JOHN COCKE

1960

8 July 1960

DO IT BY THE NUMBERS

(Digital Shorthand)

A paper to be presented to the National Convention of the Association for Computing Machinery, 23-25 August 1960, at Milwaukee, Wisconsin.

by

R.W.Bemer, I.B.M. Corporation, White Plains, New York

Note:

This draft copy is intened for internal release within the I.B.M. Corporation only.

15 February 1960
(revised 8 July 1960)

DO IT BY THE NUMBERS

(Digital Shorthand)

R.W.Bemer, I.B.M.Corporation, White Plains, New York

Abstract

Present communications systems transmit single characters in groups of coded pulses between simple terminal equipments. Since English words form only a sparse set of all possible alphabetic combinations, present methods are inefficient when computer systems are substituted for these terminals. Using numeric representations of entire words or common phrases (rather than character-by-character representations) requires approximately one-third of present transmission time. This saving is reflected in overall costs. Other benefits accrue in code and language translation schemes. Provision is made for transmission of purely numeric and/or binary streams, and for single character transmission of non-dictionary words such as the names of people or places.

General Principles

Precedent may be found inthe story of the comedians' club that sat around and laughed when a member said "38". In this case the entire story is represented by that single number. One working example is that of standard Western Union messages such as birthday greetings; not everyone realizes that the entire message is not transmitted, only its number which tells the receiver what verbal message to type out on a form. Another example is that of a person's telephone number. His name and address may be transmitted in a more compact fashion by merely using the number. The receiver, equipped with the same phone book ordered on number rather than name, can simply decode.

Overstandardization at the message level will not work generally for the infinite variety met in general transmission. The single word, delimited by blanks, is the efficient denominator in the practical sense. An example of this is the book code that children use for ciphers. Here the page number, line number and n th position on the line define a specific word. These three numbers may be compressed to a single number by using fixed sub-fields. Thus, 0312806 would indicate the 6th word of the 28th line on the 31st page. A related method would be to number all the words in the dictionary sequentially starting with 1.

Ground Rules

English unabridged dictionaries contain less than 600,000 individual entries. The average speaking vocabulary is from 1000 to 2000 words, the average writing vocabulary from 6000 to 8000. A college graduate may have from 7500 to 10000 words to use. It has been said that a person with a 3000 word vocabulary can understand 95 percent of general speech.

Since $2^{22} \sim 4{,}000{,}000$, it seems that about 22 bits should be capable of representing any word in the language, with perhaps enough freedom and overcapacity to be informational if desired. That is, they may identify words as to tense, plurals, nouns or verbs, etc.

Examples given will use the six bit representation for letters proposed in a recent draft standard issued by the Electronic Industries Association. [1]

Compression Method

In early developments, Shannon [2] represented all words by an invariant or constant number of bits. For reasons of economy this is not practical. The average length of an English word is usually taken as 5 letters, plus a delimiting blank. A minimum of 5 bits (yielding 32 combinations) is necessary to represent a character singly. Thus the average number of bits required to represent a word in this mode is 30. Reduction to 22 bits is not impressive.

Therefore, statistical frequency of usage may be used to provide representations of a variable number of bits. The problem is then how to decode the bit stream at the receiving end. This is possible by:

1. Advance knowledge of a constant byte size
2. Termination by recognition of a fixed bit pattern normally excluded from the code
3. Self-definition where the first n bits indicate how long the next group is
4. Termination by inspection of every nth bit, or bits in a single track.

Method 2 is due to Brillouin [3]. Each representation is distinguished by at least two consecutive zero bits followed by a one bits. The stream of figure 1 is decoded as an example.

1 0 1 1 1 0 0 1 0 1 1 0 1 1 1 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 1 Figure 1.

This method has the advantage of self-repair after a transmission error (except for the word in which the error occurred). It is not suitable for computer transformation of large dictionaries because of the exclusion of all address combinations with two consecutive zeros. It can be seen from Table 3 that it takes a storage of over a half-million words to handle a vocabulary of 28,635 words, at an efficiency of only .055

Method 3 is self-defining like a measuring worm, suitable to computer decoding. As an example, assume that transmission is to be in parallel on four channels (or on single wire in identifiable groups of four.) Figure 2 shows how the leading bit(s) specify the length of a single word representation.

Leading bit(s)	No. of bytes (groups)	Working bits	Number of words accommodated
0	2	7	128
10	3	10	896
110	4	13	7,168
111	6	21	2,088,960

0 x	1 x x	1 x x x	1 x x x x x		0 1 1 1
x x	0 x x	1 x x x	1 x x x x x		1 0 0 1
x x	x x x	0 x x x	1 x x x x x	etc.	0 1 0 0
x x	x x x	x x x x	x x x x x x		1 1 1 0

direction of reading ⟶

Figure 2.

Table 1. (Method 3)

Type	bits per byte	first bits	no. of bytes	working bits	number of words accommodated	percent usage (est.)	percent times bytes	bytes per word	bits per word
A	4	0	2	7	128	57	114	2.69	10.76
		10	3	10	896	21	63		
		110	4	13	7,168	19	76		
		1110	5	16	57,344	2	10		
		1111	6	20	983,040	1	6		
B	4	0	2	7	128	57	114	2,71	10.84
		10	3	10	896	21	63		
		110	4	13	7,168	19	76		
		111	6	21	2,088,960	3	18		
C	5	0	1	4	16	28	28	2.19	10.95
		10	2	8	240	36	72		
		110	3	12	3,840	26	78		
		1110	4	16	61,440	9	36		
		1111	5	21	2,031,616	1	5		
D	4	00	2	6	64	48	96	2.76	11.04
		01	3	10	960	30	90		
		10	4	14	15,360	21	84		
		11	6	22	4,177,920	1	6		
E	6	0	1	5	32	38	38	1.85	11.10
		10	2	10	992	40	80		
		110	3	15	31,744	21	63		
		111	4	21	2,064,384	1	4		
F	5	0	1	4	16	28	28	2.28	11.40
		10	2	8	240	36	72		
		110	3	12	3,840	26	78		
		111	5	22	4,190,208	10	50		
G	6	00	1	4	16	28	28	1.95	11.70
		01	2	10	1,008	50	100		
		10	3	16	64,512	21	63		
		11	4	22	4,128,768	1	4		
H	5	0	2	9	512	71	142	2.34	11.70
		10	3	13	7,680	25	75		
		110	4	17	122,880	3	12		
		111	5	22	4,063,232	1	5		
J	5	0	2	9	512	71	142	2.37	11.85
		10	3	13	7,680	25	75		
		11	5	23	8,380,416	4	20		
K	5	00	2	8	256	64	128	2.41	12.05
		01	3	13	7,936	32	96		
		10	4	18	253,952	3	12		
		11	5	23	8,126,464	1	5		
L	6	0	1	5	32	38	38	2.06	12.72
		10	2	10	992	40	80		
		11	4	22	4,193,280	22	88		
M	6	0	2	11	2,048	84	168	2.17	13.02
		10	3	16	63,488	15	45		
		11	4	22	4,128,768	1	4		

The specific example at the right in figure 2 means that the first word is decoded from the octal number 133 (or 000000000000001011011, in the full 21 bit address). There are many different methods of encoding for various byte sizes. Some are shown in Table 1. The example of Figure 2 is of Type B. The per cent usage figures are taken from Dewey's frequency study of 100,000 words [4]. The average number of bits per word may perhaps be slightly reduced from these figures by optimum adjustment to English frequency to the closest bit, rather than by the closest byte. However, this slight reduction is not warranted by the extra hardware and processing time. Note that the control patterns may be inverted or reassigned with exactly the same effect. Arbitrary change in the decoding rules is convenient for encrypting messages.

Correct positioning may be maintained for Method 3 by:

1. Using intervening pulses of different length, as in Teletype
2. Inserting synchronizing groups of all ones (1111 1111 1111) which have been excluded by the computer fromthe legitimate numbers sent
3. Checking for reasonableness of message through statistical methods
4. Guaranteeing that synchronization is never lost through self-shecking methods (addition of parity bits, error-detecting and correcting codes, etc.)

If an out-of-phase condition is likely, the proper receiving technique is to use a buffer area so the message can be re-interpreted. The amount of saving in this method will allow even the entire message to be sent twice, as an extreme measure. If the situation becomes intolerable in actual practice, Method 4 may be employed. Both Method 3 and 4 have full storage utilization, as opposed to Brillouin's Method 2.

In Method 4, a single track is reserved for a wordmark. This wordmark can delimit in either of two ways, as shown in figure 3.

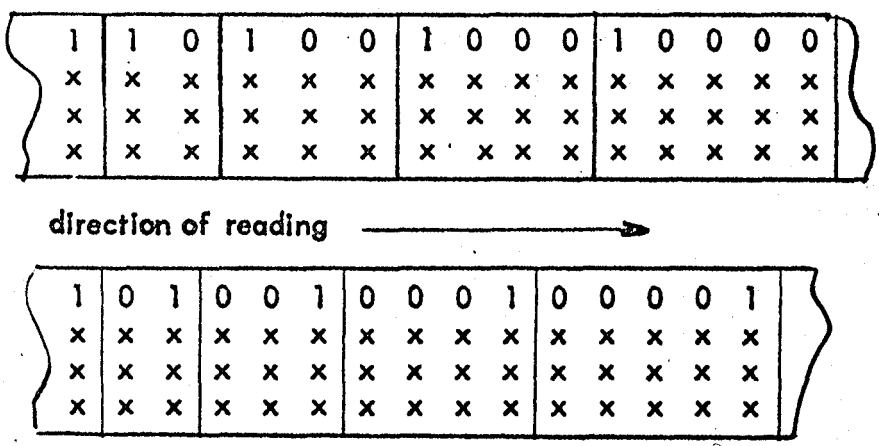

Figure 3.

At first appearance, this does not achieve the efficiency of Method 3. However, for all cases where the minimum number of bytes is two (i.e., Types A, B, D, H, J, K, M) the position adjacent to the initiating or terminal one may be used for information. This is because two one bits in succession in the wordmark track constitutes an illegal condition.

A third method may deserve study, that of using alternate ones and zeros in the wordmark track, changing at the start of each new word. The advantage of using one extra bit for information is lost in this scheme. Table 2 shows corresponding efficiency in the word-mark mode.

Method 3 Type	bits per byte	word mark bits	no. of bytes	work-ing bits
A	4	x1	2	7
		x01	3	10
		x001	4	13
		x0001	5	16
		x00001	6	19
B	4	x1	2	7
		x01	3	10
		x001	4	13
		x00001	6	19
C	5	1	1	4
		01	2	8
		001	3	12
		0001	4	16
		00001	5	20
E	6	1	1	5
		01	2	10
		001	3	15
		0001	4	20

Table 2. (Method 4)

The corresponding efficiencies in bits per word are virtually the same as those of Table 1.

For transmission on parallel wires, Method 4 is superior to Method 3 when errors occur. It is also better for single wire transmission provided synchronization can be maintained regardless of error in any bit.

Code Efficiency

Brillouin [3] sates that his code yields about 12 bits per word, very close tothe theoretical lower limit that Shannon [2] believed to be 11.82. It is obvious from Table 1 that Types A through H are all better than Shannon's limit. Having duplicated Brillouin's work with the approximation to word frequency that he used (and also the Deway frequencies), I get 10.12 bits per word with the frequency approximation that he used and 9.83 bits per word with the Dewey frequencies that Shannon used. This latter figure leads me to believe that Brillouin may have unknowingly penalized his scheme by 2 bits.

Although Baudot code is nominally 5 bits per character, the effective average is probably closer to 6 because of the extra shift characters and terminal blanks required to space words. Fieldata code does not use shift characters but does require terminal blanks and at least one extra bit, so the average is nearly seven bits per character. Thus the scheme outlined in this paper will save between 60 and 65 per cent over Baudot transmission and neqrly 70 per cent over Fieldata.

n	2^n	Z=2		Z=3		Z=4		Address Efficiency		
		U	Σ U	U	Σ U	U	Σ U	Z=2	Z=3	Z=4
1	2	1	1	1	1	1	1	.500	.500	.500
2	4	2	3	2	3	2	3	.750	.750	.750
3	8	4	7	4	7	4	7	.875	.875	.875
4	16	7	14	8	15	8	15	.875	.938	.938
5	32	12	26	15	30	16	31	.813	.938	.969
6	64	20	46	28	58	31	62	.719	.906	.969
7	128	33	79	52	110	60	122	.617	.859	.953
8	256	54	133	96	206	116	238	.520	.805	.930
9	512	88	221	177	383	224	462	.432	.748	.902
10	1024	143	364	326	709	432	894	.355	.692	.873
11	2048	232	596	600	1309	833	1727	.291	.639	.843
12	4096	376	972	1104	2413	1606	3333	.237	.589	.814
13	8192	609	1581	2031	4444	3096	6429	.193	.542	.785
14	16384	986	2567	3736	8180	5968	12397	.157	.499	.757
15	32768	1596	4163	6872	15052	11504	23901	.127	.459	.729
16	65536	2583	6746	12640	27692	22175	46076	.103	.423	.703
17	131072	4180	10926	23249	50941	42744	88820	.083	.389	.678
18	262144	6764	17690	42762	93703	82392	171212	.067	.357	.653
19	524288	10945	28635	78652	172355	158816	330028	.055	.329	.630
20	1048576	17710	46345	144664	317019	306128	636156	.045	.292	.607

Table 3.

Brillouin codes of the form 1xxxxx.000

n = number of digits to left of the decimal point
Z = number of zeros in the terminator
U = number of usable combinations in each group

$U_n = U_{n-1} + U_{n-2} + 1$ (for Z=2)

$U_n = U_{n-1} + U_{n-2} + U_{n-3} + 1$ (for Z=3)

$U_n = U_{n-1} + U_{n-2} + U_{n-3} + U_{n-4} + 1$ (for Z=4)

Address efficiency $= \frac{\Sigma U}{2^n}$

Brillouin has not published a scheme using more than two zeros as terminal indicators, and has stated that he believes it to be the most efficient possible. I have investigated the cases for three and four zeros termination, with these results:

Number of zeros	Bits per word (Shannon)	Bits per word (Dewey frequency)
2	10.12	9.83
3	10.66	10.10
4	13.65	12.03

These figures are for vocabularies of from 12,000 to 20,000 words. Table 3 shows that in this range the address efficiency of the three zero terminator is about five times as good as that for two zeros. Thus it is actually much superior for practical computer operation. The reason an extra zero does not add a full bit per word is that a higher proportion of the frequently used words may now be assigned to a denser set of addresses. The remaining choice between Brillouin's method and Methods 3 and 4 is now made as follows:

	Brillouin 00	Brillouin 000	Method 3,4	Baudot (Teletype)
Bits per word	9.83	10.10	10.62*	30.
Address efficiency	.06	.35	1.00	---

* adjusted for corresponding vocabulary size

Transformation Methods

A stored-program computer or a device with associative memory can transform a string of characters (each represented by a binary number) into a single compact and unique representation. An existing example of this statement is the conversion from binary coded decimal to pure binary numbers. The converse transformation at the receiving end requires only a simple address lookup to find the word or symbol to be printed.

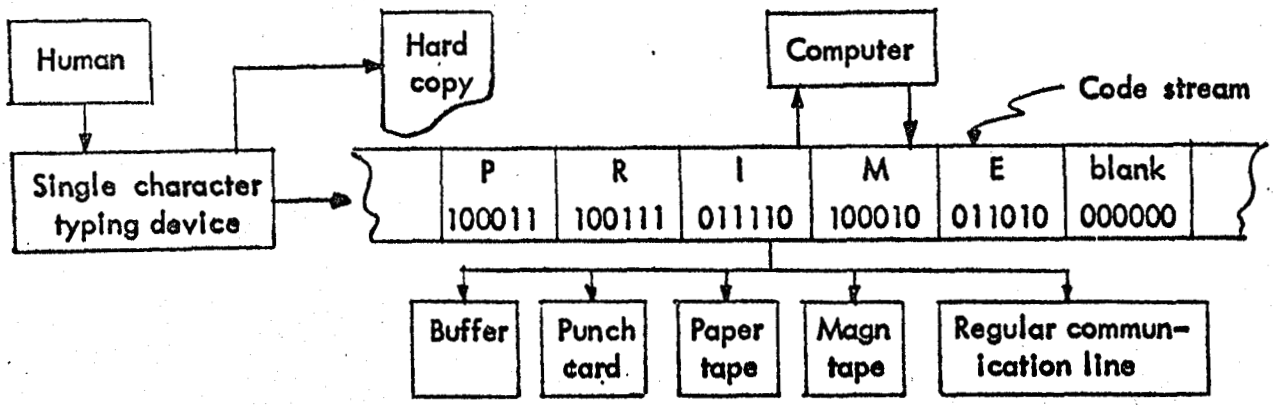

Figure 4.

The input typing or keying device produces B bits per character or letter. $B \geq 6$ in order to handle at least 26 alphabetic characters, 10 digits and other necessary characters such as punctuation. A group of letters is delimited by a blank, hyphen or other delimiter. Figure 4 shows the bit stream produced in the EIA code for the word PRIME. The blank delimits a string $(C_1, C_2, C3, \ldots . C_N)$, N being the number of letters in the group, normally a word. This string is a number R_1; in Figure 4 $R_1 = 4347364232_{octal}$. The blank triggers the unloading of the buffer to a unit or program which transforms R_1 to a number R_t, which is the compressed representation transmitted on the communications line.

As the set of English words is sparse, the set of R_1 is also sparse. Transmission efficiency increases as the set of R_t is denser. R_t is also the address used at the receiving end of the line, $(R_t) = R_1$, which may either activate a character printing device or be re-transmitted in decompressed form.

R_t is chosen to increase monotonically as frequency of the use of a word decreases. Thus the most frequently used words have lowest values of R_t and may thus have the leading zero bits truncated in the variable length mode. The experimental program on the IBM 709 has an optional tally register associated with each word. Actual usage will generate practical frequencies which may be used for reassignement of the R_t values.

R_t may represent more than one word. After initial R_t s are formed, one per word, this compressed string may be inspected by matching pairs against a list of pairs which have high enough usage frequency to warrant condensation into a single R_t. This is recursive and any number of words may be represented by a single R_t. The only requirement is for the preceding R_t to remain in a buffer for matching.

Table Lookup Method

The length of a word is expressible in 5 bits. (N_{max} in English = $28 < 2^5$ for the word antidisestablishmentarianism). The computer storage is arranged to contain the operating program, a master table of N, C_1 and subordinate tables corresponding to all these values. For each word, N and C_1 are adjoined. (N, C_1) are found in the master table and is the address of the start ofthe proper table. In figure 4 the value of N, C_1 is 0543_{octal}. (0543) = starting address for table of all five letter words starting with P. $(N, C_1 + 1)$ = ending address of table + 1. Between these limits, a binary search finds a match to the value of R_1. Associated with R_1 in the table is the corresponding R_t.

This method does not make use of frequency information. It may be desired to place the tables randomly in storage. In this case the master table must be doubled in size. Adjoin the 5 bits of N, 6 bits of C_1 and a final low-order bit which indicates the starting address by 0, the ending address by 1. Finer grouping may be had at increased cost by using the adjoint of N, C_1, C_2.

Chaining Method

The entire number R_1 is utilized directly to find the corresponding R_t. A number M is chosen such that 2^M is convenient to storage size and related to vocabulary size for optimum conversion speed. For present storage sizes, M may vary from 10 to 14. The address R_1 modulo M has the contents:

R_1, R_t, chain address

The set of numbers $R_m = R_1$ modulo M will have duplicates and will not be dense also, although denser than the set of R_1. When there are duplicates, the contents of R_m will not contain the proper R_1 except for one word. If not, the contents of the chain address are tested for a match. This proceeds recursively until a match is found on R_1; the R_t associated with that address is then used. For dense storage packing the chain addresses are chosen from a list of empty positions, that is, R_m values which the existing vocabulary does not utilize.

It is possible to apply other transformations to R_1 to reduce it to the range from 1 to 2^M. A simple extraction of M bits may be practical. For any value of R_m, the chain should be assigned in order of decreasing frequency of word usage.

Storage is assigned by referring to its representation is three lists:

1) Storage already assigned to a word (prime)
2) Storage already assigned to a word (secondary or nonprime)
3) Free List, not yet assigned

A limiting number is experimentally chosen such that only this many words are allowed unlimited prime assignment. Starting with the most frequently used word, R_m is calculated and used as an address. If this address is found on the free list, it is removed and placed on the prime storage list. The address contents are assigned, using successively larger values of R_t. If this address is not found on the free list, a duplication in R_m has occurred. Such words are held aside for assignment after the limitingnumber is reached. These remaining words are then taken again in order of highest frequency of usage, and the remainder of the free list is used in sequence to fill the chaining addresses. Each word assigned must proceed thru its chain. For example, take three words W_1, W_2 and W_3 for which the corresponding values of R_{1_1}, R_{1_2} and R_{1_3} all yield identical values of R_m.

W_1 is assigned	$(R_m) = R_{1_1}, R_{t_1},$	FLA_1 (FLA means Free List Address)
W_2 is assigned	$(FLA_1) = R_{1_2}, R_{t_2},$	FLA_2
W_3 is assigned	$(FLA_2) = R_{1_3}, R_{t_3},$	RETURN

Ends of all chains are assigned to the RETURN address. When a word must be added to the chain, it is lengthened by replacing RETURN by the next chaining address and putting RETURN in the new last word. If RETURN is ever reached in actual transmission, it indicates that this word is not yet in the dictionary. Automatic addition of this entry than occurs, upon inspection of the free list.

Immediate Applications

Present day computers operate at speeds too high for constant usage with communication lines, except under special circumstances. Until special devices are built for this express purpose, there are several ways of efficiently combining computers with existing communication lines.

1) Message Center

Since the computer should be running nearly continuously to realize maximum savings from compression, one means of achieving economy is to create message centers in such cities as Paris, New York, London, etc., where the total volume of messages may be expected to approach capacity. Since the communication volume on lines is only about a third of present volume, capacity of such lines as Atlantic cables is tripled without needing to lay down new cables.

The extreme flexibility of the computer allows a variety of modes of compression, as shown in Table 1. Some of these are suitable to the existing five bit pattern of Teletype. Thus a computer equipped with paper tape input and output could take in continuous strings of normal Teletype messages, compress them, and output a continuous string in condensed form but still suitable for transmission on regular Teletype circuits and equipment. This tape then enters the receiving computer and is either printed on its equipment or converted to the expanded tape suitable for relaying to local Teletype receivers. Large networks could thus be two-stage, with the greater proportion of distance (and cost) being traveled in the compressed form. The compression algorithm must avoid using normal control codes as any part of the numbers. The same principles apply equally to other existing and proposed paper tape formats of six bits and more.

2) Combined Message and Data Processing Center

Many computers are susceptible to external interruption of their regular operations, and can intermingle several different jobs. This is known as multi-programming. Thus a computer in regular operation can be interrupted upon demand to either encode a message for transmission or decode and print an incoming message. However, demand does not need to be heeded instantaneously. A minimum time lag would be that saved by the message compression. Depending on priority codes, lags of up to several minutes may be acceptable. This would allow for regular jobs to be interrupted at convenient points with minimum disruption.

3) Combined Message and Language Translation Center

Several language translation schemes depend partially on corresponding dictionary lookup. In this method the receiving computer can look up the corresponding word in Russian or French just as easily as in English. Since this lookup time is such a small proportion of available real time, the rest of the translation process may be carried on simultaneously. This allows messages to be sent to multiple receivers in different languages through virtually instantaneous translation.

4) Encoding for Secrecy

Secrecy comes virtually free with this code. Whereas ciphers depend upon letter substitution and are normally broken on the basis of letter frequency in a language, it is quite another thing to try to determine relative frequencies for thousands of words rather than just 26 letters. Code is the more general term, being substitution of symbols for words. The self-measuring quality inherent in zero compression is another factor which aids secrecy. Once the key is lost it becomes virtually impossible to pick up the message again.

Normally both sending and receiving computers are equipped with the same dictionary, or library. When the sender adds a new word to the dictionary, it must transmit this word to the receiver in both single character (so it will know what to print) and compressed form. Depending upon the mode of number assignment, this may cause a drastic restructuring of the entire encoding representation. In the example of the dictionary with words numbered in order, suppose a new word must be added just before "aardvark". Almost every word would have its representational number increased by one. This simple shift would be easy to detect, but the problem would be infinitely more difficult if many words were added at random places throughout the dictionary. Now imagine this ordering of numbers determined not by alphabetic ordering in the dictionary but by frequency of usage in the language.

Additions to the dictionary may be expected quite often. New users of the communications system may introduce new professional jargons. Personal or place names are used to identify many things, from army tactical positions to tropical storms. Mixed symbols such as part numbers for inventory will be used often. If these are popular or frequently used, it is more economical to add them to the compression dictionary than to send them by single characters. If not, the provision does exist to send single characters in groups of one, four, five, six or eight bits.

A particular business requiring secrecy could purchase its own special dictionary, scrambled in an unique way. Computers can store a multiplicity of these on tape files and both sending and receiving computers could select one upon the basis of a control code. Multi-programmed computers can merge several messages together for simultaneous transmission in a variety of patterns. A very simple example would be to interleave five messages so every fifth bit would belong to a specific message. Myriad varieties are possible and simple, but the unauthorized receiver would have to try every possible variety before he could make sense from any.

A valuable by-product of this method will be the ability (at last) to determine actual usage and frequency figures for both letters and words in languages. The compression program contains a counting mechanism for usage. This may be disconnected at option. This is useful to periodically rearrange the dictionary for efficiency, when operating in a standard non-secret mode. Previous counts, on sample texts of from 100,000 to 300,000 words, did not count punctuation symbols for frequency of occurrence. In this method, it is more economical to use these symbols as being identical to words.

Amortization of Computer Costs

Rough calculations of conversion time show it to be negligible by almost two orders of magnitude to the transmission costs saved. Local estimates* are 62 cents for an average message consisting of 48 words of 5 characters and blank, requiring an overall elapsed time of one minute. These 288 characters actually require only 14.4 seconds on line at 20 characters per second transmission rate. Thus if four messages could be interleaved, the cost per word would be

$$\frac{62}{4 \times 48} = \underline{.323 \text{ cents per word}}$$

A rough program planned for a 709 type of machine gives an average of 120 cycles per word for conversion at both transmitting and receiving ends. At 2 microsec/cycle this gives

.24 ms/word or over 4100 words/second

This is more than adequate to keep up with foreseeable transmission times on a real time basis. The cost, at $ 800 an hour, would be

$$\frac{80000}{3600 \times 4175} = \underline{.0053 \text{ cents per word}}$$

Net cost with computers on each end should approximate

(35% of .323) + .0053 = .1183 cents per word

Thus the overall cost may be expected (with a fully utilized computer) to be from 37% to 40% of present costs.

* M. Conlon, X2292, CHQ

Example of Possible Assignments (Type D)

Octal	Symbol	Frequency*	Octal	Symbol	Frequency*
00	(Open for contingency)		40	ARE	1200
01	Enter binary mode		41	ON	"
02	Enter 4-bit (decimal) mode**		42	OR	1100
03	Enter 6-bit mode**		43	HER	"
04	Enter 8-bit mode**		44	HAD	"
05	Blank		45	AT	"
06	,		46	FROM	1000
07	THE	15500	47	THIS	"
10	.		50	MY	1000
11	OF	9800	51	THEY	"
12	AND	7600	52	ALL	900
13	TO	5700	53	THEIR	800
14	A	5100	54	AN	"
15	IN	4300	55	SHE	"
16	THAT	3000	56	HAS	"
17	IS	2500	57	WERE	"
20	I	2300	60	ME	700
21	IT	2300	61	BEEN	"
22	;		62	HIM	"
23	FOR	1900	63	ONE	"
24	AS	"	64	SO	"
25	WITH	"	65	IF	"
26	WAS	1800	66	WILL	"
27	HIS	1700	67	THERE	"
30	HE	1700	70	WHO	700
31	BE	1500	71	NO	"
32	NOT	"	72	WE	600
33	BY	1400	73	WHEN	"
34	BUT	"	74	WHAT	"
35	HAVE	1300	75	YOUR	"
36	YOU	"	76	MORE	"
37	WHICH	"	77	(Open for contingency)	

TABLE 4

* 242,000 word sample from Cryptanalysis, H. E. Gaines, Dover, 1956.

** 4,6 and 8-bit modes have identical characters to IBM LOGICODE standard. Those modes are provided to allow single character formation. The 4-bit set is provided with a special blank following 0 - 9 . - + . Return to Normal Mode is effected in the various sets by encountering the character

01111100	in the 8-bit set	
111110	in the 6-bit set	(If 6-bit return character is out-of-phase with
1110	in the 4-bit set	the end of the byte, hold up 2 bits)

REFERENCES

1. Draft Standard, EIA Committee TR 24.4

2. Shannon, C. E. Bell System Tech. Journal 30, 50-58 (1951)

3. Brillouin, L., Science and Information Theory -- Academic Press, New York, 24-58, (1956)

4. Dewey, G., Relativ Frequency of English Speech Sounds, Harvard University Press, 1923