CPE3300-STD-CSMACD-UNI-ME-003revB
1 January 2024

DATA COMMUNICATIONS INTERFACE STANDARD

Version 3

INTEROPERABILITY STANDARD FOR A CSMA/CD WAVEFORM USING UNIPOLAR MANCHESTER CODING ON A MULTIPLE-ACCESS BUS THAT IDLES AT HIGH LEVEL

0. HISTORY

This standard has been written and maintained by multiple faculty of the Milwaukee School of Engineering for use by computer and electrical engineering students in the first computer networking and data communications course.

- Rev A – original assignment for AY2024
- Rev B – updated version fixing error in specification for collision state

1. PURPOSE

This standard establishes mandatory requirements for multiple-access baseband communications over an electrical cable. It is applicable to a computer or other device that functions as a node on the network. The requirements stated herein represent the minimum set required for interoperability.

2. APPLICABLE DOCUMENTS

The documents listed in this section are referenced in later sections of this standard. In the event of a conflict between the text of this standard and the references cited herein, the text of this standard takes precedence.

ISBN-13 978-0133506488 Data and Computer Communications, 10th ed., Stallings, W., Pearson Prentice-Hall, 2014.

3. ABBREVIATIONS AND ACRONYMS

The abbreviations and acronyms used in this standard are:

bps	bits per second
CD	collision detect
CMOS	complementary metal-oxide-semiconductor logic
CRC	cyclic redundancy check
CSMA	carrier sense multiple access
EOM	end of message
EOT	end of transmission
FCS	frame check sequence
LSB	least significant bit
ME	Manchester encoding
ms	millisecond(s)
MSB	most significant bit
Rx	receive
s	second(s)
TTL	transistor-transistor logic
Tx	transmit
UNI	unipolar
V	volts
µs	microsecond(s)

4. REQUIREMENTS

General Description: The waveform defined in this standard allows multiple devices to transmit and receive short text messages access a single electrical cable. Digital baseband signals are used on the electrical cable to transfer the messages. Each message can have up to 255 text characters sent within a data packet. Each packet adds a header containing control information. Each text character within the message field of the transmission is represented by a byte of data consisting of the ASCII representation of the character.

4.1 Physical Layer Requirements.

4.1.1 Mechanical: Each node that exchanges text messages shall include a cable that includes a plug for interfacing to the network hub. The network hub combines the transmit data signals (Tx_Data) from the network nodes plugged into the hub. The combined transmit data signal exists on a single electrical bus within the network hub, and the hub routes the bus to a set of drivers that each feed a receive data signal (Rx_Data) back to a network node.

The plug on the cable from a network node shall be an 8P8C type modular plug (also referred to as an RJ45 type plug or Ethernet cable connector plug). The network hub will have an 8P8C type socket for each connective device. An 8P8C plug with numbered pins is depicted on Figure 1. The pins used for TX (from the node to the network hub) and RX (from the network hub to the node) are identified in Table 1.

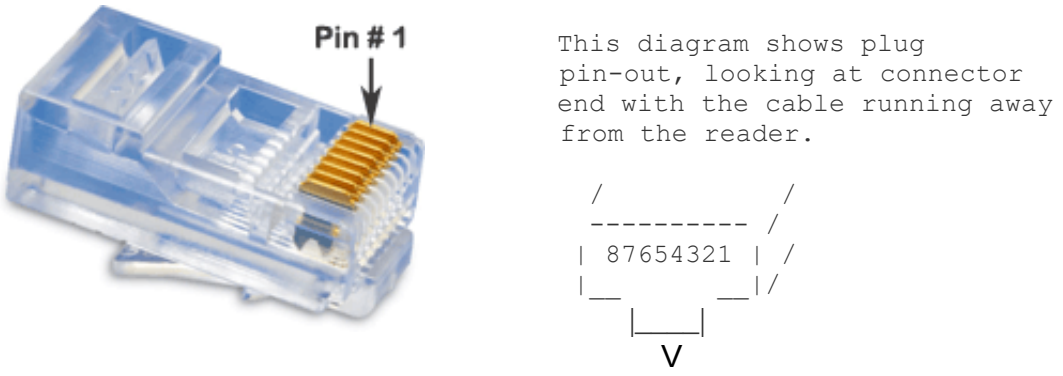

Figure 1. Mechanical plug, type 8P8C (also called RJ45)

[Note: When using the hub within the laboratory, the instructor will provide to each team a cable having an 8P8C type modular plug on one end and, on the other end, spliced wires identified for Tx_Data, Tx_Data_Ground, Rx_Data, and Rx_Data_Ground.]

Table I. Pin Usage for Plug Interfacing Node to Network Hub

PIN NUMBER	SIGNAL NAME	DESCRIPTION
1	Tx_Data	transmit data signal
2	Tx_Data_Gnd	ground for transmit data signal
3	Rx_Data	receive data signal
4	(not used)	(not used)
5	(not used)	(not used)
6	Rx_Data_Gnd	ground for receive data signal
7	(not used)	(not used)
8	(not used)	(not used)

4.1.2 Electrical requirements.

Signal voltage accuracy, Manchester (ME) coding, and data rate/clock accuracy:

- The hub supports nominal line voltage values that are the standard HCMOS voltage levels of +5V for the high level and 0V for the low level. It also supports LVCMOS voltage levels of +3.3V for the high level and 0V for the low level.
- The node shall transmit and receive data using Manchester line coding as described in Stallings. Logic-0 bits are encoded as logic-1 in the first half of the bit interval and logic-0 in the second half. Logic-1 bits are encoded as logic-0 in the first half of the bit interval and logic-1 in the second half. The idle line voltage is logic-1.
- The transmission data rate shall be 1000 bits per second (bps), corresponding to a bit time interval of 1.0000ms, and a half-bit time interval of 500.00µs. Each half-bit time interval from a transmitting node shall have a duration within ±1.32 percent of the nominal value of 500.00µs.

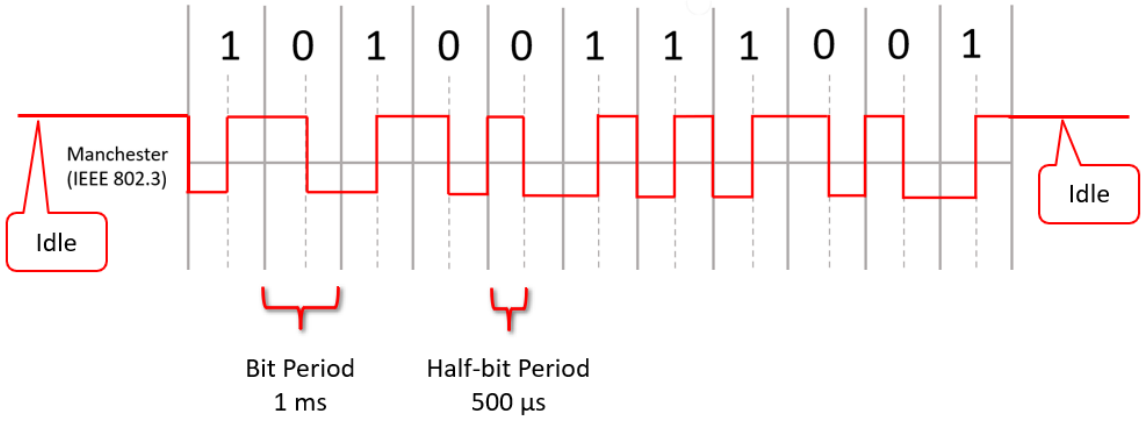

Figure 2. Depiction of Manchester encoding with idle high.

4.2 Data-link-layer requirements.

CSMA/CD: The multiple-access network will employ a bus topology. Medium access will be controlled by a Carrier Sense Multiple Access / Collision Detect (CSMA/CD) protocol implemented in each network node.

Each node shall output a constant high signal level when it is not transmitting a message.

Each node shall check the bus to ensure it is idle before transmitting data. The ***idle*** state is the condition where the bus level is logic-1 and no transitions have been detected on the bus for at least the past 1.1 bit intervals (1.11ms, if the clock period is assumed to be 1.32% long at the other, possibly transmitting node). It is suggested that a participating node declares the bus idle if the bus level has been high for a measured time of at least 1.13ms. This allows the node monitoring the bus to have a 1.32 percent error in the clock it uses to measure the time and allows for some measurement error. A time slightly longer than 1.13ms could also be used but this longer threshold should not be above 1.18ms because that would unnecessarily delay the declaration of idle state within the protocol software.

A node enters the ***collision state*** while transmitting if it detects a logic-0 on the bus for a measured time longer than 104% of a bit interval (1.04ms). A time slightly longer than 1.04ms could also be used but the detection threshold should not be above 1.14ms because that could result in an unacceptable percentage of collisions going undetected. The node shall stop transmitting when it enters the collision state and wait a random amount of time between 0s and 1.000s before checking for the idle condition to begin a retransmission. If the network has gone idle after the random backoff, the node shall attempt transmission immediately. The minimum number of retransmission attempts before timeout is ten.

Random number selection for retransmission after collision detect: The node is expected to wait a random time when a collision is detected. The random wait time should start at the time the collision state no longer exists. The random wait time shall be uniformly distributed between 0 and 1.000s using the equation

$$w = \frac{N}{N_{MAX}} * 1.000s$$

where N is selected as a uniformly distributed random integer between 1 and N_{MAX}, provided that N_{MAX} is at least 128. For example, if N_{MAX} is chosen to be 200, then the random wait time will be equal to (N)*(5.000 milliseconds).

Data packet format: Each text message of up to 255 characters shall be transmitted within a formatted data packet. The data packet shall have a header field followed by a message field followed by a cyclic redundancy check (CRC) trailer field for error detection. Each field shall contain an integer number of bytes. For each byte of

transmitted data, the eight bits shall be transmitted most significant bit (MSB) first.

Header field format: The five-byte header shall be transmitted in the order specified in Table II.

Table II. Header Format

Byte	Name	Description
1	Preamble	0x55
2	Source	The transmitter address.
3	Destination	The receiver address.
4	Length	The message field length. Valid lengths are 1 to 255 bytes.
5	CRC Flag	A CRC on/off flag. CRC off = 0x00. CRC on = 0x01.

Message field format: The data in the message field shall immediately follow the header. Each byte in the message field contains an ASCII-coded character byte.

An example packet for a single-byte message of ASCII letter 'A' between machines with address 8 and 82 is:

HEADER					MESSAGE	TRAILER
Preamble	Source	Destination	Length	CRC Flag	'A'	CRC8 FCS
0x55	0x08	0x52	0x01	0x01	0x41	0xC0

CRC trailer field format: The CRC trailer field shall consist of one byte. If the CRC check is not being used then this field shall contain the sequence 10101010 (that is, <0xAA>). If the CRC check is being used, then this field shall contain the FCS byte for the message calculated using the CRC-8 polynomial

$$x^8 + x^2 + x + 1$$

Reception of messages and use of broadcast address 0: Each node should ignore all messages unless the destination address field within the header contains either the broadcast address 0 or the node address. The node shall present the received text message on its display only if the received message contains either the broadcast address 0 or the node address. It is suggested (but not required) that any characters in a received message that are not printable characters be printed as asterisk (*) characters.

If the CRC check is being used, a failed CRC checks should be reported to the console.

Appendix I. ASCII Code.

Table A.I shows the 8-bit ASCII-coded character for each alphanumeric character and symbol.

Table A.I ASCII code.

US ASCII Character Codes (000-063)				US ASCII Character Codes (064-127)			
Binary	Decimal	Hexadecimal	Character	Binary	Decimal	Hexadecimal	Character
00000000	000	00	NUL	01000000	064	40	@
00000001	001	01	SOH	01000001	065	41	A
00000010	002	02	STX	01000010	066	42	B
00000011	003	03	ETX	01000011	067	43	C
00000100	004	04	EOT	01000100	068	44	D
00000101	005	05	ENQ	01000101	069	45	E
00000110	006	06	ACK	01000110	070	46	F
00000111	007	07	BEL	01000111	071	47	G
00001000	008	08	BS	01001000	072	48	H
00001001	009	09	HT	01001001	073	49	I
00001010	010	0A	LF	01001010	074	4A	J
00001011	011	0B	VT	01001011	075	4B	K
00001100	012	0C	FF	01001100	076	4C	L
00001101	013	0D	CR	01001101	077	4D	M
00001110	014	0E	SO	01001110	078	4E	N
00001111	015	0F	SI	01001111	079	4F	O
00010000	016	10	DLE	01010000	080	50	P
00010001	017	11	DC1	01010001	081	51	Q
00010010	018	12	DC2	01010010	082	52	R
00010011	019	13	DC3	01010011	083	53	S
00010100	020	14	DC4	01010100	084	54	T
00010101	021	15	NAK	01010101	085	55	U
00010110	022	16	SYN	01010110	086	56	V
00010111	023	17	ETB	01010111	087	57	W
00011000	024	18	CAN	01011000	088	58	X
00011001	025	19	EM	01011001	089	59	Y
00011010	026	1A	SUB	01011010	090	5A	Z
00011011	027	1B	ESC	01011011	091	5B	[
00011100	028	1C	FS	01011100	092	5C	\
00011101	029	1D	GS	01011101	093	5D	]
00011110	030	1E	RS	01011110	094	5E	^
00011111	031	1F	US	01011111	095	5F	_
00100000	032	20	SP	01100000	096	60	`
00100001	033	21	!	01100001	097	61	a
00100010	034	22	"	01100010	098	62	b

00100011	035	23	#	01100011	099	63	c
00100100	036	24	$	01100100	100	64	d
00100101	037	25	%	01100101	101	65	e
00100110	038	26	&	01100110	102	66	f
00100111	039	27	'	01100111	103	67	g
00101000	040	28	(	01101000	104	68	h
00101001	041	29	)	01101001	105	69	i
00101010	042	2A	*	01101010	106	6A	j
00101011	043	2B	+	01101011	107	6B	k
00101100	044	2C	,	01101100	108	6C	l
00101101	045	2D	-	01101101	109	6D	m
00101110	046	2E	.	01101110	110	6E	n
00101111	047	2F	/	01101111	111	6F	o
00110000	048	30	0	01110000	112	70	p
00110001	049	31	1	01110001	113	71	q
00110010	050	32	2	01110010	114	72	r
00110011	051	33	3	01110011	115	73	s
00110100	052	34	4	01110100	116	74	t
00110101	053	35	5	01110101	117	75	u
00110110	054	36	6	01110110	118	76	v
00110111	055	37	7	01110111	119	77	w
00111000	056	38	8	01111000	120	78	x
00111001	057	39	9	01111001	121	79	y
00111010	058	3A	:	01111010	122	7A	z
00111011	059	3B	;	01111011	123	7B	{
00111100	060	3C	<	01111100	124	7C	|
00111101	061	3D	=	01111101	125	7D	}
00111110	062	3E	>	01111110	126	7E	~
00111111	063	3F	?	01111111	127	7F	DEL

Appendix II. Network Addresses

Broadcast	0xFF
Hub	0x01
Invalid	0x00
To be	
assigned	