transmit either a high or a low voltage, where high represents one and low represents zero. Therefore we try to encode information using only zeros and ones.

A sequence of digits containing just zeros and ones is called a binary number. For example, here are the base ten numbers 0 through 9 represented in binary.

0 1 10 11 100 101 110 111 1000 1001

We can show 0 and 1 directly but since there is no symbol for two, we create a new column which we think of as the two's column, writing two as 10. We represent every subsequent number by adding one to the previous number and carrying to new columns whenever necessary. This is exactly like base ten, but we carry when we reach two instead of ten. In the binary system we have columns that represent increasing powers of two: one, two, four, eight … etc. The binary number 100010011 therefore represents 1 + 2 + 16 + 256 = 275. Every base ten number can be encoded into a binary number and vice versa.

ASCII – The Binary Encoding of Text

How can we use binary numbers to transmit text? ASCII encoding, (American Standard Code for Information Interchange published in 1968, pronounced *askee*) associates each character that appears on a keyboard with a unique binary sequence of length eight called a byte. These characters include all upper and lower case letters, digits, various special symbols and punctuation marks. For example, the ASCII value of 'A' is 01000001; the

ASCII value of ‘B’ is 01000010; and the ASCII value of ‘5’ is 00110101. Each byte consisting of eight binary digits (bits for short) can store $2^8 = 256$ different values, more than enough to store the various symbols on a keyboard. One bit can store two values, 0 or 1. Two bits can store 00, 01, 10 or 11. With three bits we can store eight possible sequences. With each new bit, we double the number of possibilities, by adding a zero in front of all the previous possibilities and by adding a one. This implies that *n* bits can store 2^n different configurations.

Unicode – The Binary Encoding of Written Characters

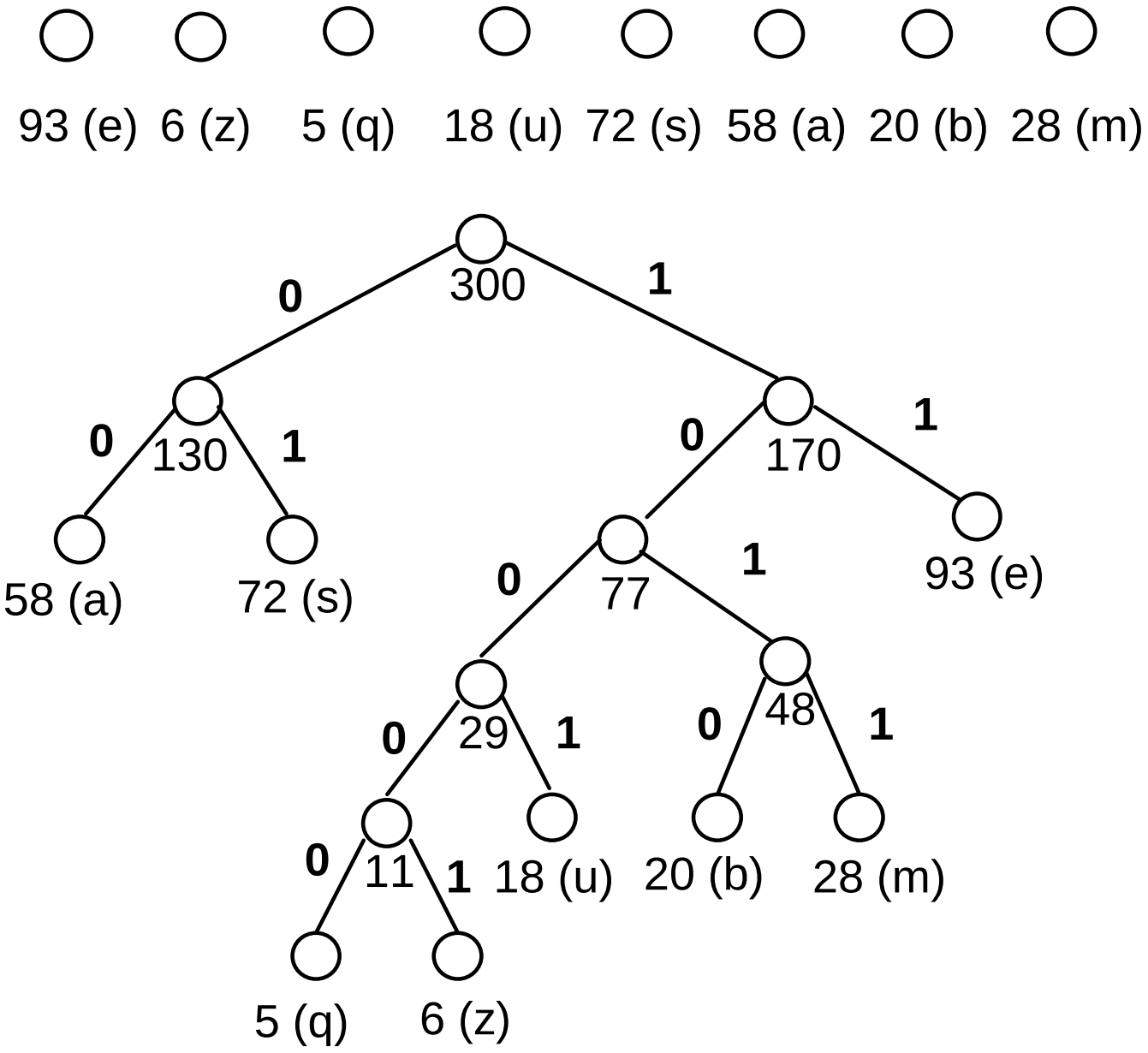
ASCII is fine for English but falls short when applied to our more diverse worldwide collection of languages and symbols. A more recent advance in written encoding is Unicode. Unicode is a universal, efficient, uniform and unambiguous way to encode the world’s written languages. It includes symbols for Hebrew, Arabic and Chinese as well hundreds of standard business symbols. Unicode is a new standard that extends ASCII. It uses two bytes instead of one, or 16 bits instead of eight. This gives Unicode the ability to encode a total of $2^{16} = 65536$ different symbols.

Huffman Encoding – Compression

Compression is a way to shorten our messages without losing any information, thereby reducing the size and transmission speed of the message. We normally encode each character of text with one byte. The inefficiency is that a *q* is encoded with the same eight bits as a *t* and yet the two letters occur with a different expected frequency. Huffman encoding takes into account the relative frequencies of the information being encoded, and can create a compression factor of up to 90%.

For example, a message contains 93 *e’s*, 6 *z’s*, 5 *q’s*, 18 *u’s*, 72 *s’s*, 58 *a’s*, 20 *b’s* and 28 *m’s*. ASCII encoding requires eight bits per character, or 8×(93+6+5+18+72+58+20+28) = 2400. Normal flat binary encoding will require three bits per character because there are eight different characters. This gives a total of 3×(93+6+5+18+72+58+20+28) = 900 bits.

Huffman assigns the encoding in the following way: *e=11, z=10001, q=10000, u=1001, s=01, a=00, b=1010, m=1011.* This gives a total message size of (2×93 + 5×6 + 5×5 + 4×18 + 2×72 + 2×58 + 4×20 + 4×28) = 765, a 15% compression. This encoding is done with a greedy algorithm that builds a tree in order to determine the optimal encoding. The algorithm starts with a separate single node tree for each character labeled by its frequency. It repeatedly combines the two subtrees with the smallest weights, labeling the new tree with the sum of the labels on the two old trees, until there is just one tree. The initial trees and the final tree are shown below.



Note that the decoding of a Huffman encoding is unambiguous because no encoded character is a prefix of another. That means that we can keep reading characters until a match occurs, decode it and continue.

Morse code is a form of compression in analog encoding that uses short and long signals to represent letters. The more commonly transmitted letters use shorter sequences while the less commonly transmitted letters use longer sequences.

Error Correction – Hamming Codes

Error detection and correction allows us to ensure that information is transmitted without loss or corruption. When we transmit digital information it is easy for some of the bits to be corrupted because of lightning storms, surges in power supply, dirty lenses and undependable transmission lines. How can we make sure that the actual data transmitted is received?

The trick is to add extra bits so that only some of the sequences are legal messages. This increases the size of the message and thereby slows down the transmission time, but allows more accurate transmission. For example, in the case of four-bit messages, the sender can simply tack on an extra copy of the four bits making eight bits total. The receiver compares the two copies and requests a retransmission if the two copies differ. For example, the sequence 10001000 is legal but 10001001 would require a retransmission. The downside is that we cannot determine which was the correct four-bit message.

There are strategies, however, that correct errors automatically without any need for retransmission. Assume that our message is broken up into groups of four bits and we want to correct any single bit error in a given group. This time we add on two extra copies of the four bits making twelve bits total. Only sequences of twelve bits with three identical copies of the first four bits are legal. There are 2^{12} possible 12-bit messages and only 2^4 are legal. If there is a single difference between any two of the three copies then we know that a single error has occurred and we correct it to be the same as the other two identical copies.

Why three copies? The three copies allow us to pinpoint where the error occurred. The idea is that *any two distinct legal messages must differ by at least three bits.* Hence an illegal 12-bit message that occurred due to a transmission error of a single bit, must have come from a unique original legal message. The decoding is unambiguous. For example, if 100011001000 is received, then the correct message is 100010001000 and there is an error in the sixth bit transmitted.

What if more than one error occurs? If two or more differences exist, we just ask for a retransmission. What if many errors occur resulting by chance in a, nonetheless, legal message? That would be undetectable. We set up this scheme to correct only one error per four-bit sequence. We could correct more than a single error if we were willing to add more redundant information and incur a proportionately larger increase in message and transmission time.

There is a more efficient way to correct single bit errors called Hamming codes. The method is elegant, practical, but complex. For an explanation of why it works see Hamming's book. Let $4+r$ be the total number of bits sent. Hamming realized that each of the 2^4 legal messages must have $4+r$ single-error illegal messages that can be created from it, and that if *any two distinct legal messages must differ by at least three bits*, then all of these illegal messages must be mutually distinct. This means that we need at least

$(4+r)\times 2^4$ distinct illegal messages and 2^4 distinct legal messages, totaling $(4+r+1)\times 2^4$ distinct messages. Since we have exactly 2^{4+r} distinct binary sequences, we need $2^{4+r} \geq (4+r+1)\times 2^4$. The reader can check that r must be at least three.

Hamming showed how to achieve this lower bound, thereby exhibiting the smallest possible size messages (seven) to do single error-correction on four-bit data messages. Here is how Hamming encodes four bits of data into seven while allowing for the correction of a single error. He writes his four data bits in positions 3, 5, 6 and 7. Then he computes the bits in positions 1, 2 and 4 depending on the number of ones in a particular set of the other positions. These extra bits are called parity bits. He sets a given parity bit to zero if the number of ones in the appropriate positions is even, and to one if the number of ones is odd. The sets of positions for the parity bits in positions 1, 2 and 4 are {3,5,7}, {3,6,7}, and {5,6,7} respectively.

For example, if the message is: 0011, then we have _ _0_111. The {3,5,7} set gives two one's, an even number so we set position one equal to zero. The {3,6,7} set is similar so position two is also zero. The {5,6,7} set has an odd number of ones so the fourth position is one. The final encoded seven bits is: 0001111.

To decode Hamming's scheme, we re-compute the parity bits and if they match up to what was received, we have a legal message. If they don't match up then the sum of the positions with incorrect parity gives the position of the single bit error. For example, if 0001111 is corrupted into 0001101, then the parity bit for position one is correct, while those for positions two and four are incorrect. This means that the error is in position 2+4 = 6. The efficiency of this method makes it the standard.

The Communication Revolution

Just 150 years ago, a routine telephone call was not possible, images were delivered in person, and newspapers brought headlines of information that may have been days old. There was no TV or radio coverage of world events. There was no world wide web for the world to share its collective knowledge.

Today we take for granted the ability to communicate with our business partners, colleagues, friends, and loved ones over long distances and at any time of the day. Via phone calls, e-mail, or fax, we transmit pictures, voice or text to almost anyone and anywhere at anytime. Communication has become faster, cheaper and more commonplace. Mathematics is used to encode, transmit and decode data in a practical and secure way. Mathematics drives the new technology and makes the communication revolution possible.

Word Count

2360

Byline

Professor Shai Simonson, Ph.D
Department of Mathematics and Computer Science
Stonehill College
shai@stonehill.edu
http://academics.stonehill.edu/compsci/SHAI.HTM

Bibliography

Ascher, Marcia and Robert, *Mathematics of the Incas : Code of the Quipu*, (Paperback - April 1997)

Cormen, T.H., Leiserson, C.E. and Rivest, R.L., *Introduction to Algorithms* , McGraw-Hill, New York, 1990, Sec. 17.3 - Huffman Codes.

Hamming, R., *Coding and Information Theory*, 2nd ed., Prentice Hall 1986.

Katz, Victor J., *A History of Mathematics: An Introduction*, 2nd Edition, Harper Collins, 1998.

Lathi, B. P., *Modern Digital and Analog Communications Systems*, Oxford University Press, 3rd Edition, April 1998.

Singh, Simon, *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*, Doubleday, 2000.

www.unicode.org, Homepage for Unicode.

http://www-groups.dcs.st-andrews.ac.uk/~history/Mathematicians/Hamming.html, Details about the life and work of R. Hamming.

Glossary

Quipu – A knotted cord used by the Incas and other cultures to encode numeric information.
Tree – A collection of dots with edges connecting them that have no looping paths.
Digital Encoding – Encoding information into a series of discrete numbers.
Analog Encoding – Encoding information with continuous quantities.