Name	DID? Y/N/C	EID? Y/N/O	Carry Flags?	Simple Node Ms	Priority Group	Type	DID/EID/Flags	Complete Value	Format	Type Byte	Flag bit A EID->1	Flag bit B Flags->0	Flag bit C 0 default	Flag bit D 1 default	CAN MTI	CAN Content	MTI Byte
	(1 bit)	(1 bit)	(1 bit)	(1 bit)	(2 bits)	(4 bits)	(3 bits)	16 bits hex	3 bits	8 bits hex	1 bit	1 bit	1 bit	1 bit	15 bits hex	Goes at start of CAN data, if present	
Base Messages																	
Initialization Complete	N	N			0	8	0	3080	1	08			1	1	1087		Full Source Node ID
Verify Node ID Number	Y	N			0	10	4	30A4	6	dest NIDa	-	-	-	-	6ddd	0A	MTI byte 0x0A
Verify Node ID Number	N	N		Y	0	10	0	10A0	0	0A			1	1	00A7		
Verified Node ID Number	N	N			0	11	0	30B0	1	0B			1	1	10B7		Full Source Node ID
Optional Interaction Rejected	Y	N			0	12	4	30C4	6	dest NIDa	-	-	-	-	6ddd	0C	MTI byte 0x0C, MTI, error, optional information
Terminate Due to Error	Y	N			0	13	4	30D4	6	dest NIDa	-	-	-	-	6ddd	0D	MTI byte 0x0D, MTI, error, optional information
Protocol Support Messages																	
Protocol Support Inquiry	Y	N		Y	1	14	4	12E4	6	dest NIDa	-	-	-	-	6ddd	2E	MTI byte 0x2E
Protocol Support Reply	Y	N		Y	1	15	4	12F4	6	dest NIDa	-	-	-	-	6ddd	2F	MTI byte 0x2F, protocol flags
Event Exchange Messages																	
Identify Consumers	N	Y		Y	1	4	2	1242	0	24	Y		1	1	024F		EventID (no room for DestID!)
Consumer Identify Range	N	Y			1	5	2	3252	1	25	Y		1	1	125F		EventID w mask (no room for DestID!)
Consumer Identified	N	Y	Y		1	6	3	3263	1	26	Y	Y	valid	uncertain	126B		EventID (no room for DestID!)
Identify Producers	N	Y		Y	1	8	2	1282	0	28	Y		1	1	028F		EventID (no room for DestID!)
Producer Identify Range	N	Y			1	9	2	3292	1	29	Y		1	1	129F		EventID w mask (no room for DestID!)
Producer Identified	N	Y	Y		1	10	3	32A3	1	2A	Y	Y	valid	uncertain	12AB		EventID (no room for DestID!)
Identify Events	Y	N			1	11	4	32B4	6	dest NIDa	-	-	-	-	6ddd	2B	MTI byte 0x2B
Identify Events	N	N		Y	1	11	0	12B0	0	2B			1	1	02B7		
Learn Event	N	Y		Y	1	12	2	12C2	0	2C	Y		1	1	02CF		EventID
Producer/Consumer Event Report	N	Y		Y	1	13	2	12D2	0	2D	Y		1	1	02DF		EventID
Datagram Messages																	
Datagram (General)	Y	N			2	0	4	3404	4,5	dest NIDa	-	-	-	-	4/5ddd		Data (0-8 bytes)
Datagram Received OK	Y	N			2	12	4	34C4	6	dest NIDa	-	-	-	-	6ddd	4C	MTI byte
Datagram Rejected	Y	N			2	13	4	34D4	6	dest NIDa	-	-	-	-	6ddd	4D	MTI byte, error code
Stream Messages																	
Stream Initiate Request	Y	N			2	14	4	34E4	6	dest NIDa	-	-	-	-	6ddd	4E	MTI byte, buffer size (2 bytes), Source Stream ID (1 byte), reserved byte, flags (tagged=0x80)
Stream Initiate Reply	Y	N			2	15	4	34F4	6	dest NIDa	-	-	-	-	6ddd	4F	MTI byte 0x4B,buffer size (2 bytes), Source Stream ID (1 byte), Dest Stream ID, flags (tagged=0x80; error info)
Stream Data Send	Y	N			3	9	4	3694	7	dest NIDa	-	-	-	-	7ddd		(stream IDs inferred on CAN); 8 bytes data
Stream Data Proceed	Y	N			3	10	4	36A4	6	dest NIDa	-	-	-	-	6ddd	6A	MTI byte, Stream IDs (2 bytes)
Stream Data Complete	Y	N			3	11	4	36B4	6	dest NIDa	-	-	-	-	6ddd	6B	MTI byte, Stream IDs (2 bytes); optional length (4 bytes)

Full 16 Bit MTI calculation

CAN Calculation

This goes in OpenLCB CAN frame Variable Field
Logical OR this with 0x18000 to get top 17 bits of CAN header
ddd' refers to destination address

Y means carries flags in CAN header

0 gets more priority

coding
2=carries EID
4=carries DID
1=carries flags

Full value must be checked!

0=simple MTI
1=complex MTI

4=DestID datagram
5=DestID datagram last segment
6=DestID non-Stream
7=DestID stream data

If flags not specified, send and check 1 bits

d=dest NIDa
f=flags

Places these appear in code:

prototypes/C/libraries/OlcbTestCAN/obj/test
prototypes/C/libraries/OlcbCommonCAN/OpenLcbCan.h
prototypes/C/libraries/OpenLCB/OLCB_CAN_Buffer.cpp

prototypes/ObjectiveC/OpenLcbLib/OlcbMtiDefinitions.h
prototypes/ObjectiveC/OpenLcbLib/OlcbTestDefinitions.h
prototypes/ObjectiveC/OpenLcbLib/MtiReformat.c

prototypes/java/src/org/openlcb/can/MessageBuilder.java

prototypes/Arduino/libraries/OpenLCB/OpenLcbCan.h
prototypes/CBUS-PIC/canlib/frametypes.c