	Byte 0 Byte 1	1 1 1 1 2 2 2 2 2 2	Byte 4 Byte 5	Byte 6 Byte 7
	0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0	1 2 3 4 5 6 7 0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
		ATOMIC Reque	est Packets	
Byte 0 >	ackID $\searrow \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	destinationID	sourcelD	transaction wrsize srcTID
Byte 8 >	Extended Address [0:15]	Address [0:29]	ta yam sbs	double-word 0 [0:15]
Byte 16 >		double-word 0 [16:63]		Final CRC
	NWRIT	E, NWRITE_R, CASTOUT, FLUSH wit	h data Reguest Packets (Where	N < 9)
Byte 0 >	ackID $\searrow \stackrel{\text{ld}}{\bowtie}$ prio $\begin{array}{c ccccccccccccccccccccccccccccccccccc$	destinationID	sourcelD	transaction wrsize srcTID
Byte 8 >	Extended Address [0:15]	Address [0:29]	법 xam 및 sbs	double-word 0 [0:15]
Byte 16 >		double-word 0 [16:63]	> ***	double-word 1 [0:15]
Byte (16 + (N-2)*8) >		double-word <i>n</i> -1 [16:63]		double-word n [0:15]
Byte (16 + (N-1)*8) >	double-word n [16:63]			Final CRC
	NIMIDIT	E NIMPITE D. CASTOLIT FLUSH J	th data Daguast Daglasts (Mhars	N = 0)
		TE, NWRITE_R, CASTOUT, FLUSH wit	•	
Byte 0 >	ackID S by prio tt ftype 0 0 1 0 1	destinationID	sourceID ਸ਼ੁੱਸ਼ xam ਬ੍ਰਿੰਡ sbs	transaction wrsize srcTID
Byte 8 > Byte 16 >	Extended Address [0:15]	Address [0:29]	sbs	double-word 0 [0:15]
-,		double-word 0 [16:63]		double-word 1 [0:15]
Byte 72 >	• double-word 7 [16:63] double-word 8 [0:15]			
Byte 80 >	Early CRC double-word 8 [16:63]			
Byte 88 >	Final CRC	Logic 0 Pad	double Hold o [10.05]	
			i I	
			I I	
		re, nwrite_r, castout, flush wi	th data Request Packets (Where	e N > 9)
Byte 0 >	ackID S b prio tt ftype 0 0 1 0 1	destinationID	sourceID	transaction wrsize srcTID
Byte 8 >	Extended Address [0:15]	Address [0:29]	da sbs	double-word 0 [0:15]
Byte 16 >		double-word 0 [16:63]		double-word 1 [0:15]
		•	i	
Byte 72 >		double-word 7 [16:63] double-word 8 [0:15]		
Byte 80 >	Early CRC double-word 8 [16:63]			
Byte 88 >		double-word	9 [0:63]	
Byte (16 + (N-1)*8) >		double-word n [0:63]		
Byte (16 + N*8) >	Final CRC	Logic 0 Pad		

Notes: N is the number of double-words in the payload. n = N-1