16.186/December 1970*
Laboratory Report/IBM Confidential

EVALUATION OF ERROR-DETECTION CODES FOR 2400-BAUD DIGITAL TRANSMISSION OVER VHF RADIO LINKS

F. B. Wood

International Business Machines Corporation
Advanced Systems Development Division
Los Gatos, California

ABSTRACT

The tests described in Laboratory Report 16.171**verified the feasibility of 2400-baud digital transmission over VHF radio as a means of increasing the capacity of voice mobile public safety communication channels. Data derived from these tests on a phase modulation-synchronous detection system included the percentage of error-free messages for each transmission path, plus error distributions for the messages containing errors. The present study was undertaken because automatic error detection and correction by retransmission will be essential in applications such as the proposed Mobile Terminal System. This study used computer simulation to examine several recommended error-detection codes in relation to the observed error distributions.

In the 33,000 error patterns processed, simple parity checks missed 6% of the messages in error; longitudinal redundancy checks missed 2%; and a combination of simple parity and LRC missed 1.4%. Nine polynomial codes were tested and all proved very successful. The SABRE/PARS code (polynomial CCC:$X^6 + X^5 + 1$) missed only 0.1%, and the other eight polynomial codes detected every message in error. In addition to the SABRE/PARS code, two other IBM codes were tested--CRC-12 and CRC-16, plus a proposed CCITT code, two Fire codes, a Bose-Chaudhuri-Hocquenghem code, and two interlaced Hamming codes.

*Draft completed December 1969.
**Wood, F. B., "Reliability of 2400-Baud Digital Transmission over VHF Mobile Radio Links," Los Gatos, California, September 15, 1968.

LOCATOR TERMS FOR THE IBM SUBJECT INDEX

Communications, land-based mobile
Terminals, portable
05 Computer Application
Codes, error detection

CONTENTS

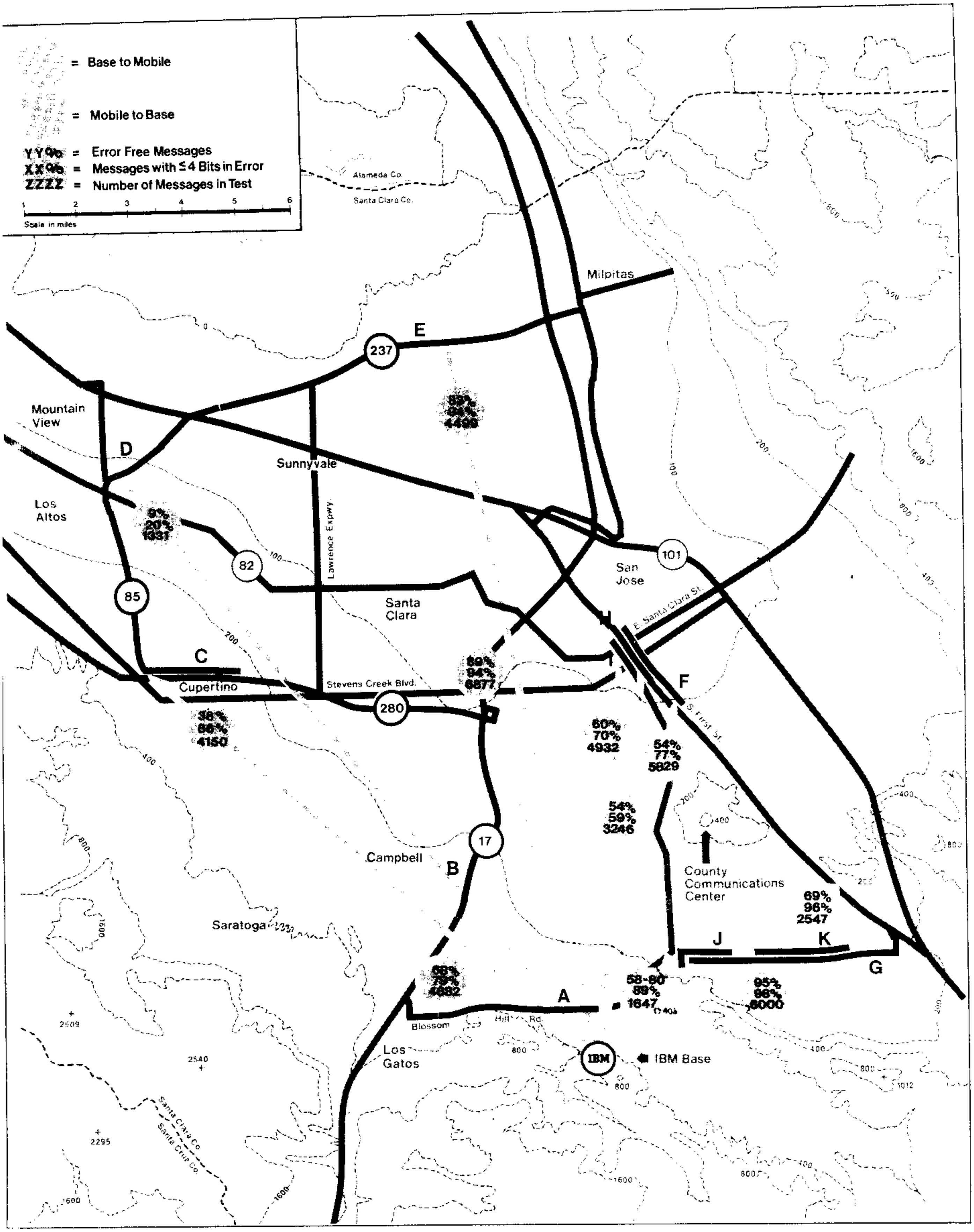

Fig. 1. The Santa Clara Valley, Showing Transmission Paths and Test Results (Bold black lines with letter codes are the test routes followed by the mobile unit. The test routes are further described in Table 2.)

I. INTRODUCTION

In the summer of 1967, 2400-baud digital data transmission was tested over VHF radio links in a series of experiments involving a mobile unit in contact with a base station at the Los Gatos Laboratory. Those tests (described in Laboratory Report 16.171[1]) confirmed the feasibility of using digital transmission to increase the message capacity of overcrowded voice communication channels. The demands of public safety communication systems (police, fire, ambulance) and of commercial radio-dispatched services create a need for more efficient use of the available channels.

A Mobile Terminal System has been proposed which could improve communications in a police department, for instance, by transmitting most messages to and from mobile units in computer-coded form. Since a high degree of accuracy would be essential in such a system, error detection and automatic retransmission would also be essential. The digital transmission tests were therefore planned so that complete error data would be available as a realistic basis for the choice of error detection codes for a working system.

The mobile unit, a station wagon, cruised around the Santa Clara Valley so that a number of different transmission paths could be tested (see Fig. 1). The 2400-baud signals were transmitted over 151 MHz FM radio links. From selected sample areas, prerecorded phase-modulated audio waveforms were transmitted from the station wagon to the Laboratory. A tape recorder in the station wagon recorded the messages transmitted from the base station, i.e., recorded the analog waveforms from the discriminator of the VHF radio receiver. Each message consisted of 127 bits as shown in Fig. 2. Later, in the Laboratory, the tapes were read through a demodulator into a special terminal control unit on the paper tape channel of an IBM 1620 computer.

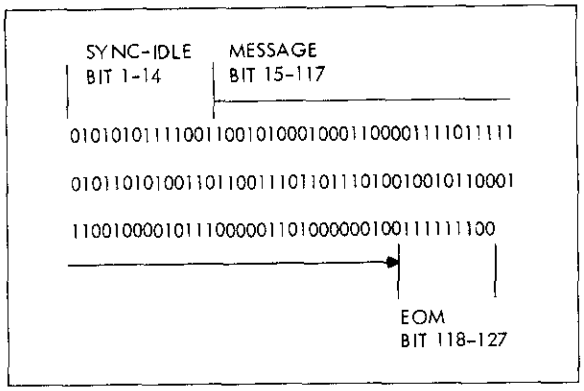

Fig. 2. 127-Bit Standard Message Used in Tests

Table 1. Summary of Test Results

TEST	NUMBER OF MESSAGES	% ERROR FREE	% ≤ 4 ERRORS
BASE TO MOBILE			
A Country Road, 1 to 3 miles	4682	69	80
B Intercity Freeway, 4 to 10 miles	6877	89	94
C City Street and Depressed Highway, 10 to 13 miles	4150	38	66
D Industrial Freeway, 14 miles (Intermodulation Noise)	1331	9	20
E Country Highway, 13 miles	4499	83	94
F Downtown Street, 6 to 7 miles	5829	54	77
G Suburban Through Street, 2 to 5 miles	6000	95	98
MOBILE TO BASE			
H Downtown Street, 6 to 7 miles (14-Bit Sync Idle)	3246	54	59
I Downtown Street, 6 to 7 miles (15-Bit Sync Idle)	4932	60	70
J Suburban Through Street, 2 to 5 miles (14-Bit Sync Idle)	2547	69	96
K Suburban Through Street, 2 to 5 miles (15-Bit Sync Idle)	1647	58-80	89

Both sets of messages--those to and from the mobile unit--were stored on disk files by a message control program. A 1620 message analysis program compared each message as received with the standard message transmitted. If a message had any error in text or discrepancy in control characters, cards were punched with the complete message as received by the computer, plus flag codes indicating any input/output correction procedures that were necessary to blank out illegal characters generated by the errors. The cards also identified the particular test and message involved, and contained a tabulation of the error count. Table 1 summarizes the test results in terms of error-free messages and messages containing four errors or less. Table 2 indicates the effects of various transmission paths in Santa Clara County.

The cards were then fed to an IBM 1800 computer for statistical analysis of the error distributions (see Report 16.171 for the results). After the analysis the error data was in the form of condensed-format cards which included the octal pattern of each error. The data was then stored on magnetic tape to be used as the raw material for the evaluation of various error detection codes in the study reported here. That material remains available to check error detectability with additional codes if that seems advisable.

II. CHOICE OF ERROR DETECTION CODES TO BE STUDIED

The plan was to use the accumulated error data and simulate the effect of various codes to determine their relative effectiveness. In choosing the codes to be tested, we were influenced by the nature of the projected application, the experience of others with error detection codes in similar communication systems, and the particular characteristics of the set of error data provided by the transmission experiments.

Our simulation was organized on two levels: (1) on the basis of the number of bits per character (8, 7, or 6), and (2) on the basis of 15 error-detection codes that might be considered.

Simulation of an eight-bit character code was included because of the existence of IBM EDCDIC and ASCII-8 codes, and the possibility that the United States of America Standards Institute may establish a USASII-8 code for information interchange.

Seven-bit character codes were simulated because the present USASII standard for information interchange is a seven-bit code. It was felt that, if a standard were set for police communications, it would likely be USASCII or some future standard.

Table 2. Error Distributions for Different Transmission Paths

	BASE TO MOBILE							MOBILE TO BASE			
	TEST A Country Road, 1 to 3 mi.	TEST B Intercity Freeway 4 to 10 mi.	TEST C City St. & Depressed Highway, 10 to 13 mi.	TEST D Industrl. Freeway 14 mi. (Intermodulation Noise)	TEST E Country Highway 13 mi.	TEST F Downtwn Street 6 to 7 mi.	TEST G Suburb. Thru-Street 2 to 5 mi.	TEST H Downtwn Street 6 to 7 mi. (14-Bit Sync Idle)	TEST I Downtwn Street 6 to 7 mi. (15-Bit Sync Idle)	TEST J Suburb. Thru-Street 2 to 5 mi. (14-Bit Sync Idle)	TEST K Suburb. Thru-Street 2 to 5 mi. (15-Bit Sync Idle)
Messages Sent	4682	6877	4150	1331	4499	5829	6000	3246	4932	2547	1647
Short Messages (%)											
126 bits	6.14	1.60	9.00	18.30	2.16	6.90	0.70	14.90	10.60	2.12	3.8
63 bits	5.34	1.05	8.83	23.05	1.86	6.60	0.55	14.64	10.88	1.18	3.3
31 bits	3.23	0.44	7.15	21.00	0	1.80	0.06	3.99	1.60	0.08	0.4
Error-Free Messages (%)	68.40	88.60	37.90	9.45	83.00	54.00	94.80	53.90	60.20	69.06	57.5
Sync Bad, But Text OK	0.15	0.01	0.41	0.02	0.07	0.01	0.07	0.18	1.50	0	22.9
Recoverable Messages	68.55	58.61	38.31	9.47	83.07	54.01	94.87	54.08	61.70	69.06	80.4
ERROR DISTRIBUTIONS %											
No. of Errors 1	4.75	2.53	10.11	3.46	5.15	10.02	2.29	2.14	5.25	19.70	6.3
Cumulative %	73.31	91.14	48.42	12.93	88.22	64.03	97.16	56.22	66.95	88.70	86.7
2	3.78	1.98	10.64	3.38	3.70	7.60	0.74	1.34	1.75	5.81	1.3
Cumulative %	77.09	93.12	59.06	16.31	91.92	71.63	97.90	57.56	68.70	94.57	88.0
3-4	2.46	1.56	6.68	3.90	1.68	5.20	0.35	1.04	1.20	1.38	1.3
Cumulative %	79.55	94.68	65.74	20.21	93.60	76.65	98.25	58.60	69.90	95.89	89.3
5-8	2.29	1.43	4.26	5.78	1.12	4.70	0.38	1.77	0.95	0.16	0.9
9-10	1.11	0.34	1.62	2.55	0.40	1.30	0.09	0.64	0.41	0.04	0.1
17-32	0.42	0.04	0.85	1.50	0.04	0.10	0	0.09	0.08	0.39	0.1
33-64	2.14	0.38	2.92	7.59	0.61	1.50	0.55	5.38	6.10 + 1.50[s]	0.31	6.5 + 22.9[s]
65-127	0.04	0	0.01	0.01	0	0	0	0.03	0	0	0

Six-bit character codes were considered because there are computer-communication systems in the field that might be modified for police communications using six-bit codes. We also found some opinion that a smaller character set such as 47 character plus a few control characters might be adequate for police use.

A. CODES SIMULATED BY REASON OF COMPATIBILITY WITH EXISTING HARDWARE AND SOFTWARE

Although it may be desirable to design new communication adapters to connect police radio systems to a computer, it is important to know the performance of existing systems that might be adapted to police use.

1. Parity and Longitudinal Redundancy (LRC) Codes

Since there are many computer systems with either six-bit or seven-bit codes, we simulated adding a parity bit to each character and dividing the 127-bit test messages both into eight-bit groups and seven-bit groups. A common method of providing a higher level of error detection than simple parity is to add a longitudinal redundancy check at the end of each message or block of characters. In these simulations we included tests of parity bits on each character, longitudinal redundancy checks at the end of messages, and the two combined.

2. Codes Available with IBM Binary Synchronous Communication (BSC) Adapters

For BSC, two polynomial codes are provided, namely, CRC-16 and CRC-12. CRC-16 uses two eight-bit characters for a cyclic code check at the end of each block. This code is derived from the product of two primitive polynomials:

$$(X + 1)(X^{15} + X + 1) = (X^{16} + X^{15} + X^{2} + 1), \text{ or } 11000000000000101.$$

The properties of this code as summarized by Gorog[2] are as follows:

(1) It detects any three errors (minimum distance of four) in a message of length less than or equal to $2^{15}-1$. Minimum distance of four means that every allowable message in the set differs in at least four bit positions from every other message in the set.

(2) It also detects any group of two bursts of two errors each in a message of the same length as above.

(3) It requires minimum hardware since with only two non-zero coefficients (in addition to the end terms X^{16} and X^{0}) only two internal Exclusive-OR circuits (EOR's) are required. (For the theorems from which the properties of the code can be derived, see Ref. 3.)

CRC-12, specified for six-bit Transcode, is a 12-bit polynomial code derived as follows:

$$(X + 1)(X^{11} + X^{2} + 1) = (X^{12} + X^{11} + X^{3} + X^{2} + X + 1),$$

or 110000000~~1~~1111.

This code has the first and second properties listed for CRC-16, but requires more hardware. An alternate code suggested but not tried has the capabilities of CRC-12 but requires fewer internal EOR circuits (two less than needed by CRC-12). The alternate code is derived in this way:

$$(X^{12} + X^{10} + X^{5} + 1), \text{ or } 1010000100001.$$

This and other alternatives can, of course, be evaluated on the basis of the accumulated error data if that is considered desirable in the future.

3. Code in Use in Airline-Reservation System

A six-bit polynomial code designated as CCC is used in the IBM-PARS (Programmed Airline Reservation System). Since the adaptation of PARS is one way of developing a police resource allocation system, it is desirable to know the performance of this model:

$$(X^{6} + X^{5} + 1), \text{ or } 1100001.$$

B. CODE CONSIDERED BY REASON OF BEING A POTENTIAL STANDARD

There are a number of 16-bit cyclic codes which have all the capabilities listed for CRC-16 plus some additional advantages. One such code was proposed by Standard Elektrik, A.G. and Siemens, A.G. in their joint contribution to the CCITT:

$$(X^{16} + X^{12} + X^{5} + 1), \text{ or } 10001000000100001.$$

This code can detect two larger error bursts (of three, four, or maybe five bits in shorter message lengths, depending on the message length chosen) in addition to those detectable by the CRC-16 code.

C. CODES CONSIDERED BY REASON OF HIGHER DETECTION CAPABILITY

Certain ground rules for the choice of error-detection codes were suggested by Tang[4] on examination of the particular error distributions which occurred in the digital transmission tests. After noting that error detection is the main goal, since retranmission will be requested when errors are detected, Tang makes three main points:

(1) "The 64-character or 512-bit memory to be used implies a code length no longer than 512 bits. Preferably the code length should be equal to an integral fraction of 512 bits, such as 128 bits or 256 bits."
(2) "It is necessary to detect more than 5 bit errors in a block of 127 bits."
(3) "There is no indication that only one or two short bursts usually occur in such a block. However, it seems reasonable to assume that errors tend to cluster."

On the basis of this analysis, Tang suggests a number of possible codes. (These codes are typically referred to by the name of the person who devised them.)

1. Fire Codes

Fire codes offer many possibilities, such as these two examples:

a. $g(X) = (X^{10} + 1)(X^6 + X + 1)$, with 16 check bits.

The natural code length $N = 630 > 512$; it can detect two bursts of length b_1 and b_2 where $b_1 + b_2 \leqslant 11$.

b. $g(X) = (X^{19} + 1)(X^5 + X^2 + 1)$, with 24 check bits.

The natural code length $N = 589 > 512$; it can detect single bursts of length 24 or less, or combinations of two bursts whose sum $(b_1 + b_2)$ is less than or equal to 20 bits and the shorter burst is less than or equal to 6 bits in length.

2. Bose-Chaudhuri-Hocquenghem (BCH) Codes

BCH codes also offer various possibilities, such as:

$$g(X) = (X^8 + X^4 + X^3 + X^2 + 1)(X^8 + X^6 + X^5 + X^4 + X^2 + X + 1)(X^8 + X^7 + X^6 + X^5 + X^4 + X^2 + 1),$$ with 24 check bits.

This code with a natural length $N = 255$ can detect 6 random bit errors. Adding an overall parity bit to the above code will increase by one the code length, number of check bits and number of detectable errors.

3. Hamming Codes

Two interleaved Hamming codes are suggested:

a. "An augmented Hamming (128, 120) code can be obtained by generating a code with $g(X) = X^7 + X^3 + 1$ and add an overall parity bit. Interleaving this code at a 4-bit period, one has a (512, 480) code capable of detecting two bursts of length 4 and also combinations of many random errors. There are 32 check bits."[4]

b. "Augmenting a Hamming code generated by $g(X) = X^5 + X^2 + 1$ with an overall parity, one obtains a (32, 26) code. This code can be interleaved at an 8-bit period to yield a (256, 208) code capable of detecting two bursts of length 8 or many combinations of random errors. There are 48 check bits."[4]

4. Reed-Solomon Codes

Reed-Solomon codes are also possible candidates:

"With 8-bit characters, the natural code length of R-S codes is $N = 8 \times 255 > 512$ bits. One can denote the 255 non-zero characters by the powers of a certain 'primitive element', α. The polynomial $(g(X) = (X-\alpha)(X-\alpha^2)----(X-\alpha^t)$ then generates a code capable of detecting t character errors. There are t check characters or 8t check bits. For instance, to detect 5 character errors, one needs 40 check bits."[4]

Table 3. Characteristics of Polynomial Codes Selected.

List No.	Polynomial	Position Last Digit
Code 1	11000000000000101	17
Code 2	10010000001000001	17
Code 3	10000110001000011	17
Code 4	1001010000000000000100101	25
Code 5	1101111001100001011111111	25
Code 6	110011011	9
Code 7	1101111	7
Code 8	1100000001111	13
Code 9	1100001	7

D. SUMMARY OF CODES CHOSEN FOR SIMULATION IN THIS STUDY

The different codes simulated in the computer programs in this study are identified as follows:

Code A - Single parity bit on 7-bit code making 8-bit byte including parity.
Code B - Longitudinal redundancy check on string of 8-bit bytes.
Code C - Single parity bit on 6-bit code group making 7-bit character.
Code D - Longitudinal redundancy check on string of 7-bit characters.
Code E - Combination of parity and LRC using both Codes A and B.
Code F - Combination of parity and LRC using both Codes C and D.

The polynomial codes were identified in the simulation programs by hexadecimal numbers D1 through D9, which are labeled Code 1 through Code 9 and the undetected error counts were labeled by the alphanumeric equivalents J through R of the hex numbers D1 through D9.

Code 1 (D1 or J) - IBM Cyclic Redundancy Code CRC=16 using two eight-bit bytes for a polynomial redundancy check
Code 2 (D2 or K) - Proposed Comite Consultative Internationale Telephone et Telegraf (CCITT) Cyclic Code of 16 bits
Code 3 (D3 or L) - A Fire Code of 16 check bits for burst error detection
Code 4 (D4 or M) - A second example of a Fire Code with 24 check bits
Code 5 (D5 or N) - A Bose-Chaudhuri-Hocquenghem (BCH) error detection code of 24 check bits
Code 6 (D6 or O) - Four Hamming Codes of eight check bits interlaced
Code 7 (D7 or P) - Eight Hamming Codes of 5-bit code plus parity interleaved for a 48-bit check word
Code 8 (D8 or Q) - IBM Cyclic Redundancy Code CRC-12 using two six-bit characters
Code 9 (D9 or R) - IBM Cyclic Redundancy Code CCC using one six-bit check character as in SABRE/PARS

Table 3 characterizes the nine polynomial codes, indicating the binary representation of the code polynomial, the position of the last digit, and the degree of interlace. The last digit position is used to start the modulo-2 division at the right place in the simulated decoding of the error patterns. An interlace number of one means there was no interlace, because only one polynomial was used in the decoding. The interlace values of 4 and 8, respectively, for Codes 6 and 7 indicate that 4 and 8 identical polynomials were used in the simulated decoding, with the four polynomials operating on bits k, k+1, k+2, and k+3, respectively, for Code 6.

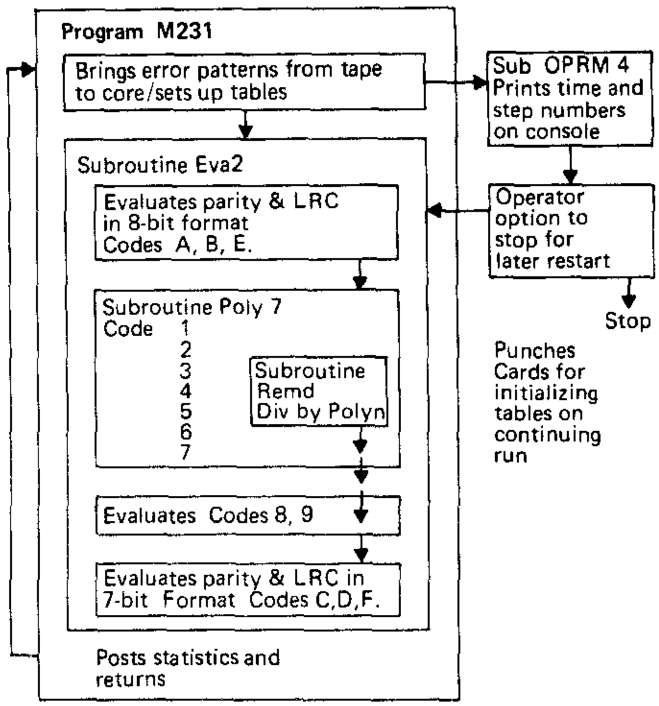

Fig. 3. Outline of Computer Program to Evaluate Error Detecting Codes

III. SIMULATION OF ERROR DETECTION

As Fig. 3 indicates, the first step in the simulation program is to bring a batch of error messages (up to 150) from tape to core. The size of the batch was determined by limitations on the transfer of full message patterns from the 1620 computer to the 1800 computer for condensing the error pattern data prior to entry into the System/360 Mod 50. The selection of the number of messages in a batch was influenced by the original message format, characteristics of the card reader on the 1800 computer, requirements for time-sharing of the 1800, and other factors.

The simulation program requires large core and long CPU time on the Mod 50. In order to prevent the simulation program from blocking ordinary job shop time sharing under MVT (Multiple Variable Task Time

Sharing) features were included to permit cancellation of the job at selected points where partial results could be saved for restarting the program later. The Subroutine OPRM4 was provided to type out time and step information at the console so that the operator could decide when to cancel the simulation program in the interest of providing fair time-sharing for other projects. When cancelled, the program punches control cards to be read in as data cards when the program is restarted.

In simulating the parity check, the program counts the number of errors, if any, in each pseudo-character in the message. If there are none a period (.) is stored in the parity string. If one or more errors is counted, the modulo-2 value of the count (i.e., '0' or '1') is stored for that character. The results can be seen in Fig. 4, which is a debugging printout for one 128-bit message (127-bit message plus an extra '0'). It shows the error-detecting capabilities of all the codes in relation to that particular error pattern or message. The parity string just referred to is the seventh line, which begins with "VRC(PAR)."

The error pattern being analyzed is displayed in octal form in the third line (starting with 'K'). The header information means that the error pattern is from test number 23029, message number 12, and that two bits were found to be in error. (These numbers come after 'K', 'P', and 'R', respectively.)

As pointed out in the figure, the error pattern in this case is

```
    1  0  2    --octal in the third line, or
  001000010    --binary in lines 6, 43 and 46
```

When the message is divided into groups representing eight-bit characters, the two errors fall into different characters and both are detected. This can be read from (again) the seventh line of the printout which represents Code A and Code B. On the seventh line after "VRC(PAR)", each period (.) indicated no error in that pseudo-character. A one (1) indicates a detectable error.

When the message is divided into seven-bit groups (see lower part of figure, lines 43 and 44) the two errors fall into one group, so that the parity string contains all periods with exception of one '0', and this means that the parity check fails.

The error detection failures are summarized on cards V and X. At the end of each batch of 150 messages, grand totals are entered on cards W and Y, as shown in Figs. 5 and 6. For example, on line 2 of card X, the '1' to the right of 'C' means that in this batch of messages prior to message No. 12, Code C had one failure. After the failure shown in line 44, we note that in line 52 the number to the right of 'C' has changed from '1' to '2'.

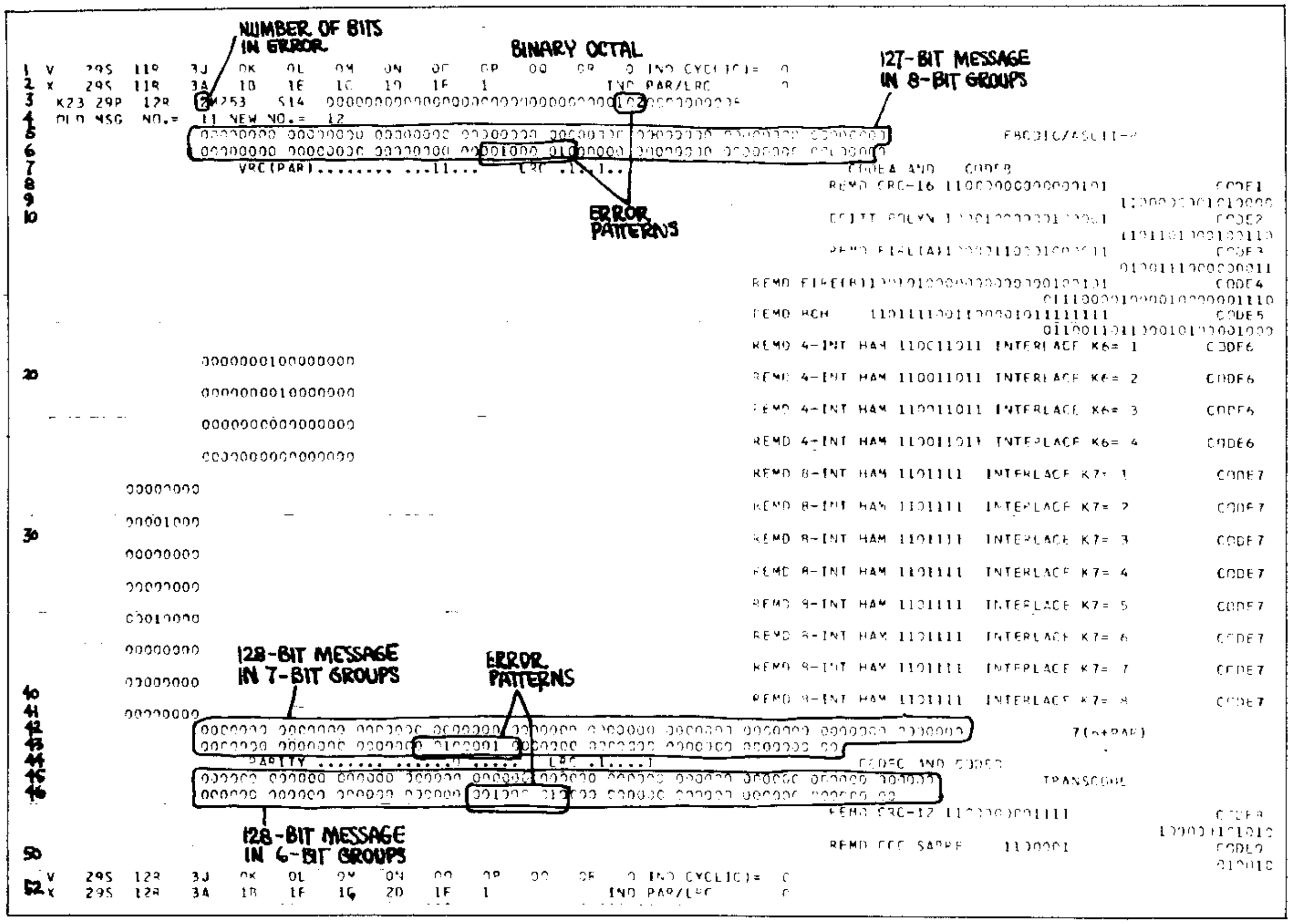

Fig. 4. Debugging Output for One Cycle of Error-Detection Simulation

ERROR DETECTABILITY OF PARITY AND LRC CODES

TEST NO. MSG ERR/UNDETECTED ERRORS BY CODE

Test No.	Msg	Err	Undetected								
Y23665	S4101	R2975A	164B	30E	25C	109D	33F	22	TOT PAR/LRC	FILE 1	1
Y23217	S5399	R4499A	204B	30E	21C	21CD	26F	21	TOT PAR/LRC	FILE 2	0
Y23263	S5829	R3223A	487B	80E	68C	467D	93F	68	TOT PAR/LRC	FILE 3	0
Y23124	S6870	R6112A	151B	21E	8C	143D	13F	10	TOT PAR/LRC	FILE 4	0
Y23315	S6900	R6544	61B	17E	16C	62D	21F	19	TOT PAR/LRC	FILE 5	0
Y23158	S4150	R1518A	509B	73E	49C	509D	69F	48	TOT PAR/LRC	FILE 6	0
Y25522	S3246	R1764A	137B	131F	91C	159D	112F	91	TOT PAR/LRC	FILE 7	0
Y25633	S4932	R2275A	197B	200E	159C	213D	187F	159	TOT PAR/LRC	FILE 8	0
Y25444	S2547	R1756A	34B	69F	1C	11D	2F	1	TOT PAR/LRC	FILE 9	0
Y25585	S1647	R 947A	22B	4E	3C	17D	3F	2	TOT PAR/LRC	FILE 10	0
Y23166	S1181	R 130A	123B	33E	28C	149D	33F	28	TOT PAR/LRC	FILE 11	0
TOTALS	46802	33543	2081	696	475	2093	605	477			

"Y-CARD"

TEST NUMBER

5829 MESSAGES IN TEST

6112 MESSAGES CONTAINED ERRORS

151 ERRORS WERE NOT DETECTED BY CODER

Fig. 5. Failures of Parity and LRC Codes

ERROR DETECTABILITY OF POLYNOMIAL CODES

TEST NO. MSG ERR/UNDETECTED ERRORS BY CODE

"W-CARD"

TEST NO. MSG ERR											FILE
W23665S4101R2975J	OK	OL	OM	ON	OO	OP	OQ	OR	0	TOT(CYCLIC)= 0	FILE 1
W23217S5390R4499J	OK	OL	OM	ON	OO	OP	OQ	OR	0	TOT(CYCLIC)= 0	FILE 2
W23263S5829R3223J	OK	OL	OM	ON	OO	OP	OQ	OR	12	TOT(CYCLIC)= 0	FILE 3
W23124S6870R6112J	OK	OL	OM	ON	OO	OP	OQ	OR	1	TOT(CYCLIC)= 0	FILE 4
W23315S6900R6544J	OK	OL	OM	ON	OO	OP	OQ	OR	1	TOT(CYCLIC)= 0	FILE 5
W23158S4150R1618J	OK	OL	OM	ON	OO	OP	OQ	OR	7	TOT(CYCLIC)= 0	FILE 6
W25522S3246R1764J	OK	OL	OM	ON	OO	OP	OQ	OR	1	TOT(CYCLIC)= 0	FILE 7
W25633S4932R2975J	OK	OL	OM	ON	OO	OP	OQ	OR	2	TOT(CYCLIC)= 0	FILE 8
W25444S2547R1756J	OK	OL	OM	ON	OO	OP	OQ	OR	2	TOT(CYCLIC)= 0	FILE 9
W25585S1647R 947J	OK	OL	OM	ON	OO	OP	OQ	OR	1	TOT(CYCLIC)= 0	FILE 10
W23166S1181R 130J	OK	OL	OM	ON	OO	OP	OQ	OR	5	TOT(CYCLIC)= 0	FILE 11
TOTALS	0	0	0	0	0	0	0	0	32		

4150 MESSAGES

1618 IN ERROR

7 NOT DETECTED BY CODER

Fig. 6. Cumulated Failures for Polynomial Codes

Subroutine POLY7 divides the error pattern by the polynomial corresponding to the code and saves the remainder which is determined by subroutine REMD. In Fig. 4, the remainder is printed on the line following the line labeled with the code number. For example, division by the CRC-16 polynomial gives a remainder of 1100000001010000 at the right hand end of line 9. If the remainder is all zeros (possible, of course, in the event that the error pattern equals the code polynomial or is a multiple of the code polynomial), the code does not detect that there is an error in the message.

Two sample hand calculations of the remainders are given in Appendix A as Figs. A1 and A2. As may be noted in Fig. 4, the remainders are computed for each interlaced section for the interlaced Hamming codes 6 and 7.

The evaluation of codes 8 and 9 follows the algorithm of POLY7 but with different constants. These codes were handled under EVA2 instead of POLY7 because of a maximum size limitation on the size of the subroutines. To evaluate these two codes, the program regroups the error message into six-bit characters.

For the seven-bit character parity and LRC simulation, subroutine EVA regroups the error pattern and then applies the same parity and LRC check procedures as for the eight-bit case.

IV. RESULTS AND CONCLUSIONS

Figure 5 lists the failures of parity and LRC codes, derived from the Y-card counters indicated in Fig. 4. Figure 6 shows the cumulated failures for the polynomial codes, from the W-card counters also marked in Fig. 4.

Only one polynomial code failed to detect all the errors, namely, the SABRE/PARS code. This code missed 32 messages of the 33,000 messages in error, for a failure rate of 0.1%. The undetected error rates are not significantly different for seven-bits-plus-parity and six-bits-plus-parity. Simple parity checks missed 6% of the messages in error; longitudinal redundancy checks missed 2%; and the parity/LRC combination missed 1.4%.

These results lead to the conclusion that for the digital transmission method under consideration--phase modulation with synchronous detection of 2400-baud signals on 151 MHz FM radio links--polynomial codes of 12th degree or higher are needed to assure near-perfect error detection. The sample of 33,000 error patterns is not large enough to give us the error rate for 12-bit polynomial codes. However, we can make a first approximation to the rates for higher order codes by the following approximate formulas:

Given an undetected error rate for CCC (6th order polynomial) of 1 in 1000 messages in error, the estimated rate of a 12th order polynomial is 1 in 64,000 messages in error (that is, 1000×2^6), and the estimated rate for a 16th order polynomial is 1 in 1,024,000 messages in error (that is, $64,000 \times 2^4$).

It should be noted that these estimates are limited to the particular conditions of the 2400-baud tests described in IBM Laboratory Report 16.171. For other conditions, one would need to make new tests or at the least project the effect of changed conditions on the error rate and error pattern distribution. It should also be noted that the present study does not go into the relative effectiveness of different codes of the same order.

REFERENCES

1. Wood, F. B., "Reliability of 2400-Baud Digital Transmission over VHF Mobile Radio Links," IBM Advanced Systems Development Division Laboratory Report 16.171, Los Gatos, California, September 15, 1968.
2. Gorog, Etienne, private communication, May 1968.
3. Peterson, W. W. Error Detecting Codes. John Wiley, New York; and MIT Press, 1961.
4. Tang, D. T., private communication, July 19, 1968.

APPENDIX A. SAMPLE DIVISION BY CODE POLYNOMIALS

As a check on the simulated operation of the polynomial error detection codes, two samples from Fig. 4 are worked out in detail. These provide a means of verifying the accuracy of subroutines POLY7 and REMD. Figure A1 is an example of dividing by the polynomial for Code 1; Figure A2, for Code 2.

Code 1

```
                                                1111 1000 0000 0010 000  Dividend
1100 0000  0000 0010 1 | 1000 0100 0000 0000 0000 0000 0000 0000 000  Error
Polynomial Divisor       1100 0000 0000 0010 1                          pattern
                          100 0100 0000 0010 10
                          110 0000 0000 0001 01
                           10 0100 0000 0011 110
                           11 0000 0000 0000 101
                            1 0100 0000 0011 0110
                            1 1000 0000 0000 0101
                              1100 0000 0011 0011 0
                              1100 0000 0000 0010 1
                                             11 0001 1000 0000 000
See Fig. 4, Line Code 1 and                  11 0000 0000 0000 101
remainder on line below Code 1.                   1 1000 0000 1010 000  Remainder
```

Fig. A1. Example of Division by Code 1 Polynomial.

Code 2

```
                                               1000 1100 1101 1100 110  Dividend
1000 1000 0001 0000 1 | 1000 0100 0000 0000 0000 0000 0000 0000 000  Error
Code Polynomial         1000 1000 0001 0000 1                          pattern
                             1100 0001 0000 1
                             1000 1000 0001 0000 1
                              100 1001 0001 1000 1
                              100 0100 0000 1000 01
                                  1101 0001 0000 11
                                  1000 1000 0001 0000 1
                                   101 1001 0001 1100 1
                                   100 0100 0000 1000 01
                                     1 1101 0001 0100 11
                                     1 0001 0000 0010 0001
                                       1100 0001 0110 1101
                                       1000 1000 0001 0000 1
                                        100 1001 0111 1101 1
                                        100 0100 0000 1000 01
                                            1101 0111 0101 11
                                            1000 1000 0001 0000 1
                                             101 1111 0100 1100 1
                                             100 0100 0000 1000 01
                                               1 1011 0100 0100 001  Remainder
```

Fig. A2. Example of Division of Code 2 Polynomial.

APPENDIX B. ERROR PATTERNS NOT DETECTED BY CODE 9 (SABRE/PARS)

The 32 error patterns that Code 9 failed to detect in the simulation error detection are listed in Table B. The number after 'K' is the test number, the number after 'P' is the message number, and the number after 'R' is the number of digits in error. A letter 'M' before the symbol 'S14' indicates that the end-of-message control character was missing, i.e., lost or slipped out of position. The 43-digit octal string represents 129 binary digits. The first 127 digits represent the error pattern. The last two are dummies inserted by the binary-to-octal conversion. The last octal digit is overlaid by symbol 181 to represent the end-of-message character. When the EOM character is lost, '18' is artificially placed in octal digits 42 and 43. When a section of the message is lost, the lost part is filled in with 8's.

Figures B1 and B2 illustrate the division by the polynomial to obtain zero remainder yielding an undetected error. Figure B1 is for 'K23340 P 128' in File 3; Figure B2 is for 'K23141 P 2' in File 6.

```
                      1        Code 9
1100001 | 00 1100 001          Oct '141'
             1100 001
                    0
```

Fig. B1. Polynomial Division for 'K23340 P 128' in File 3

```
                 1001111
1100001 | 011010011011110     Oct '32336'
           1100001
             1000101
             1100001
              1001001
              1100001
               1010001
               1100001
                1100001
                1100001
                      0
```

Fig. B2. Polynomial Division for 'K23141 P 2' in File 6.

Table B. Error Patterns Not Detected by CCC Code

TEST NO.	MSG	ERR LN	*		ERRCR PATTERN IN OCTAL	END=8
						FILE 1
						FILE 2
						FILE 3
K23227P	39R	2M253		S14	0000002000000000000000000002000000000000008	
K23229P	96R	3M253		S14	0000000000000000000000000000220000000004008	
K23233P	112R	5M253	M	S14	0000000000000000000000000000001020100000618	
K23238P	80R	4M253		S14	0000000400000000000000101000001000000000008	
K23238P	90R	4M253		S14	0000000400000000000000101000001000000000008	
K23240P	12R	11M253		S14	0000000014040000000400000000000200000000008	
K23240P	128R	3M253		S14	0000000000000000000000001410000000000000008	
K23242P	21R	6M253		S14	0000000000000000601400000000000060000000008	
K23243P	32R	4M253	M	S14	0000000000030000000000000000001000000004018	
K23255P	52R	5M253		S14	0000012000000000030000000000010000000000008	
K23255P	74R	4M253		S14	0000000000000000000000000010000001000014008	
K23257P	55R	0M171	M	S14	0000000000040000000000000055118888888888368888	
						FILE 4
K23109P	75R	4M253	C	S14	0000000000000000034000000400000000000000008	
						FILE 5
K23309P	12R	2M253		S14	0000010000000000000000000010000000000000008	
						FILE 6
K23132P	115R	54M253	M	S14	0000127462421417372641547335113071027015018	
K23133P	74R	54M253	M	S14	0000127462421417372641547335113071027015018	
K23134P	67R	54M253	M	S14	0000127462421417372641547335013071027015018	
K23136P	60R	3M253		S14	0000000000000600000000000000000000000040008	
K23141P	2R	9M253		S14	0000032336000000000000000000000000000000008	
K23151P	4R	5M253	M	S14	0000000000000000000004000050000000300000018	
K23156P	95R	50M253	M	S14	0000076355664413441104600233606570044430376	
						FILE 7
K25522P	131R	53M253	M	S14	0000005243034761523316662216162056046111778	
						FILE 8
K25602P	128R	3M253		S15	0000000000000002010000004000000000000000008	
K25633P	9R	12M253		S15	0000002070000002372401100000000000000000008	
						FILE 9
K25439P	37R	21M253	M	S14	0000000000000000000000005111113071027015018	
K25444P	89R	3M253		S14	0000000000000000000000000001410000000000008	
						FILE 10
K25579P	147R	56M253	M	S15	0000027462421417372651547326162056032011778	
						FILE 11
K23161P	11R	12M253		S14	0000000070000200000100000140620060000000008	
K23161P	56R	6M253		S14	0000000200000000000300000040000000000140008	
K23163P	21R	7M253		S14	0000000020400010000000000010000020000006008	
K23163P	63R	4M253	M	S14	0000000000000000000004000000001030000000018	
K23166P	58R	55M253	M	S14	0000073351130710070150047742536312106077758	