STANDARD OF
THE JAPAN
ELECTRICAL
MANUFACTURERS'
ASSOCIATION

JEM 1479

First edition
2000-11-28

Third edition
2012-09-27

Protocol Specification for FA Control Network Standard

*In case of any doubt arising as to this **English edition**, the original JEM 1479 in Japanese shall prevail.*

一般社団法人日本電機工業会

Reference number
JEM 1479:2012(J)

CONTENTS

Page

The Japan Electrical Manufacturers' Association Standard

PROTOCOL SPECIFICATION

FOR FA CONTROL NETWORK STANDARD

FOREWORD FOR THE ENGLISH EDITION

This document is English translation of JEM 1479 that is written in Japanese for convenience to users and developers of FL-net who feel difficult to understand Japanese language. Therefore, the JEM 1479 (Japanese) is original while this document is for reference only.

This document may be revised while the original is unchanging to improve English expression without noticeable version change. The Working Group on Specification of the Special Committee for FL-net Promotion at JEMA, which worked on the translation, welcomes comments and questions on English expression, which will help it for future improvement.

FOREWORD

This is the standard of the Japan Electrical Manufacturers' Association deliberated by the Working Group on Specification, the Special Committee for FL-net Promotion and the Standardization Committee, and revised by the Steering Technical Committee.

It should be remarked that some parts of this standard may conflict with patent rights to technical properties, patent right registration applications after application public release, patent rights for utility articles, or patent right registration application for utility articles after application public release. The Japan Electrical Manufacturers' Association accepts no responsibility for confirmations of such patent rights to technical properties, patent right applications after application public release, patent rights for utility articles, or patent right applications for utility articles after application public release.

PROTOCOL SPECIFICATION FOR FA CONTROL NETWORK STANDARD

Introduction

This standard applies to portable type a.c. generating sets driven by diesel engine or gasoline engine having rated output power above 3 kVA and less than 10 kVA.

This document provides protocol specifications for controller-level to device level networks for efficiently executing data exchange between various control units in manufacturing systems of factory CIMs such as programmable logic controllers (hereinafter referred to as PLC), robot controllers (hereinafter referred to as RC), numerical control units (hereinafter referred to as NC), FA controllers, and personal computers for control purposes (hereinafter referred to as personal computers).

Figure 1 shows the positioning of the FA control network covered by this document in the OSI/CIM model.

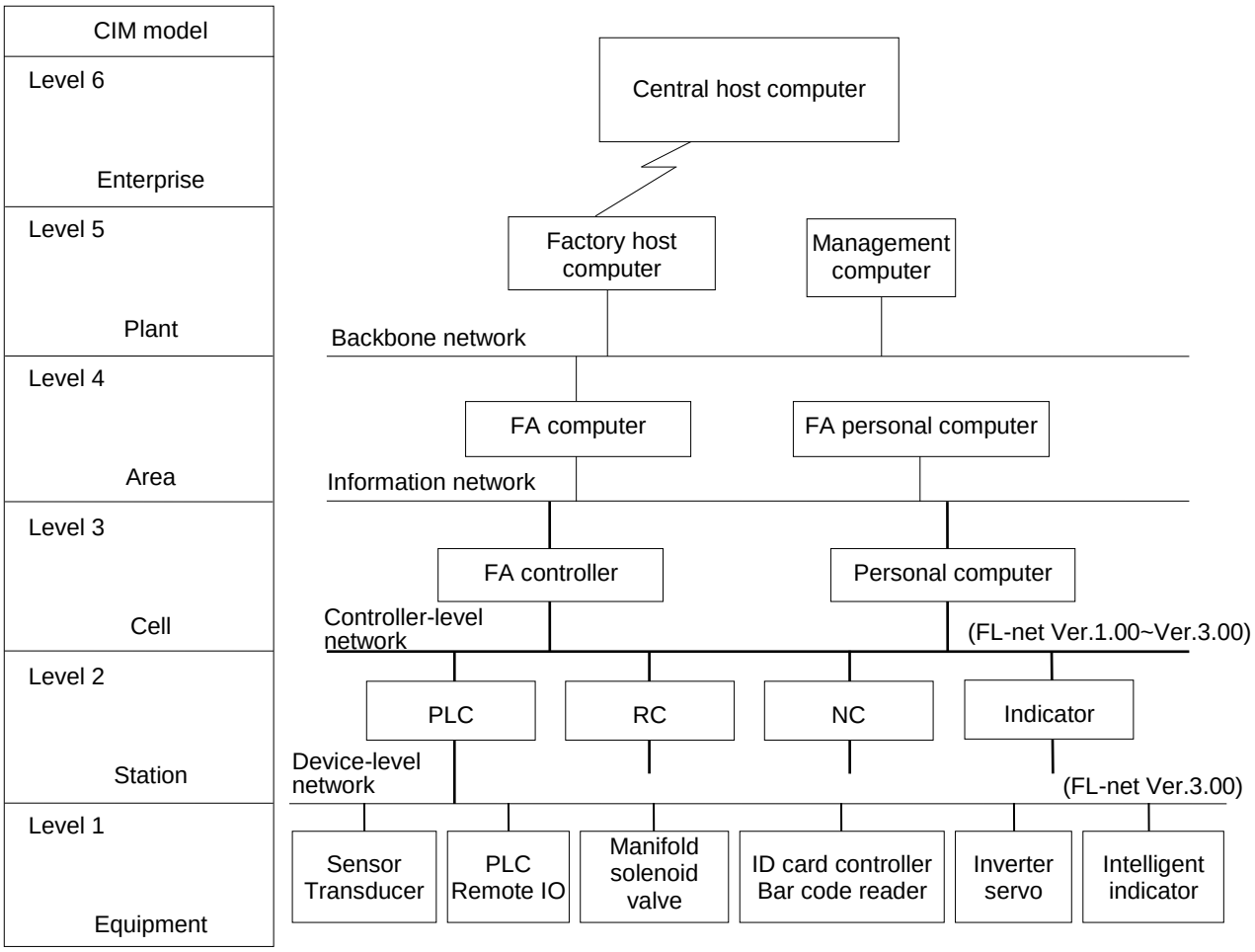


Figure 1 - Positioning of FL-net in OSI/CIM Model

1 Scope

This document prescribes the protocol specifications for the FA control network (FL-net (OPCN-2)[1)] (hereinafter referred to as FL-net)) which covers two level networks; the controller-level network between PLC, RC, NC, and FA controllers or personal computers, and the device level network for device level communication between PLC, FA controllers, personal computers, and input/output devices and drive controllers.

The FL-net has the following versions.

— JEM 1479: 2000 version is expressed as Ver. 1.00.

— Devices manufactured and certified based on JEM 1479: 2000 are expressed as Ver. 1.00 devices.

— JIS B 3521: 2004 and JEM 1479: 2002 versions are expressed as Ver. 2.00.

— Devices manufactured and certified based on JIS B 3521: 2004 are expressed as Ver. 2.00 devices.

— JEM 1479: 2012 version is expressed as Ver. 3.01.

— Devices manufactured based on JEM 1479:2012 and certified at the implementation classes 3 to 7 are expressed as Ver. 3.01 devices.

The Ver. 1.00 product and the Ver. 2.00 product have no compatibility in connectivity, and the specification of the Ver. 3.01 products includes that of Ver. 2.00 and Ver.3.00 products. In this document, description of FL-net Ver. 3.01 is added to the functions specific to Ver. 3.01.

The compatibility of the FL-net is maintained by implementing this specification, which prescribes network specifications for the FA control network services and protocols, in equipment and/or devices. However, verification of the compatibility and interconnectivity of implemented products is not included within the scope of this document.

Note 1) Stands for FA Link network (Open Programmable Controller Network - level2).

2 References

The following standards are incorporated to be part of this document by reference. The latest edition of the references is applied (including addenda).

JIS B 3521	Protocol specification for FA control network standard
JIS X 5252	Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications
JEM-TR 214	Device profile common specification for FA control network [FL-net(OPCN-2)]

3 Terms and definitions

The definitions of the main terms used in this document shall be based on the definition of JIS B 3521 and the following.

3.1 FA link protocol

A protocol defined to implement the FL-net network. The FA link protocol corresponds to the data link layer, the network layer, the transport layer, and the application layer, and prescribes the protocols for token management, cyclic transmission, and message transmission so that each node can equally perform transmission.

3.2 FA link

An entity to implement the FA link protocol.

3.3 FA link header

The header information part of a frame defined in the FA link protocol.

3.4 token

A temporary permission granted to use media on the network. A token is successively passed from one node to another.

3.5 valid linking state

A state in which a token circulates around the several nodes on the network.

3.6 network startup state

The state of a node attempting to participate in the network when not in the valid linking state.

3.7 in-ring startup state

The state of a node attempting to participate in the network when in the valid linking state.

3.8 cyclic transmission

A cyclic transmission system that a node transmits data stored in the common memory area, which is specified every time a token is obtained, to all the nodes by broadcast. For details, refer to 9.3.2.

3.9 message transmission

An aperiodic transmission system for notification of events generated between nodes. For details, refer to 9.3.3.

3.10 cyclic frame

A frame used when performing cyclic transmission.

3.11 token frame

A frame to send and receive a token between nodes.

3.12 message frame

A frame used when performing message transmission.

3.13 trigger frame

A frame that a node in the network startup state sends notification that the timer for waiting time for the transmission of the participation request frames in each node gets started.

3.14 participation request frame

A frame for a node to notify its existence when the node participates the network.

3.15 ACK(message acknowledge)

Information used for confirmation of receipt of message frames.

3.16 version of sequence number

The static number that is used to determine whether the message data frame has previously received or not. The version of sequence number is created when a node participates in the FL-net, and the value is maintained until the node leaves the network.

3.17 sequence number

The sequential number that is used to determine whether the message data frame has previously received or not. The sequence number is initialized when starting, and is incremented every time a message transmission is completed.

3.18 TCD : transaction code

A Code to identify each frame defined in this network. The transaction codes include the

numbers reserved by the FA link protocol and the numbers which users can freely define. For details, refer to 8.2.

3.19 node

Data stations used in this network.

3.20 node number

The number to identify nodes linked to this network. The numbers from 1 to 254 can be specified. In the FA link protocol, these numbers are used as information for destination addresses and source addresses in FA link headers.

3.21 common memory

Virtual memory for cyclic transmission defined in this network. There are two types: area 1 and area 2; each of which has the size of 512 words and 8,192 words, respectively. For details, refer to 7.1.

3.22 word

The unit for the amount of data. One word means two octets.

3.23 virtual address space

A memory space which can be accessed in message transmission.

3.24 token mode

Mode to indicates whether independent token frames are supported.

3.25 major version

The version of the FA link protocol is expressed as a major version in the range of four bits using integral values without the digits after the decimal point. When this version number is 0(null), it is regarded as 1.

Example: The major version of Ver. 3.01 is 3.

3.26 minor version

The version of the FA link protocol is expressed as a minor version in the range of four bits using the digits after the decimal point.

Example 1: The minor version of Ver. 2.01 is 01.

Example 2: The minor version of Ver. 2.10 is 10.

3.27 double word

A unit for the amount of data. One double word means four octets.

3.28 final cyclic frame

A cyclic frame transmitted just before a token frame.

3.29 server function

A function to receive request messages and to send response messages in message transmission.

3.30 client function

A function to transmit request messages and to receive response messages in message transmission.

3.31 protocol type

Information for distinguishing from similar industrial Ethernet protocols.

3.32 device level network function

A function between a master node and slave nodes(devices) to exchange IO data that is assigned in the common memory.

3.33 master node

A node to receive data generated by slave nodes or deliver data to be processed to slaves in the device level network configuration, and to function as a controller in the system. It is sometimes written simply as a master.

3.34 slave node

A node, in the device level network configuration, which receives data distributed by the master node, or transmits data to be collected by the master, and which functions as a device (terminal) in the system. It is sometimes written simply as a slave.

3.35 input data

Data assigned in the common memory that sent from slave nodes to a master node.

3.36 output data

Data assigned in the common memory that sent from a master node to slave nodes.

3.37 setting tool

A tool for setting node information, IO allocation, etc.

3.38 solicitation frame

Setting frame from a master to a slave in the device level network function.

3.39 UDP/IP, TCP/IP superimpose function

A function to mix general Ethernet(UDP, TCP, etc.) transmission with the FL-net network.

3.40 load measure function

A function to measure the network load of a node.

3.41 general purpose communication

General purpose Ethernet communication (such as UDP, TCP, etc.) which are not prescribed in the FA link protocol.

3.42 IO define

A definition of IO allocation of the common memory by a master node to slave nodes.

3.43 multi master

A function to constitute a FL-net network with multiple configuration that consist of a master node and a number of slaves(it does not mean multiple master nodes can be assigned to one slave node).

3.44 alternate transmission order

The transmission order for transmitting either a message frame or a solicitation frame when holding a token.

4 Notation and Abbreviation

4.1 Notation

The following notation shall be used.

a) Decimal shall be used unless specifically notified in this document. Hexadecimal is written in any one of 2-digit, 4-digit, or 8-digit notation. 8-digit hexadecimal notation is separated with "_" (underscore) for every 4 digits.

Example 1: Decimal	12	0	4782	65 546
Example 2: Hexadecimal	16#FF	16#00	16#12AE	16#0001_000A

b) The # symbol is added in front of the number of node numbers used in this document.

Example: Node number 1 Node #1

c) The version of the protocol specification prescribed in this document is expressed as Ver.*.*. The asterisk (*) symbol indicates any decimal value from 0 to 15. The integral value without the digits after the decimal point is a major version, and the digits after the decimal point shall be a minor version.

Only a major version is used about the general term of a protocol (Ver.2 protocol etc.).

4.2 Abbreviation

The abbreviated terms used in this document shall be shown in Table 1.

Table 1 - List of abbreviations and terms

Abbreviations	Terms
3CWT	three circulation waiting time
AWT	message acknowledge waiting time
BCT	broadcast transmission
BSIZE	current block size
CBN	current fragment block number
C_AD1	common memory area 1 data top address
C_AD2	common memory area 2 data top address
C_SZ1	common memory area 1 data size
C_SZ2	common memory area 2 data size
DA	destination address
DNA	Node number of destination side
H_TYPE	header type
LKS	FA link status
MFT	allowable minimum frame interval time
MODE	FA link protocol version (major version, minor version), token mode
MSN	manufacturer model name of node information
M_ADD	message offset-address in virtual address space
M_CTL	message control
M_RLT	message result
M_SZ	message data size in virtual address space
NDN	node name of node information
PAT	participation request frame acceptance time
PDU	protocol data unit
PPT	peer to peer transmission
PRI	message priority
PWT	participation request frame transmission waiting time
P_TYPE	protocol type
RCT	allowable refresh cycle time
RMT	refresh cycle measurement time
RPA	response data by ACK
SA	source address
SDU	service data unit

Table 1 - List of abbreviations and terms (Continued)

Abbreviations	Terms
SEQ	sequence number
SNA	Node number of source side
TBN	total fragment block number
TCD	transaction code
TDT	joining token detection time
TFL	total frame octet length of header and data
TM	token mode
TW	token watchdog time
TrWT	trigger frame transmission waiting time
ULS	upper layer status
VDN	vender code of node information
VER	program version
V_SEQ	version of sequence number
SFBTFI	solicitation frame batch transmission frame interval
SFTWT	solicitation frame transmission waiting time
Ver.2.00	FL-net Version2.00
Ver.3.01	FL-net Version3.01

5 Product Categories

Table 2 shows functions that are required when developing FL-net products. For functions using IO, refer to the category of the slave and the master. In case of superimposing UDP, TCP protocols, FL-net Ver.3.00 nodes are recommended to implement token retention time measurement and a sender log function (to be used for investigation when a node shows abnormal behavior by superimposing).

Table 2 - Summary list of functions by product category

Product category	Implementation class	Function category	Basic function		FL-net message transmission function		Device level network function			Command server for setting tool				
		Node types	Token participation	Cyclic transmission	Required message response	Optional message function	IO data exchange	Flexible slave solicitation transmission	Solicitation response	Read node state	Node setting	Load measurement	IO allocation	Others
Node before FL-net Ver. 2.00	Class 1	Ver. 2.00 standard nodes, such as PLCs, personal computers	R	R	R	O	N	N	N	N	N	N	N	N
	Class 2	Ver. 2.00 nodes (Setting tool)	R	P	R	O	N	N	N	N	N	N	N	N
FL-net Ver. 3.00 master node	Class 3	Controllers and flexible master, such as PLCs, personal computers	R	R	R	O	O	O	N	R	R	R	O	O
	Class 4	Flexible master, such as PLCs, personal computers	R	N	R	O	R	R	N	R	R	O	R	O
	Class 5	A fixed master, such as PLCs, personal computers	R	N	R	O	R	N	N	P	O	O	N	O
FL-net Ver. 3.00 slave node	Class 6	Flexible slave input/output devices	R	N	R	O	R	N	R	P	R	O	N	O
	Class 7	Fixed slave input/output devices	R	N	R	O	R	N	N	P	O	O	N	O

R: Required P: Partially required O: Optional N: Not necessary
Note: Refer to Table 3 for the list of functions included in the product categories.

Table 3 - Detailed list of functions by product category

Function	Type 1	Type 2	Command	TCD	Class No.						
					1	2	3	4	5	6	7
Common function	Token participation		Token frame	65 000	R[a)]	R[a)]	R[a)]	R[a)]	R[a)]	R[a)]	R[a)]
			Participation request frame	65 002							
			Trigger frame	65 012							
Cyclic	Client		Cyclic frame	65 001	R[b)]	R[c)]	R[b)]	N	N	N	N
	Server				R[b)]	O[b)]	R[b))]	N	N	N	N
FL-net message	Server	Read/Write message	Network parameter read frame (response)	65 207	R[a)]	R[a)]	R[a)]	R[a)]	R[a)]	R[a)]	R[a)]
			Profile read frame (response)	65 211							
			Message return frame (response)	65 215							
			Log data read frame (response)	65 213							
			Log data clear frame (response)	65 214	R[d)]	R[d)]	R[d)]	R[d)]	R[d)]	R[d)]	R[d)]
			Transparent message frame	10 000～59 999	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]
			Byte block read frame (response)	65 203	O[a)]	O[a)]	O[a)]	O[a)]	O[a)]	O[a)]	O[a)]
			Byte block write frame (response)	65 204							
			Word block read frame (response)	65 205							
			Word block write frame (response)	65 206							
			Network parameter write frame (response)	65 208							
		Operation command message	Stop command frame (response)	65 209							
			Operation command frame (response)	65 210							
		Vendor specific	Vendor specific message frame (response)	65 216	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]
	Client	Read/Write message	Transparent message frame	10 000～59 999	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]
			Log data clear frame (request)	65 014							
			Byte block read frame (request)	65 003	O[a)]	O[a)]	O[a)]	O[a)]	O[a)]	O[a)]	O[a)]
			Byte block write frame (request)	65 004							
			Word block read frame (request)	65 005							
			Word block write frame (request)	65 006							
			Network parameter read frame (request)	65 007							
			Network parameter write frame (request)	65 008							
			Profile read frame (request)	65 011							
			Log data read frame (request)	65 013							
			Message return frame (request)	65 015							
		Operation command message	Stop command frame (request)	65 009							
			Operation command frame (request)	65 010							
		Vendor specific	Vendor specific message frame (request)	65 016	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]	O[e)]

Table 3 - Detailed list of functions by product category (continued)

Function	Type 1	Type 2	Command	TCD	Class No.						
					1	2	3	4	5	6	7
Device level network	Data exchange	Transmission	Cyclic frame	65 001	N	N	O[b)]	R[b)]	R[b)]	R[b)]	R[b)]
		Reception									
	Client	Solicitation	Solicitation frame	65 017	N	N	O[b)]	R[b)]	N	N	N
	Server	Solicitation	Solicitation frame	65 017	N	N	N	N	N	R[b)]	N
Command server	Server (UDP)	Read/Write message	Byte block read frame (response)	65 203	N	N	O	O	O	O	O
			Byte block write frame (response)	65 204							
			Word block read frame (response)	65 205							
			Word block write frame (response)	65 206							
			Network parameter read frame (response)	65 207							
			Network parameter write frame (response)	65 208							
			Profile read frame (response)	65 211							
			Log data read frame (response)	65 213							
			Log data clear frame (response)	65 214							
			Message return frame (response)	65 215							
		Operation command message	Stop command frame (response)	65 209	N	N	O	O	O	O	O
			Operation command frame (response)	65 210							
		IO allocation	IO allocation setting frame (response)	65 218	N	N	O	R	N	N	N
			IO allocation read frame (response)	65 219							
		Load measurement	Start frame of token retention time measurement (response)	65 220	N	N	R	O	O	O	O
			End frame of token retention time measurement (response)	65 221							
			Start frame of measurement in general purpose communication data sender log (response)	65 222							
			End frame of measurement in general purpose communication data sender log (response)	65 223							
		Node setting	Configuration parameter setting frame (response)	65 224	N	N	R	R	O	R	O
		Read node status	Read frame of participating node management information parameter (response)	65 225	N	N	R	R	R	N	N
			Read frame of self node management information parameter (response) (Read node information)	65 226	N	N	R	R	R	R	R
			Read frame of self node configuration information parameter (response) (Read node information in the memory in the node)	65 227	N	N	R	R	O	R	O
		Reset	Node reset frame (response)	65 228	N	N	O	O	O	R	R

Table 3 - Detailed list of functions by product category (continued)

Function	Type 1	Type 2	Command	TCD	Class No.						
					1	2	3	4	5	6	7
Command server (Continued)	Server (TCP)	Read/Write message	Byte block read frame (response)	65 203	N	N	O	O	O	O	O
			Byte block write frame (response)	65 204							
			Word block read frame (response)	65 205							
			Word block write frame (response)	65 206							
			Network parameter read frame (response)	65 207	N	N	O	O	O	O	O
			Network parameter write frame (response)	65 208							
			Profile read frame (response)	65 211							
			Log data read frame (response)	65 213							
			Log data clear frame (response)	65 214							
			Message return frame (response)	65 215							
		Operation command message	Stop command frame (response)	65 209	N	N	O	O	O	O	O
			Operation command frame (response)	65 210							
		IO allocation	IO allocation setting frame (response)	65 218	N	N	O	O	N	N	N
			IO allocation read frame (response)	65 219							
		Load measurement	Start frame of token retention time measurement (response)	65 220	N	N	O	O	O	O	O
			End frame of token retention time measurement (response)	65 221							
			Start frame of measurement in general purpose communication data sender log (response)	65 222							
			End frame of measurement in general purpose communication data sender log (response)	65 223							
		Node setting	Configuration parameter setting frame (response)	65 224	N	N	O	O	O	O	O
		Read node state	Read frame of participating node management information parameter (response)	65 225	N	N	O	O	O	N	N
			Read frame of self node management information parameter (response) (Read node information)	65 226	N	N	O	O	O	O	O
			Read frame of self node configuration information parameter (response) (Read node information in the memory in the node)	65 227							
		Reset	Node reset frame (response)	65 228	N	N	O	O	O	O	O

R: Required O: Optional (included as an authentication item when implemented) N: Not necessary
Only 1 to 1 communication supported unless otherwise specified.
Note: The implementation of command server functions in TCP shall be optional.
Note[a)] Only 1 to 1
[b)] Support only 1 to n communication
[c)] Required, but the data size is 0
[d)] Support both 1 to 1 communication and 1 to n communication
[e)] Optionally support 1 to 1 communication and 1 to n communication

6 Basic Specifications

6.1 Relation between OSI Reference Model and FL-net

The relation between the OSI reference model and the FL-net protocol specification shall be shown in Figure 2. The presentation layer and the session layer prescribed in the OSI reference model shall not be used in this document according to the attribute of the control system within the scope. This document defines the transport layer, the network layer, and the data link layer as the FA link lower layer. Any function other than a broadcast communication function shall not be prescribed for the FA link lower layer.

6.1.1 JIS X 5252 is used for the physical layer.

The FA link layer is defined as the upper layer of these specifications. The FA link prescribes the cyclic transmission for collecting and distributing data of each node at a fixed interval and the message transmission for event notification in addition to the network management function. The upper layer of the FA link is the upper application. The upper layer described in this document indicates the upper application.

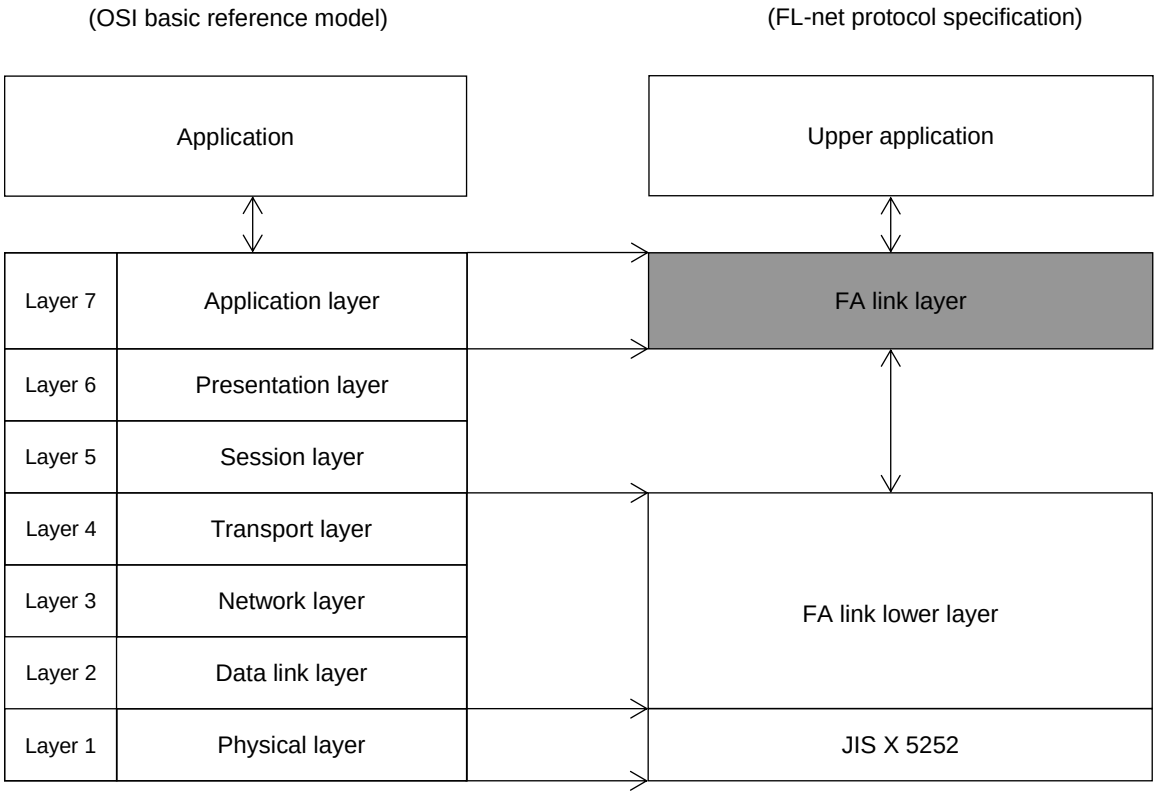


Figure 2 – Relation between OSI reference model and FL-net protocol specification

6.2 The number of nodes to be connected

The number of nodes which can be connected to the network shall be 254 nodes at the maximum.

6.3 UDP and TCP superimposition

FL-net Ver.3.00 is allowed to coexist with UDP and TCP communication that are used by setting tools, monitoring and information systems, an Internet connectivity, OA equipment, etc. on the FL-net network.

The UDP and TCP communication can be overlaid with or without the token of general OA equipment or each node. However, UDP and TCP superimposition increases the network load. Since the FA link protocol does not implement a function to adjust the load of overlaid UDP and TCP communication, a load measure function has been added to allow the user to consider the network load.

Using the load measure function, the user who uses the UDP/IP, TCP/IP superimpose function in the FL-net system can adjust the parameters for the FL-net communication performance after understanding of the influence on the real-time performances (cyclic transmission etc.) required by the system.

An example of the system configuration for the overlaid UDP and TCP shall be shown in Figure 3.

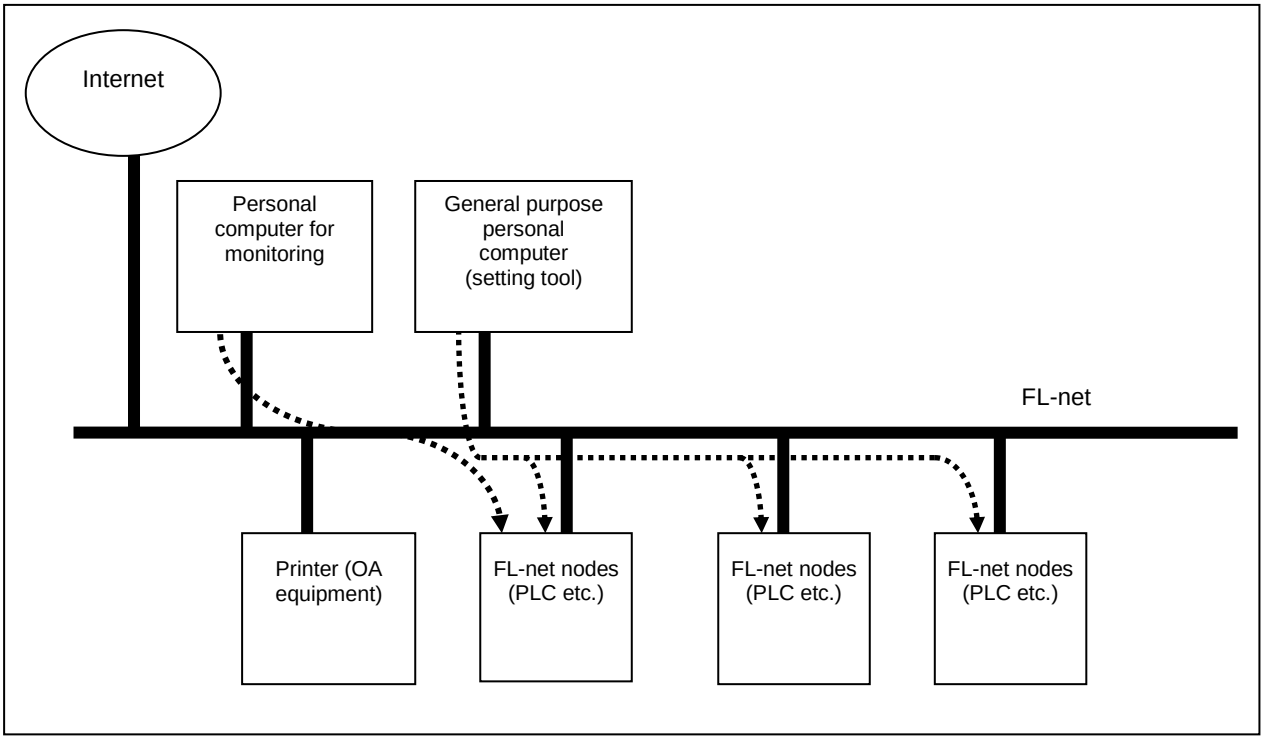


Figure 3 – Example of the system configuration for the overlaid UDP and TCP

In addition, the overlaid UDP and TCP communication enables access to FL-net Ver. 3.00 nodes from general purpose personal computers outside of the segment via standard routers. An example of access between different segments via a router installed on the network shall be shown in Figure 4.

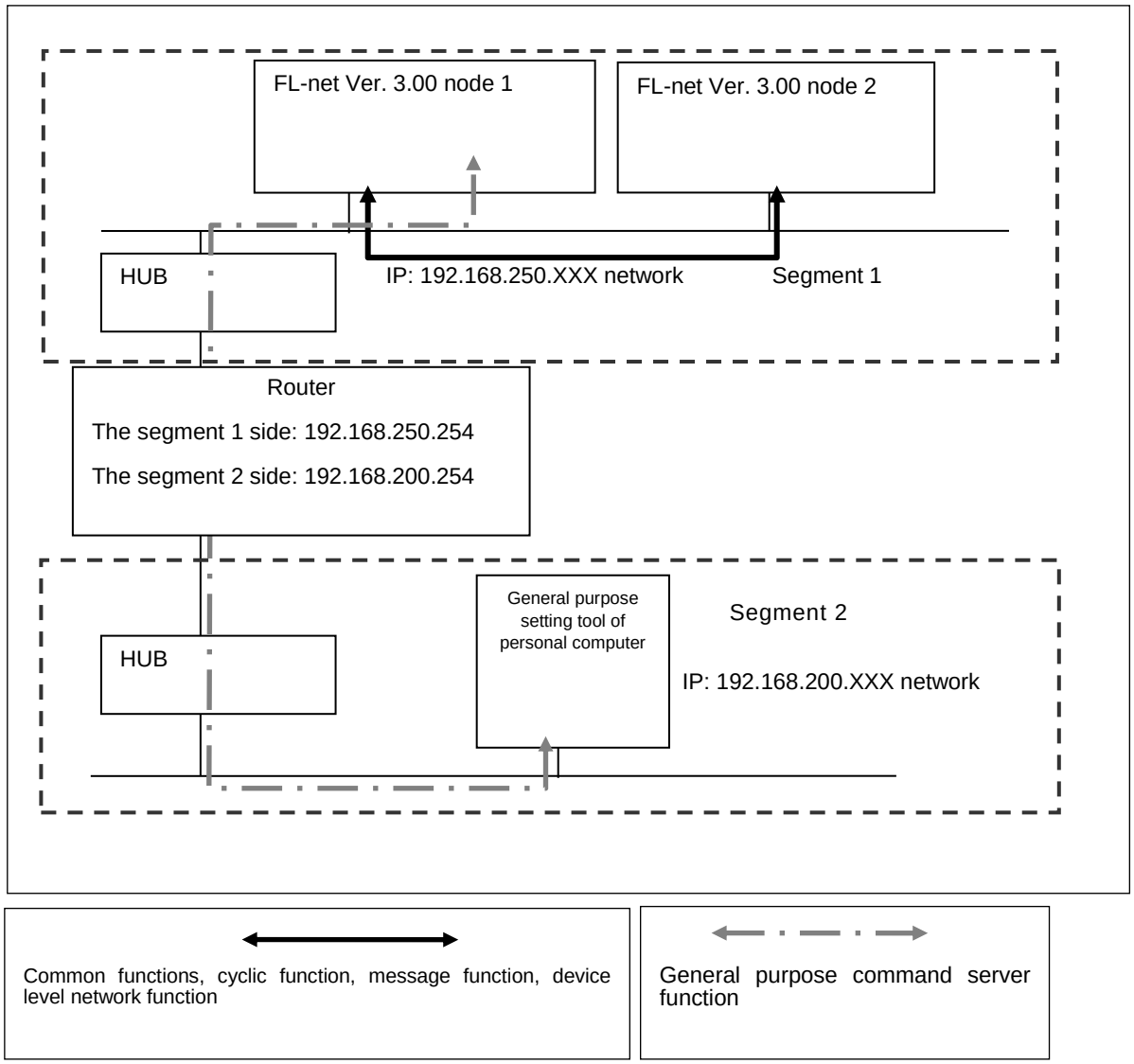


Figure 4 - Example of access from different segment (connected via a router)

Examples of IP address setting shall be shown in Table 4.

Table 4 - Examples of IP address setting

Device		IP address	Subnet mask address	Gateway address
FL-net Ver. 3.00 node 1		192.168.250.1	255.255.255.0	192.168.250.254
FL-net Ver. 3.00 node 2		192.168.250.2	255.255.255.0	192.168.250.254
General purpose setting tool of personal computer		192.168.200.100	255.255.255.0	192.168.200.254
Router	Segment 1	192.168.250.254	255.255.255.0	—
	Segment 2	192.168.200.254	255.255.255.0	—

6.4 Load Measure Function

The load measure function is a function in FL-net Ver. 3.00 to record the log of the network load for each node in order to consider the load on the network due to the overlaid UDP and TCP.

The load measure function includes two processing: token retention time measurement processing and general purpose communication data sender log processing.

a) Token retention time measurement processing

Measures the minimum and maximum values of the time for which a node holds a token, and the elapsed time until the maximum value is measured from the start of measurement, and saves those values in a logging area. In addition, followed information are saved in the log as factor to have influence on the token retention time(For details, refer to 8.4.4.1); the latest time of token cancellation, the latest time of token reissuing, the latest time of token retention timeout, the latest time of token monitoring timeout, number of tokens while measuring the token retention time, detection time for the maximum refresh cycle, detection time for error in receiving cyclic frame, and detection time for error in receiving message frame.

b) General purpose communication data sender log processing

Measures the number of reception for the first ten frames per sender IP when the node receives a general purpose communication frame. (For details, refer to 8.4.4.2).

6.5 General purpose Command Server Function

The setting tool for FL-net Ver. 3.00 can configure and read various parameters.

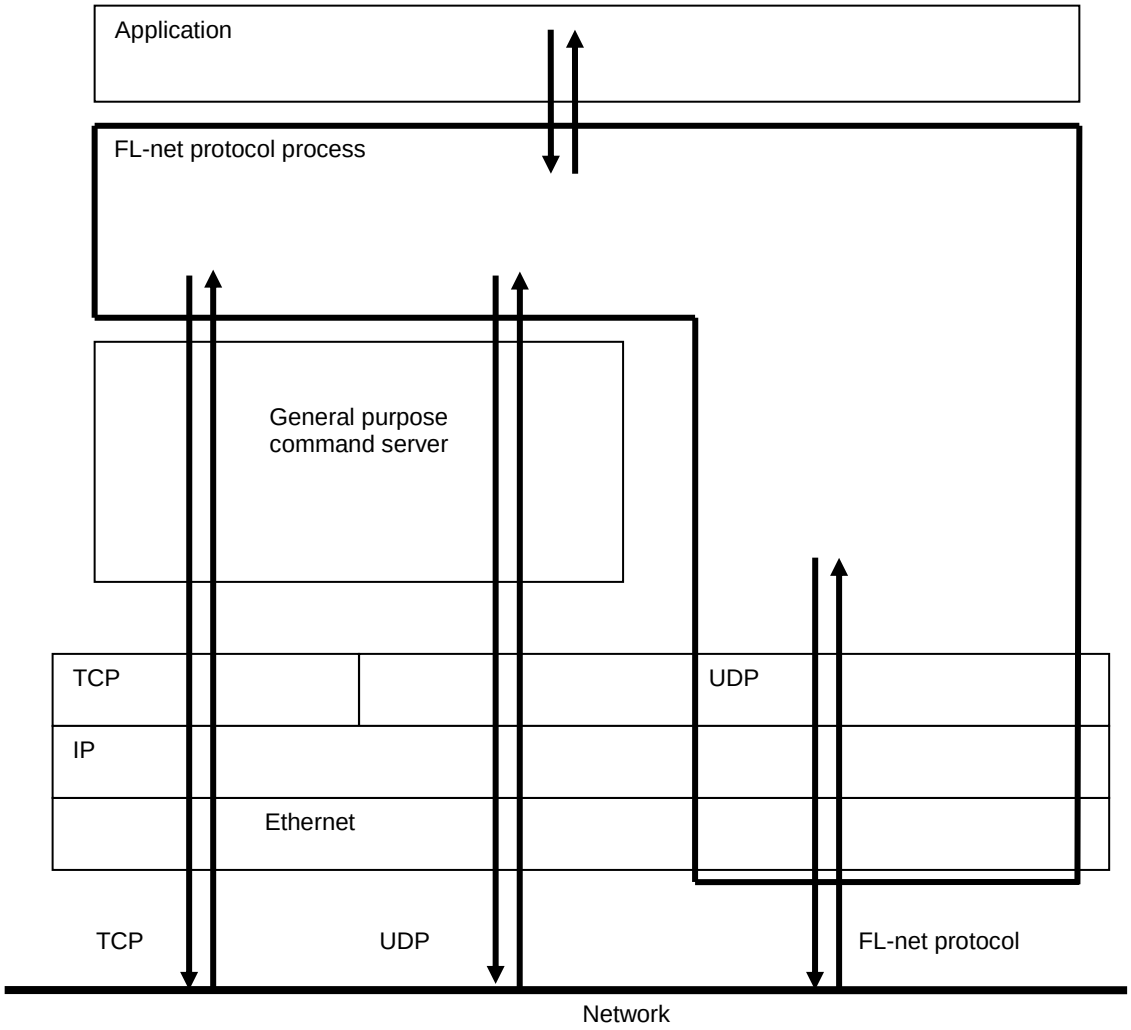
The command server processes requests that received with Ethernet TCP or UDP protocol and returns response.

The request from a setting tool is received by TCP or an UDP protocol from a network, is processed according to the request TCD by application, a general purpose command server, or FL-net protocol processing, and transmits a response from the same course as the time of reception.

When a response data sending request is received from a message sending process and a system message process, it is transmitted via the received path.

Nodes without functions for each command of the command server functions are required to send the unimplemented response of M_RLT (= 2) within one second, or send no response (For M_RLT, refer to Part 9.4.6).

The outline of the general purpose command server interface shall be shown in Figure 5.



Note: Refer to Chapter 5 for implementation level details of the general purpose command server functions.

Figure 5 - Outline of general purpose command server interface

6.6 Device level network function

FL-net Ver. 3.00 can perform device-level I/O data exchange between I/O devices directly connected to the FL-net network by the master/slave method using the common memory.

An example of configuration of the device level network system shall be shown in Figure 6.

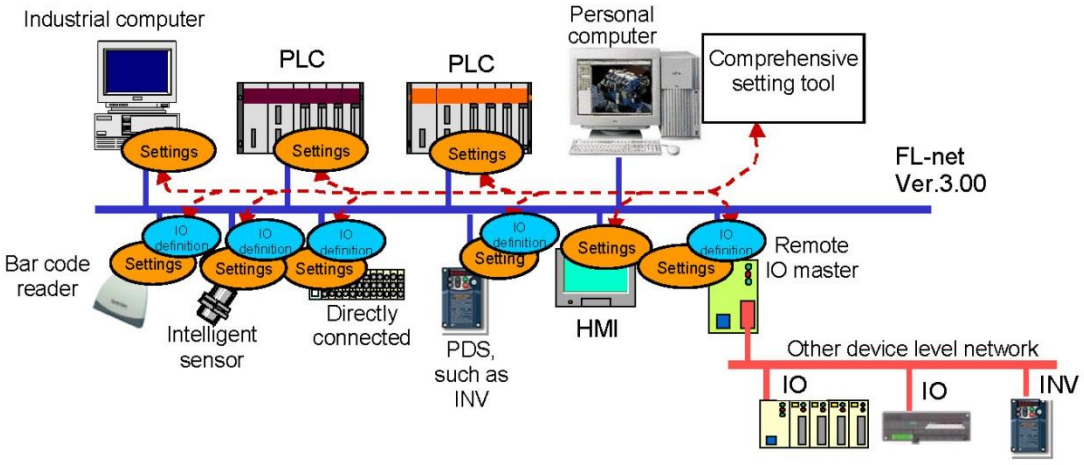


Figure 6 - Example of configuration of the device level network system

The device level network function is achieved by prescribing the memory map of the common memory shared by each node using the conventional cyclic frames.

There are two setting modes of IO data exchange: a fixed setting mode with fixed common memory allocation for simple user settings and an optional setting mode that allows the user to freely define common memory allocation for flexible system configuration.

6.6.1 Setting Mode

The setting mode shall be as follows.

a) Fixed setting mode

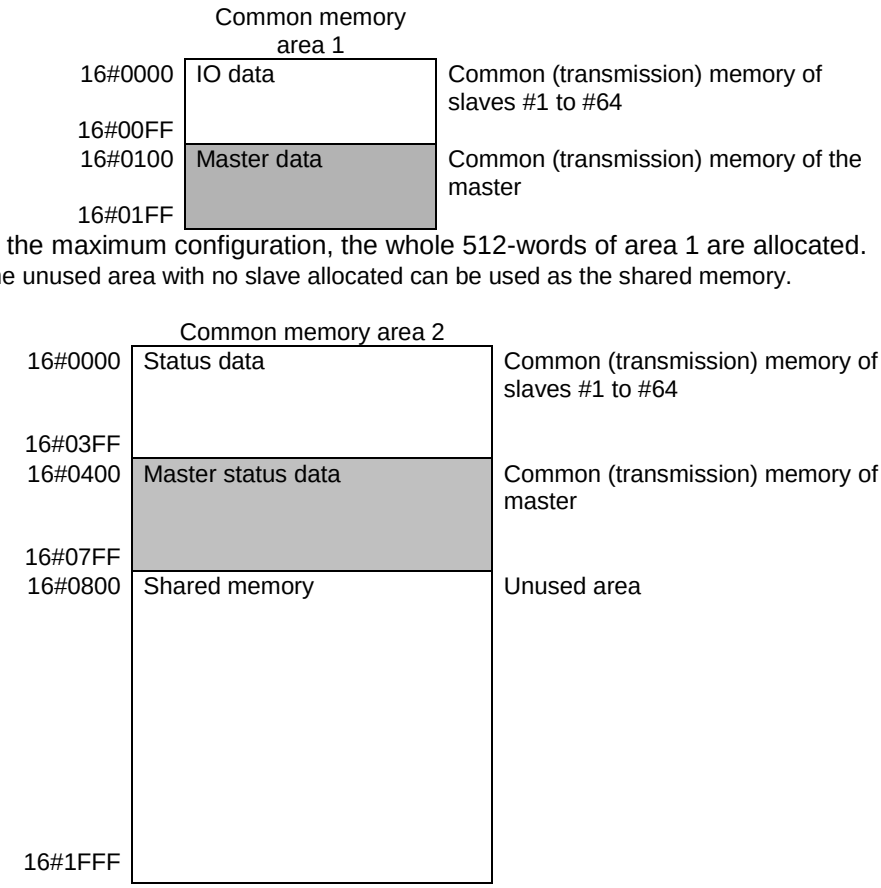
Fixed setting mode Is basically designed to operate with the minimum settings.

The node numbers of default behavior are configured as 100 for the master and in the range from 1 to 64 for slaves. This determines unique addresses of the common memory (For details, refer to 8.6).

The slaves can identify the existence of the master with the master node number 100, and the master and slaves input and output via the common memory corresponding to the node number. When slave nodes have a master node number setting function, the system configuration which can set up a master node arbitrarily and has two or more master nodes in one system is possible. Also in this case, the common memory of the slaves is fixed with the node numbers and two or more masters cannot output to the same slave.

The size of input and output is fixed to 4 words for inputs and 4 words for output, but the IO area is extensible to 4 or more words by the settings of the number of IOs of the slaves and the slave type in the master status areas from the master.

The common memory allocation in the fixed setting mode shall be shown in Figure 7.



・At the maximum configuration, the whole 512-words of area 1 are allocated.
・The unused area with no slave allocated can be used as the shared memory.

・At the maximum configuration, the 2000 words of the 8000-word of the area 2 are allocated.
・The unused area with no slave allocated can be used as the shared memory.

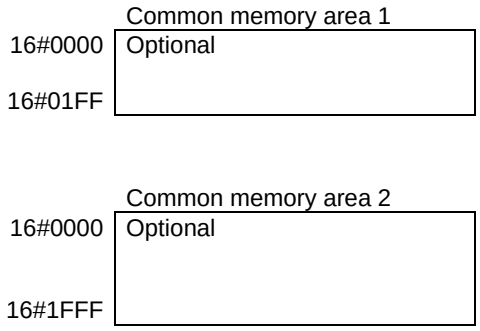
Figure 7 - Common memory allocation in the fixed setting mode

b) Flexible setting mode

Allows the user to arbitrarily set the node numbers of the master and the slaves and the common memory allocation. However, the common memory area assigned to one node shall be a continuous area. Although the configuration with two or more masters (multi-master) is possible, two or more masters cannot output to the same slave.

When the user configures a master for the setting of each slaves, the master distributes configuration information to each slaves using the solicitation frames.

The common memory allocation in the flexible setting mode shall be shown in Figure 8.



・The unused area with no slave allocated can be used as the shared memory.

Figure 8 - Common memory allocation in the flexible setting mode

6.6.2 Solicitation frame

The solicitation frames is for defining common memory allocation of each slave station in the flexible setting mode, and are transmitted from the master to corresponding slaves.

The slave node number, fixed settings and flexible settings identification flags, IO data areas, and status areas are set in the data part of the solicitation frame. (For details, refer to 10.4.1).

The data part of the solicitation frame shall be shown in Figure 9.

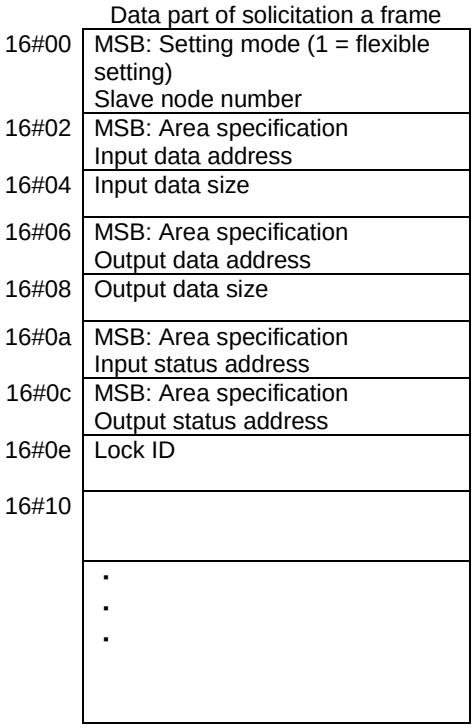


Figure 9 - Data part of the solicitation frame

The data size of a solicitation frame is 8 words(16 octets) per node.

One transmission shall be up to 64 nodes (1024 octets) according to the defined transmission frame size.

When the transmission exceeds 64 nodes, the solicitation frame is transmitted in multiple times (up to four times).

7 Service

This chapter prescribes services provided and required by the FL-net. The services prescribed here shall be as follows.

a) Services to the upper layer

The services to the upper layer shall be as follows.

1) Read and write to the common memory data

2) Message

— Byte block read service

— Byte block write service

— Word block read service

— Word block write service

— Network parameter read service

— Network parameter write service

— Stop command service

— Operation command service

— Profile read service

— Transparent message service

— Log data read service

— Log data clear service

— Message return service

— Vendor specific message service

3) Network management

— Configuration parameter setting

— Read self node management information parameter

— Read participating node management information parameter

— Read network management information parameter

— Read message sequence number management information

— Read node status

4) Device level network

— Load measurement

— IO allocation

b) Services to the lower layer

The service to the lower layers shall be as follows.

— SAP for transmission

— SAP for cyclic transmission

— SAP for message transmission and reception

— SAP for receiving participation requests

— Transmission of solicitation frames

7.1 Common Memory

7.1.1 Outline

The common memory provides a function which can be treated as a common global memory among nodes which perform cyclic transmission. This service shall be optionally implemented.

The common memory has two types of areas including the area 1 and the area 2; the area 1

has the data capacity of 512 words, and the area 2 has the data capacity of 8,192 words. The capacity for the common memory prepared by the communication part of one node shall be the fixed size of 512 words + 8,192 words = 8,704 words. However, the nodes which do not implement this service need not prepare a common memory.

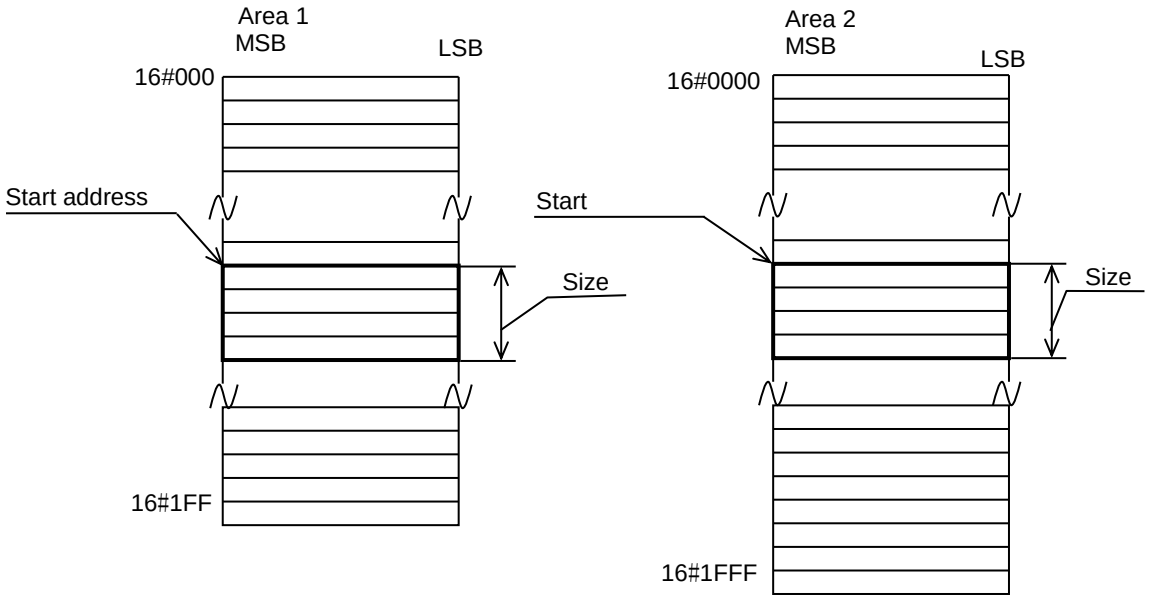
Any sizes can be specified within the maximum space of the area 1 and the area 2 as the transmission area of one node in the area of the common memory.

7.1.2 Common Memory Allocation

The node can allocate transmission areas to each of the area 1 and the area 2 of the common memory. The transmission area of each node shall not overlap with the transmission area of other nodes.

Each node can set any sizes by word as the transmission area of the node within the range from 0 to 512 words for the area 1 and from 0 to 8,192 words for the area 2. The transmission area of each node is set with the start address and the size of the area.

An example of the common memory allocation shall be shown in Figure 10.



Note: The bit length shall be 16 bits.

Figure 10 - Example of the common memory allocation

7.1.3 Data Sharing Function

Each node provides a function to share the same data across the system by broadcasting the data of the transmission area of the node allocated in the common memory at a constant frequency. Each node on the network avoids overlapping the transmission area of each node when allocating and using the common memory.

In operation of the common memory, the area in the common memory allocated to a certain node is used as a transmission area of the node, and is used as a receiving area for other nodes (refer to Figure 11). In addition, a node may use the common memory only as a receiving area (example: node #01 in Figure 12).

The nodes which have not implemented the common memory discard all the received data (example: node #01 in Figure 13).

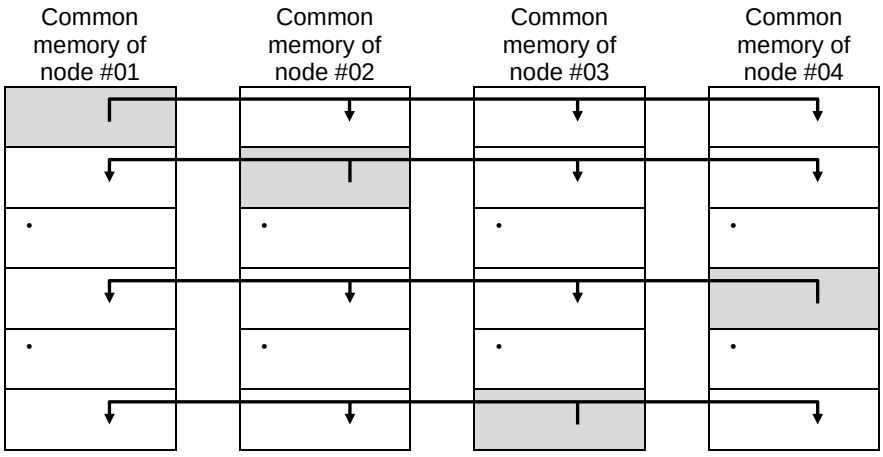


Figure 11 - Example of data sharing function

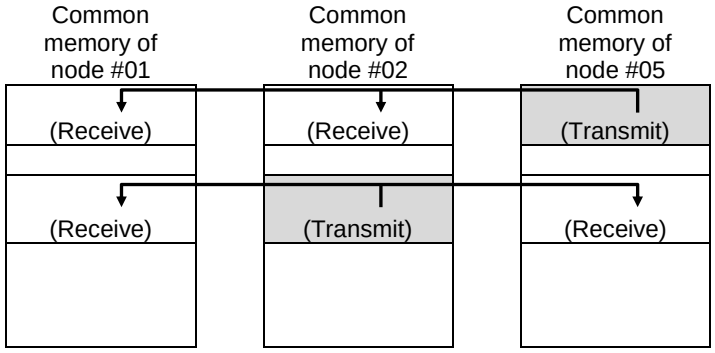


Figure 12 - Example of a node which only performs receiving operation

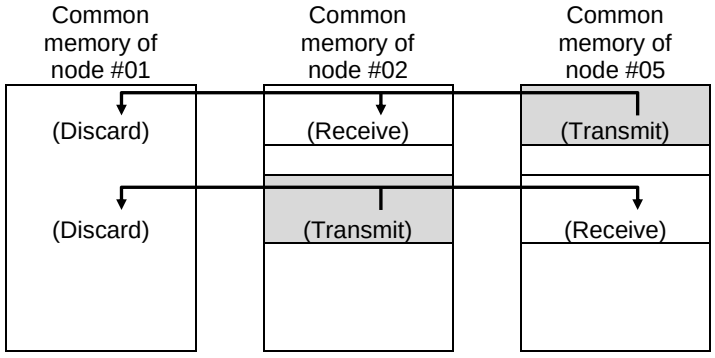


Figure 13 - Example of a node without a common memory

7.1.4 Data transmission

Data transmission shall be as follows.

a) Write request to the common memory

The upper layer sets data in the common memory by performing a write request to the applicable area in the common memory. The parameter of the write request shall be as follows.

— The address of the common memory

— Data size

— Data

b) Request to send

The upper layer requests to transmission data in the transmission area in the common memory to the FA link layer.

The FA link layer which received the request to send transfers the data of the node allocated in the common memory to the transmission buffer, prepares for transmission, and transmits in sequence when the node obtains a token. When the data size from the transmission node is larger than the size which can be transmitted in one frame (1024 octets), the data in the buffer is divided into two or more frames for transmission. Each node transmits all the transmission data in the common memory each time it obtains a token. An example of the data flow in the common memory shall be shown in Figure 14.

7.1.5 Data reception

Data reception shall be performed as follows.

a) Read request to the common memory

The upper layer reads the data in the common memory by performing read request to the applicable area in the common memory. The parameters of the read request shall be as follows.

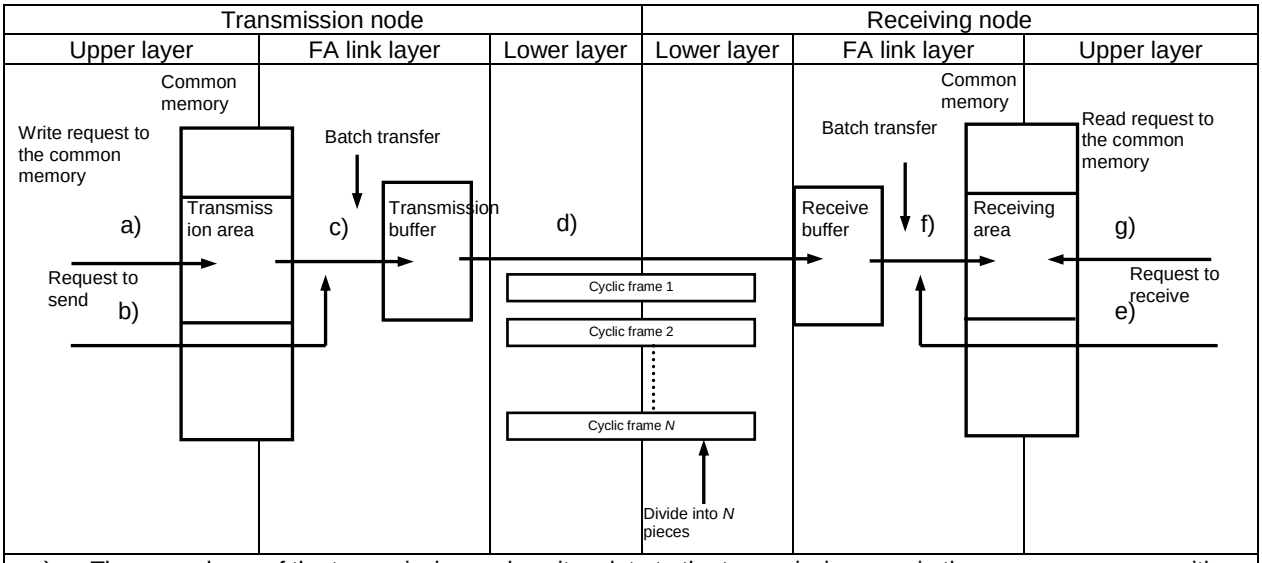
— Common memory address

— Data size

b) Request to receive

The upper layer requires the FA link layer to transfer received data to the applicable area in the common memory

Each node transfers the data to the applicable area in the common memory when a cyclic frame is received. At this time, the receiving node updates the corresponding area while synchronizing with the upper layer when it finished receiving all the cyclic data from one node. In other words, even when cyclic data is divided into several frames for transmission, data transfer to the common memory area is performed when all the frames transmitted from one node is received. When any one of all the frames divided for transmission from the node is missing, all the data transmitted from the node are discarded. An example of the data flow in the common memory shall be shown in Figure 14.



a) The upper layer of the transmission node writes data to the transmission area in the common memory with the write request to the common memory.

b) The upper layer of the transmission node sends a request to send to the FA link layer when all the data have been written in the transmission area.

c) The FA link layer performs batch transfer of the data from the transmission area to the transmission buffer when the request to send is received from the upper layer.

d) The FA link layer transmits data in the transmission buffer to other nodes by broadcast as cyclic frames when its node received and is holding a token. At this time, if the amount of the transmission data is larger than 1024 octets, the data in the buffer is divided into two or more frames for transmission. The FA link layer of the receiving node temporarily stores all the received cyclic frames in the receive buffer.

e) The upper layer of the receiving node sends a request to receive to the FA link layer.

f) The FA link layer of the receiving node performs batch transfer of the data in the receive buffer to an applicable receiving area with the request to receive from the upper layer.

g) The upper layer of the receiving node reads data from the receiving area in the common memory with a read request to the common memory.

Figure 14 - Example of data flow in the common memory

7.2 Message Transmission

7.2.1 Outline

Message transmission is a function to exchange aperiodic data generated between nodes.

The maximum amount of message data that can be transmitted at one time is 1024 octets.

The message transmission function includes peer to peer transmission which specifies a destination node and 1 to n broadcast for all nodes. The 1 to 1 message transmission prescribes a function to confirm reception and a response message for checking whether or not the destination has received the data in order to maintain the reliability of delivery of message data.

7.2.2 Type of Service

This document provides services from a) to n) as message transmission. Among the provided services, the transparent message service, the log data clear service, and the vendor specific message service prescribe 1 to 1 and broadcast transmission, and others prescribe only peer to peer transmission. Each of the message transmission service except for the transparent message service defines requests and responses. The type of each message is defined with the transaction codes. In addition, each message has two transaction codes: one for request and one for response. However, the transaction codes for response and request for

transparent messages are defined by users as necessary.

a) Byte block read service

b) Byte block write service

c) Word block read service

d) Word block write service

e) Network parameter read service

f) Network parameter write service

g) Stop command service

h) Operation command service

i) Profile read service

j) Transparent message service

k) Log data read service

l) Log data clear service

m) Message return service

n) Vendor specific message service

7.2.3 Byte Block Read Service

The byte block read service shall be as follows.

a) Function

This service reads byte block data of a destination node from a source node. This service uses the area defined in the virtual address space.

The virtual address space of the byte block shall be shown in Figure 15. The virtual address space is a 32 bit address space, and each node defines its contents.

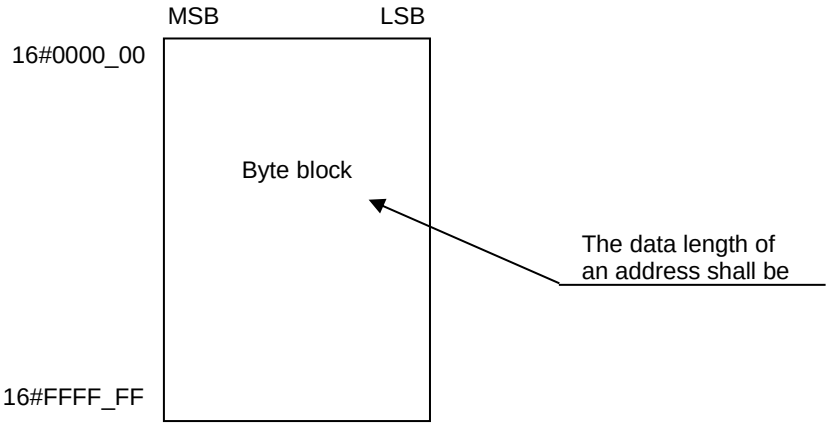
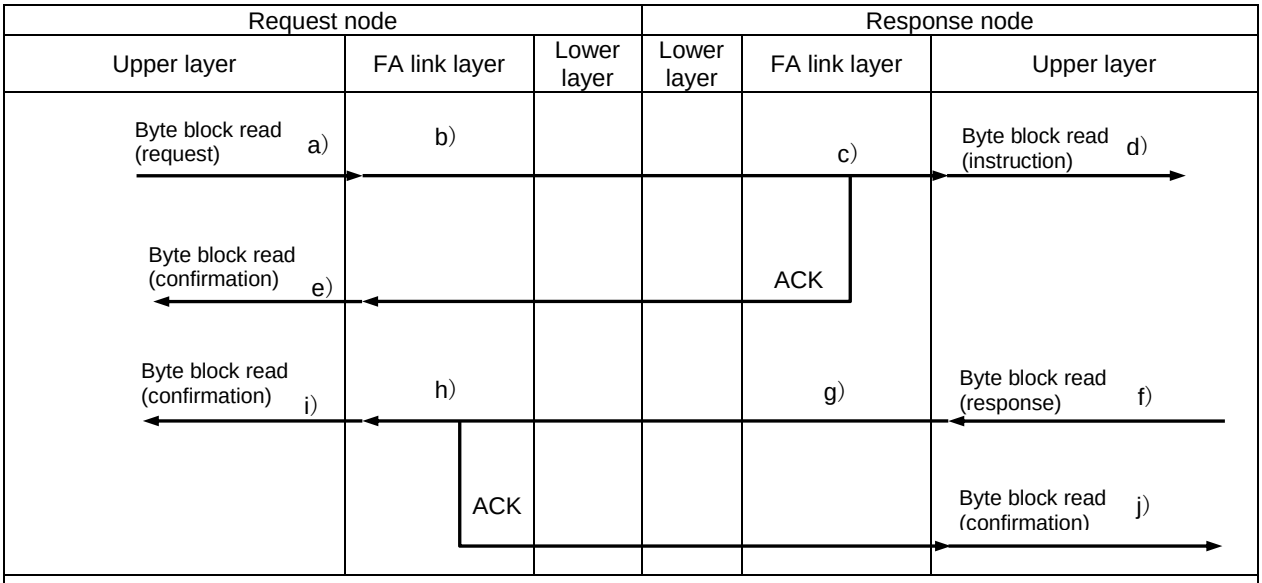


Figure 15 - Virtual address space of a byte block

b) Protocol

The sequence of the byte block read service shall be shown in Figure 16.



a) The upper layer of the request node sets the following parameters and sends a byte block read request to the FA link layer.
   - The number of the request destination node: DNA
   - The offset address of the byte block to be requested (address for the virtual address space): M_ADD
   - The size of the byte block to be requested (data size for the virtual address space): M_SZ

b) The FA link layer of the request node takes in a PDU (request) from the upper layer as an SDU and sends a request to send to the lower layer.

c) The FA link layer of the response node requests the lower layer to transmit an ACK to the PDU (instruction) received from the lower layer and provides the PDU (instruction) to the upper layer.
When the response process of the response node is completed before the response node obtains the token, the following f), g), and h) may be performed in advance before transmitting the ACK for the request.

d) The upper layer of the response node takes in the PDU (instruction) as an SDU, and starts a process for the request.

e) The upper layer of the request node checks whether or not the response node received the request with the ACK received from the lower layer.

f) The upper layer of the response node sets the following parameters with the completion of the byte block read process and responses to the FA link layer about the byte block read request.
   - Source node number: DNA
   - The offset address of the byte block to respond (address for the virtual address space): M_ADD
   - The size of the byte block to respond (data size for the virtual address space): M_SZ
   - The message result: M_RLT
   - Response data (byte block data which was read): data

g) The FA link layer of the response node takes in a PDU (response) from the upper layer as an SDU and sends a request to send to the lower layer.

h) The FA link layer of the request node requests the lower layer to transmit the ACK to the PDU (response) received from the lower layer, and provides the PDU (confirmation) to the upper layer. The ACK does not contain response data.

i) The upper layer of the request node takes in the PDU (confirmation) as an SDU and receives the response data for the byte block read request.

j) The upper layer of the response node checks whether or not the request node received the response data from the ACK received from the lower layer.

Figure 16 - The sequence of the byte block read service

7.2.4 Byte Block Write Service

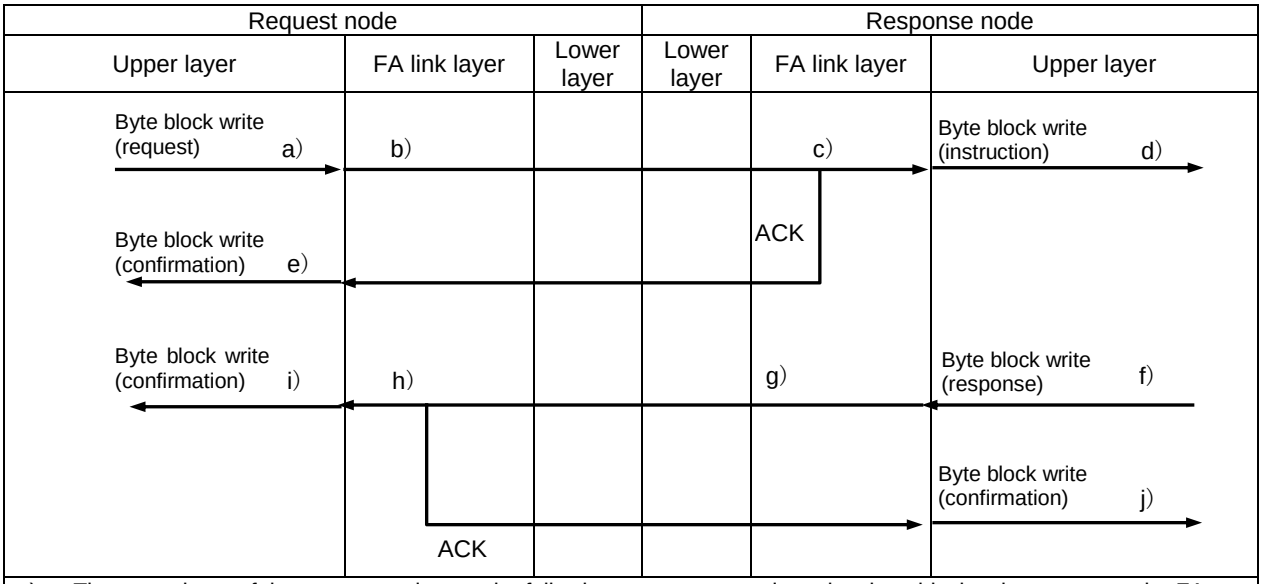
The byte block write service shall be as follows.

a) Function

This service writes byte block data from the source node to the destination node. This service uses the area defined in the virtual address space. For the virtual address space, refer to 7.2.3.

b) Protocol

The sequence of the byte block write service shall be shown in Figure 17.



a) The upper layer of the request node sets the following parameters and sends a byte block write request to the FA link layer.
   — The number of the destination node: DNA
   — The offset address of the byte block to be requested (address position for the virtual address space): M_ADD
   — The size of the byte block to be requested (data size for the virtual address space): M_SZ
   — Requested data (byte block data to be written): data

b) The FA link layer of the request node takes in a PDU (request) from the upper layer as an SDU and sends a request to send to the lower layer.

c) The FA link layer of the response node requests the lower layer to transmit an ACK to the PDU (instruction) received from the lower layer and provides the PDU (instruction) to the upper layer. At this time, the ACK does not contain requested data.
   The response process of the response node is completed before the response node obtains the token, the following f), g), and h) may be performed in advance before transmitting the ACK for the request.

d) The upper layer of the response node takes in the PDU (instruction) as an SDU, and starts a process for the request.

e) The upper layer of the request node checks whether or not the response node received the request from the ACK received from the lower layer.

f) The upper layer of the response node sets the following parameters with the completion of the byte block write process and responses to the FA link layer about the byte block write request.
   — Source node number: DNA
   — The offset address of the responded byte block (address position for the virtual address space): M_ADD
   — The size of the responded byte block (data size for the virtual address space): M_SZ
   — The message result: M_RLT

g) The FA link layer of the response node takes in the PDU (response) from the upper layer as an SDU and sends a request to send to the lower layer.

h) The FA link layer of the request node request the lower layer to transmit an ACK to the PDU (response) received from the lower layer and provides the PDU (confirmation) to the upper layer.

i) The upper layer of the request node takes in the PDU (confirmation) as an SDU, and receives a process completion for the byte block write request.

j) The upper layer of the response node checks whether or not the request node received the process completion with the ACK received from the lower layer.

Figure 17 - The sequence of byte block write service

7.2.5 Word Block Read Service

The word block read service shall be as follows.

a) Function

This service reads word block data of a requested node from a requesting node. It uses areas defined in virtual address space. Figure 18 shows the virtual address space for a word block. Virtual address space is 32-bit address space, and its content is defined by each node.

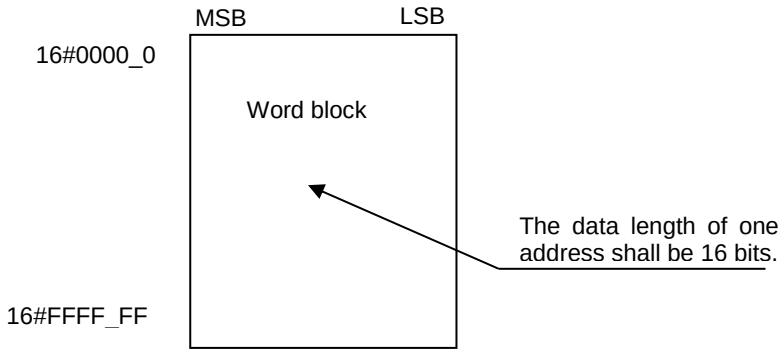
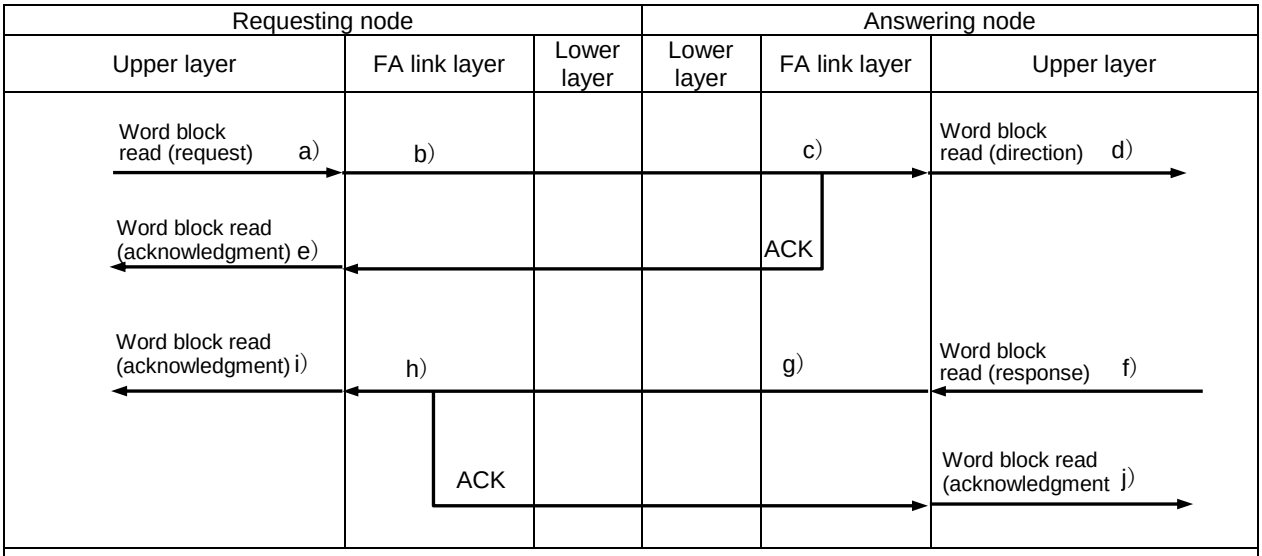


Figure 18 - Virtual Address Space for a Word Block

b) Protocol

The sequence of the word block read service shall be shown in Figure 19.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for a word block read.
   - Node number of destination side: DNA
   - Offset address of the word block to request (address in virtual address space): M_ADD
   - Size of the word block to request (data size in virtual address space): M_SZ

b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.

c) The FA link layer of the answering node requests the lower layer to transmit ACK for the PDU (direction) received from the lower layer, and passes the PDU (direction) to the upper layer.
   If the answering node finishes the answering process before it acquires the token, it may conduct the processes f), g), and h) before sending ACK for the request.

d) The upper layer of the answering node captures the PDU (direction) as an SDU and starts a process for the request.

e) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.

f) The upper layer of the answering node sets the following parameters by completing the process of the word block read and responds to the FA link layer for a word block read request.
   - Node number of source side: DNA
   - Offset address of the word block to respond to (address in virtual address space): M_ADD
   - Size of the word block to respond to (data size in virtual address space): M_SZ
   - Message result: M_RLT
   - Response data (read word block data): data

g) The FA link layer of the answering node captures the PDU (response) as an SDU from the upper layer and requests the lower layer for transmission.

h) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer. ACK does not contain response data.

i) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives response data for the word block read request.

j) The upper layer of the answering node verifies with the ACK received from the lower layer whether the requesting node received the response data.

Figure 19 - Sequence of the Word Block Read Service

7.2.6 Word Block Write Service

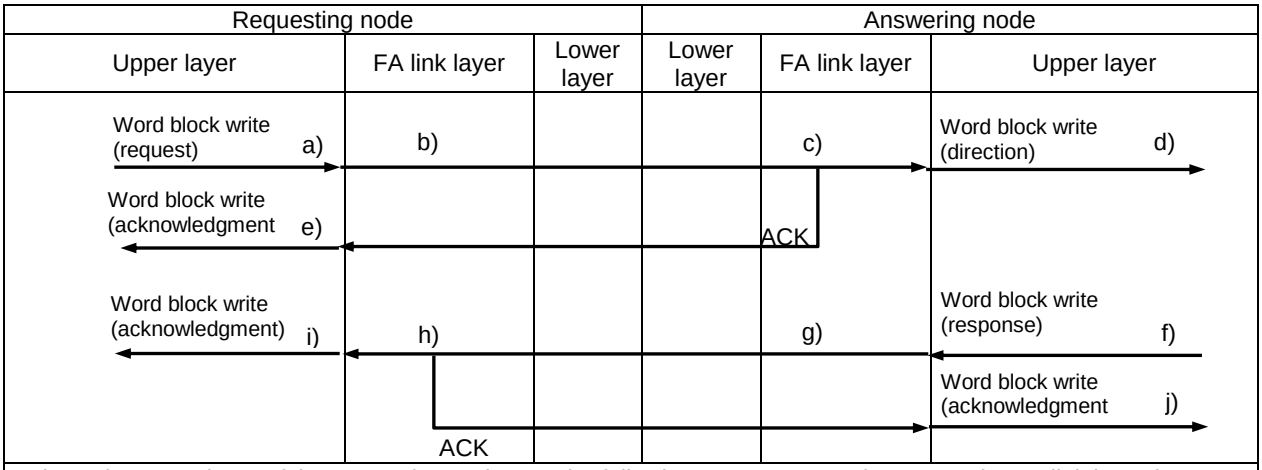
The word block write service shall be as follows:

a) Function

This service writes word block data from a requesting node to a requested node. It uses areas defined in virtual address space. For virtual address space, refer to 7.2.5.

b) Protocol

The sequence of the word block write service shall be shown in Figure 20.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for a word block write.
   - Node number of destination side: DNA
   - Offset address of the word block to request (address position in virtual address space): M_ADD
   - Size of the word block to request (data size in virtual address space): M_SZ
   - Request data (word block data to write): data

b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.

c) The FA link layer of the answering node requests the lower layer to transmit ACK for the PDU (direction) received from the lower layer, and passes the PDU (direction) to the upper layer. In this case, ACK does not contain response data.
   If the answering node finishes the answering process before it acquires the token, it may conduct the processes f), g), and h) before sending ACK for the request.

d) The upper layer of the answering node captures the PDU (direction) as an SDU and starts a process for the request.

e) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.

f) The upper layer of the answering node sets the following parameters by completing the process of the word block write and responds to the FA link layer for a word block write request.
   - Node number of source side: DNA
   - Offset address of the word block to which it responded (address position in virtual address space): M_ADD
   - Size of the word block to which it responded (data size in virtual address space): M_SZ
   - Message result: M_RLT

g) The FA link layer of the answering node captures the PDU (response) as an SDU from the upper layer and requests the lower layer for transmission.

h) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer.

i) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives a process completion for the word block write request.

j) The upper layer of the answering node verifies with the ACK received from the lower layer whether the requesting node received the process completion.

Figure 20 - Sequence of the Word Block Write Service

7.2.7 Network Parameter Read Service

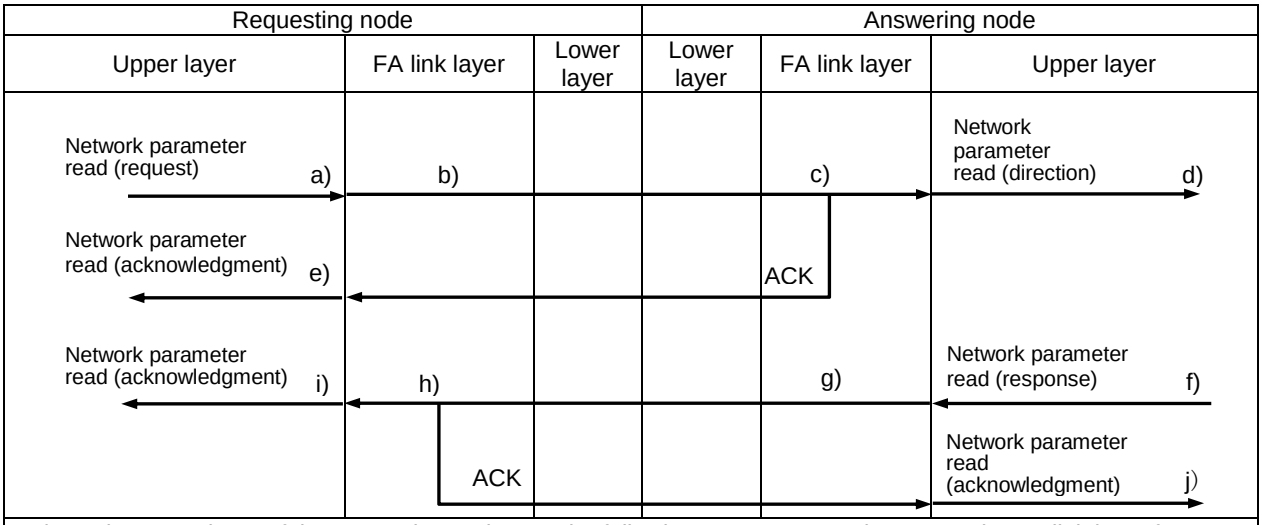
The network parameter read service shall be as follows:

a) Function

This service reads network parameters of a requested node from a requesting node.

b) Protocol

The sequence of the network parameter read service shall be shown in Figure 21.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for a network parameter read.
   — Node number of destination side: DNA

b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.

c) The FA link layer of the answering node requests the lower layer to transmit ACK for the PDU (direction) received from the lower layer, and passes the PDU (direction) to the upper layer.
   If the answering node finishes the answering process before it acquires the token, it may conduct the processes f), g), and h) before sending ACK for the request.

d) The upper layer of the answering node captures the PDU (direction) as an SDU and starts a process for the request.

e) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.

f) The upper layer of the answering node sets the following parameters by completing the process of the network parameter read and responds to the FA link layer for a network parameter read request.
   — Node number of source side: DNA
   — Message result: M_RLT
   — Response data (network parameter of the self node): data
      — Node name of node information: Optionally set by the user.
      — Vender code of node information: Set optionally by the vendor.
      — Manufacturer model name of node information: Optionally set by the manufacturer.
      — Area 1 data top address
      — Area 1 data size
      — Area 2 data top address
      — Area 2 data size
      — Token watchdog time
      — Allowable minimum frame interval time
      — FA link status
      — Protocol type
      — Upper layer status
      — Allowable refresh cycle time setting value
      — Refresh cycle measurement (current value)
      — Refresh cycle measurement (maximum value)
      — Refresh cycle measurement (minimum value)

Figure 21 - Sequence of the Network Parameter Read Service

g) The FA link layer of the answering node captures the PDU (response) as an SDU from the upper layer and requests the lower layer for transmission.
h) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer. ACK does not contain response data.
i) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives response data for the network parameter read request.
j) The upper layer of the answering node verifies with the ACK received from the lower layer whether the requesting node received the response data.

Figure 21 - Sequence of the Network Parameter Read Service (continued)

7.2.8 Network Parameter Write Service

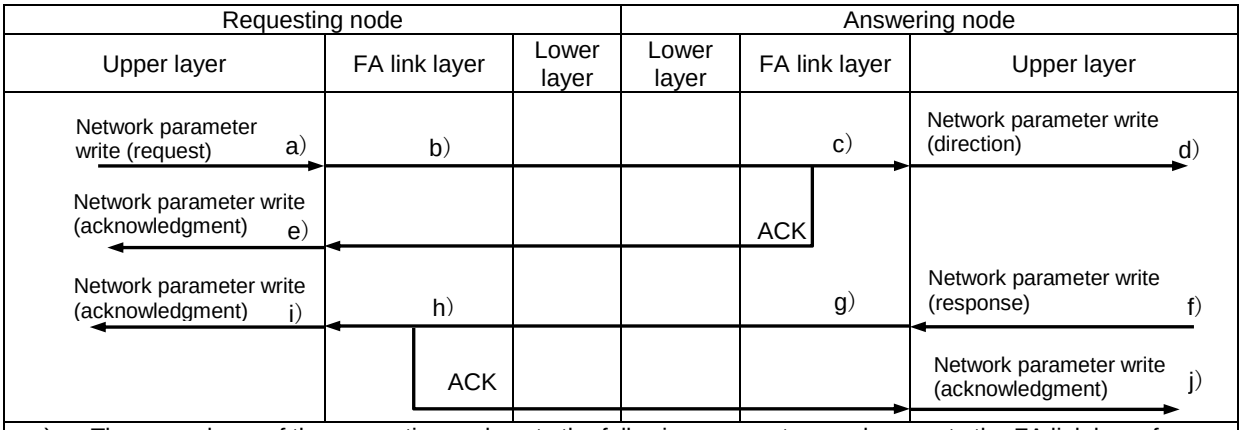
The network parameter write service shall be as follows:

a) Function

This service writes network parameters from a requesting node to a requested node.

b) Protocol

The sequence of the network parameter write service shall be shown in Figure 22.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for a network parameter write.
   - Node number of destination side: DNA
   - Request data (network parameters to write): data
     - Setting parameter flags:
       - 16#01: For writing only the data top address and data size of area 1 and area 2
       - 16#02: For writing only the node name of node information
       - 16#03: For writing the data top address, data size, and node name of node information of area 1 and area 2
     - Area 1 data top address
     - Area 1 data size
     - Area 2 data top address
     - Area 2 data size
     - Node name of node information: Optionally set by the user.

b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.

c) The FA link layer of the answering node requests the lower layer to transmit ACK for the PDU (direction) received from the lower layer, and passes the PDU (direction) to the upper layer. In this case, ACK does not contain response data.
If the answering node finishes the answering process before it acquires the token, it may conduct the processes f), g), and h) before sending ACK for the request.

d) The upper layer of the answering node captures the PDU (direction) as an SDU and starts a process for the request.

e) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.

f) The upper layer of the answering node sets the following parameters by completing the process of the network parameter write and responds to the FA link layer for a network parameter write request.
   - Node number of source side: DNA
   - Message result: M_RLT

g) The FA link layer of the answering node captures the PDU (response) as an SDU from the upper layer and requests the lower layer for transmission.

h) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer.

i) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives a process completion for the network parameter write request.

j) The upper layer of the answering node verifies with the ACK received from the lower layer whether the requesting node received the process completion.

Figure 22 - Sequence of the Network Parameter Write Service

7.2.9 Stop Command Service

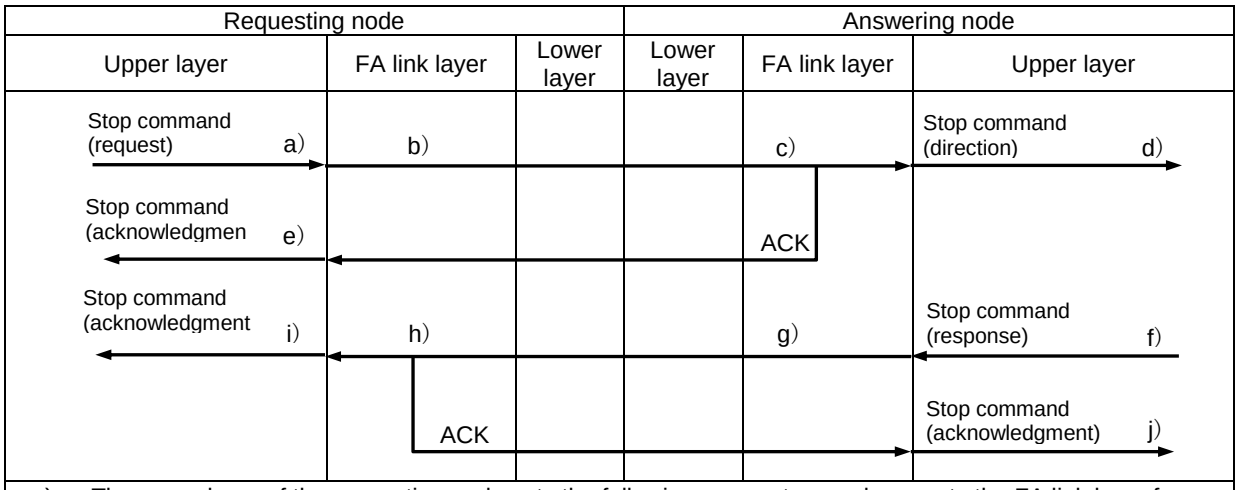
The stop command service shall be as follows:

a) Function

This service requests a requested node for the stop command from a requesting node.

b) Protocol

The sequence of the stop command service shall be shown in Figure 23.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for the stop command.
   — Node number of destination side: DNA
b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.
c) The FA link layer of the answering node requests the lower layer to transmit ACK for the PDU (direction) received from the lower layer, and passes the PDU (direction) to the upper layer.
   If the answering node finishes the answering process before it acquires the token, it may conduct the processes f), g), and h) before sending ACK for the request.
d) The upper layer of the answering node captures the PDU (direction) as an SDU and starts a process for the request.
e) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.
f) The upper layer of the answering node sets the following parameters by completing the process of the stop command and responds to the FA link layer for the stop command request.
   — Node number of source side: DNA
   — Message result: M_RLT
g) The FA link layer of the answering node captures the PDU (response) as an SDU from the upper layer and requests the lower layer for transmission.
h) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer.
i) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives a process completion for the stop command request.
j) The upper layer of the answering node verifies with the ACK received from the lower layer whether the requesting node received the process completion.

Figure 23 - Sequence of the Stop Command Service

7.2.10 Operation Command Service

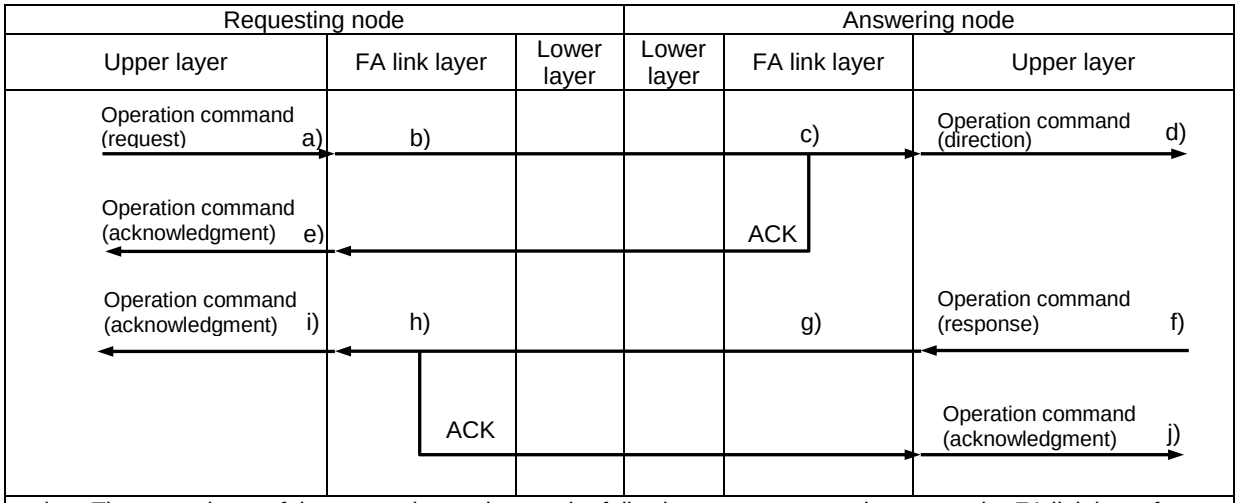
The operation command service shall be as follows:

a) Function

This service requests a requested node for the operation command from a requesting node.

b) Protocol

The sequence of the operation command service shall be shown in Figure 24.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for the operation command.
   — Node number of destination side: DNA
b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.
c) The FA link layer of the answering node requests the lower layer to transmit ACK for the PDU (direction) received from the lower layer, and passes the PDU (direction) to the upper layer.
   If the answering node finishes the answering process before it acquires the token, it may conduct the processes f), g), and h) before sending ACK for the request.
d) The upper layer of the answering node captures the PDU (direction) as an SDU and starts a process for the request.
e) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.
f) The upper layer of the answering node sets the following parameters by completing the process of the operation command and responds to the FA link layer for the operation command request.
   — Node number of source side: DNA
   — Message result: M_RLT
g) The FA link layer of the answering node captures the PDU (response) as an SDU from the upper layer and requests the lower layer for transmission.
h) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer.
i) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives a process completion for the operation command request.
j) The upper layer of the answering node verifies with the ACK received from the lower layer whether the requesting node received the process completion.

Figure 24 - Sequence of the Operation Command Service

7.2.11 Profile Read Service

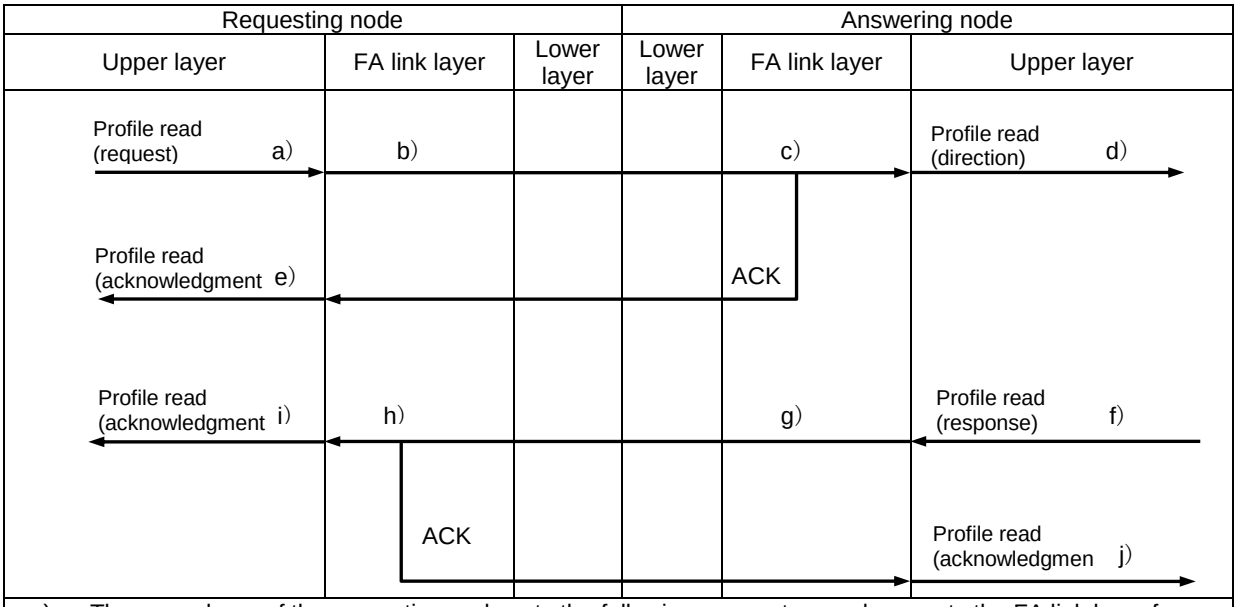
The profile read service shall be as follows:

a) Function

This service reads the profile of a requested node from a requesting node.

b) Protocol

The sequence of the profile read service shall be shown in Figure 25.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for the profile read.
   — Node number of destination side: DNA
b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.
c) The FA link layer of the answering node requests the lower layer to transmit ACK for the PDU (direction) received from the lower layer, and passes the PDU (direction) to the upper layer.
   If the answering node finishes the answering process before it acquires the token, it may conduct the processes f), g), and h) before sending ACK for the request.
d) The upper layer of the answering node captures the PDU (direction) as an SDU and starts a process for the request.
e) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.
f) The upper layer of the answering node sets the following parameters by completing the process of the profile read and responds to the FA link layer for a profile read request.
   — Node number of source side: DNA
   — Message result: M_RLT
   — Response data (read profiles): data
     The content of the data is beyond this standard (Refer to JEM-TR 214 (device profile common specifications)).
g) The FA link layer of the answering node captures the PDU (response) as an SDU from the upper layer and requests the lower layer for transmission.
h) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer. ACK does not contain response data.
i) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives response data for the profile read request.
j) The upper layer of the answering node verifies with the ACK received from the lower layer whether the requesting node received the response data.

Figure 25 - Sequence of the Profile Read Service

7.2.12 Transparent message service

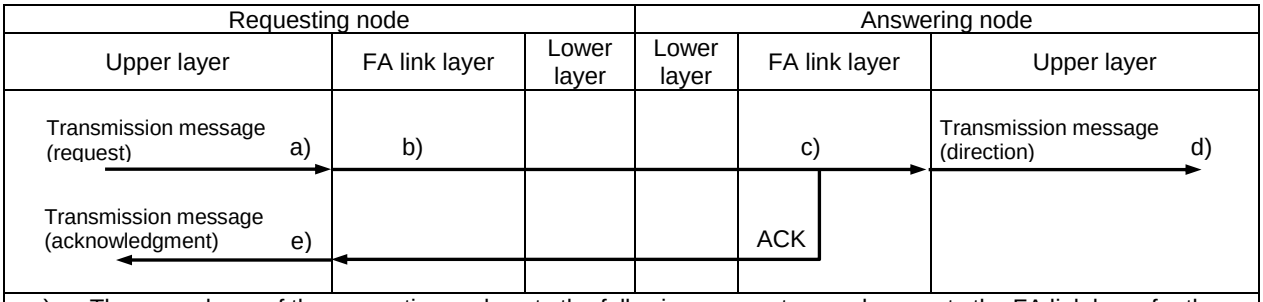
The transparent message service shall be as follows:

a) Function

This service sends user-defined service from a requesting node to a requested node. It does not define responses. It uses a user-defined transparent message for a response.

b) Protocol

The sequence of the transparent message shall be shown in Figure 26.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for the transparent message.
   - Transmission form: one-to-one transmission (PPT)/one-to-*n* transmission (BCT)
   - Node number of destination side: DNA
   - User data (optionally defined by the user): data

b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.

c) The FA link layer of the answering node requests the lower layer to transmit ACK for the PDU (direction) received from the lower layer, and passes the PDU (direction) to the upper layer.

d) The upper layer of the answering node captures the PDU (direction) as an SDU and starts a process for the request.

e) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.

Note For one-to-*n* transmission, there is no ACK [c)] or transparent message (acknowledgment) [e)].

Figure 26 - Sequence of the Transparent Message Service

7.2.13 Log Data Read Service

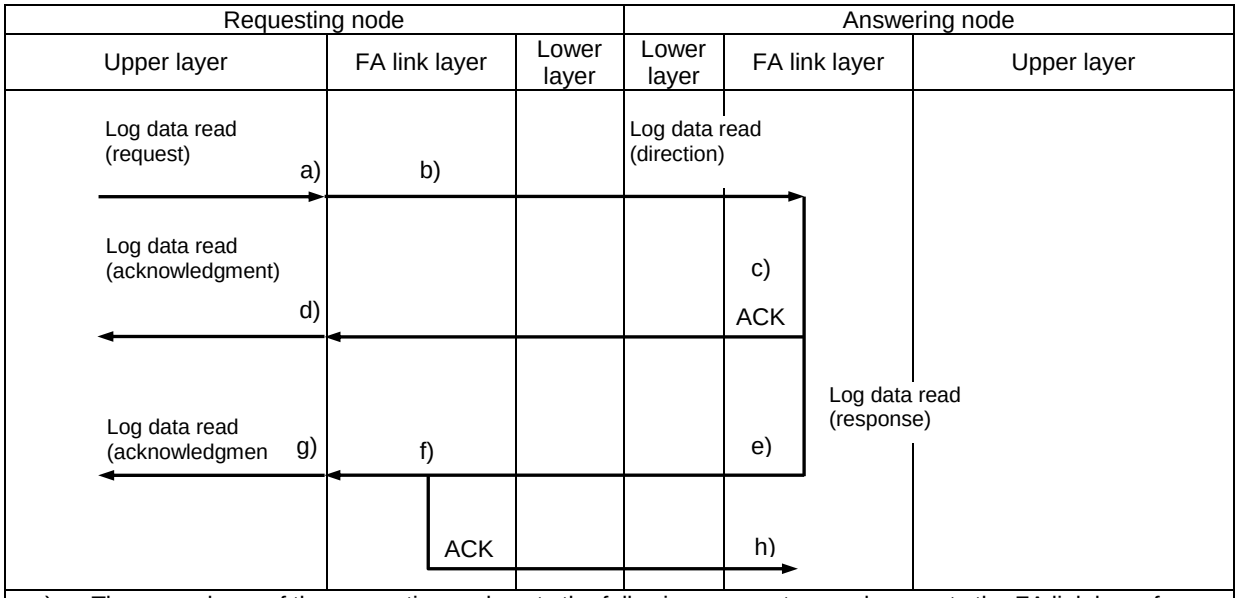
The log data read service shall be as follows:

a) Function

This service reads log data of a requested node from a requesting node.

b) Protocol

The sequence of the log data read service shall be shown in Figure 27.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for the log data read.
   — Node number of destination side: DNA

b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.

c) The FA link layer captures the PDU (direction) received from the lower layer as an SDU and starts the process for the log data read request, requesting the lower layer to send ACK for the PDU (direction) received from the lower layer.
   If the answering node finishes the answering process before it acquires the token, it may conduct the processes e) and f) before sending ACK for the request.

d) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.

e) The FA link layer of the answering node sets the following parameters by completing the process of the log data read and responds to the lower layer for a log data read request.
   — Node number of source side: DNA
   — Message result: M_RLT
   — Response data (read log data): data
     For the content of the data, refer to Annex B.

f) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer. ACK does not contain response data.

g) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives response data for the log data read request.

h) The FA link layer of the answering node discards the ACK received from the lower layer.

Figure 27 - Sequence of the Log Data Read Service

7.2.14 Log Data Clear Service

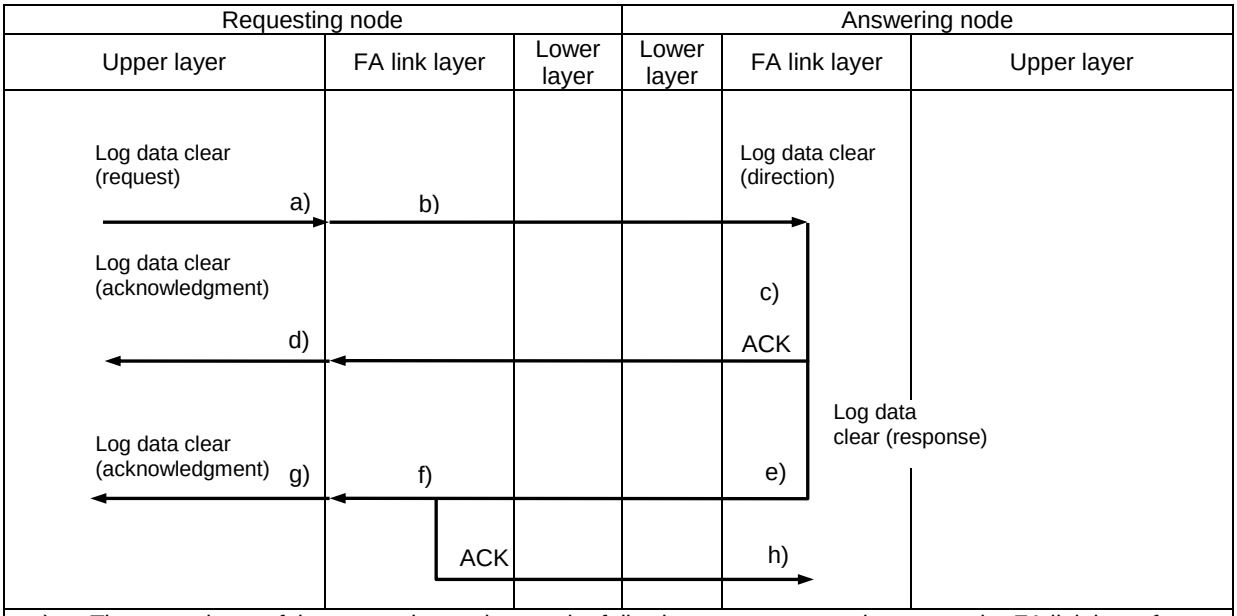
The log data clear service shall be as follows:

a) Function

This service clears log data of a requested node from a requesting node.

b) Protocol

The sequence of the log data clear service shall be shown in Figure 28.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for the log data clear.
   - Transmission form: one-to-one transmission (PPT)/one-to-*n* transmission (BCT)
   - Node number of destination side: DNA

b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.

c) The FA link layer captures the PDU (request) received from the lower layer as an SDU and starts the process for the log data clear request, requesting the lower layer to send ACK for the PDU (direction) received from the lower layer.
   If the answering node finishes the answering process before it acquires the token, it may conduct the processes e) and f) before sending ACK for the request.

d) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.

e) The FA link layer of the answering node sets the following parameters by completing the process of the log data clear and responds to the lower layer for a log data clear request.
   - Node number of source side: DNA
   - Message result: M_RLT

f) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer. ACK does not contain response data.

g) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives response data for the log data clear request.

h) The FA link layer of the answering node discards the ACK received from the lower layer.

Note For one-to-*n* transmission, ACK [c)], d), and subsequent processes are not conducted.

Figure 28 - Sequence of the Log Data Clear Service

7.2.15 Message Return Service

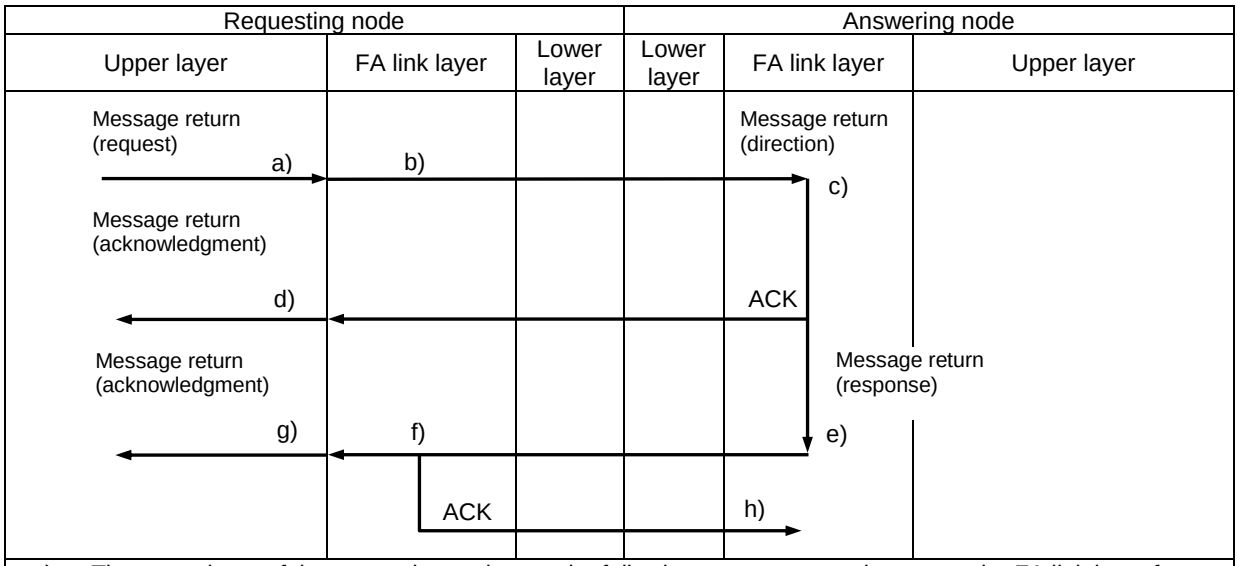
The message return service shall be as follows:

a) Function

This service sends the received message data by return and conducts a message transmission test.

b) Protocol

The sequence of the message return service shall be shown in Figure 29.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for a message return.
   - Node number of destination side: DNA
   - Return data: data

b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.

c) The FA link layer captures the PDU (request) received from the lower layer as an SDU and starts the process for the message return request, requesting the lower layer to send ACK for the PDU (direction) received from the lower layer.
   If the answering node finishes the answering process before it acquires the token, it may conduct the processes e) and f) before sending ACK for the request.

d) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.

e) The FA link layer of the answering node sets the following parameters by completing the process of the message return and responds to the lower layer for a message return request.
   - Node number of source side: DNA
   - Message result: M_RLT
   - Response data (return data): data

f) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer. ACK does not contain response data.

g) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives response data for the message return request.

h) The FA link layer of the answering node discards the ACK received from the lower layer.

Figure 29 - Sequence of the Message Return Service

7.2.16 Vendor-specific Message Service

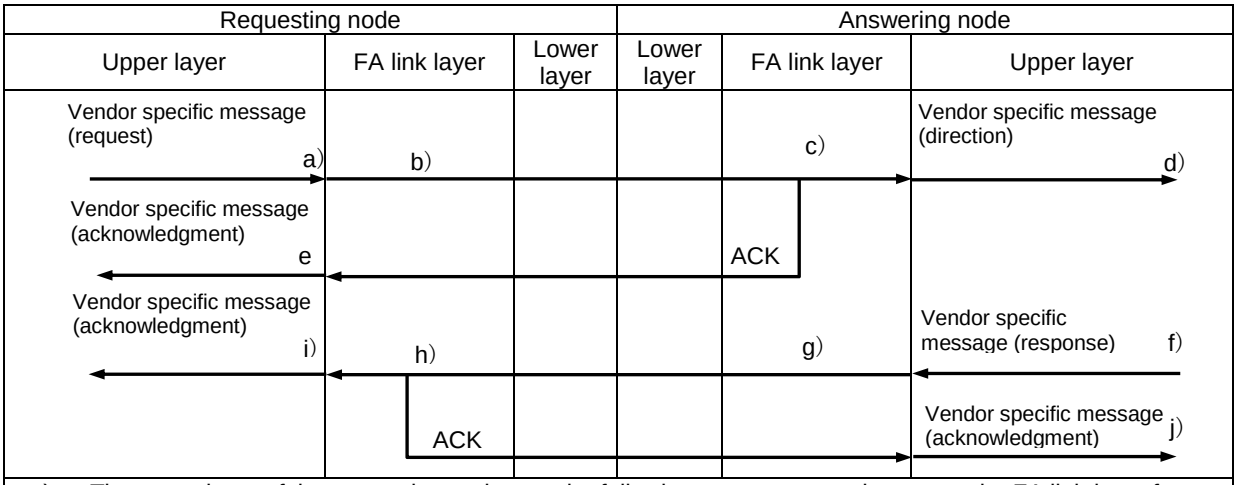
The vendor-specific message service shall be as follows:

a) Function

This service requests a requested node for a vendor-specific message from a requesting node.

b) Protocol

The sequence of the vendor-specific message service shall be shown in Figure 30.



a) The upper layer of the requesting node sets the following parameters and requests the FA link layer for a vendor-specific message.
   - Transmission form: one-to-one transmission (PPT)/one-to-*n* transmission (BCT)
   - Node number of destination side: DNA
   - Vender code of node information: VDN
   - Sub code: SCODE
   - Message data: data

b) The FA link layer of the requesting node captures the PDU (request) as an SDU from the upper layer and requests the lower layer for transmission.

c) The FA link layer of the answering node requests the lower layer to transmit ACK for the PDU (direction) received from the lower layer, and passes the PDU (direction) to the upper layer.
If the answering node finishes the answering process before it acquires the token, it may conduct the processes f), g), and h) before sending ACK for the request.

d) The upper layer of the answering node captures the PDU (direction) as an SDU and starts a process for the request.

e) The upper layer of the requesting node verifies with the ACK received from the lower layer whether the answering node received the request.

f) The upper layer of the answering node sets the following parameters by completing the process of the vendor-specific message and responds to the FA link layer for a vendor-specific message request.
   - Node number of source side: DNA
   - Message result: M_RLT
   - Vender code of node information: VDN
   - Sub code: SCODE
   - Message data: data

g) The FA link layer of the answering node captures the PDU (response) as an SDU from the upper layer and requests the lower layer for transmission.

h) The FA link layer of the requesting node requests the lower layer to transmit ACK for the PDU (response) received from the lower layer, and passes the PDU (acknowledgment) to the upper layer. ACK does not contain response data.

i) The upper layer of the requesting node captures the PDU (acknowledgment) as an SDU and receives response data for the vendor-specific message request.

j) The upper layer of the answering node verifies with the ACK received from the lower layer whether the requesting node received the response data.

Note For one-to-*n* transmission, ACK [c)], e), and subsequent processes are not conducted.

Figure 30 - Sequence of the Vendor-specific Message Service

7.3 Network Management Service

7.3.1 Setting Configuration Parameters

Configuration parameters are used to define the operations of the FA link. Set the configuration parameters as per the following:

a) Node number

The upper layer sets the Node number for the FA link layer as the initial value. The range of the Node number shall be 1 to 254. If the Node number is not set, the node will not participate in the network. If the node detects an overlapping Node number when it is trying to participate in the network, it will not send any frames. If the node is participating in the network, this parameter shall not be changed.

b) Common memory area 1 and 2 data top address and data size

Can be set from the upper layer and the network. If the node is participating in the network, this parameter shall not be changed.

If this parameter is not set from the upper layer, the FA link layer determines that the node has no transmission area and participates in the network. If the network parameter write service and configuration parameter setting service are used from the network to set this parameter, the node leaves the network after receiving the message and participates in the network again in an in-ring startup state.

The ranges of common memory areas 1 and 2 data top address and data size are as per the following. When these parameters are set, the FA link layer will set the common memory setting completion flag to the FA link status.

1) Common memory area 1 data top address (16#0000-16#01FF)

2) Common memory area 1 data size (16#0000-16#0200)

3) Common memory area 2 data top address (16#0000-16#1FFF)

4) Common memory area 2 data size (16#0000-16#2000)

c) Node name of node information

Can be set from the upper layer and the network.

If the network parameter write service and configuration parameter setting service are used from the network to set this parameter, the node leaves the network after receiving the message and participates in the network again in an in-ring startup state. If only the node name is changed, the node does not leave the network and continues the communication.

Set the node name by using up to 10 ASCII characters.

d) Token watchdog time

The time is a node-specific value and may be set to the node as the initial value from the upper layer. If this value is not set, the node will not participate in the network. If the node is participating in the network, this parameter shall not be changed.

Set the token watchdog time in the range of 1 to 255 in steps of 1 ms.

e) Allowable minimum frame interval time

The interval is a node-specific value and may be set to the node as the initial value from the upper layer. If this value is not set, the node will not participate in the network. If the node is participating in the network, this parameter shall not be changed.

Set the allowable minimum frame interval time in the range of 0 to 50 in steps of 100 μs.

7.3.2 Reading the Local Node Management Information Parameters

The self node management information parameters provide data on the settings and

operations of the self node set by the upper layer or the network parameter write service and the configuration parameter setting service when starting the node.

The FA link layer uses the information of the self node management information parameters as information of the participation request frame and network parameter read. A value within parentheses indicates a data length.

a) Parameter

The parameters are as per the following:

1) Node number (1 octet): 1-254

2) Common memory area 1 data top address (2 octets): Word address (16#0000-16#01FF)

3) Common memory area 1 data size (2 octets): Size (16#0000-16#0200)

4) Common memory area 2 data top address (2 octets): Word address (16#0000-16#1FFF)

5) Common memory area 2 data size (2 octets): Size (16#0000-16#2000)

6) Upper layer status (2 octets): Operating information, error information, etc. Refer to 7.3.6 a) for details.

7) Token watchdog time (1 octet): 1-255: by 1 ms

8) Allowable minimum frame interval time (1 octet): 0-50: by 100 μs

9) Vendor code of node information (10 octets): Name of the vendor

10) Manufacturer model name (10 octets): The Manufacturer model name of the node

11) Node name of node information (10 octets): Name of the node set by the user

12) Protocol type (1 octet): 16#80 fixed

13) FA link status (1 octet): Participation, leaving, etc. Refer to 7.3.6 b) for details.

14) Self node status (1 octet): Detecting overlapping Node numbers, etc. Refer to 7.3.6 c) for details.

7.3.3 Reading the Participating Node Management Information Parameters

The participating node management information parameters provide the information of the condition of each node participating in the network. The FA link layer manages the parameter data for each node participating in the network.

a) Parameter

The parameters are as per the following:

1) Node number (1 octet): 1-254

2) Upper layer status (2 octets): Operating information, error information, etc. Refer to 7.3.6 a) for details.

3) Common memory area 1 data top address (2 octets): Word address (16#0000-16#01FF)

4) Common memory area 1 data size (2 octets): Size (16#0000-16#0200)

5) Common memory area 2 data top address (2 octets): Word address (16#0000-16#1FFF)

6) Common memory area 2 data size (2 octets): Size (16#0000-16#2000)

7) Allowable refresh cycle time (2 octets): 0-65535: by 1 ms

8) Token watchdog time (1 octet): 1-255: by 1 ms

9) Allowable minimum frame interval time (1 octet): 0-50: by 100 μs

10) FA link status (1 octet): Information of participation, leaving, etc. Refer to 7.3.6 b) for details.

The information of parameters are contained in the frame received by each node, except the information of participation/leaving and communication invalidity detection contained in "10) FA link status." Each node updates the participating node management information parameters when it receives the final cyclic frame from a participation request frame or an unrecognized node. However, "2) Upper layer status" and "7) Allowable refresh cycle time" update the parameters whenever they receive the final cyclic frame.

A node changes the information of participation/leaving and communication invalidity detection contained in the FA link status of the applicable node to the participation status when it receives a participation request frame or token frame from each node. When it recognizes a leaving node by three continuous detections of the receive timeout of a token frame from each node, it changes the information of participation/leaving contained in the FA link status of the applicable node to the leaving status.

7.3.4 Reading the Network Management Information Parameters

The network management information parameters provide information of the state common to the network.

a) Parameter

The parameters are as per the following:

1) Token holding Node number (1 octet): The number of the node currently holding the token (1-254)

2) Allowable minimum frame interval time (1 octet): 0-50: by 100 μs

3) Allowable refresh cycle time (2 octets): 0-65535: by 1 ms

4) Refresh cycle measurement time (current value) (2 octets): 0-65535: by 1 ms

5) Refresh cycle measurement time (maximum value) (2 octets): 0-65535: by 1 ms

6) Refresh cycle measurement time (minimum value) (2 octets): 0-65535: by 1 ms

7.3.5 Reading the Message Sequence Number Management Information

The message sequence number management information provides information of the sequence number and version of sequence number in a message transmission.

a) For transmission

Prepare the following for the transmission table:

1) Version of sequence number (4 octets): 16#0000_0001-16#FFFF_FFFF

2) Sequence number (one-to-*n* transmission) (4 octets): 16#0000_0001-16#FFFF_FFFF

3) Sequence number (one-to-one transmission) (4 octets) x 256: 16#0000_0001-16#FFFF_FFFF

b) For receiving

Prepare the following 254 (number of nodes) 12-octettables as a receiving table. If the sequence number is "0," it indicates that no message has been received yet.

1) Version of sequence number (4 octets): 16#0000_0000-16#FFFF_FFFF

2) Sequence number (one-to-*n* transmission) (4 octets): 16#0000_0000-16#FFFF_FFFF

3) Sequence number (one-to-one transmission) (4 octets): 16#0000_0000-16#FFFF_FFFF

7.3.6 Reading the Node Status

The node status has three types: upper layer status, FA link status, and self node status.

a) Upper layer status

The upper layer status consists of the operating information and error information of the upper layer (Refer to Table 5.).

1) Operating information of the upper layer

1.1) RUN

Indicates that the "upper layer program" is operating.

1.2) STOP

Indicates that the "upper layer program" is halted.

Note 1 An "upper layer program" is a program that has the upper layer of the FA link and interfaces such as a controller (e.g. PLC).

Note 2 The definition of "operating" and "halted" shall be determined by the upper layer and beyond this standard.

2) Error information of the upper layer

2.1) NORMAL

Indicates that the upper layer is normal and the cyclic data and message data are valid.

2.2) WARNING

Indicates that an abnormal condition where operation can be continued is occurring in the upper layer but the cyclic data and message data are valid.

2.3) ALARM

Indicates that an abnormal condition where operation cannot be continued is occurring in the upper layer and the cyclic data and message data are not valid.

2.4) Error content of the upper layer

Indicates the error content of the "upper layer program."

Table 5 - Conditions of the Upper Layer

Error Information	Operating Information	
	RUN	STOP
NORMAL	The "upper layer program" is operating and the cyclic data and message data are valid.	The "upper layer program" is halted and the cyclic data and message data are valid.
WARNING	The "upper layer program" is operating and has an anomaly, but the cyclic data and message data are valid.	The "upper layer program" is halted and has an anomaly, but the cyclic data and message data are valid.
ALARM	The cyclic data and message data are not valid.	The cyclic data and message data are not valid.

b) FA link status

Of the FA link status information, the information of participation/leaving and communication invalidity detection of each node is managed for each node. Other information is contained in frames in the network. For the configuration on the frame structure, refer to 10.1.1.

1) Address overlap detection flag

Indicates that there is an overlap in the common memory area settings for the node connected to the network.

2) Common memory setting completion flag

Indicates that the common memory settings for the node are complete.

3) Common memory data validity notice flag

Indicates that the cyclic data is valid.

4) Upper layer actuating signal error flag

Indicates that updates of survival signals in the upper layer cannot be recognized.

5) Communication invalidity detection flag

Indicates that frames with a different token mode exist in the network.

6) Node participation/leaving flag

Indicates whether the node is participating in the network or not.

c) Self node status

Information of the self node to provide for the upper layer

1) Node number overlap flag

Indicates that a node with the same node number as the setting value for the self node exists in the network.

2) Token watchdog time error flag

Indicates that a transmission process does not finish within the token watchdog time set to the self node.

3) Receiving waiting status flag

Indicates a state of waiting for receiving a frame without receiving any frames in the initialization of the network.

4) Initialization error flag

Indicates that there was an error in the initial settings or resetting parameters.

7.4 Load Measure Service

This service conducts the start and termination request of load measure for the Node Ver. 3.00 via UDP or TCP communication from the setting tool.

The measurement result is added as a data division to a response frame for the termination request.

The measurement items include the token retention time and general purpose communication data sender log.

Table 6 shows a list of the transaction codes used by the load measure service.

Note An unimplemented node shall conduct an unimplemented response of M_RLT (message result in the FA link header) = 2 within one second or conduct no response.

Table 6 - Load Measure Service Transaction Code List

Name	Transaction Code
Start frame of token retention time measurement (request)	65 020
End frame of token retention time measurement (request)	65 021
Start frame of measurement in general purpose communication data sender log (request)	65 022
End frame of measurement in general purpose communication data sender log (request)	65 023
Start frame of token retention time measurement (response)	65 220
End frame of token retention time measurement (response)	65 221
Start frame of measurement in general purpose communication data sender log (response)	65 222
End frame of measurement in general purpose communication data sender log (response)	65 223
Note Use "55004" for the port number at the time of transmission and receiving.	

7.5 IO Allocation Setting Service

7.5.1 Overview

There are two methods of IO allocation settings: downloading by directly connecting a vendor-specific tool and downloading from the setting tool through UDP or TCP protocol communication via the network. Figure 31 shows an example of settings via the network.

Table 7 shows a list of the transaction codes used by the IO allocation settings.

Note An unimplemented node shall conduct an unimplemented response of M_RLT (message result in the FA link header) = 2 within one second or conduct no response.

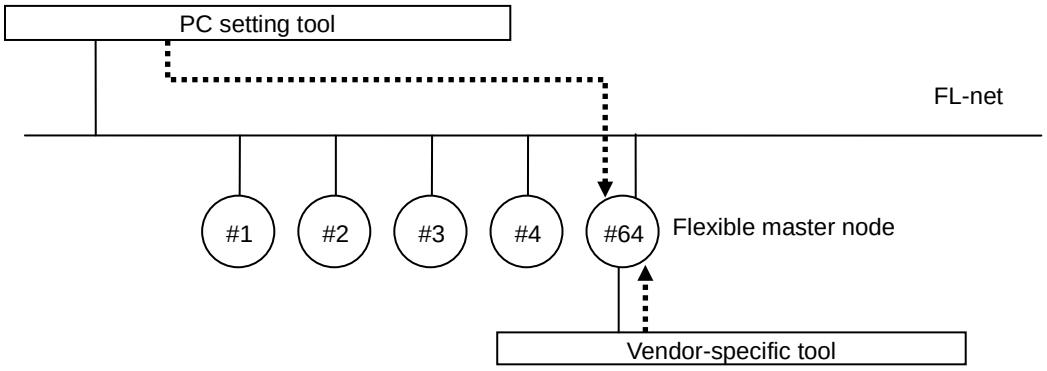


Figure 31 - IO Allocation Setting Service

Table 7 - IO Allocation Setting Service Transaction Code List

Name	Transaction Code
IO allocation setting frame (request)	65 018
IO allocation setting frame (response)	65 218
Note Use "55004" for the port number at the time of transmission and receiving.	

7.5.2 IO allocation setting

IO allocation setting shall be as follows.

a) Fixed setting

Default with which no operation for IO allocation setting is required.

b) Flexible setting

The user can define each master node with respect to the number of slaves (total number) under the control of the master node, ID nos. of these slave nodes, IO data address and size, and IO status address.

Flexible setting, shall be shown in Figure 32.

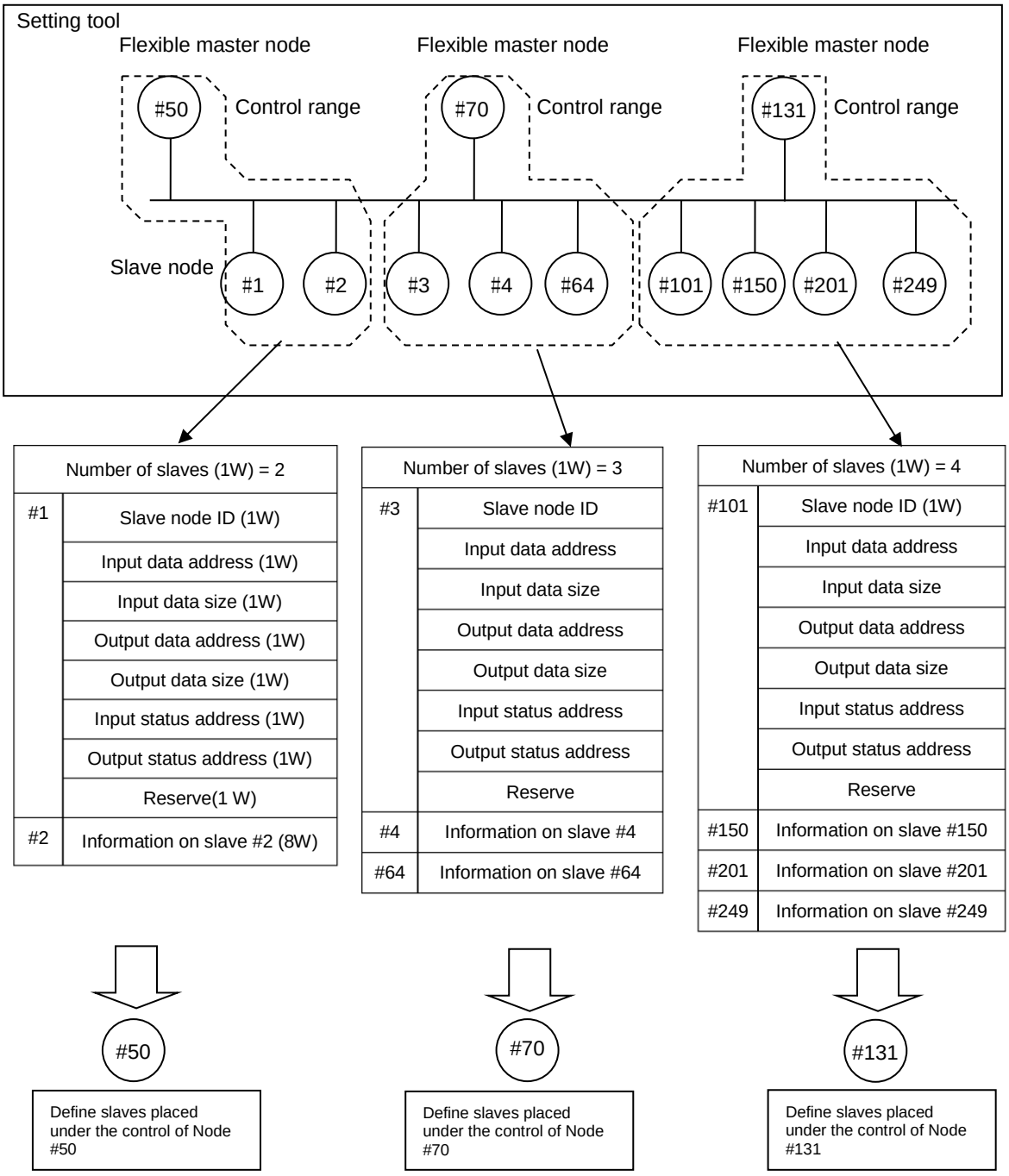


Figure - 32 IO Allocation by Flexible Setting mode

7.6 Service to lower layers

This service available on the interface between FA link layer and lower layers is to permit the FA link layer and FA link entities on the same layer to exchange protocol data units.

The lower layers shall prepare four different SAPs available for the FA link layer: transmission,

receiving a cyclic data, receiving a message, and receiving a request for participation. The FA link layer and lower layers execute request or commanding operations under each SAP to exchange information.

7.6.1 Function of the “transmission” SAP

The “transmission” SAP provides the FA link layer a request for transmission.

a) Parameters

Uses the following parameters

1) Address of the request receiver

2) SAP of the request receiver

3) SDU

7.6.2 Function of the “receiving a cyclic data” SAP

The “receiving a cyclic data” SAP provides the FA link layer instructions for receiving cyclic frames.

a) Parameters

Uses the following parameter

1) SDU

7.6.3 Function of the “receiving a message” SAP

The “receiving a message” SAP provides the FA link layer instructions for receiving a message.

a) Parameters

Uses the following Parameter

1) SDU

7.6.4 Function of the “receiving a request for participation” SAP

The “receiving a request for participation” SAP provides the FA link layer instructions for receiving a frame containing a request for participation.

a) Parameters

Uses the following parameter

1) SDU

7.7 Solicitation frame transmission service

7.7.1 Overview

This service provided by the master is to give the slaves information on IO allocation settings.

Max. 1024 octets of data can be sent at a time. If a solicitation data exceeds 1024 octets (if the solicitation frames to be sent to the slaves exceed 64 nodes), the master shall divide the frames into groups and transmit them separately.

This solicitation frame transmission is a broadcast service over all slaves, which prevents the master from checking acknowledgement from each slave. Therefore the application shall check such acknowledgement by checking whether the slave is participating in the network, and by referencing the master node ID contained in the status area of each slave.

For a multi-master system, it is necessary to check duplicated use of slaves by referencing the master node ID contained in the status area of each slave.

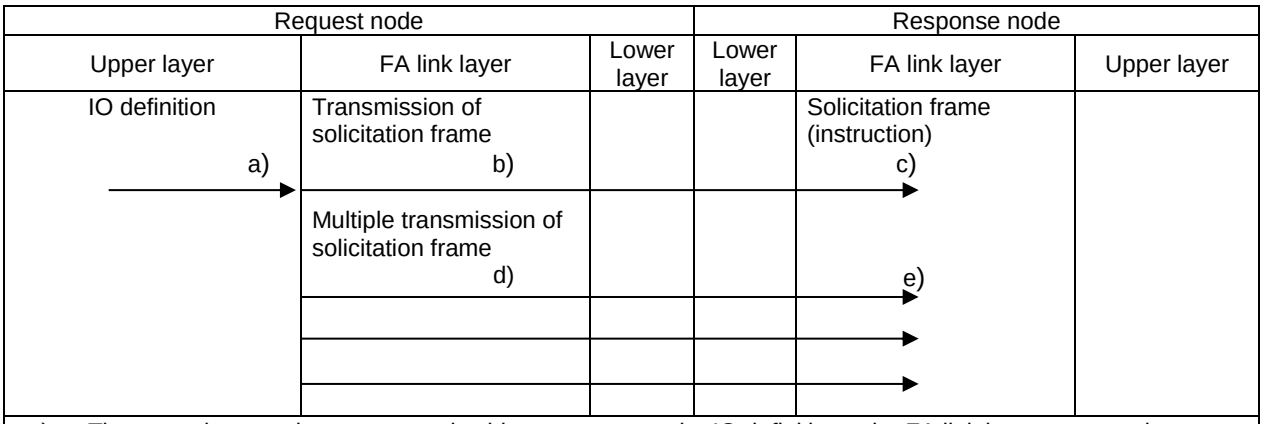
Table 8 shows the list of solicitation frame transmission service transaction codes.

Table - 8 List of Solicitation Frame Transmission Service Transaction Codes

Name	Transaction code
Solicitation frame	65 017
Note: The UDP port number shall be 55 003 for transmission and 55 002 for receiving operation.	

7.7.2 Protocol

The solicitation frame transmission service sequence shall be shown in Figure 33



a) The upper layer at the request node side requests, per the IO definitions, the FA link layer to transmit a solicitation frame.

b) The FA link layer at the request node side generates, per the IO definitions, a solicitation frame and requests the lower layer to transmit the frame after the participation request frame acceptance times out (i.e. during the network startup state), or during retention of the token (i.e. during the valid linking state).

c) The FA link layer at the response node side executes the relevant processes in accordance with the IO definitions intended for the node contained in the solicitation frames that the FA link layer have received from the lower layer.

d) If the IO definitions exceed 64 nodes, the FA link layer at the request node side provides the lower layer a request for transmitting the remaining solicitation frame after the solicitation frame batch transmission frame interval (SFBTFI) is up (i.e. during the network startup state) or during retention of the token, and after the solicitation frame transmission waiting time (SFTWT) is up [i.e. during the valid linking state [a)]].

e) The FA link layer at the response node side executes the relevant processes in accordance with the IO definitions intended for the node chosen among those contained in the solicitation frames that the FA link layer have received from the lower layer. When a slave received two or more IO definitions, use the one received last.

Note: A solicitation frame may be divided into pieces depending on the size so that they can be transmitted separately. Each node shall pick up only the IO definitions intended for its self node from each segment of the frame. The remaining segments that do not contain such IO definitions can be discarded. Each slave does not have to retain the divided data or put them together.

Note [a)] Transmitting a solicitation frame in the valid linking state means that the master shall transmit the frame while it has a token.

To transmit a message frame or solicitation frame, satisfy the following four conditions.

1) If the refresh measuring time (RMT) is not greater than 90% of the maximum allowable refresh cycle time (RCT), the message frame or solicitation frame that shall currently be transmitted first according to the alternate transmission order can be transmitted in the one-frame transmission mode.
2) If the refresh measuring time (RMT) is greater than 90% of the maximum allowable refresh cycle time (RCT) but smaller than RCT, and if the solicitation frame shall be transmitted first according to the current alternate transmission order, the solicitation frame can be transmitted in the one-frame transmission mode.
3) If the refresh measuring time (RMT) is greater than the maximum allowable refresh cycle time (RCT), neither message frame nor solicitation frame can be transmitted irrespective of which of message frame or solicitation frame shall be transmitted first according to the current alternate transmission order.
4) If there are no message frames or solicitation frames to be transmitted, no frames will be transmitted irrespective of which of message frame or solicitation frame shall be transmitted first according to the current alternate transmission order.

Figure 33 Solicitation Frame Transmission Service Sequence

The transmission order for solicitation frames is determined as follows. Solicitation frame transmission orders . shall be shown in Figure 34.

a) In the case that a solicitation frame shall be transmitted according to the current alternate transmission order, where no cyclic data or message data exist and yet-to-participate slaves exist, transmit the solicitation frame, cyclic frame without data and token frame in this order.

b) In the case that a solicitation frame shall be transmitted first according to the current alternate transmission order, where both cyclic data and yet-to-participate slaves exist,

transmit the solicitation frame, cyclic frame and token frame in this order.

c) In the case that a solicitation frame shall be transmitted first according to the current alternate transmission order, where both divided cyclic data and yet-to-participate slave exist, transmit the solicitation frame, divided cyclic frame and token frame in this order.

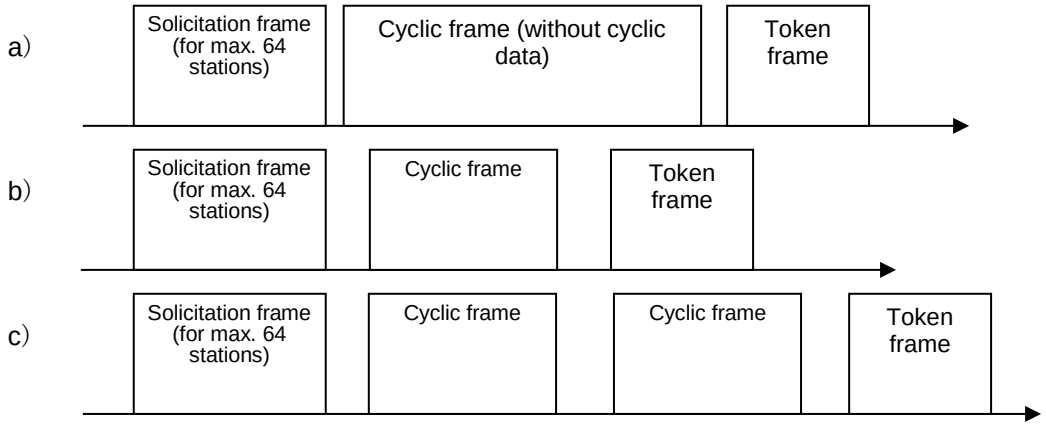


Figure - 34 Transmission Order of Solicitation Frame

8 Functions

8.1 Node ID (no.)

FL-net specifies node nos. as their identification information. Node nos. are used for “SNA” and “DNA” presentation in the FA link header (see 10.1).

— FL-net uses node nos. from 1 to 254. However node nos. from 250 to 254 are reserved for maintenance purpose.

— Node no. 0 shall not be used.

— Node no. 255 shall be used for DNA for broadcast message.

— For token frames, use DNA for presenting the Node number of the next token transmission destination.

8.2 Protocol data unit

The FA link protocol uses “frame” as a protocol data unit (PDU).

Data transmitted via a network are defined for each layer of the network. The frame structure is as shown in Figure 35.

The authenticity of the frames flowing in the network is guaranteed at the layers lower.than the transport layer. Therefore this standard checks the authenticity of the frames by detecting the authenticity of FA link information (FA link header) only. Frame types depend on the TCD in the FA link header. The frame types defined by the FA link shall be shown in Table 9.

For the details of the FA link header, see 10.1.

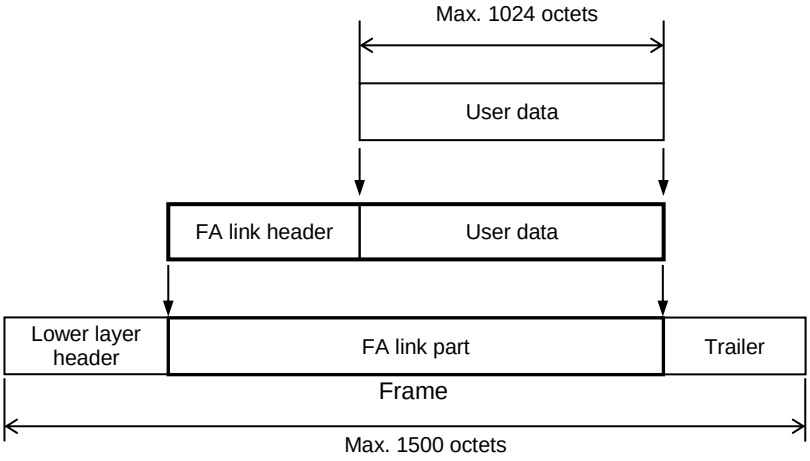


Figure - 35 Frame Structure

Table - 9 Fames Defined by FA Link

Name	Transaction code
(Reserved)	0 to 9999
Transparent message frame	10000 to 59999
(Reserved)	60000 to 64999
Token frame	65000
Cyclic frame	65001
Participation request frame	65002
Byte block read frame (request)	65003
Byte block write frame (request)	65004
Word block read frame (request)	65005
Word block write frame (request)	65006
Network parameter read frame (request)	65007
Network parameter write frame (request)	65008
Stop command frame (request)	65009
Operation command frame (request)	65010
Profile read frame (request)	65011
Trigger frame	65012
Log data read frame (request)	65013
Log data clear frame (request)	65014
Message return frame (request)	65015
Vendor specific message frame (request)	65016
Solicitation frame	65017
IO allocation setting frame (request)	65018
IO allocation read frame (request)	65019
Start frame of token retention time measurement (request)	65020
End frame of token retention time measurement (request)	65021
Start frame of measurement in general purpose communication data sender log (request)	65022
End frame of measurement in general purpose communication data sender log (request)	65023

Table 9 Fames Defined by FA Link (continued)

Name	Transaction code
Configuration parameter setting frame (request)	65024
Read frame of participating node management information parameter (request)	65025
Read frame of self node management information parameter (request)	65026
Self node setting information parameter read frame (request)	65027
Node reset frame (request)	65028
(Reserved) (for future extension)	65029 to 65202
Byte block read frame (response)	65203
Byte block write frame (response)	65204
Word block read frame (response)	65205
Word block write frame (response)	65206
Network parameter read frame (response)	65207
Network parameter write frame (response)	65208
Stop command frame (response)	65209
Operation command frame (response)	65210
Profile read frame (response)	65211
(Reserved)	65212
Log data read frame (response)	65213
Log data clear frame (response)	65214
Message return frame (response)	65215
Vendor specific message frame (response)	65216
(Reserved)	65217
IO allocation setting frame (response)	65218
IO allocation read frame (response)	65219
Start frame of token retention time measurement (response)	65220
End frame of token retention time measurement (response)	65221
Start frame of measurement in general purpose communication data sender log (response)	65222
End frame of measurement in general purpose communication data sender log (response)	65223
Configuration parameter setting frame (response)	65224
Read frame of participating node management information parameter (response)	65225
Read frame of self node management information parameter (response)	65226
Self node setting information parameter read frame (response)	65227
Node reset frame (response)	65228
(Reserved) (for future extension)	65229 to 65399
(Reserved)	65400 to 65535

8.3 Octet order

The frames defined by the FA link uses the rules for determining the order of octets as shown in Figure 36.

a) Octet order for header part

— The header of the token frame uses Big endian to order octets.

— The header of the cyclic frame uses Big endian to order octets.

— The header of the message frame uses Big endian to order octets.

— The participation request frame uses Big endian to order octets.

— The trigger frame uses Big endian to order octets.

— The ACK data uses Big endian to order octets.

— The header of the solicitation frame uses Big endian to order octets.

b) Octet order for data part

— The data part of the cyclic frame uses Little endian to order octets.

— The data part of the message frame uses Little endian to order octets.

— Note that the data (parameters) part of the profile read frame uses Big endian to order octets.

— The data part of the solicitation frame uses Little endian to order octets.

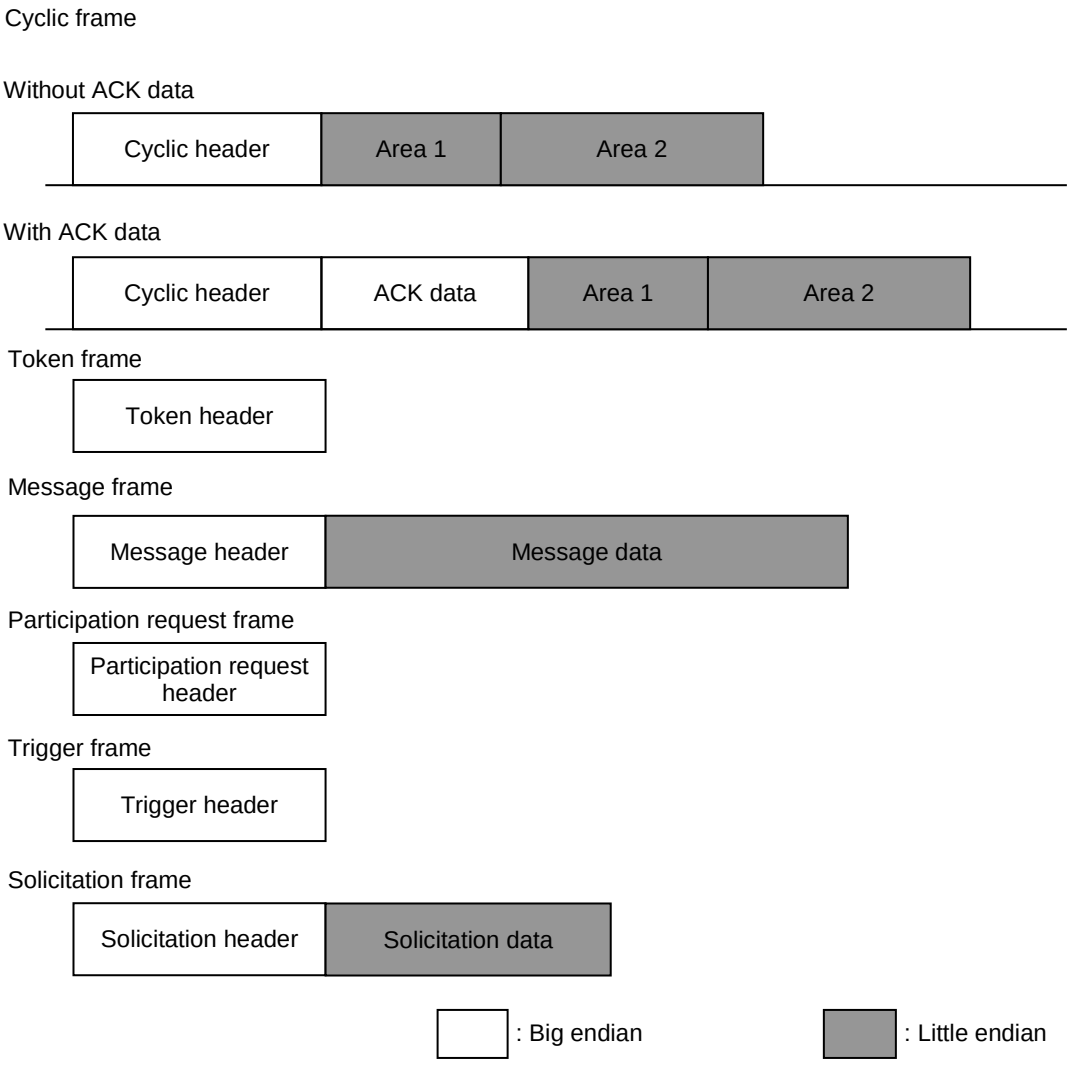


Figure - 36 Octet Order for FA Link Frame

8.4 General purpose command server

This is an interface function for setting and reading in various parameters from setting tools, and for requesting start and stop of load measuring functions.

For commands available for the general purpose command server, see Table 3.

For the details of the frames of the commands, see part 10.

8.4.1 Communication sequence for the general purpose command server and setting tools

The general purpose command server and setting tools use TCP and UDP. Their communication sequence is, as shown in Figure 37, basically such that a response shall occur following a request irrespective of which of these protocols is used.

This contains a watchdog function for detecting missing of data to allow the setting tool to resend the request for transmission.

Note 1: Whether or not to resend depends on the setting tool and vendor application. The protocol does not specify rules for the resending operation.

Note 2: Response time limit is determined by the setting tool.

Note 3: When the setting tool received a response from a node to a request after the response time limit, setting tool discards the response. A judgment of an excess of response time limit is made by the sequence number (SEQ) set to each command header part at the time of a request.

Note 4: The number of connections for each of TCP and UDP shall be 1 or larger respectively. The maximum number of connections depends on the vendor.

Note 5: The node's operation when the number of connections exceeds the limit is considered as vendor dependence.

Note 6: Nodes that do not support the commands from the general purpose command server shall return M_RLT (message result in the FA link header) = 2 command not supported within 1 second, or return nothing.

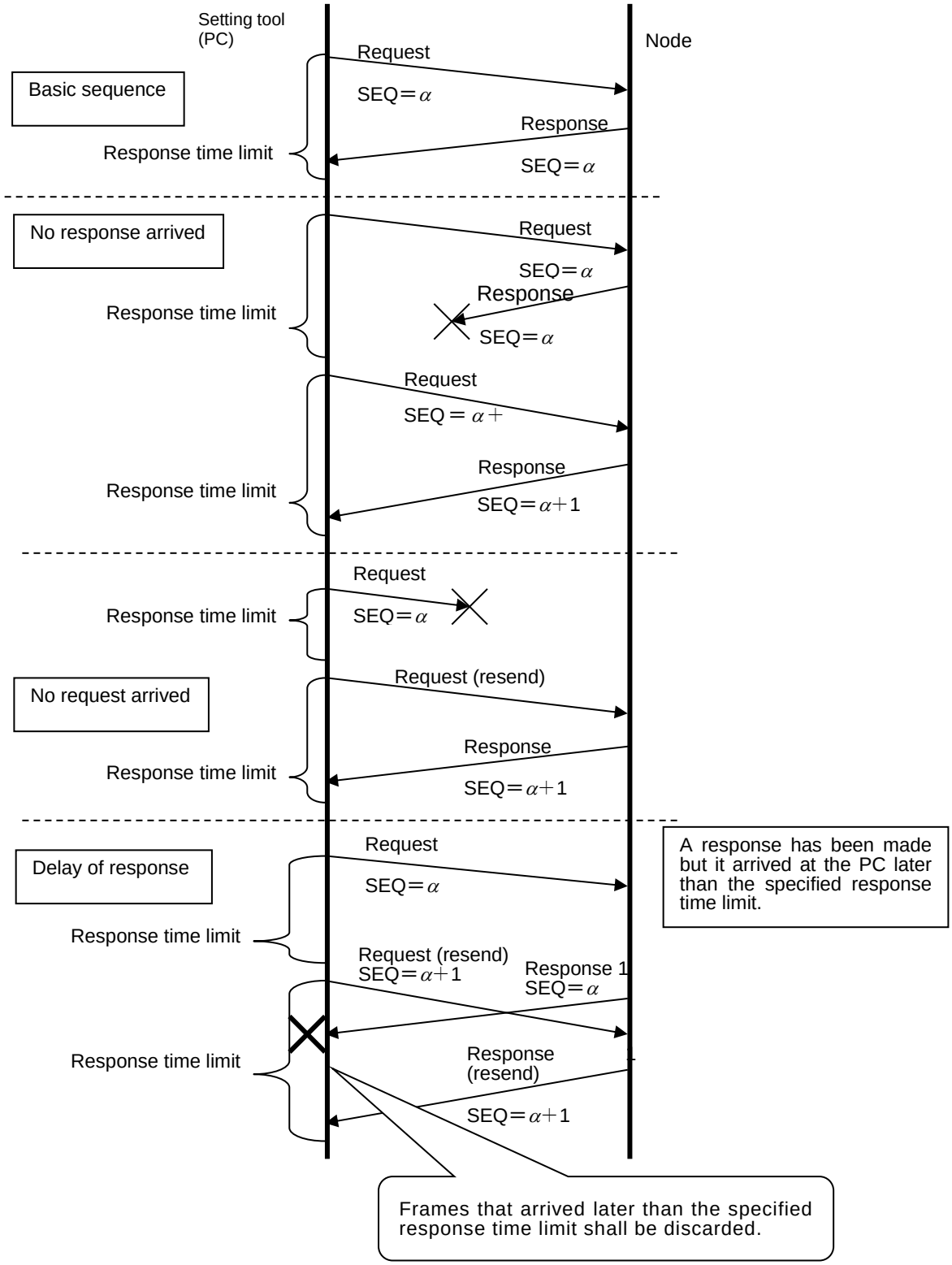
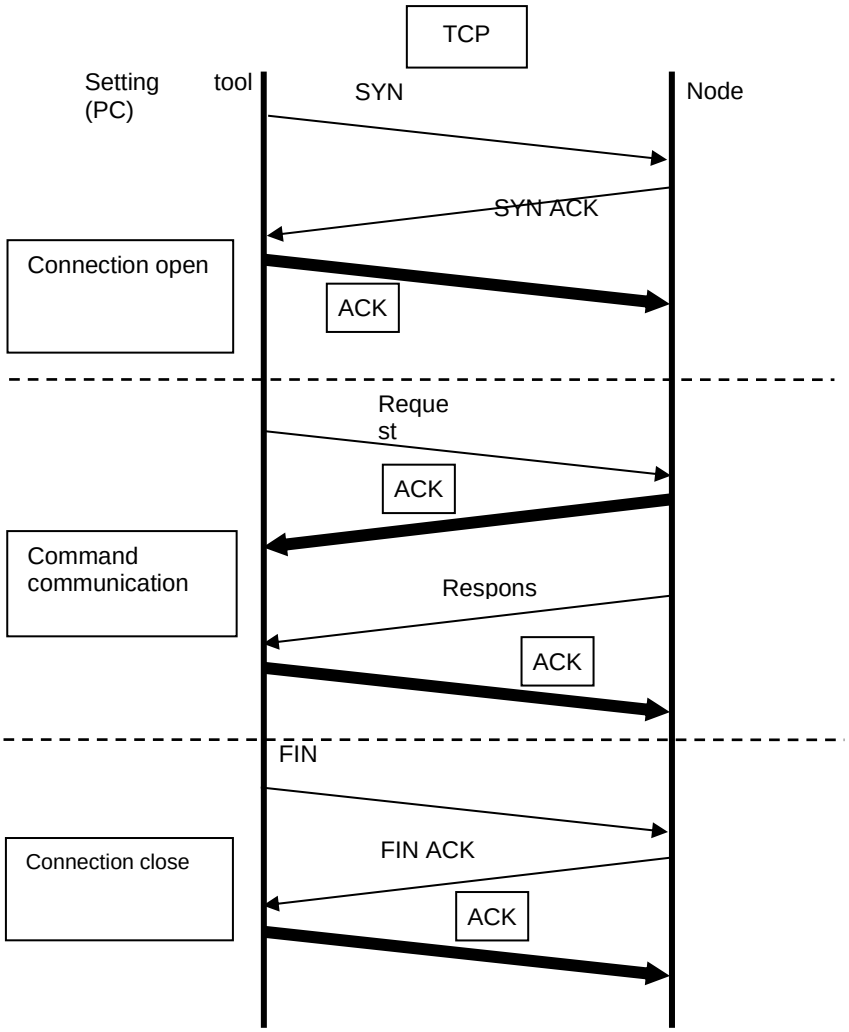


Figure 37 Basic Sequence of Response to Request

8.4.2 Communication sequence for TCP

8.4.2.1 Sequence for TCP

When using TCP, the setting tool uses the sequence shown in Figure 38.



Note: The frames written in bold are checked by TCP to determine whether they have arrived.

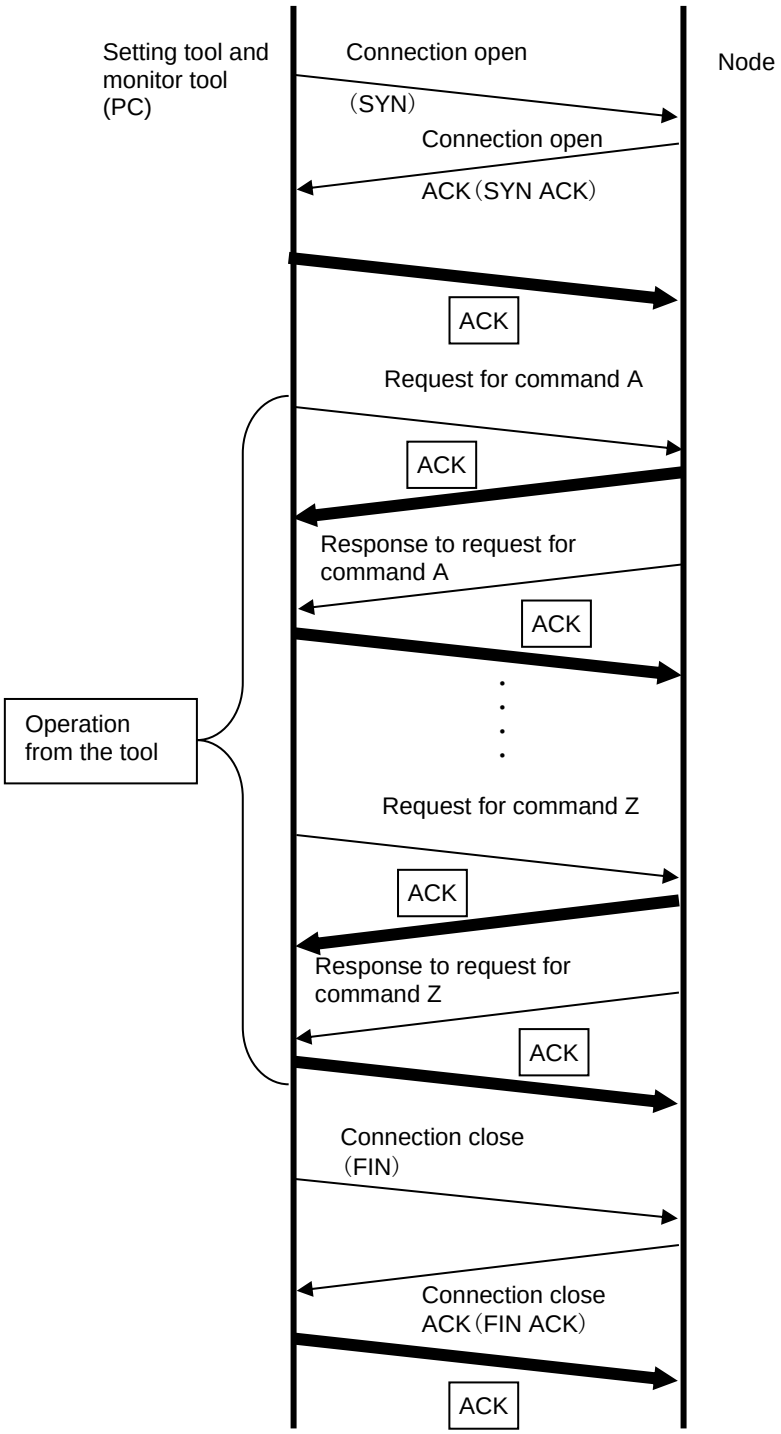
Figure 38 Sequence for TCP

8.4.2.2 Connection open/close sequence

When using TCP, the setting tool executes “connection open” before starting communication and executes “connection close” after the communication.

For any operation of the tool (e.g. operation of pressing a button associated with a request for executing multiple commands), “connection open” shall occur prior to the request for the first command and “connection close” shall occur after the final request.

The “connection open/close” sequence shall be shown in Figure 39.



Note: The frames written in bold are checked by TCP to determine whether they have arrived.

Figure 39 Connection Open/Close Sequence

8.4.3 Communication sequence for UDP

Unlike TCP, no “connection open” or “connection close” sequence is required for starting and ending communication. Under UDP, communication shall begin with transmission of a frame containing a request for command execution.

When using UDP, the general purpose command server operates in the unicast mode.

When using UDP, the setting tool uses the sequence shall be shown in Figure 40.

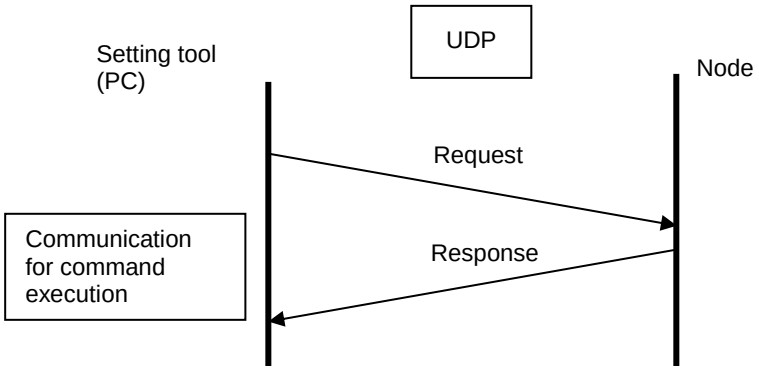


Figure 40 Sequence for UDP

8.4.4 Load measuring function

This function measures the following items. The information obtained by this function is available for analyzing errors (see Table 10). All of the measurements are stored in the log data area.

a) Missing of token

b) Token retention time

c) Refresh cycle time

d) Number of the events of receiving cyclic data

e) Number of errors in receiving message data

f) Source IP address for TCP general purpose communication to the self node, and the number of receiving events

Table 10 Error Events and Corresponding Measuring Items

Event 1)	Event 2)	Load measuring item
Disturbed token	Missing of token	Number of the events of discarding token
		Time when the event of discarding token was detected last
		Number of the events of reissuing token
		Time when reissuance of token occurred last
	Extension of token retention time	Number of the events of token retention timeout
		Time when the token retention timeout event occurred last
		Number of the events of token watchdog timeout
		Time when the token watchdog timeout event occurred last
		Maximum token retention time
		Minimum token retention time
		Detection time for the maximum token retention time
		Measuring time for token retention time
		Number of tokens while measuring the token retention time
	Extension of refresh cycle time	Detection time for the maximum refresh cycle
Missing cyclic data		Number of the events of receiving cyclic frame
		Number of errors in receiving cyclic frame
		Detection time for error in receiving cyclic frame
Missing message data		Number of errors in receiving message frame
		Detection time for error in receiving message frame
Higher communication frequency of other nodes than FL-net	Communication to self node	Measuring time for general purpose communication data sender log
		General purpose communication data source IP address (up to the first 10 IP addresses)
		Number of the events of receiving general purpose communication data (up to the first 10 IP addresses)

8.4.4.1 Token retention time measuring function

This is to measure token retention time to determine whether or not communication frequency of nodes other than those of FL-net is higher (see Figure 41).

It measures the minimum and maximum values of the time for which a node holds a token, and the elapsed time until the maximum value is measured from the start of measurement, and saves those values in a logging area.(for measuring items, see Table 12).

The measurement of token retention time starts when receiving a command for starting measurement from the setting tool and ends when receiving a command for ending the measurement (see Figure 42).

The measuring result is appended to the response data for the ending command.

If another start command is received during measurement, the current measurement shall be stopped and new measurement shall be started. If an ending command is received again after the measurement is ended, the latest measurement shall be appended to the response for the command. Note that setting tools are not managed for issuance of commands, which means that the node can accept an ending command or another start command from other than the setting tool that has now issued the start command for the current measurement session.

In the event that power interruption or resetting of the node occurs during measurement, the measurement data shall be discarded.

Measurement data can be compared with the preliminarily measured data of token retention time in normal operation to know the difference in the intensity of load to determine whether an error has occurred.

The transaction codes for measurement starting and ending commands shallbe shown in Table 11.

Table 11 Transaction Codes for Measuring Token Retention Time

Name	Transaction code
Start frame of token retention time measurement (request)	65 020
End frame of token retention time measurement (request)	65 021
Start frame of token retention time measurement (response)	65 220
End frame of token retention time measurement (response)	65 221
Note: use port no. ”55 004” for transmission and receiving operation	

Table 12 List of Measuring Items for Token Retention Time

Measurement item	Unit	Added by Ver. 3.00 [a)]
Number of the events of discarding token	Time [b)]	―
Time when the event of discarding token was detected last	sec [c)]	○
Number of the events of reissuing token	Time [b)]	―
Time when reissuance of token occurred last	sec [c)]	○
Number of the events of token retention timeout	Time [b)]	―
Time when the token retention timeout event occurred last	sec [c)]	○
Number of the events of token watchdog timeout	Time [b)]	―
Time when the token watchdog timeout event occurred last	sec [c)]	○
Maximum token retention time	µSec (1/1 000 000 sec)	○
Minimum token retention time	µSec (1/1 000 000 sec)	○
Detection time for the maximum token retention time	sec [c)]	○
Measuring time for token retention time	sec [c)]	○
Number of tokens while measuring the token retention time	Time [b)]	○
Detection time for the maximum refresh cycle	sec [c)]	○
Number of the events of receiving cyclic frame	Time [b)]	―
Number of errors in receiving cyclic frame	Time [b)]	―
Detection time for error in receiving cyclic frame	sec [c)]	○
Number of errors in receiving message frame	Time [b)]	―
Detection time for error in receiving message frame	sec [c)]	○
Note [a)] O: New items since Ver. 3.00― : Log data items since Ver. 2.00 [b)] The number is as counted by a free-run counter. [c)] Elapse since start of measurement (sec.)		

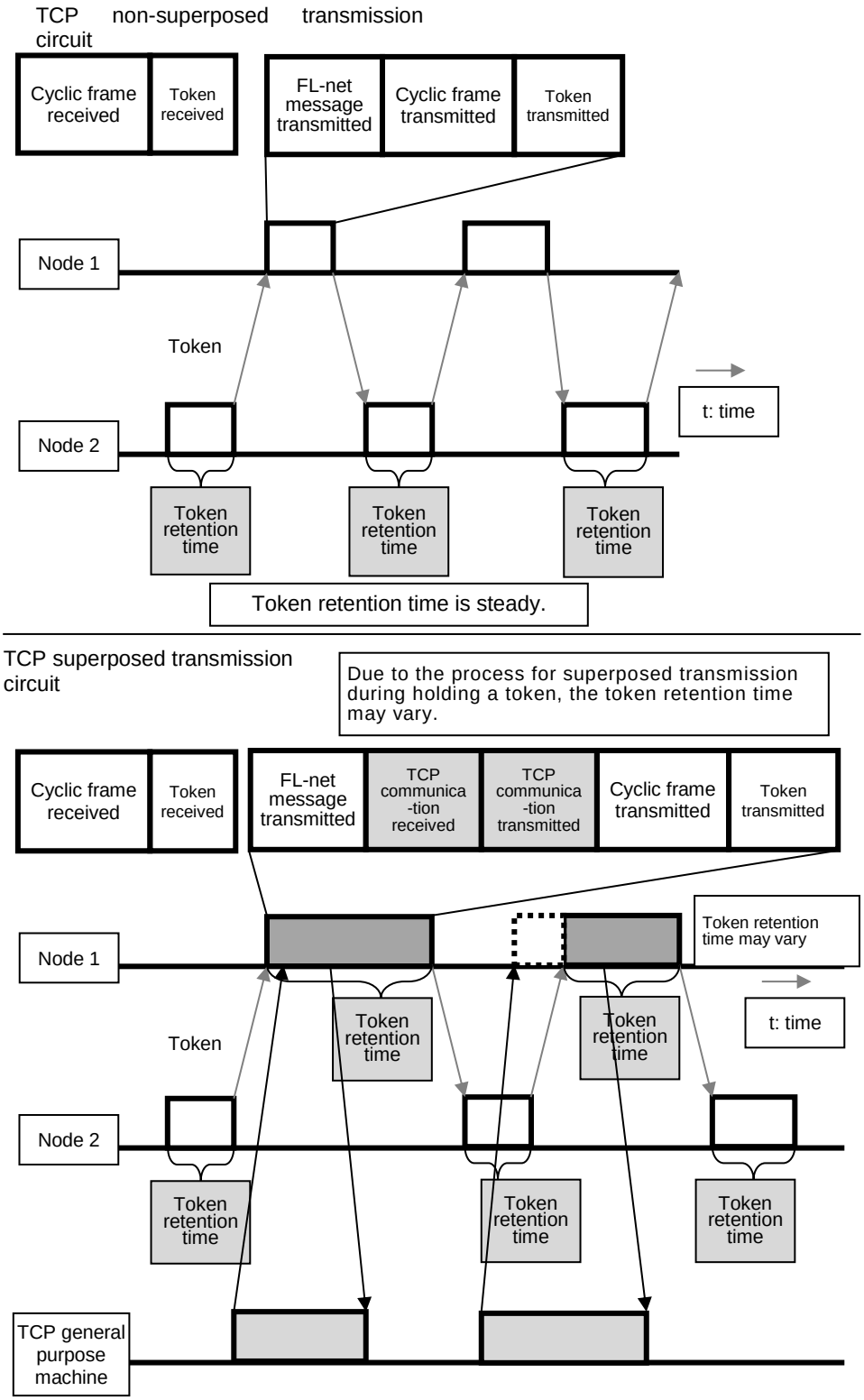


Figure 41 Load Measurement by Token Retention Time

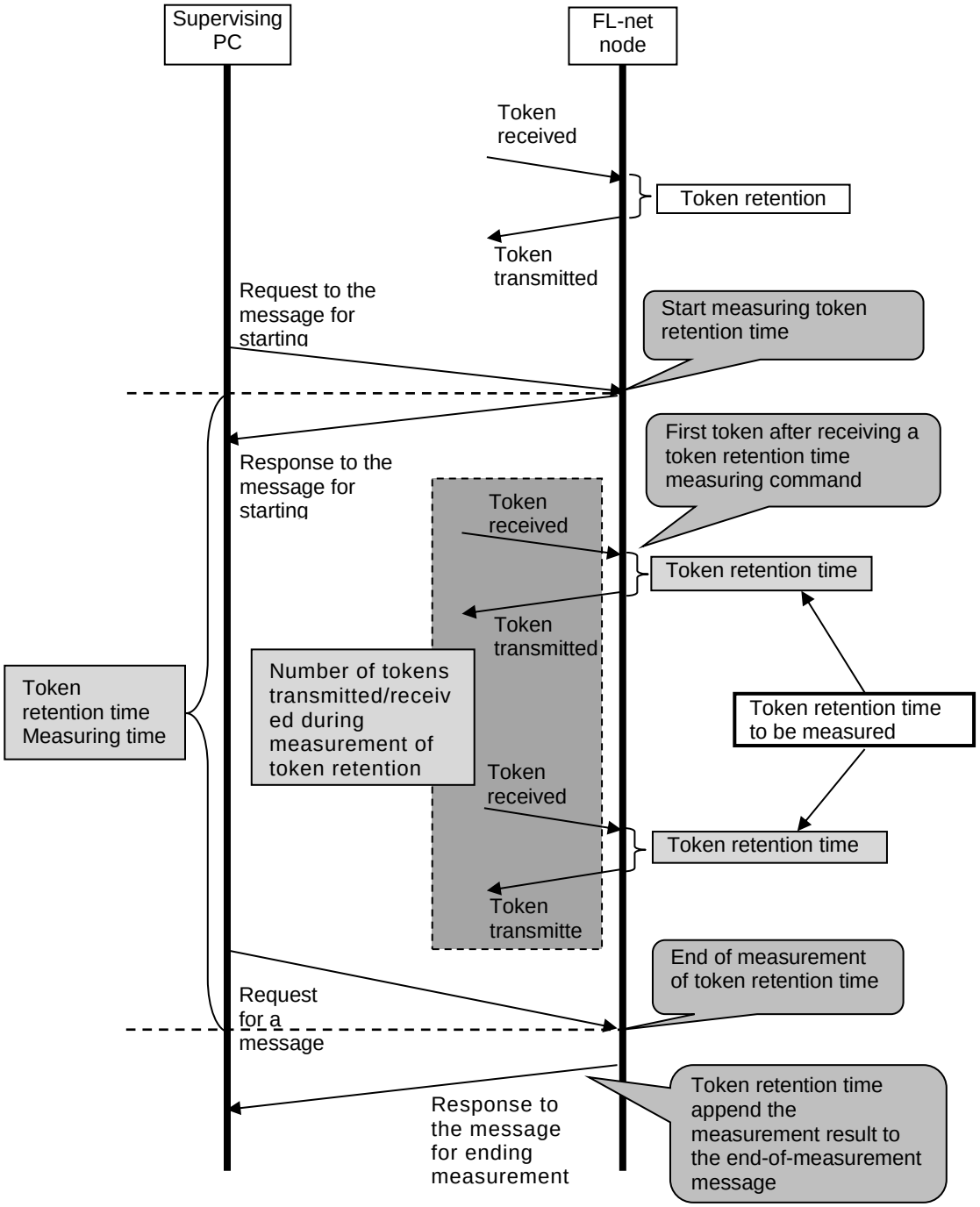


Figure 42 Approximate View of Token Retention Time Measuring Sequence

8.4.4.2 General purpose communication data sender logging function

This measures how many times the node receives the general purpose communication data frames from each of the source IP addresses. However it does not count the event exceeding the first 10 events for each source (for measuring items, see Table 14).

This function starts with receiving a measurement start command from the setting tool and continues until a measurement ending command is received (see Table 13).

The measurement result shall be appended to the response for the measurement ending command.

If another start command is received during measurement, the current measurement shall be

discarded and new measurement shall be started. If an ending command is received again after the measurement is ended, the latest measurement shall be appended to the response data. Note that setting tools are not managed for issuance of commands, which means that the node can accept an ending command or another start command from other than the setting tool that has now issued the start command for the current measurement session.

In the event that power interruption or resetting of the node occurs during measurement, the measurement data shall be discarded.

The transaction codes for measurement starting and ending commands shall be shown in Table 13.

Table 13 Transaction Codes for General Purpose Data Sender Log

Name	Transaction code
Frame for starting measurement for general purpose data sender log (request)	65 022
Frame for ending measurement for general purpose data sender log (request)	65 023
Frame for starting measurement for general purpose data sender log (response)	65 222
Frame for ending measurement for general purpose data sender log (response)	65 223

Table 14 List of Measuring Items for General Purpose Data Sender Log

Measuring item	Unit	Added by Ver. 3.00 [a)]
Measuring time for general purpose communication data sender log	Sec [b)]	○
IP 1	—	○
IP 1 receive counter	Time	○
IP 2	—	○
IP 2 receive counter	Time	○
IP 3	—	○
IP 3 receive counter	Time	○
IP 4	—	○
IP 4 receive counter	Time	○
IP 5	—	○
IP 5 receive counter	Time	○
IP 6	—	○
IP 6 receive counter	Time	○
IP 7	—	○
IP 7 receive counter	Time	○
IP 8	—	○
IP 8 receive counter	Time	○
IP 9	—	○
IP 9 receive counter	Time	○
IP 10	—	○
IP 10 receive counter	Time	○

Note 1: Each measuring item consists of a 32-bit member.
Note 2: The sender log measuring time counter and receive counter are reset to 0 when a request for starting measurement is received, and then start.
Note 3: Layers used for acquisition depend on the vendor.
Note [a)] O: New items since Ver. 3.00
—: Log data items since Ver. 2.00
[b)] Elapse since start of measurement (sec.)

An example of operation of the general purpose data sender log function shall be shown in Figure 43.

If the self node receives general purpose communication data from 11 IP addresses (IP1 through IP11) during measurement, the data from addresses up to 10 IP addresses (IP1 through IP10) are stored in the log while the data from IP11 is discarded.

The receive counter for an IP address increments by 1 every time when receiving data from the IP address (no receive counters for 11th or later IP addresses exist).

These receive counters do not support overflow of the count: after it reaches 16#FFFF_FFFF, it is cleared to 0.

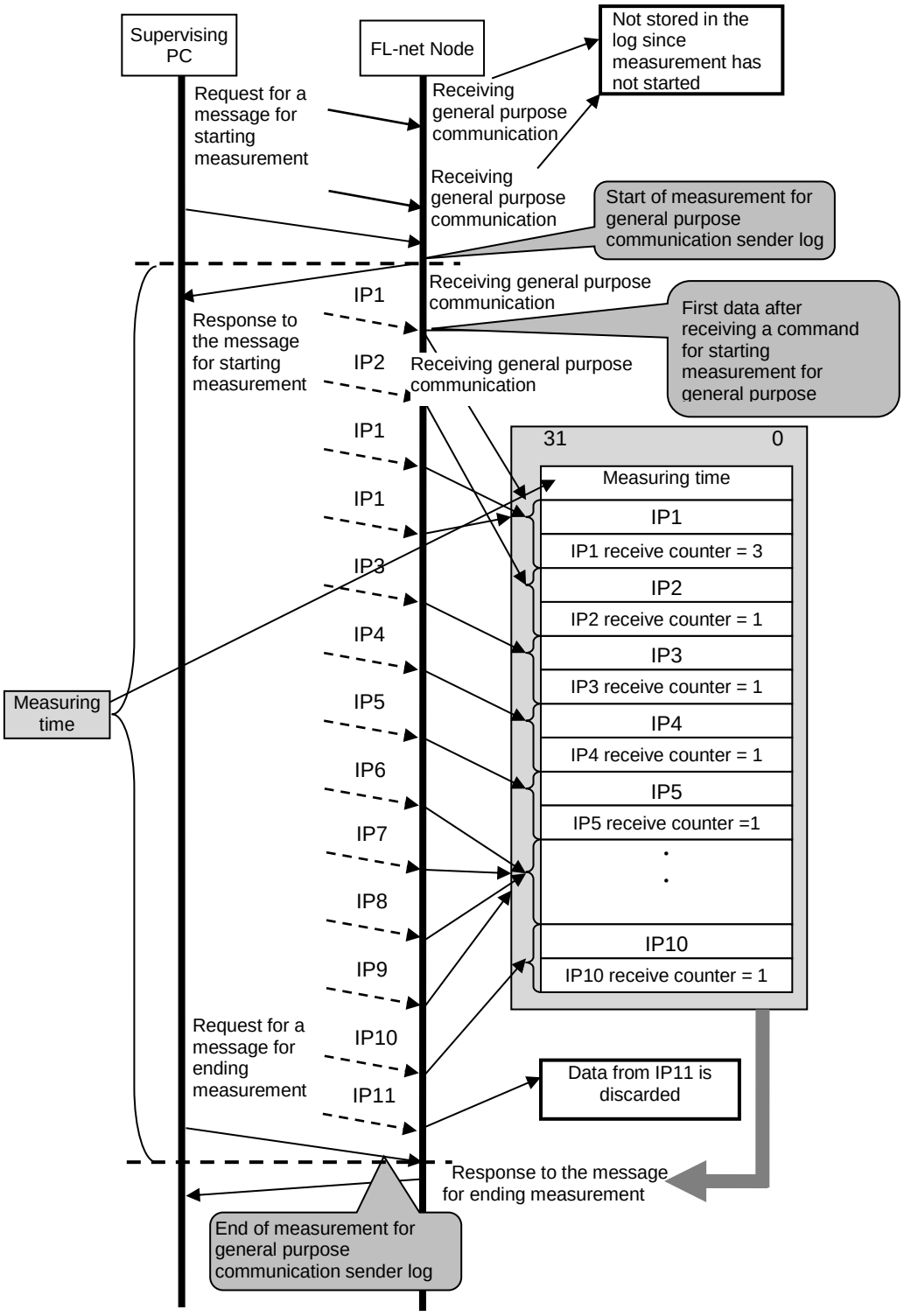


Figure 43 Operation of General Purpose Communication Data Sender Logging Function (Example)

8.4.4.3 Result notification

Load measurement results stored in the log data area can be retrieved by using an existing log data read message. Load measurement results are stored also in the log area.

Of the load measurement results stored in the log, those returned as response data are denoted by a) and b) in Table 15. Items a) concern token retention time. Items b) concern general purpose communication data sender log.

The log data area that the load measuring function uses shall be shown in Table 15.

Table 15 Log Data Area Used by the Load Measuring Function

Offset (decimal) (octet)	Name	Description	Implementation
76	Number of the events of receiving cyclic frame [a)]	Data about frame type	Optional
96	Number of errors in receiving cyclic frame [a)]	Data about cyclic transmission	Mandatory
116	Detection time for error in receiving cyclic frame[a)]	Time when cyclic transmission/reception error is detected	Optional
152	Detection time for the maximum refresh cycle [a)]	Time when maximum value during refresh cycle is detected	Optional
180	Detection time for error in receiving message frame[a)]	Time when the event of failing to transmit/receive message frame is detected	Optional
244	Number of the events of discarding token [a)]	Token related data	Mandatory
248	Number of the events of reissuing token [a)]	Token related data	Mandatory
252	Time when the event of discarding token was detected last [a)]	Time when the number of the events of discarding token is detected	Optional
256	Time when reissuance of token occurred last [a)]	Time when the number of the events of reissuing token is detected	Optional
260	Time when the token retention timeout event occurred last [a)]	Time when the token retention timeout counter increments	Optional
264	Number of the events of token retention timeout [a)]	Token related data	Optional
268	Number of the events of token watchdog timeout [a)]	Token related data	Optional
272	Time when the token watchdog timeout event occurred last [a)]	Time when the token watchdog timeout counter increments	Optional
276	Maximum token retention time [a)]	Maximum token retention time measured during the period that starts with a measurement start command and ends with a measurement end command	Optional
280	Minimum token retention time [a)]	Minimum token retention time measured during the period that starts with a measurement start command and ends with a measurement end command	Optional
284	Detection time for the maximum token retention time [a)]	Time when the maximum token retention time is updated	Optional
312	Measuring time for token retention time [a)]	Period for measuring token retention time that starts with a measurement start command and ends with a measurement end command	Optional
316	Number of tokens while measuring the token retention time [a)]	Number of tokens received during measurement of token retention time	Optional
332	Measuring time for general purpose communication data sender log[b)]	Period for measuring general purpose communication data sender log that starts with a measurement start command and ends with a measurement end command	Optional
368	IP 1[b)]	IP address of General purpose communication data sender log 1	Optional
372	IP 1 receive counter [b)]	Number of the events of receiving data from the IP address of General purpose communication data sender log 1	Optional
376	IP 2 [b)]	IP address of General purpose communication data sender log 2	Optional
380	IP 2 receive counter [b)]	Number of the events of receiving data from the IP address of General purpose communication data sender log 2	Optional
384	IP 3 [b)]	IP address of General purpose communication data sender log 3	Optional

Table 15 Log Data Area Used by the Load Measuring Function (continued)

Offset (decimal) (octet)	Name	Description	Implementation
388	IP 3 receive counter [b)]	Number of the events of receiving data from the IP address of General purpose communication data sender log 3	Optional
392	IP 4 [b)]	IP address of General purpose communication data sender log 4	Optional
396	IP 4 receive counter [b)]	Number of the events of receiving data from the IP address of General purpose communication data sender log 4	Optional
400	IP 5 [b)]	IP address of General purpose communication data sender log 5	Optional
404	IP 5 receive counter [b)]	Number of the events of receiving data from the IP address of General purpose communication data sender log 5	Optional
408	IP 6 [b)]	IP address of General purpose communication data sender log 6	Optional
412	IP 6 receive counter [b)]	Number of the events of receiving data from the IP address of General purpose communication data sender log 6	Optional
416	IP 7 [b)]	IP address of General purpose communication data sender log 7	Optional
420	IP 7 receive counter [b)]	Number of the events of receiving data from the IP address of General purpose communication data sender log 7	Optional
424	IP 8 [b)]	IP address of General purpose communication data sender log 8	Optional
428	IP 8 receive counter [b)]	Number of the events of receiving data from the IP address of General purpose communication data sender log 8	Optional
432	IP 9 [b)]	IP address of General purpose communication data sender log 9	Optional
436	IP 9 receive counter [b)]	Number of the events of receiving data from the IP address of General purpose communication data sender log 9	Optional
440	IP 10 [b)]	IP address of General purpose communication data sender log 10	Optional
444	IP 10 receive counter [b)]	Number of the events of receiving data from the IP address of General purpose communication data sender log 10	Optional

Note: The data is stored using Little endian to order octets.
Note [a)] Measurement items for token retention time
[b)] Measurement items for general purpose communication data sender log

8.4.4.4 Behavior when receiving a measurement start command during measurement

If another start command is received during measurement, the current measurement shall be discarded and new measurement shall be started. If an ending command is received after the measurement is ended, the latest measurement shall be appended to the response for the command. Note that setting tools are not managed for issuance of commands, which means that the node can accept an ending command or another start command from other than the setting tool that has now issued the start command for the current measurement session.

Figure 44 shows how measurement start commands are handled while measurement is already in operation.

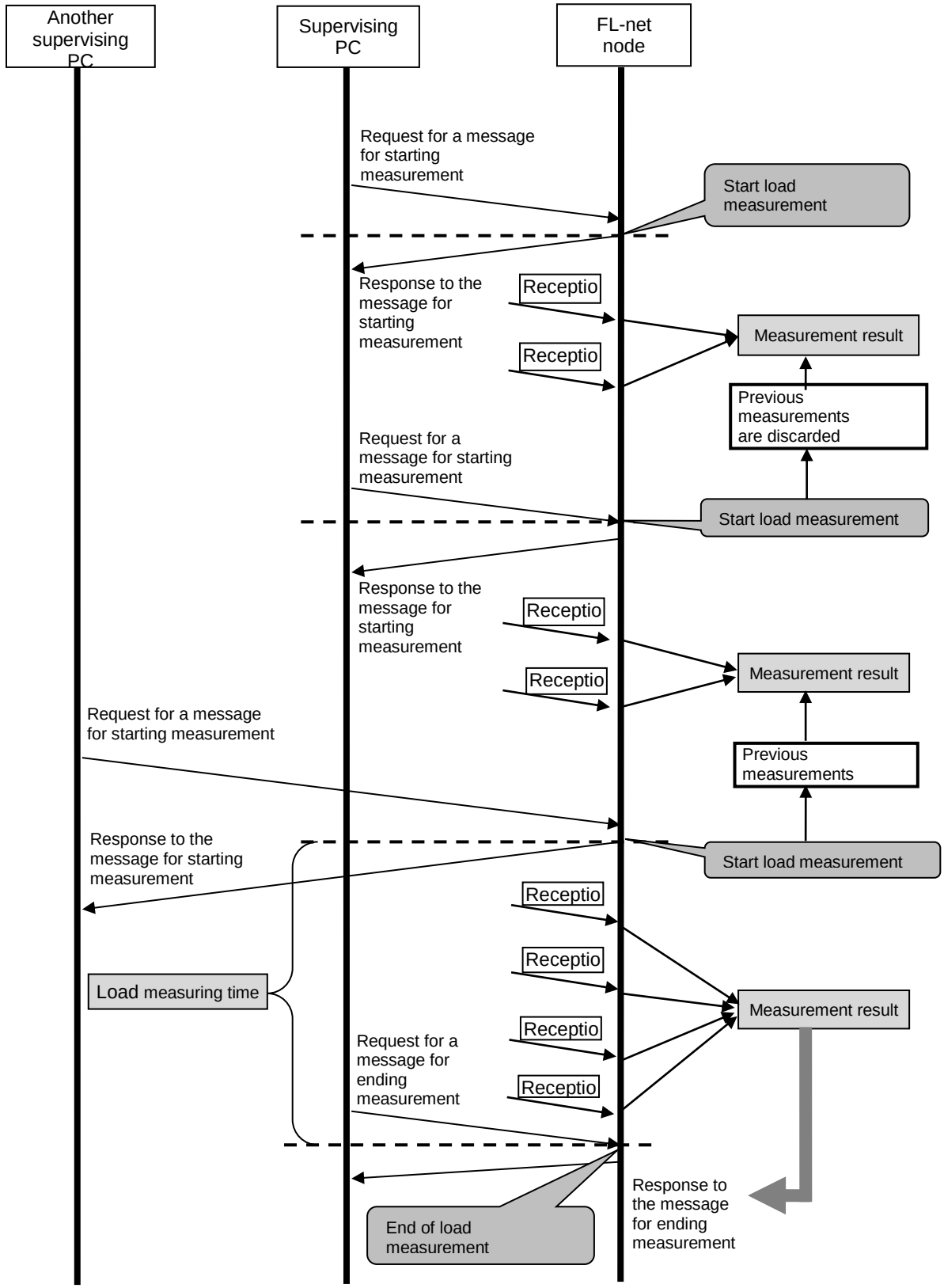


Figure 44 Handling of Measurement Start Command that Occurs during Measurement

8.4.4.5 Use of Load Measurement Result

Given that the system configuration shown in Figure 45 suffers fluctuation of the refresh frequency when the PC 2 transmits general purpose communication data to Node 1 at higher intervals, which causes Node 1 to sometimes disconnect from the system, the trouble can be analyzed using the load measuring function, as shown in Figure 45.

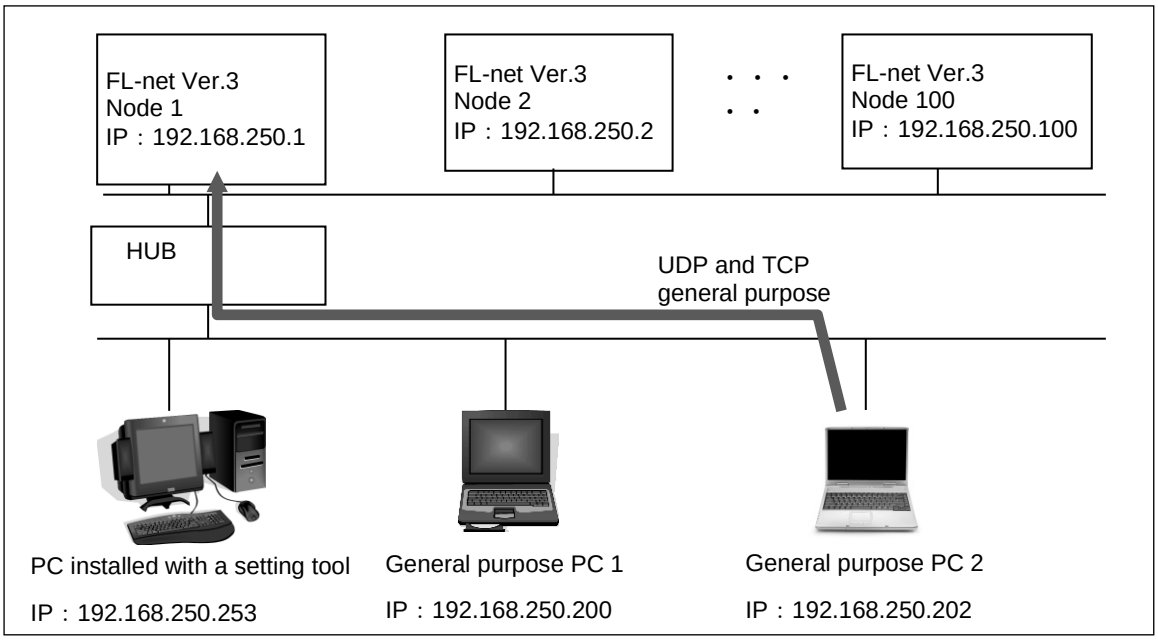


Figure 45 System Configuration for Measuring Token Retention Time

For communication with Node 1, perform measurement for token retention time and general purpose communication data sender log to obtain results as shown in Tables 16 and 17. Based on the result, the trouble can be analyzed as follows.

a) Measurements of token retention time indicate (see Table 16)

— Token retention timeout events have occurred frequently, which suggests that the process cannot be completed within the token retention time.

— Based on the fact that the latest token retention timeout event occurred very recently, we can suppose that the timeout events have occurred frequently.

— The large difference between the maximum and minimum token retention time is an evidence of larger network load.

b) Measurements for general purpose communication data sender log indicate (see Table 17)

— The PC 2 (IP1: 192.168.250.202) initiated communication many times, which suggests that general purpose communication has occurred very frequently.

The abovementioned findings indicate that the very frequent general purpose communication from PC 2 to Node 1 has caused Node 1 to undergo large loads.

Possible measures for mitigating this problem include reducing the network load and changing the relevant settings so that such large loads cannot be considered as an error.

— Measure 1: reduce the frequency of communication from PC 2 to Node 1.

— Measure 2: review the token watchdog time (TW) from a system-wide viewpoint and adjust it according to the measured load condition.

Table 16 Measurements of Token Retention Time for Node 1

Measuring item	Measurement	Note
Number of the events of discarding token	0 times	Not occurred
Time when the event of discarding token was detected last	0 Sec	
Number of the events of reissuing token	0 times	Not occurred
Time when reissuance of token occurred last	0 Sec	
Number of the events of token retention timeout	300 times	Token retention timeout events occur very frequently
Time when the token retention timeout event occurred last	595 Sec	
Number of the events of token watchdog timeout	0 times	Not occurred
Time when the token watchdog timeout event occurred last	0 Sec	
Maximum token retention time	250 000 μSec	The difference between the maximum and minimum values is large
Minimum token retention time	1 000 μSec	
Detection time for the maximum token retention time	30 Sec	
Measuring time for token retention time	600 Sec	Measured for 10 minutes
Number of tokens while measuring the token retention time	1 234 567 times	Number of tokens during measurement
Detection time for the maximum refresh cycle	80 Sec	–
Number of the events of receiving cyclic frame	123 400 times	–
Number of errors in receiving cyclic frame	0 times	Not occurred
Detection time for error in receiving cyclic frame	0 Sec	
Number of errors in receiving message frame	0 times	Not occurred
Detection time for error in receiving message frame	0 Sec	

Table 17 Measurements of General Purpose Communication Data Sender Log for Node 1

Measuring item	Measurement	Note
Measuring time for general purpose communication data sender log	600 Sec	Measured for 10 minutes
IP 1	192.168.250.202	PC 2 initiated communication very frequently
IP 1 receive counter	100000 times	
IP 2	192.168.250.200	PC 1 initiated communication infrequently
IP 2 receive counter	3 times	
IP 3	192.168.250.253	The setting tool initiated communication infrequently
IP 3 receive counter	10 times	
IP 4	0.0.0.0	–
IP 41 receive counter	0 times	
IP 5	0.0.0.0	–
IP 5 receive counter	0 times	
IP 6	0.0.0.0	–
IP 6 receive counter	0 times	
IP 7	0.0.0.0	–
IP 7 receive counter	0 times	
IP 8	0.0.0.0	–
IP 8 receive counter	0 times	
IP 9	0.0.0.0	–
IP 9 receive counter	0 times	
IP 10	0.0.0.0	–
IP 10 receive counter	0 times	

8.5 System Configuration of Device Level Network Function

8.5.1 System Configuration of Fixed Setting Mode

The system configuration of the fixed setting mode shall be as follows.

a) Basic Configuration of the Fixed Setting Mode Includes one master and N slaves as shown in Figure 46.

The master shall have the node number 100[2)], the self area of the common memory from 16#0100 to 16#01FF. The slaves shall have the node numbers from 1 to 64, the self area of the common memory from 16#0000 to 16#00FF. The common memory shall be allocated to the fixed address by the node number as self areas of the master node and slave nodes. The slaves write input data to their own self areas and read the output data from the master's self area corresponding to the slave node number. The input output area allocated to the common memory shall be fixed to 4 words for inputs and 4 words for output. However, the valid input and output data shall be the actual number of IO of slaves.

Note [2)] The master node number, which is 100 by default in the fixed setting mode, can be changed in the range from 65 to 249, and is notified to the slave using the status area so that the slaves can check the master number of the self node. Even when the master node number is changed, the self area of the common memory in the master shall be from 16#0100 to 16#01FF.

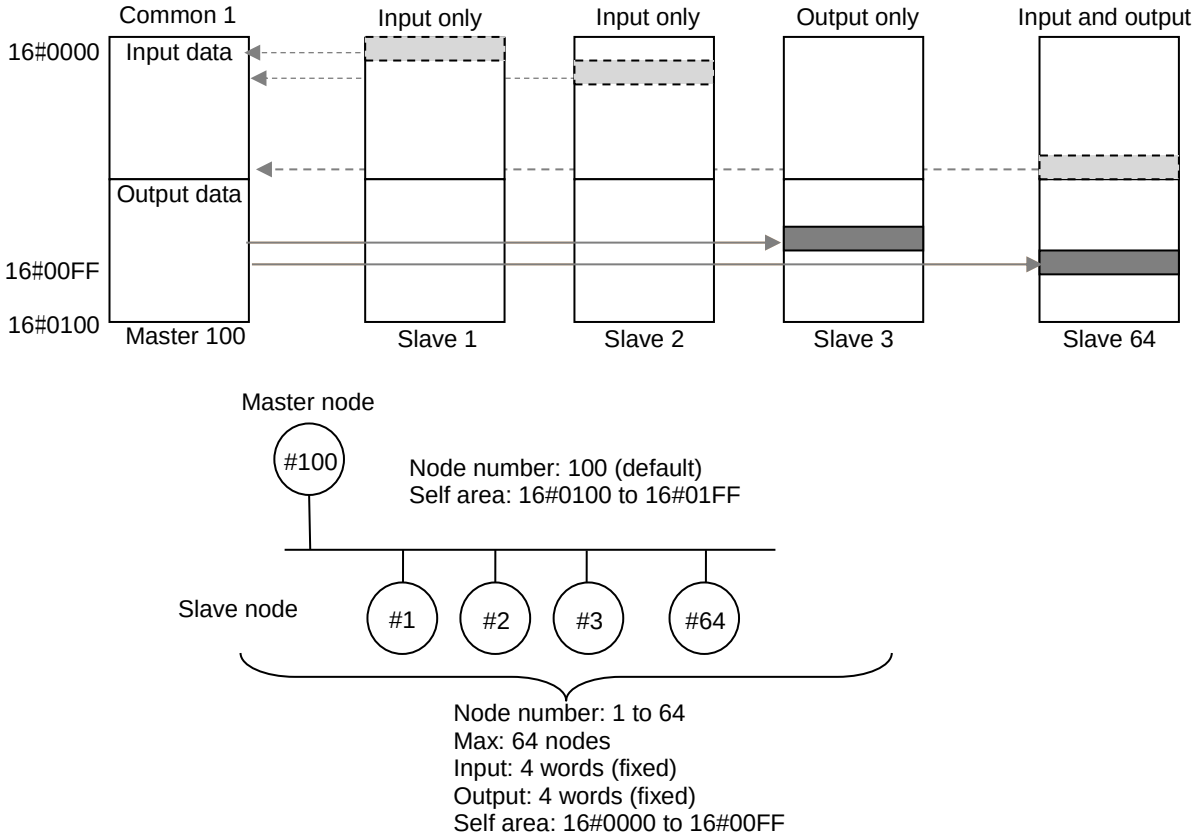


Figure 46 - System configuration of the fixed setting mode

b) IO Area Extension Configuration in the Fixed Setting Mode

Slaves have an IO area extension function to define areas over 4 words and settings of the status area can specify whether to use or not use the subsequent areas.

This setting allows, as shown in Figure 47, the slave node to use two or more areas in order to exchange the IO data with the master. Although the IO area can be extended within the range of the slave area, the node in the duplicate area cannot be used.

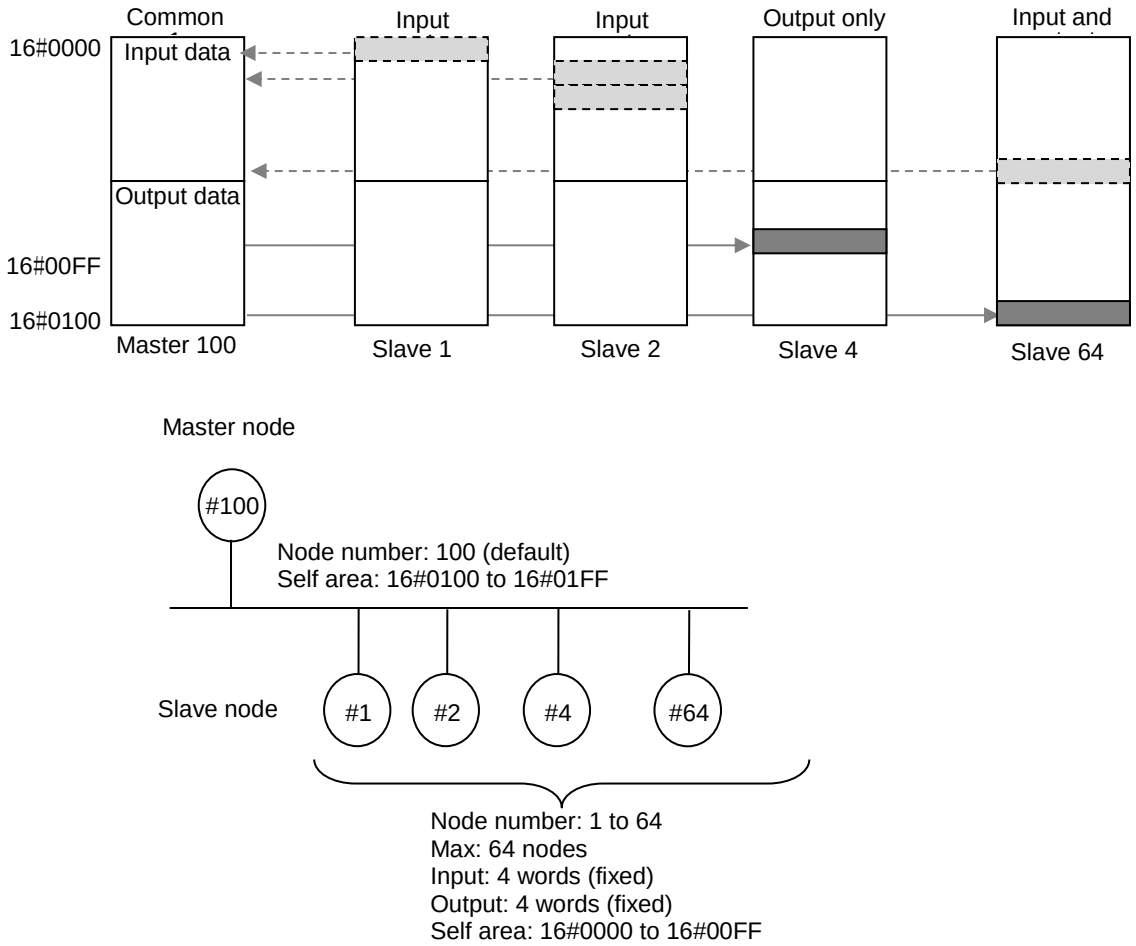


Figure 47 – IO Area Extension Configuration in the Fixed Setting Mode

8.5.2 The System Configuration of the Flexible Setting Mode

a) Basic Configuration of the Flexible Setting Mode

Consists of one master and N slaves.

As shown in Figure 48, the master shall be arbitrarily able to set the node numbers from 1 to 249 and the self area of the common memory from the common memory area ,1, 2. In the same manner, the slave can arbitrarily set the node numbers from 1 to 249 and the self area of the common memory from the common memory area ,1, 2. The slaves write input data to their own self areas and read the output data from the master's self area set by solicitation frames.

The size of the input output area shall be able to be set in words according to the actual number of the IOs.

The common memory settings of the slaves controlled by the master are distributed to the slaves using the solicitation frames.

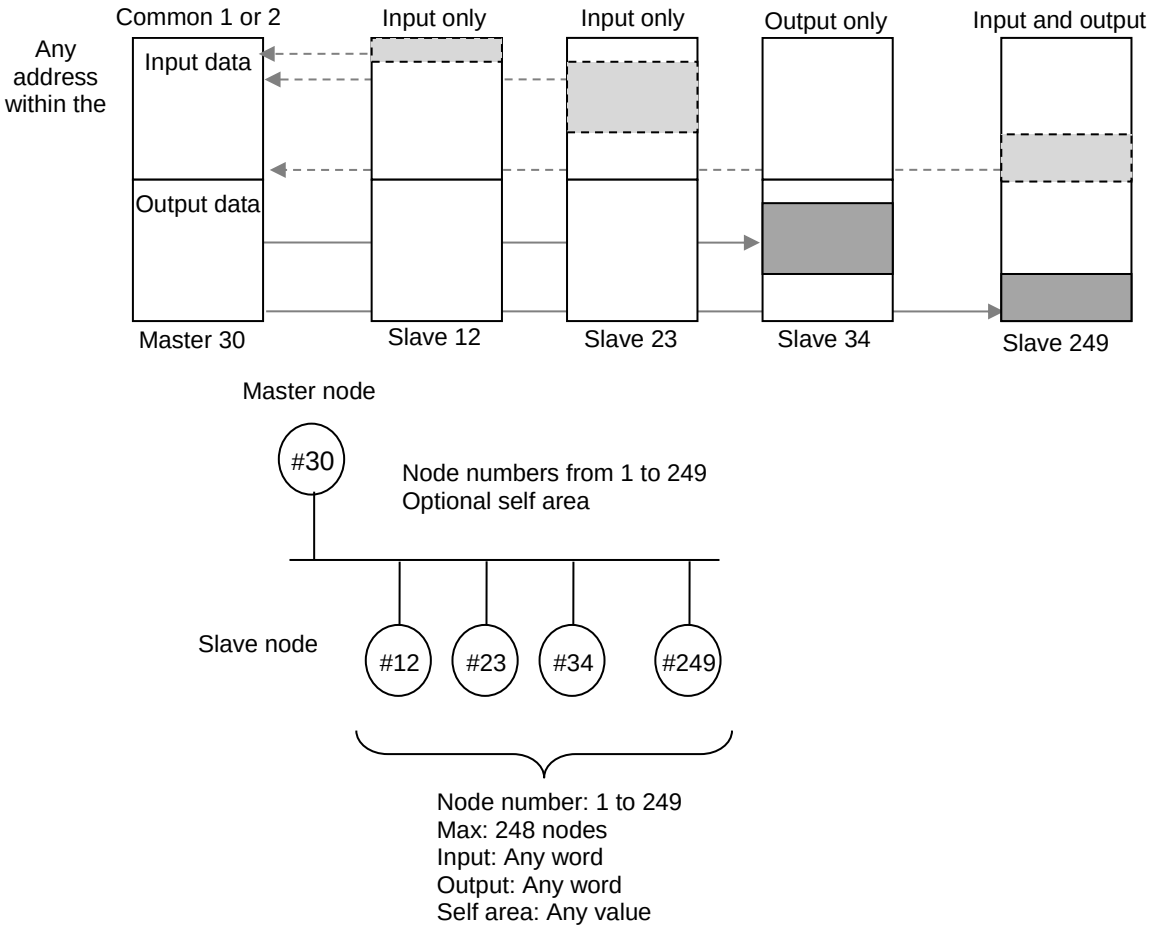


Figure 48 - Basic configuration of the flexible setting mode

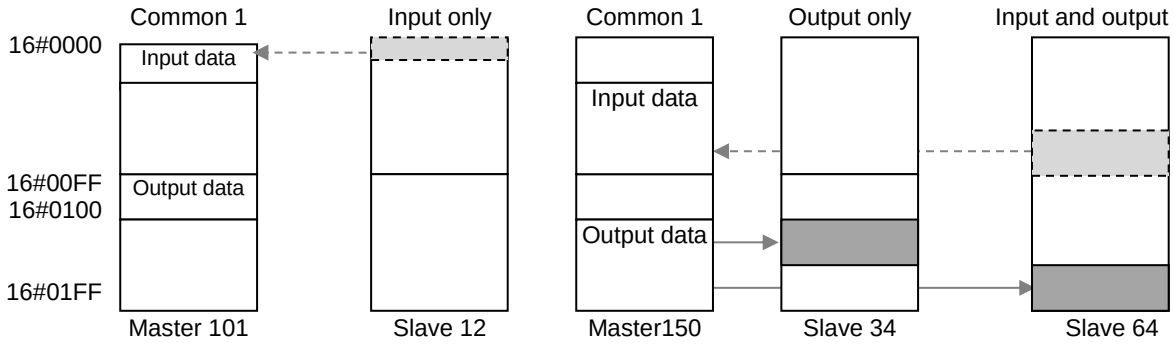
8.5.3 Multi Master

A multi master is a function to implement two or more configurations including one master node and several slave nodes on the FL-net network as shown in Figure. 49, and can operate in both the fixed setting mode and the flexible setting mode.

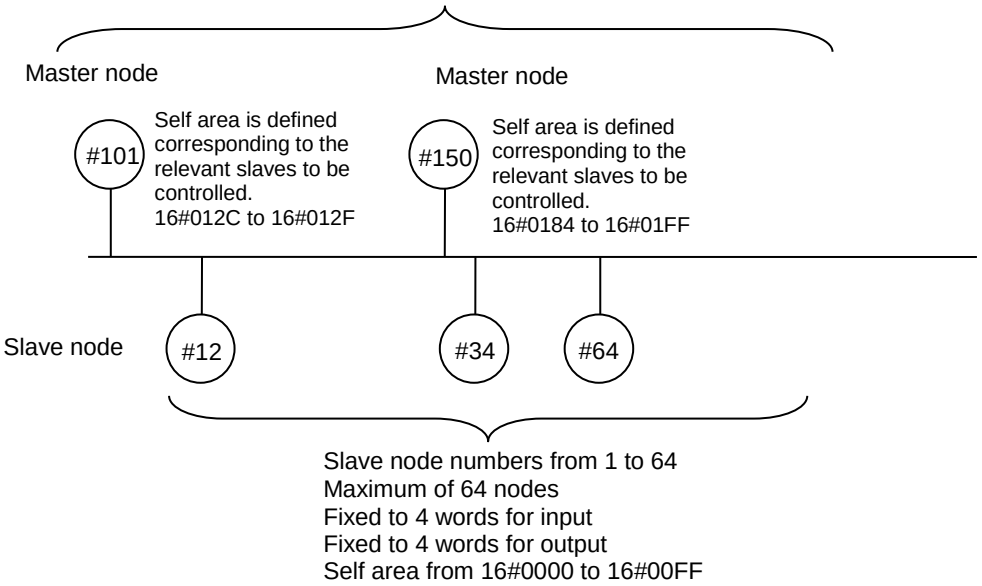
a) Multi master in the fixed setting mode

A multi master system can be built by changing the master node number from default setting (100) and allocating the master self areas divided by several master nodes.

Since the input output areas of the slaves are fixed, each master allocates a corresponding slave to their own output area. Note that the area must be continuous.



Master node number: 65 to 249
Slave node number: 1 to 64, max: 64 nodes
Self area: 16#0100 to 16#01FF (configured by Master)
Self area size (configured by Master)
Slave nodes to be controlled (configured by Master)



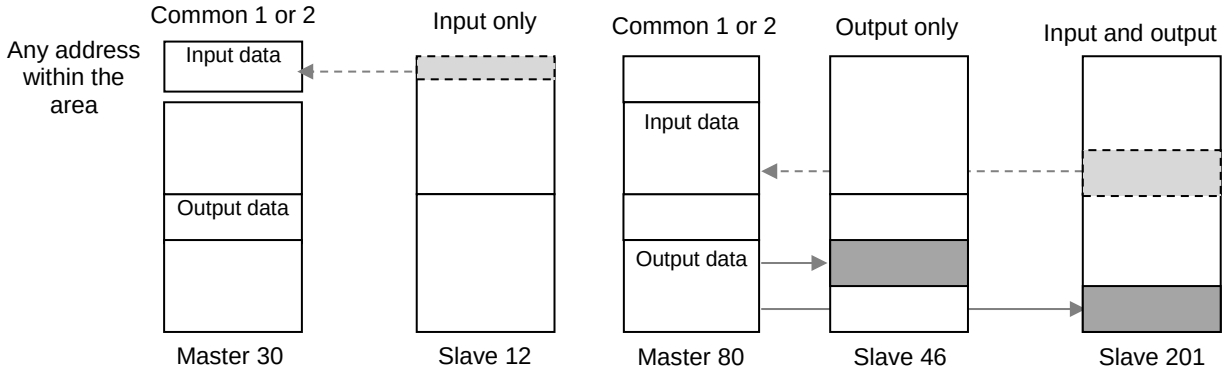
Slave node numbers from 1 to 64
Maximum of 64 nodes
Fixed to 4 words for input
Fixed to 4 words for output
Self area from 16#0000 to 16#00FF

Figure 49 - The fixed setting mode of the multi master

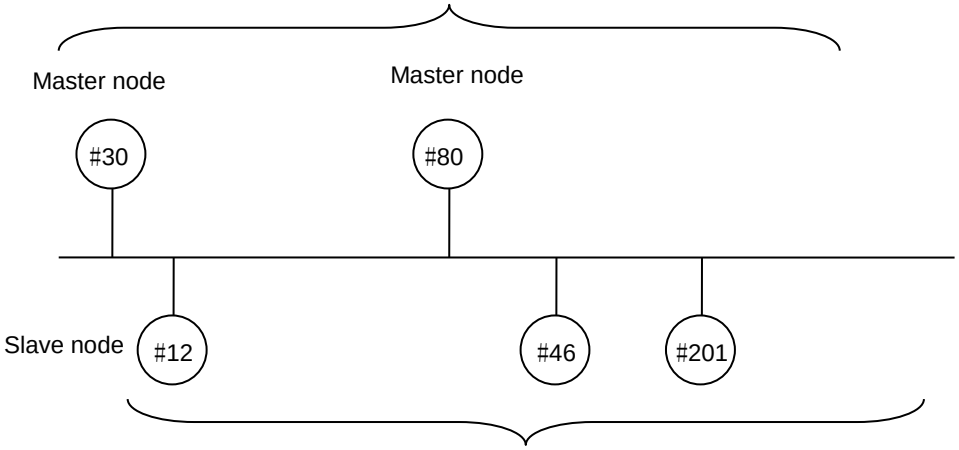
b) Multi master in the flexible setting mode

In the flexible setting mode, as shown in Figure 50, two or more master node can be defined in the range from 1 to 249 of the node number.

Each flexible master can arbitrarily set its node number in the range of 1 to 249, and allocate self memory area in common memory area 1 and 2. In the same manner, each flexible slave can arbitrarily set its node number in the range of 1 to 249, and allocate self memory area in common memory area 1 and 2. Each master send the combination of the master and slaves to corresponding slaves with solicitation frame.



Master node number: 1 to 249
Max: 249 nodes (incl. slave nodes)
Self area: Any value (configured by Master)
Self area size (configured by Master)
Slave nodes to be controlled (configured by Master)



Slave node number: 1 to 249
Max: 249 nodes (incl. master nodes)
Input: Any word (configured by Master)
Output: Any word (configured by Master)
Self area: Any value (configured by Master)

Figure 50 - The flexible setting mode of the multi master

c) Coexistence of the flexible setting mode and the fixed setting mode

The multi master with a mixture of different modes requires settings for each master and slave as shown in Table 18.

Table 18 - Coexistence of the flexible setting mode and the fixed setting mode

	Flexible master	Flexible slave	Fixed master	Fixed slave	Notes
Flexible master		X			Single master (opt): Setting needed
	X	X			Multi master (opt): Setting needed
		X		X	Slave (fixed): Mapping fixed
		X	X	X	Adjustment required to prevent duplicate mapping
Fixed master				X	Single master (fix): Setting not necessary
			X	X	Based on the setting of Master (fix): Setting needed
		X		X	Operation mode of Slave (fixed) needed to be set on Slave (flex)

8.5.4 Coexistence of Nodes of Ver.2 and Ver.3

Both Ver. 3 nodes and Ver. 2 nodes can reside in the same network as follows.

a) Fixed setting mode

As shown in Figure 51, the common memory area not used by the master and slave can be used by other nodes as a shared memory.

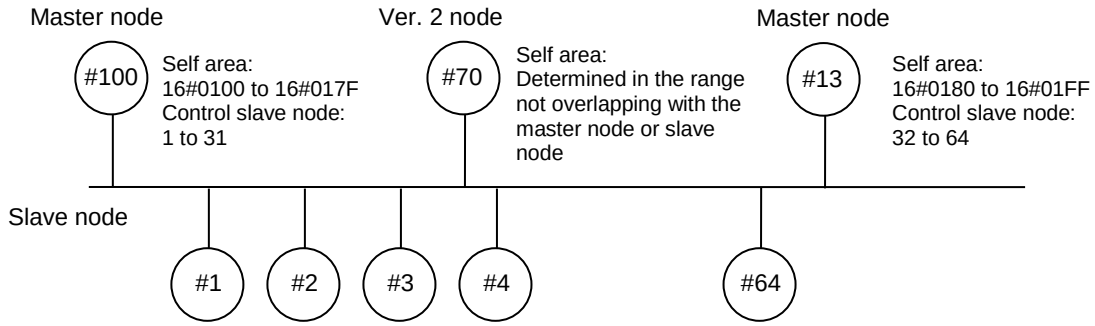


Figure 51 - The fixed setting mode

b) Flexible setting mode

As shown in Figure 52, the common memory area not used by the master and slave can be used by other nodes as a shared memory.

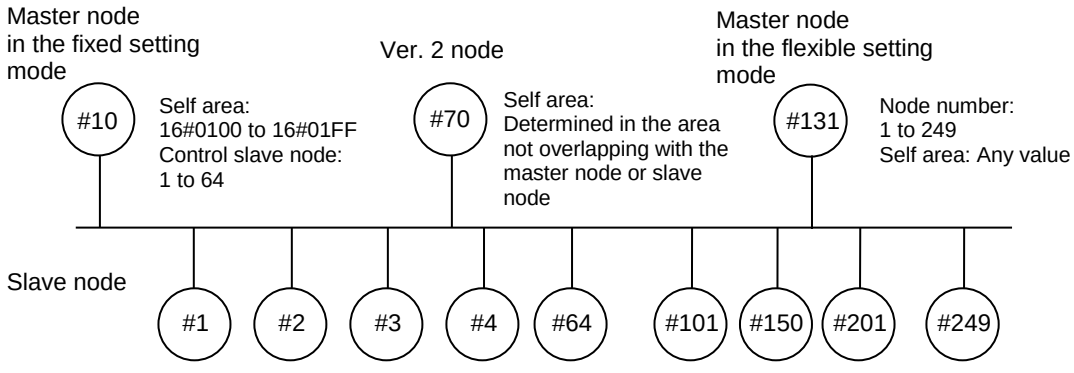


Figure 52 - The flexible setting mode

8.5.5 Limitation

This part describes the limitation on the fixed and flexible setting modes.

a) Fixed setting mode

The master node can use input from irrelevant slave nodes, but cannot output to irrelevant slaves as shown in Figure 53. The slave nodes which can be controlled by one master node shall have continuous node numbers. (Ex. In Figure 52, slave node #10 and #20 cannot be controlled by master #70 and #131.)

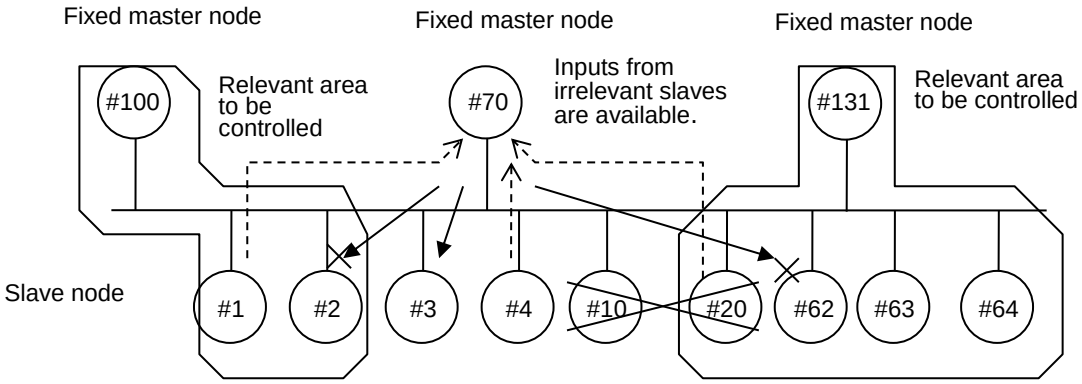


Figure 53 - The fixed setting mode

b) Flexible setting mode

The master node can, use input from irrelevant slave nodes, but cannot output to irrelevant slaves as shown in Figure 54. The control range here means the input output area of the slave node that master node in the flexible setting mode is configured as the self master.

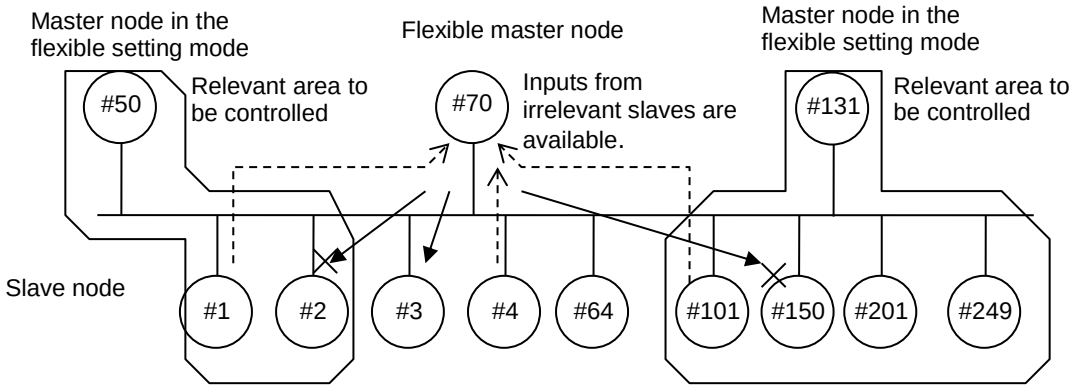


Figure 54 - The flexible setting mode

8.6 IO Definition

8.6.1 IO Data Allocation in the Common Memory

Slave nodes are categorized to the fixed setting mode and flexible setting mode by the setting of the allocation of IO data and status data to the common memory area 1 and 2.

In the fixed setting mode, the common memory occupied by each node is uniquely determined with the node numbers, and in the flexible setting mode, the common memory occupied by each slave is determined by the solicitation frames from the master.

IO data shall be allocated in the common memory as follows.

a) Allocation in the fixed setting mode

The IO data and the status data are fixedly allocated to the common memory area 1 and 2. (refer to Figure 55).

A configuration is consists of one fixed master(node number 100) and 64 slave nodes (node number 1 to 64). The IO data is allocated to the common memory area 1 and the status data is allocated to the common memory area 2.

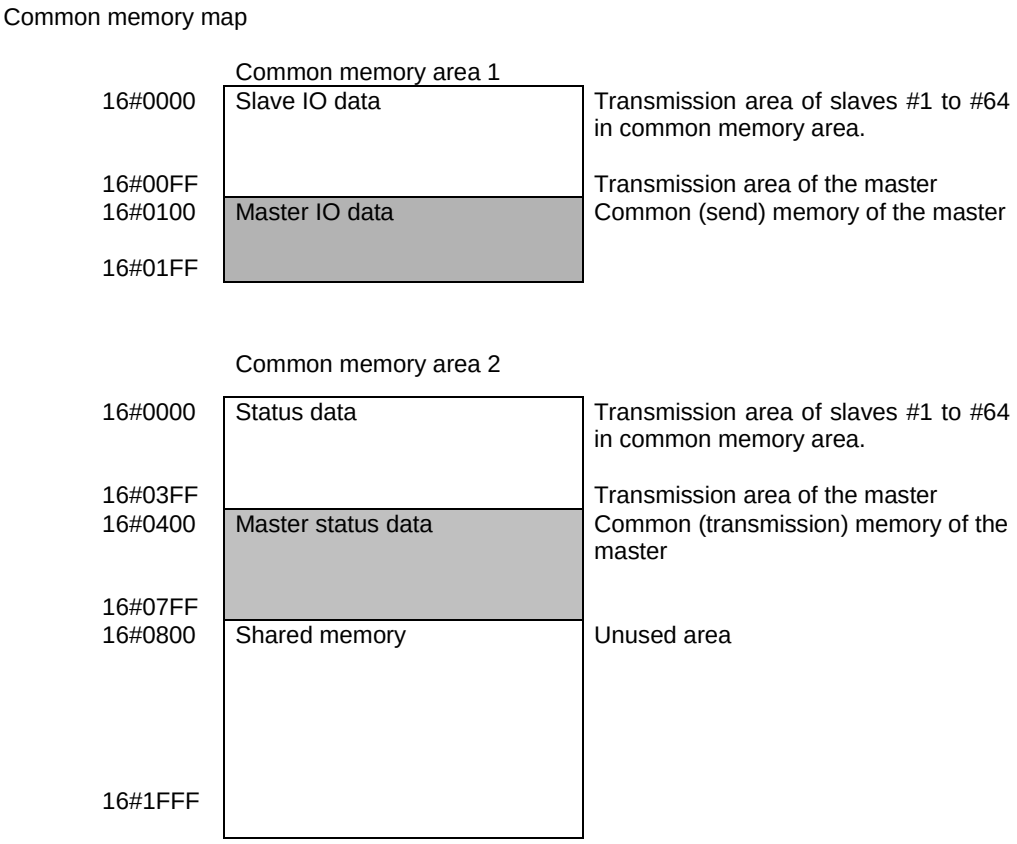


Figure 55 - The common memory allocation in the fixed setting mode

1) Common Memory Area 1 (IO data)

One slave node shall occupy 4 words of the common memory, and the master node shall occupy the maximum of 256 words of the common memory.

Common memory area 1 in the fixed setting mode shall be allocated as shown in FIgure 56.

Input data: Data from a slave to the master

Output data: Data from the master to a slave

16#0000	Input data (slave #1)	Send memory of the slave #1
16#0004	Input data (slave #2)	Send memory of the slave #2
16#0008	Input data (slave #3)	Send memory of the slave #3
	・ ・	
16#00F8	Input data (slave #63)	Send memory of the slave #63
16#00FC	Input data (slave #64)	Send memory of the slave #64
16#0100	Output data (Slave #1)	Send memory of the master node #100
16#0104	Output data (Slave #2)	
16#0108	Output data (Slave #3)	
	・	
16#01F8	Output data (Slave #63)	
16#01FC	Output data (Slave #64)	

Figure 56 - The common memory 1 allocation in the fixed setting mode

The continuous IO area per slave node can be only extended by setting a subsequent area usage flag of the actual configuration slave type in the status area. The subsequent area usage flag can be individually set for input and output, but slave node numbers overlapped with the extended area cannot be used.

Allocation of the common memory 1 under the fixed setting mode when two or more are occupied shall be shown in Figure 57.

16#0000	Input data (slave #1)	Send memory of the slave #1
16#0004	Input data (slave #1)	Send memory of the slave #1
16#0008	Input data (slave #3)	Send memory of the slave #3
	・ ・	
16#00F8	Input data (slave #63)	Send memory of the slave #63
16#00FC	Input data (slave #64)	Send memory of the slave #64
16#0100	Output data (Slave #1)	Send memory of the master node #100
16#0104	Output data (Slave #1)	
16#0108	Output data (Slave #3)	
16#010C	Output data (Slave #4)	
	・	
16#01F8	Output data (Slave #63)	
16#01FC	Output data (Slave #64)	

Figure 57 - The common memory 1 allocation in the fixed setting mode

2) The Common Memory Area 2 (Status data)

As status data, one slave node shall occupy 16 words, and the master node shall occupy the maximum of 1,024 words of the common memory area 2.

The common memory area 2 in the fixed setting mode shall be allocated as shown in Figure 58.

16#0000	Input status (slave #1)	Transmission memory of the slave #1
16#0010	Input status (slave #2)	Transmission memory of the slave #2
16#0020	Input status (slave #3)	Transmission memory of the slave #3
	. . .	
16#03E0	Input status (slave #63)	Transmission memory of the slave #63
16#03F0	Input status (slave #64)	Transmission memory of the slave #64
16#0400	Output status (slave #1)	Transmission memory of the master node #100
16#0410	Output status (slave #2)	
16#0420	Output status (slave #3)	
	. . .	
16#07E0	Output status (slave #63)	
16#07F0	Output status (slave #64)	

Figure 58 - Common memory area 2 allocation under the fixed setting mode

b) Allocation in the flexible setting mode

In the flexible setting mode, user can freely allocate the IO data and the status data for each node in common memory area 1 and 2.

The configuration is consist of the flexible master node (any node number) and the flexible slave node (any node number), and the IO data and the status data are allocated to the common memory area 1 or 2.

Corresponding combination of the master and slaves and allocation of the common memory occupied by slave nodes shall be set by the solicitation frames.

The area occupied by the master is allocated to continuous areas.

The common memory in the flexible setting mode shall be allocated as shown in Figure 59.

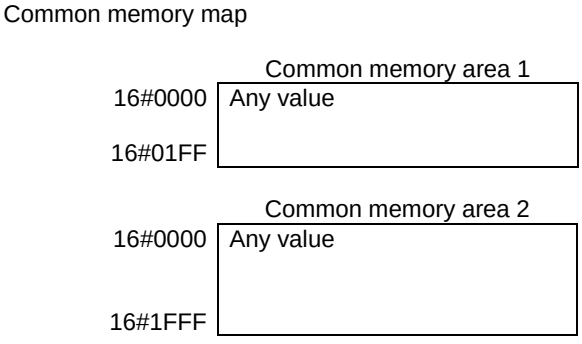


Figure 59 - Common memory allocation in the flexible setting mode

1) Example of the Common Memory Allocation

1.1) In the case the IO area and the status area of the slaves are allocated individually in common memory area 1 and 2.

The common memory in the flexible setting mode shall be allocated as shown in Figure 60.

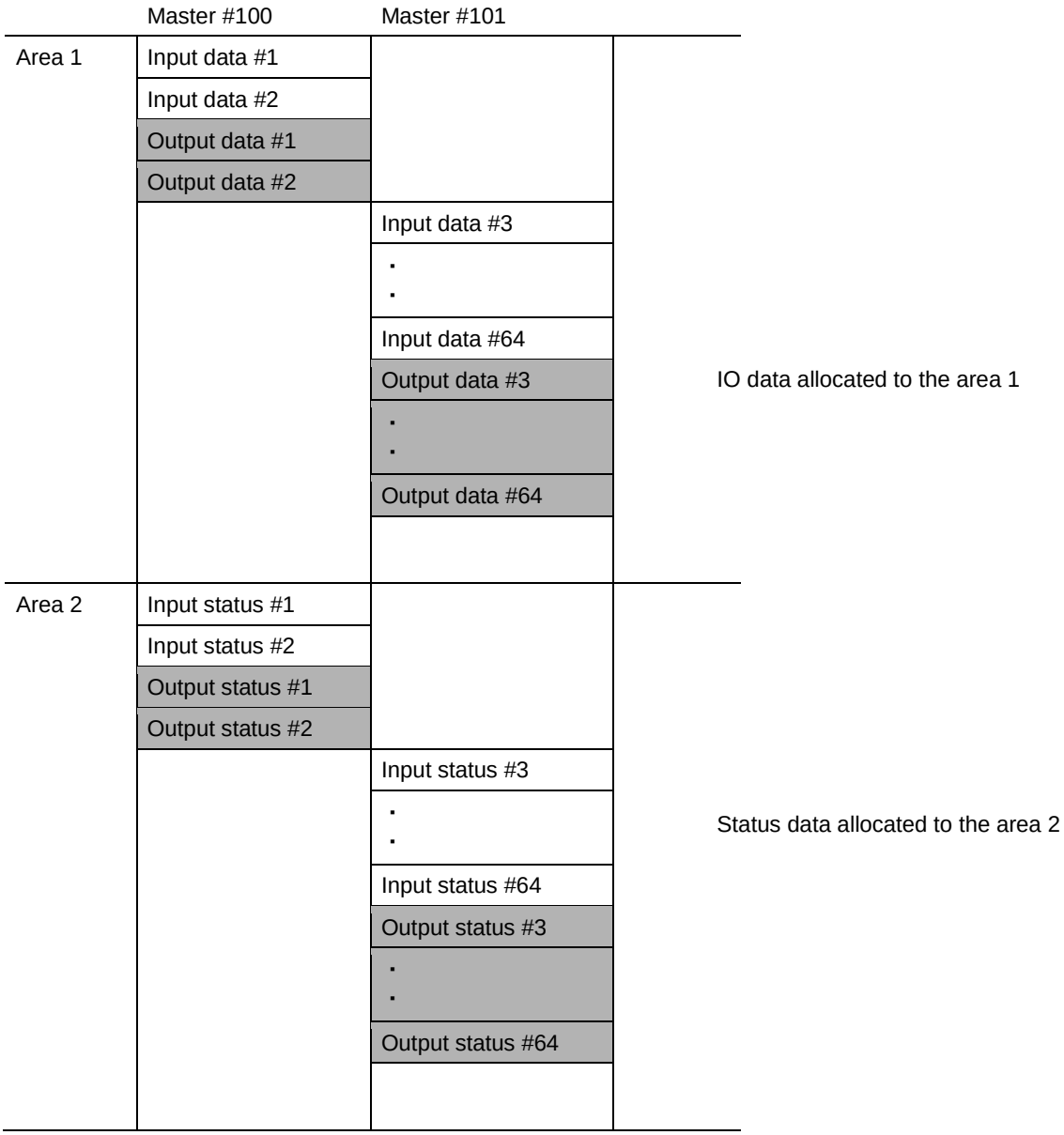


Figure 60 - The common memory allocation in the flexible setting mode (pattern 1)

1.2) In the case the IO area and the status area of the slaves are allocated together in common memory area 1 and 2

The common memory allocation in the flexible setting mode shall be as shown in Figure 61.

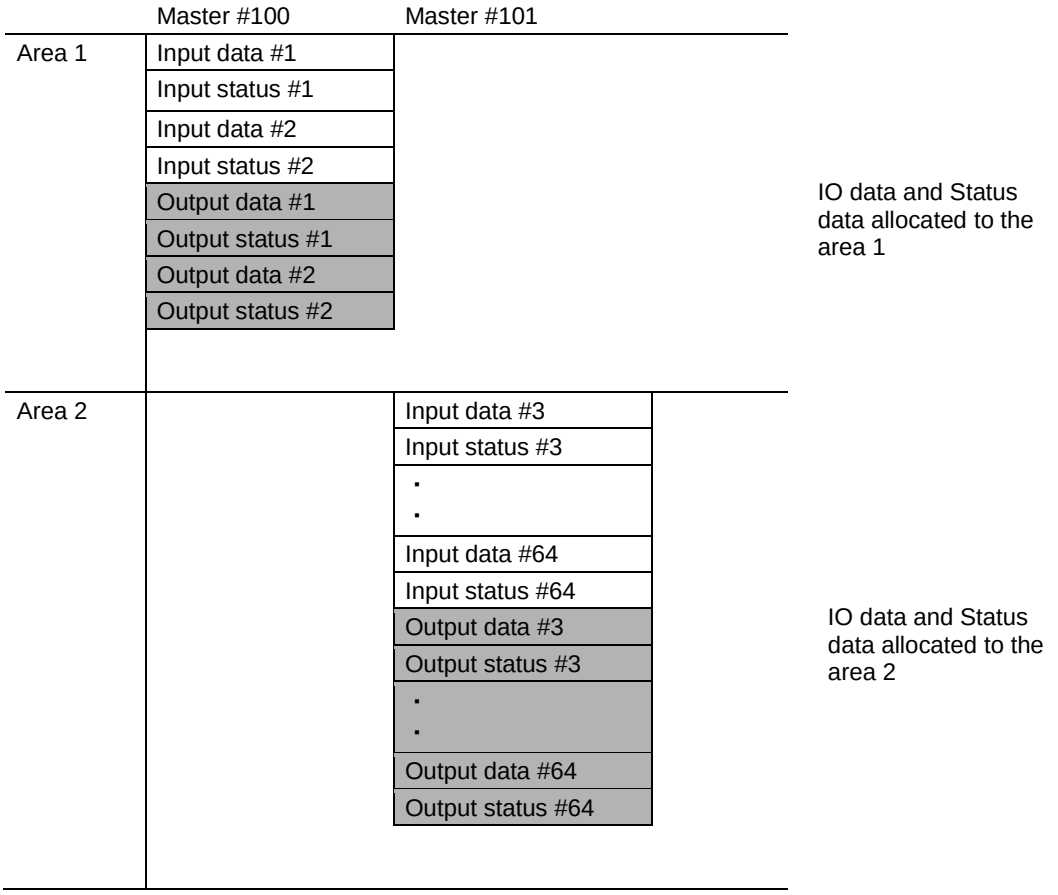


Figure 61 - The common memory allocation in the flexible setting mode(pattern 2)

2) Allocation Procedure of the Common Memory

The common memory allocation shall be set to flexible master with IO allocation setting frames (Figure 62), and shall be distributed from flexible master to flexible slave with solicitation frames (Figure 63).

For details of the IO allocation and solicitation frames, refer to 10.4 and 10.5.

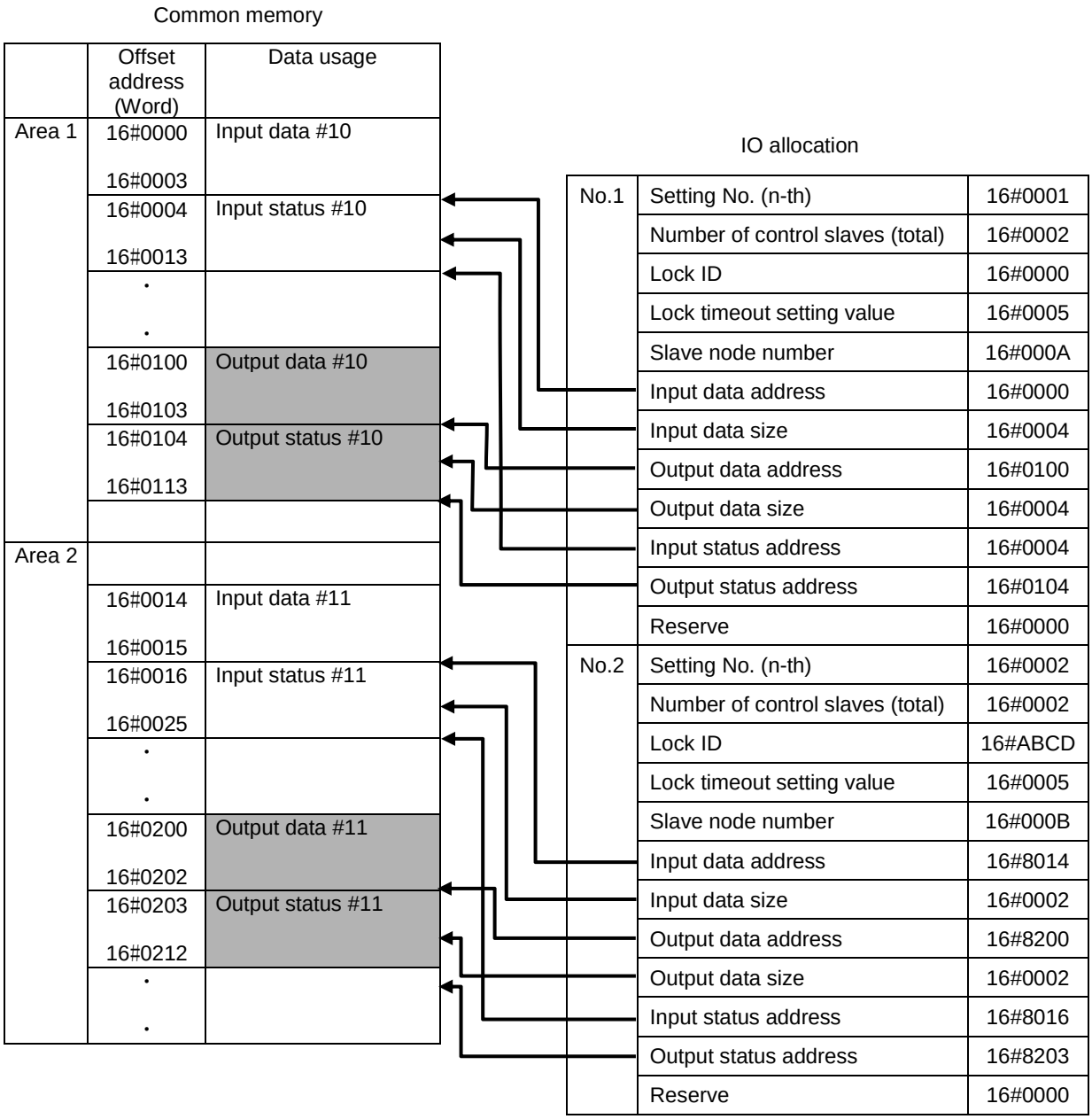


Figure 62 - IO allocation values for flexible master

Data part of the solicitation frame

	Offset (octet)	Items	Setting values
Setting 1	0	Slave node number	16#000A
	2	Input data address	16#0000
	4	Input data size	16#0004
	6	Output data address	16#0100
	8	Output data size	16#0004
	10	Input status address	16#0004
	12	Output status address	16#0104
	14	Lock ID	16#0069
Setting 2	16	Slave node number	16#000B
	18	Input data address	16#8014
	20	Input data size	16#0002
	22	Output data address	16#8200
	24	Output data size	16#0002
	26	Input status address	16#8016
	28	Output status address	16#8203
	30	Lock ID	16#0069

An information set of slave

The IP address of flexible master is assumed to be 192.168.100.105

An information set of slave (For slave #11)

The IP address of flexible master is assumed to be 192.168.100.105

Figure 63 - Solicitation frames generated from IO allocation setting

c) Relation between the solicitation frames and the common memory in the fixed and the flexible setting modes

The solicitation frames and the common memory in the fixed setting mode and the flexible setting mode shall be related as shown in Figure 64.

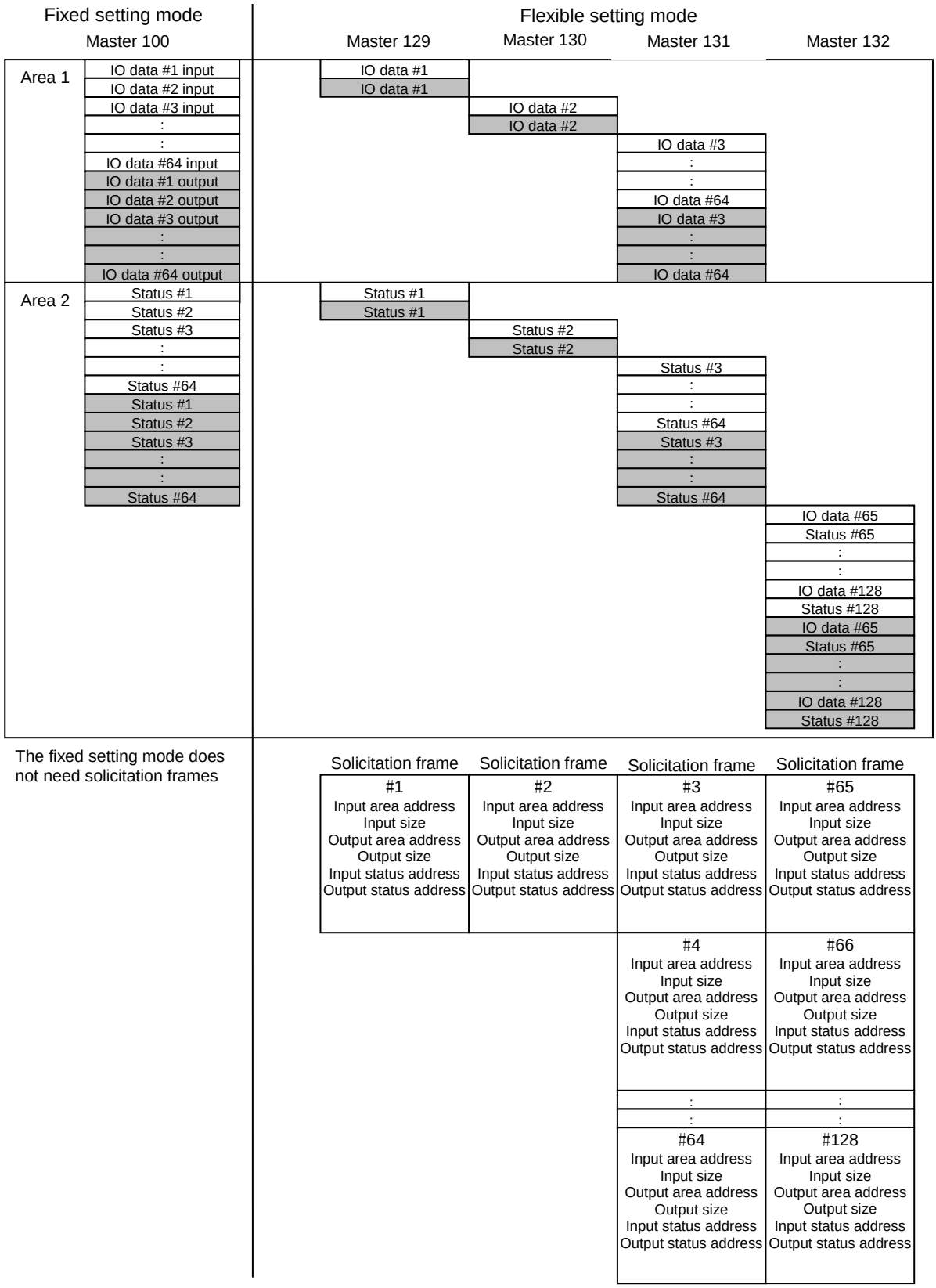


Figure 64 - Relation between solicitation frames and the common memory

8.6.2 Status Areas

This part describes the configuration of the status areas for the master and slaves in the fixed setting mode and the flexible setting mode.

When the subsequent area mentioned later is used, each parameter in the subsequent areas shall be as follows.

— Setting values of execution slave type and specified slave type: Valid (required)

— Setting values of the remote control area, the slave status, the simple setting area, the simple setting confirmation area, the reservation area, and the general-purpose status area: Invalid (ignored)

8.6.2.1 Slave Status Areas in the Fixed

This is an area to notify the master of the status of slave (refer to Figure 65).

The address is an address of the common memory area 2.

Slave output

Area 2				
Address	Usage	Size (Word)	Write	Notes
16#0000	Slave status	1	Slave #1	Indicates the state transition of a slave with the state number
16#0001	Actual slave type	1		Indicates the slave configuration and the number of IO
16#0002	Simple setting confirmation area	1		Master node number (flexible and fixed)
		1		Slave operation setting confirmation status
		6		Data copied from the simple setting area specified from the master
16#000A	General-purpose status area	6		Used for the status notified from a slave..
16#0010	Slave status	1	Slave #2	Same as above
16#0011	Actual slave type	1		
16#0012	Simple setting confirmation area	8		
16#001A	General-purpose status area	6		
16#0020	Slave status	1	Slave #3	Same as above
16#0021	Actual slave type	1		
16#0022	Simple setting confirmation area	8		
16#002A	General-purpose status area	6		
:				
16#03E0	Slave status	1	Slave #63	Same as above
16#03E1	Actual slave type	1		
16#03E2	Simple setting confirmation area	8		
16#03EA	General-purpose status area	6		
16#03F0	Slave status	1	Slave #64	Same as above
16#03F1	Actual slave type	1		
16#03F2	Simple setting confirmation area	8		
16#03FA	General-purpose status area	6		

Figure 65 - Slave status areas in the fixed setting mode

The slave status area in the fixed setting mode shall be as follows.

a) Slave status Indicates the state transition of a slave with the state number.

0: Not joined, 1: Stopped, 2: Connecting, 3: In operation (during input and output), 4: No master, 5: Self node removed, 6: Setting error

b) Actual slave type The actual configuration of a slave (the number of IOs of a slave) is shown below.

A slave compares the specified slave type notified from the master with the actual slave type. If they are not matched, the slave status shall be "6: Setting error".

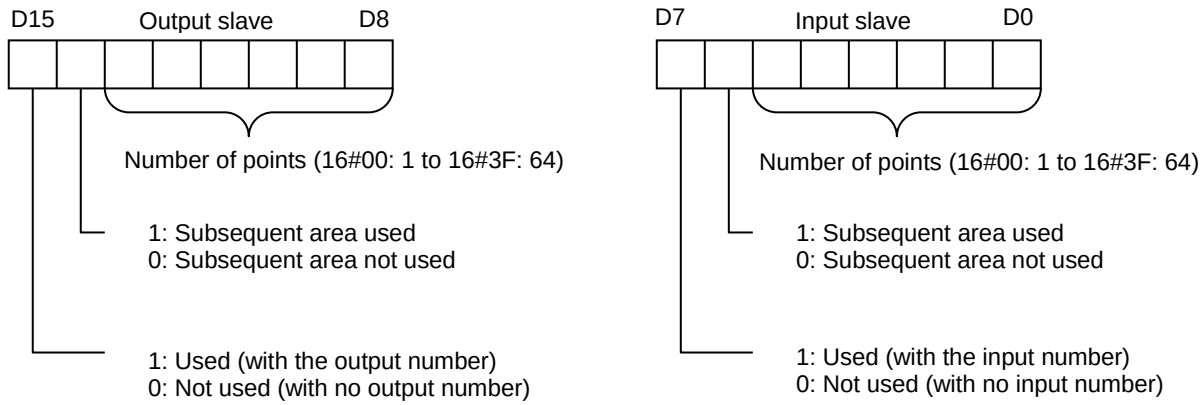
D15 to D8: Output slave, D7 to D0: Input slave

D15/D7: 0: Not used (0), 1: Used

D14/D6: 0: The subsequent area not used, 1: Used

D13 to D8/D5 to D0: Number of points (“16#00: 1” to “16#3F: 64”)

For details, refer to Figure 66. An example of actual slave type shall be as shown in Figure 67.



Note As for "Subsequent area used", when a slave occupies the area over the maximum of 64 points, the subsequent area can also be used with the slave by setting the "Subsequent area used". However, only the continuous area can be used without any blank area.

For example, in case of a module with 127 point outputs and 129 point inputs, actual slave type shown in Figure 67 is set, and the three status areas (16 words x 3= 48 words) are occupied for 129 point input setting by using the subsequent areas.

Figure 66 - The status of the actual slave type (Slave status in the fixed setting mode)

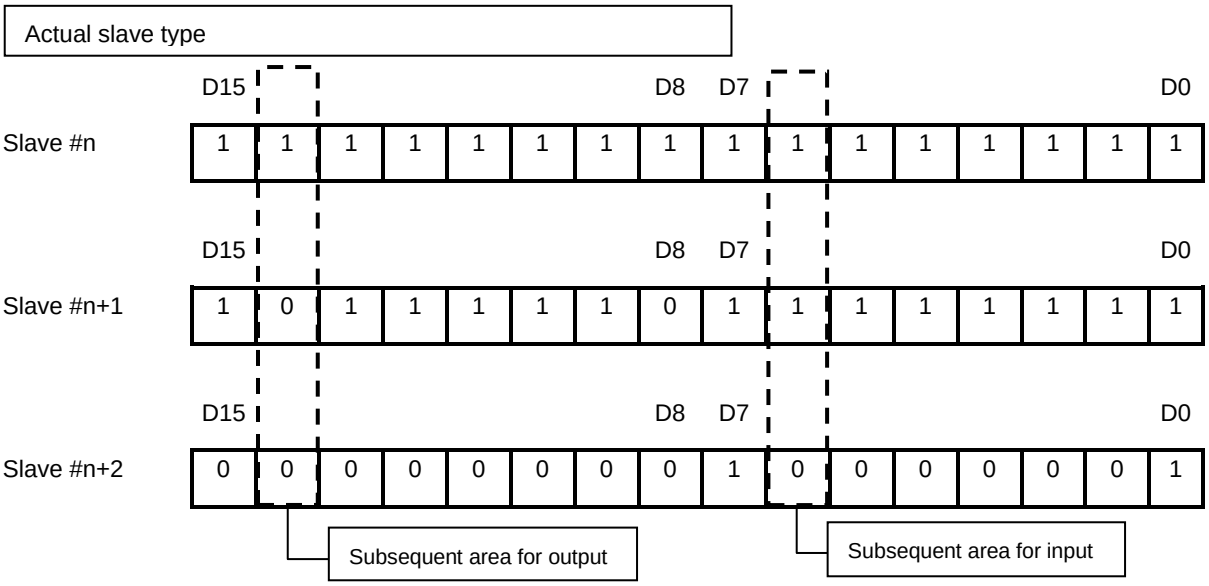


Figure 67 - An example of the status of the actual slave type (Slave status in the fixed setting mode)

c) Simple Setting Confirmation Area

Eight word area for checking whether the simple setting from the master has been reflected, including the following:

1) Master node number: Indicates the master node number controlling slaves.

2) Slave operation setting confirmation status (refer to Figure 68)

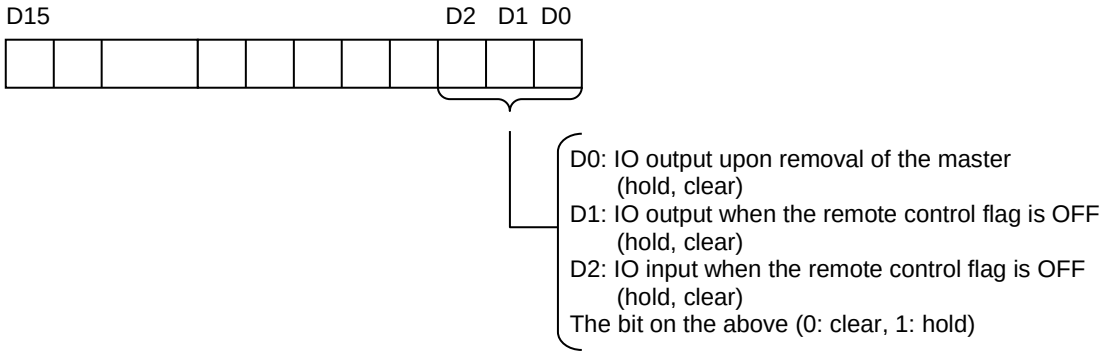


Figure 68 - Slave operation setting confirmation status in the simple setting confirmation area (Slave status in the fixed setting mode)

3) he contents retrieved in the simple setting area specified from the master shall be reflected.

d) General-purpose status area Is used for the status notified from a slave.

8.6.2.2 Master Status Area in the Fixed Setting Mode

The master status area is to notify the slaves of the status of the master (Refer to Figure 69).

Slaves receive the master's operating instructions from the master status area.

The address is an address of the common memory area 2.

Since the master status is dynamically changed, it is checked periodically to reflect to slave operation.

Remote control area and slave operating instructions can be dynamically changed, and operation when other data is dynamically changed shall depend on implementation.

Area 2				
Address	Usage	Size (Word)	Write	Notes
16#0400	Remote control area (for #1)	1	Master	
16#0401	Specified slave type (for #1)	1		
16#0402	Simple setting area (for #1)	1		Master node number instructions (fixed setting mode only) Setting ranges are 0, and from 65 to 249 0=100 (default)[a]
		1		a) Slave operating instructions D0: IO output upon removal of the master (hold, clear) D1: IO output when the remote control flag is OFF (hold, clear) D2: IO input when the remote control flag is OFF (hold, clear) The bit on the above (0: clear, 1: hold)
		6		Contents can be defined with slave products
16#040A	Reserved area (for #1)	6		Shall be 16#0000 (fixed) in the current specification and confirmation on the slave side is not necessary
16#0410	Remote control area (for #2)	1	Master	Same as above
16#0411	Specified slave type (for #2)	1		
16#0412	Simple setting area (for #2)	8		
16#041A	Reserved area (for #2)	6		
16#0420	Remote control area (for #3)	1	Master	Same as above
16#0421	Specified slave type (for #3)	1		
16#0422	Simple setting area (for #3)	8		
16#042A	Reserved area (for #3)	6		
:				
16#07E0	Remote control area (for #63)	1	Master	Same as above
16#07E1	Specified slave type (for #63)	1		
16#07E2	Simple setting area (for #63)	8		
16#07EA	Reserved area (for #63)	6		
16#07F0	Remote control area (for #64)	1	Master	Same as above
16#07F1	Specified slave type (for #64)	1		
16#07F2	Simple setting area (for #64)	8		
16#07FA	Reserved area (for #64)	6		

Note[a)] The master shall set the self node number to the master node number instrution area of the corresponding slave node. However, the node numbers (areas) of the slave nodes controlled by one master shall be continuous.

Figure 69 - Master status areas in the fixed setting mode

The master status areas in the fixed setting mode shall be as follows.

a) Remote control area Is an area to set remote control of slaves (Refer to Figure 70).

Slaves start and stop operation based on operating instructions from the master.

Remote stop: Instructs to stop the slave from operating.

Remote operation: Instructs to start the slave operating. Slaves shall start operating when the change of this bit from 0 to 1 is detected during the slave status “1: Stopped”.

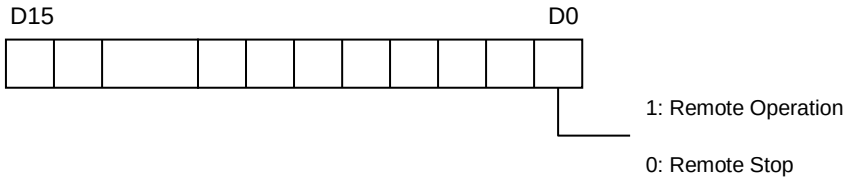


Figure 70 - The status of the remote control area (Master status in the fixed setting mode)

b) Specified slave type

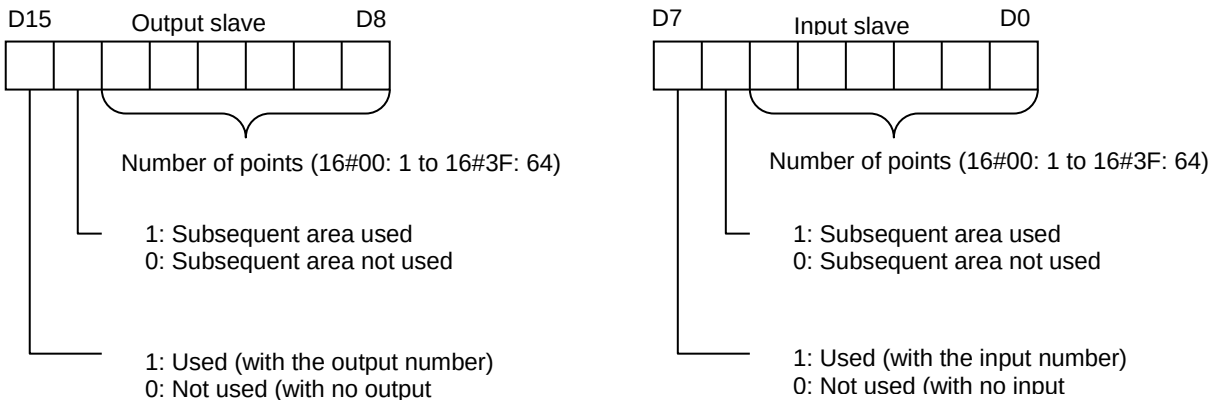
D15 to D8: Output slave, D7 to D0: Input slave

D15/D7: 0: Not used (0), 1: Used

D14/D6: 0: The subsequent area not used, 1:Used

D13 to D8/D5 to D0: Number of points (16#00: 1 to 16#3F: 64)

For details, refer to Figure 71. An example of specified slave type shall be as shown in Figure 72.



Note As for "Subsequent area used", when a slave occupies the area over the maximum of 64 points, the subsequent area can also be used with the slave by setting "Subsequent area used". However, only the continuous area can be used without any blank area.

For example, in case of a module with 127 point outputs and 129 point inputs, the specified slave type shown in Figure 72 is set, and the three status areas (16 words x 3= 48 words) are occupied for 129 point input setting by using the subsequent areas. (The specified slave type from the master indicates an expected value of the number of slave IOs. The slave checks consistency with the expected value, and if it is inconsistent, the slave status shall be the setting error).

Figure 71 - The status of specified slave type (Master status in the fixed setting mode)

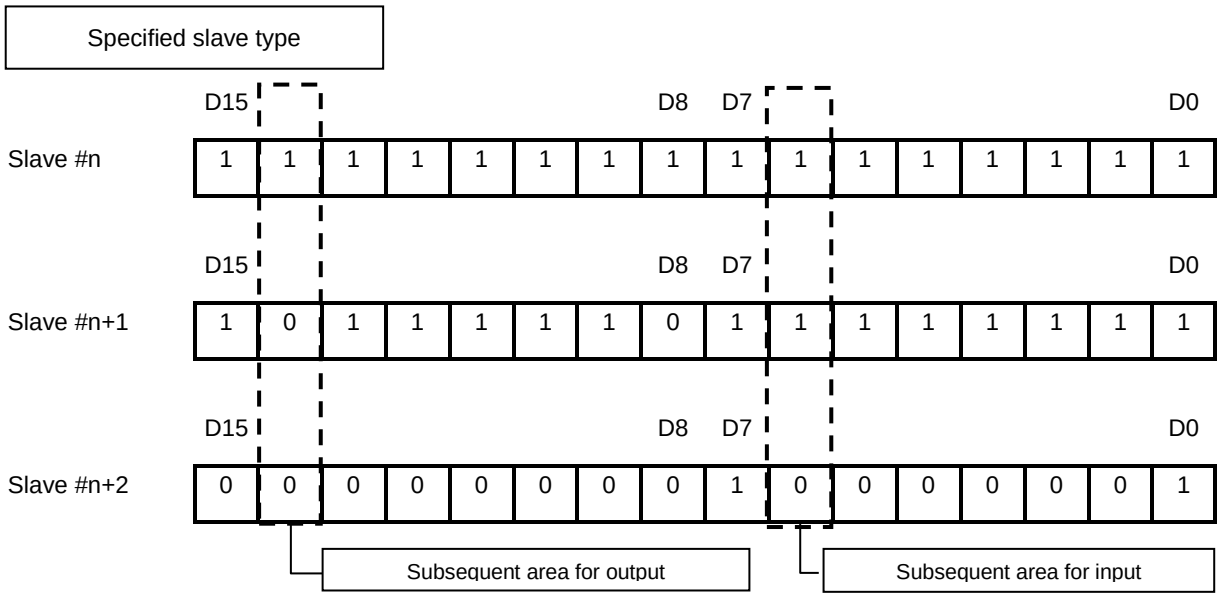


Figure 72 - An example of the status of specified slave type (Master status in the fixed setting mode)

c) Simple setting area

Is an 8-word area for setting from the master, including the followings:

1) Master node number: Specifies the master node number to control the slave in the fixed setting mode.

— 0 specifies the master node number 100 by default.

— Master node number 65 or more can be set.

2) Slave operating instructions (refer to Figure 73)

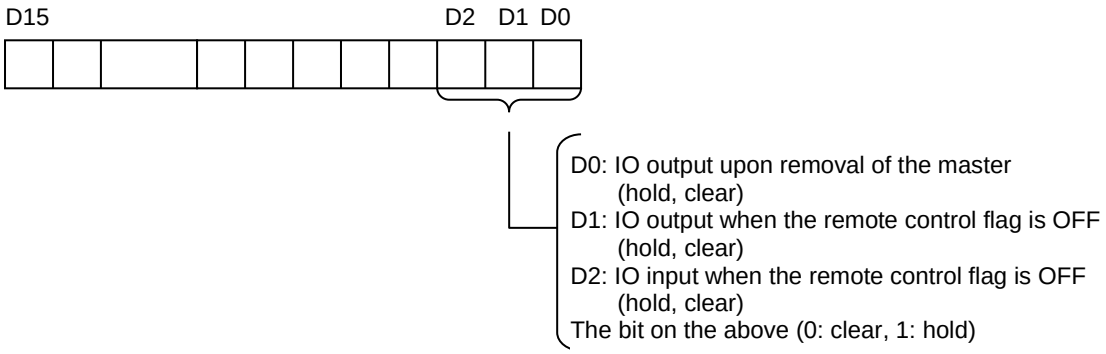


Figure 73 - The status of slave operating instructions in simple setting area (Master status in the fixed setting mode)

3) An area instructed from the master and defined for each slave

8.6.2.3 Slave Status Areas in the Optional Setting Mode

This is an area to notify the master of the status of slave (refer to Figure 74).

The address is a word address configured with the input status address in the data part of solicitation frames from the master.

Slave output ↓

Area 1 or area 2				
Address	Usage	Size (Word)	Write	Notes
16#0000	Slave status	1	Slave #n	Indicates the state transition of a slave with the status number
16#0001	Actual slave type	1		Indicates the slave configuration and the Number of points
16#0002	Simple setting confirmation area	1		Master node number (flexible and fixed)
		1		Slave operation setting confirmation status
		6		Contents taken in the simple setting area specified from the master
16#000A	General-purpose status area	6		Used to notify of the status from a slave.
16#0010	Slave status	1	Slave #n'	Same as above [a)]
16#0011	Actual slave type	1		
16#0012	Simple setting confirmation area	8		
16#001A	General-purpose status area	6		
・ ・	・ ・	・ ・	・ ・	Same as above [a)]

Note [a)] These areas are used when the subsequent areas are used (only when exceeding 64 points).

Figure 74 - Slave status areas in the flexible setting mode

Slave status areas in the flexible setting mode shall be as follows.

a) Slave status

Indicates the state transition of a slave with the state number.

0: Not joined, 1: Stopped, 2: Connecting, 3: In operation (during input and output), 4: No master, 5: Self Node removed, 6: Abnormal Settings(when the slave type specified from the master is inconsistent with the actual slave type)

b) Actual slave type

The actual configuration of a slave (the number of IOs of a slave) is shown below.

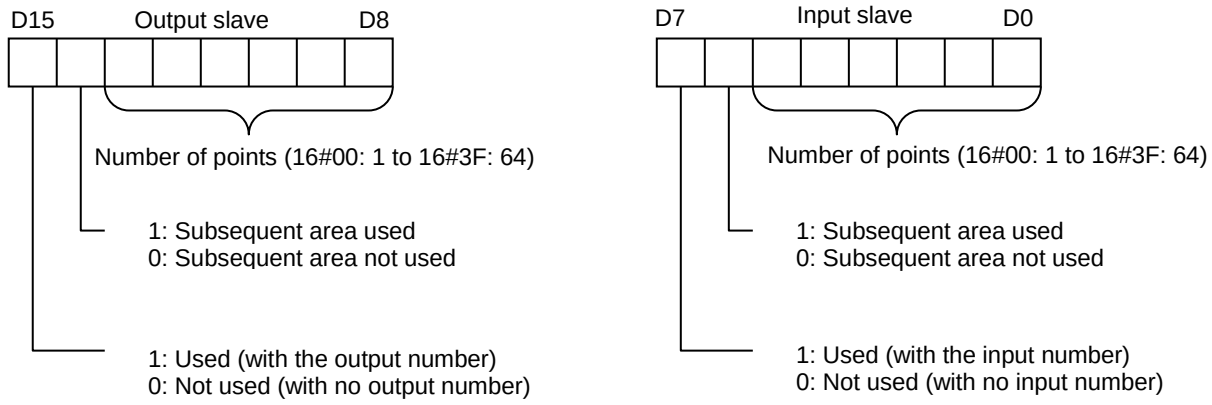
D15 to D8: Output slave, D7 to D0: Input slave

D15/D7: 0: Not used (0), 1:Used

D14/D6: 0:Subsequent area not used, 1: Used

D13 to D8/D5 to D0: Number of points (16#00: 1 to 16#3F: 64)

For details, refer to Figure 75. An example of actual slave type shall be as shown in Figure 76.



Notes As for “Subsequent area used”, when a slave occupies the area over the maximum of 64 points, the subsequent area can also be used with the slave by setting "Subsequent area used". However, only the continuous area can be used without any blank area.

For example, in case of a module with 127 point outputs and 129 point inputs, when the output data size is set to be 8 words and the input data size is set to be 9 words in the solicitation frame from flexible master, the actual configuration shall be set as follows.

Figure 75 - The status of the actual slave type (Slave status in the flexible setting mode)

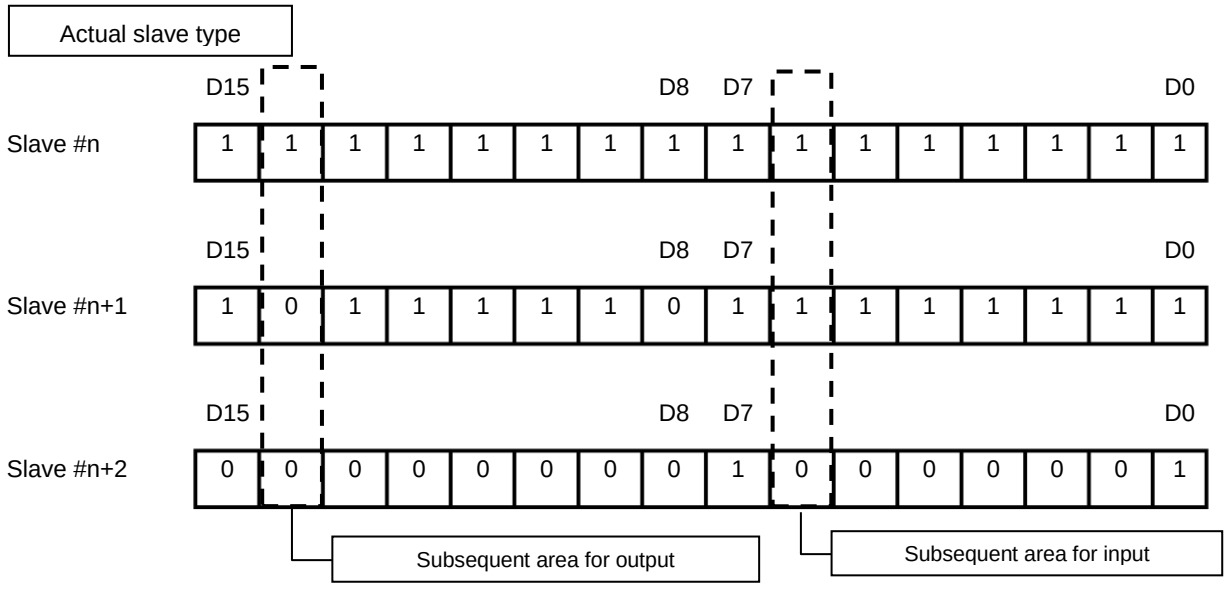


Figure 76 - An example of the status of the actual slave type (flexible setting mode slave status)

The status area to be used shall be 16 words x 3 = 48 words because the status areas for three are needed for setting the subsequent area of the input slave, considering maximum 128 points for outputs from the output data size and maximum 144 points for outputs from input data size configured in the solicitation frames.

c) Simple setting confirmation area

Is an 8-word area for checking whether the simple setting from the master has been reflected, including the followings:

1) Master node number: Indicates the master node number controlling slaves.

2) Slave operation setting confirmation status (refer to Figure 77)

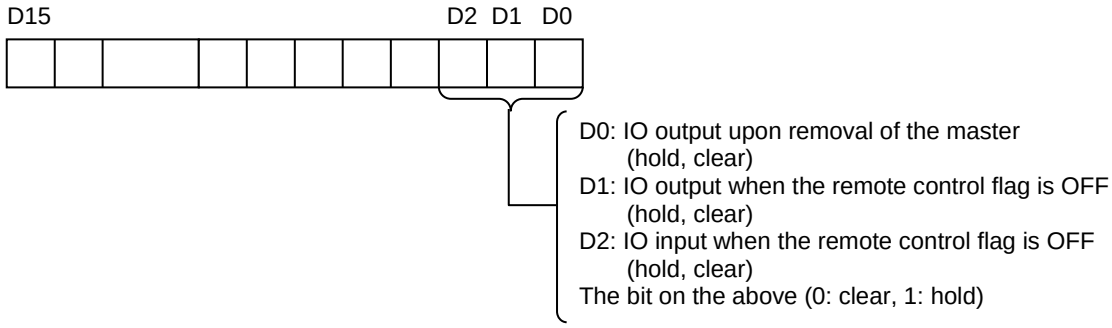


Figure 77 - Slave operation setting confirmation status in the simple setting confirmation area (Slave status in the flexible setting mode)

3) The contents retrieved in the simple setting area specified from the master shall be reflected.

d) General-purpose status area Is used for the status notified from a slave.

8.6.2.4 Master Status Area in the Optional Setting Mode

The master status area is to notify the slave of the status of the master (Refer to Figure 78).

The address is an offset word address from the output status address in the data part of the IO allocation setting frame (A status area is allocated for each slave).

Since the master status is dynamically changed, it is checked periodically to reflect to slave operation.

Area 1 or area 2				
Address	Usage	Size(Word)	Write	Notes
16#0000	Remote control area (for #n)	1	Master	
16#0001	Specified slave type (for #n)	1		
16#0002	Simple setting area (for #n)	1		Master node number instructions[a)]
		1		Slave operating instructions D0: IO output upon removal of the master (hold, clear) D1: IO output when the remote control flag is OFF (hold, clear) D2: IO input when the remote control flag is OFF (hold, clear) 0= clear and 1= hold for the above-mentioned bit.
		6		Contents can be defined with slave products
16#000A	Reserved area (for #n)	6		Shall be 16#0000 (fixed) in the current specification and confirmation on the slave side is not necessary
16#0010	Remote control area (for #n')	1	Master	Same as above[b)]
16#0011	Specified slave type (for #n')	1		
16#0012	Simple setting area (for #n')	8		
16#001A	Reserved area (for #n')	6		
・ ・	・ ・	・ ・		Same as above[b)]

Note [a)] The master shall set the self node number to the master node number instrution area of the corresponding slave node.

[b)] These areas are used when the subsequent areas are used (only when exceeding 64 points).

Figure 78 - The flexible setting mode Master status areas

The master status areas in the flexible setting mode shall be as follows.

a) Remote control area

An area to set remote control of slaves (Refer to Figure 79).

Slaves start and stop operation based on operating instructions from the master.

Remote stop: Instructs to stop the slave from operating.

Remote operation: Instructs to start the slave operating. Slaves shall start operating when the change of this bit from 0 to 1 is detected during the slave status “1: Stopped”.

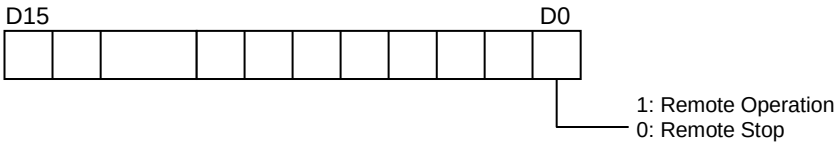


Figure 79- The status of the remote control area (Master status in the fixed setting mode)

b) Specified slave type

Indicates the expected configuration of the slave expected by the master.

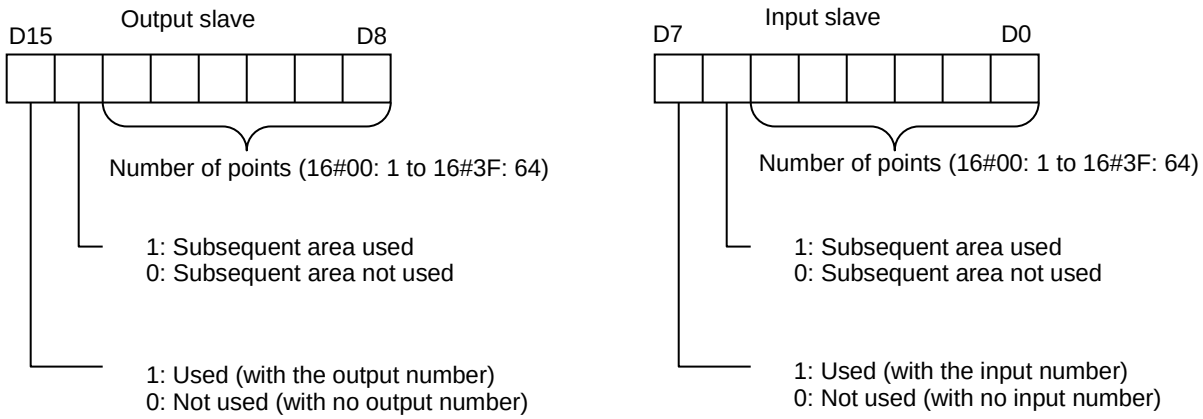
D15 to D8: Output slave, D7 to D0: Input slave

D15/D7: 0: Not used (0), 1: Used

D14/D6: 0: The subsequent area not used, 1: Used

D13 to D8/D5 to D0: Number of points (16#00: 1 to 16#3F: 64)

For details, refer to Figure 80. An example of specified slave type shall be as shown in Figure 81.



Notes As for "Subsequent area used", when a slave occupies the area over the maximum of 64 points, the subsequent area can also be used with the slave by setting "Subsequent area used". However, only the continuous area can be used without any blank area. For example, in case of a module with 127 point outputs and 129 point inputs, the specified slave type shown in Figure 81 is set, and the three status areas (16 words x 3= 48 words) are occupied for 129 point input setting by using the subsequent areas. (The specified slave type from the master indicates an expected value of the number of slave IOs. The slave checks consistency with the expected value, and if it is inconsistent, the slave status shall be the setting error).

Figure 80 - The status of specified slave type (Master status in the flexible setting mode)

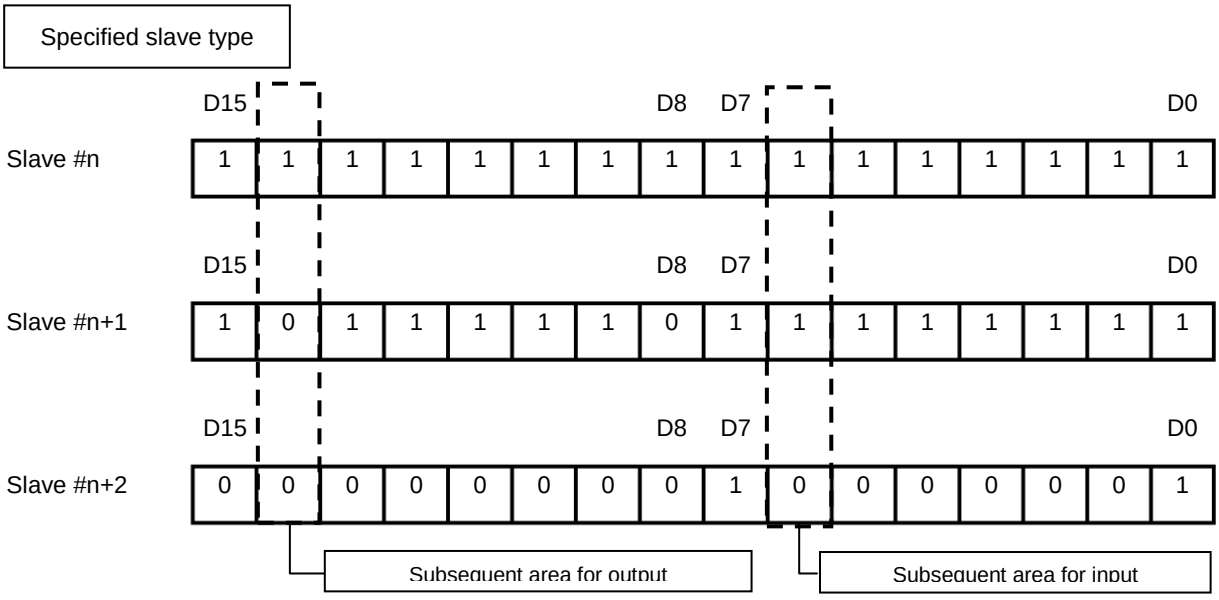


Figure 81 - An example of the status of specified slave type (Master status in the flexible setting mode)

c) Simple setting area

Is an 8-word area for setting from the master, including the followings.

1) Master node number: Specifies the master node number to control the slave in the flexible setting mode. The number from 1 to 249 can be set.

2) Slave operating instructions (refer to Figure 82)

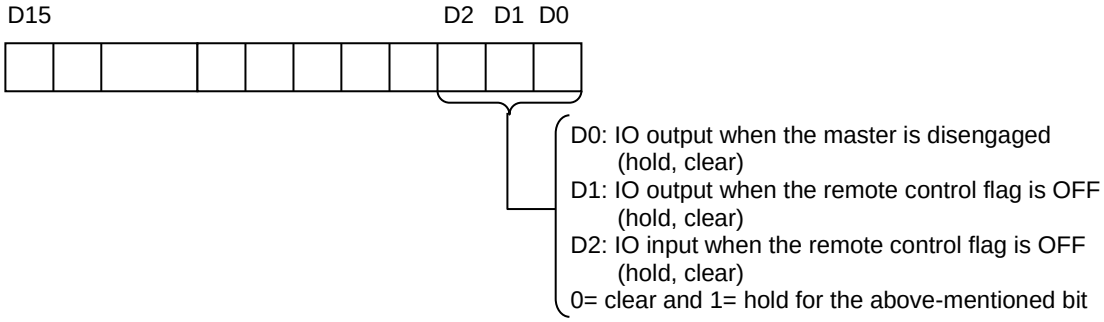


Figure 82 - Simple setting area The status of slave operating instructions (flexible setting mode master status)

3) n area instructed from the master and defined for each slave.

8.7 Timer

8.7.1 Token Watchdog Time (TW)

Token Watchdog Time (TW) is the maximum time from a token obtained by a node to the token to be issued. The TW that each node has is notified to all nodes as information in the frame header. Each node uses the TW value in the frame as monitoring time from the token obtained to the token to be issued by the appropriate node.

The value of the token watchdog time shall be from 1 to 255 ms (incremented by 1 ms).

8.7.2 Token Retention Time (THT)

Token Retention Time (THT) is a time that a node guarantees the issue of a token. Each node shall issue a token within this period of time. When internal processing takes longer than this period of time, a token is not issued in order to prevent from having two or more tokens. The THT value shall not exceed the TW value (incremented by 1 ms)

8.7.3 Allowable Minimum Frame Interval Time (MFT)

Allowable Minimum Frame Interval Time (MFT) is the minimum interval ensured from the end of a frame to the subsequent frame to be sent.

The MFT is used in the following two cases.

a) Period of time from when a token to the self node is received to when the self node transmits a certain frame.

b) Period of time from frame transmission of a self node to the next frame transmission (cyclic division, transmission of a message) (Period of time from the end of the previous frame to the beginning of the next frame).

The MFT value obtained by each node is notified to all the nodes as header information of frames. Each node calculates the maximum value based on the information, and uses it as an MFT value. The MFT value shall be from 0 to 5,000 microseconds (incremented by 100 microseconds).

The MFT shall not be applied between a token frame and the previous cyclic frame. The time between these frames shall be at the minimum.

8.7.4 Allowable Refresh cycle time (RCT)

Allowable Refresh cycle time (RCT) is the 120% value of the time for a token to circulate around at a single time if a node does not receive a message frame from when the node obtains the token addressed to the node to when the node obtains the token addressed to the node next time.

The RCT on the the self node frame shall be "0" in the network startup state. The RCT value

shall be set to frames after token frames addressed to the self node are received 3 times. Therefore, a message is not transmitted until the token circulates around 3 times.

Nodes joining in in-ring startup state shall set the RCT value to the frame after the token frames addressed to the self node are received 3 times. However, if RCT value cannot be specified, the allowable refresh cycle time of the lastly received cyclic frame is used.

Through the monitoring process above, the RCT is dynamically determined based on the number of nodes joining the network. The RCT value shall be incremented by 1 ms.

8.7.5 Refresh Cycle Measurement Time (RMT)

Refresh Cycle Measurement Time (RMT) is a period of time between when a certain node obtains a token addressed to the node and when it obtains the token addressed to the node next time. The RMT value shall be incremented by 1 ms.

8.7.6 Joining Token Detection Time (TDT)

Joining Token Detection Time (TDT) is a period of time to monitor whether or not a node is in the valid linking state when it participates in the network. Except when a token frame or a trigger frame is received, a node transmits no frames until this time period is exceeded. The TDT value shall be fixed (3,000 ms).

8.7.7 Participation Request Frame Transmission Waiting Time (PWT)

Participation Request Frame Transmission Waiting Time (PWT) is a period of time after the node in the network startup state receives or sends a trigger frame until the node sends a participation request frame. The participation request frames are transmitted with the timer value based on the node number for each node to prevent a overlap of the other newly participating node.

PWT = n (node number) X 4

n : Node number

8.7.8 Participation Request Frame Acceptance Time (PAT)

Participation Request Frame Acceptance Time (PAT) is a waiting time after the node in the network startup state receives or sends a trigger frame until other nodes send a participation request frame. The PAT value shall be fixed (1,200 ms).

8.7.9 Trigger Frame Transmission Waiting Time (TrWT)

Trigger Frame Transmission Waiting Time (TrWT) is a period of time after the TDT is exceeded until the node transmits a trigger frame.

TrWT = R X 4

R : Remainder when the node number is divided by 8

8.7.10 Three Circulation Waiting Time (3CWT)

Three Circulation Waiting Time (3CWT) is a time to monitor the token to circulate around 3 times. The node waits for the token to circulate around 3 times when participating in the network in the in-ring startup state. The 3CWT value shall be fixed (3,000 ms).

8.7.11 Acknowledge Waiting Time (AWT)

Message acknowledge Waiting Time (AWT) is a waiting time after the node sends 1 to 1 message until an ACK comes back from the destination node. The AWT value shall be fixed (100 ms).

8.7.12 Solicitation Frame Batch Transmission Frame Interval (SFBTFI)

Solicitation Frame Batch Transmission Frame Interval (SFBTFI) is a time interval between frames in case flexible master divides a solicitation frame and transmits to flexible slave when not in the valid linking state.

The SFBTFI value shall be fixed to 5,000 (microseconds), which is the maximum of the MFT.

8.7.13 Solicitation Frame Transmission Waiting Time (SFTWT)

Solicitation Frame Transmission Waiting Time 2 (SFTWT) is a timer used when flexible master transmits a solicitation frame in the valid linking state.

Flexible master holding a token shall transmit a solicitation frame and start the timer when not all of flexible slaves managed by the self node have participated in the network.

When the SFTWT times out and not all of the slaves managed by the self node have participated in the network, flexible master transmits the solicitation frame again and repeats the transmission process until all the slaves participate in the network..

When the solicitation frames are divided for transfer, the timer shall be started after the transmission.

The SFTWT value shall be as follows.

$$\text{SFTWT} = \text{SFTWT} = (\text{TDT}) + (3\text{CWT}) + (\text{PWT})$$

$$= 3\,000 + 3\,000 + 1\,200$$

$$= 7{,}200 \text{ (ms) fixed.}$$

9 Operation

9.1 Management of Transmission Right

9.1.1 Outline

Access control to the FL-net network is performed with a token. A node can perform transmission basically only when it is holding a token.

A node shall be able to perform transmission without a token only when reissuing a token when a token is lost, or when transmitting a trigger frame when trying to participate in the network, a participation request frame, and a solicitation frame at the batch transmission.

— In the FL-net, one token shall go around among nodes participating in the network. A node shall hold the transmission right over the network after receiving this token until it hands over the token to the next node. In this process, the time to hold a token shall not exceed the token retention time.

— The frame which can be transmitted when holding the token shall be two or more cyclic frames, one or less message frame, or one or less solicitation frame.

— When the transmission right is released, the token frame shall be transmitted.

— The token is monitored by the timer of each node and shall be automatically reissued when it has not gone around the network for a certain period of time.

— A function to unite tokens is provided when there are two or more tokens in the network.

9.1.2 Flow of Token

A token frame includes a node number of destination side (DNA) of the token and a node number of source side (SNA) of the token. Each node becomes a token holding node when the node number of destination side in the received token frame matches the self node number.

The order of the rotation of a token is determined by the node number. Each node participating in the network circulates the token in ascending order of the node number. The

last node with the maximum node number in ascending order of the node numbers passes the token to the node with the minimum node number. The flow of the token shall be as shown in Figure 83.

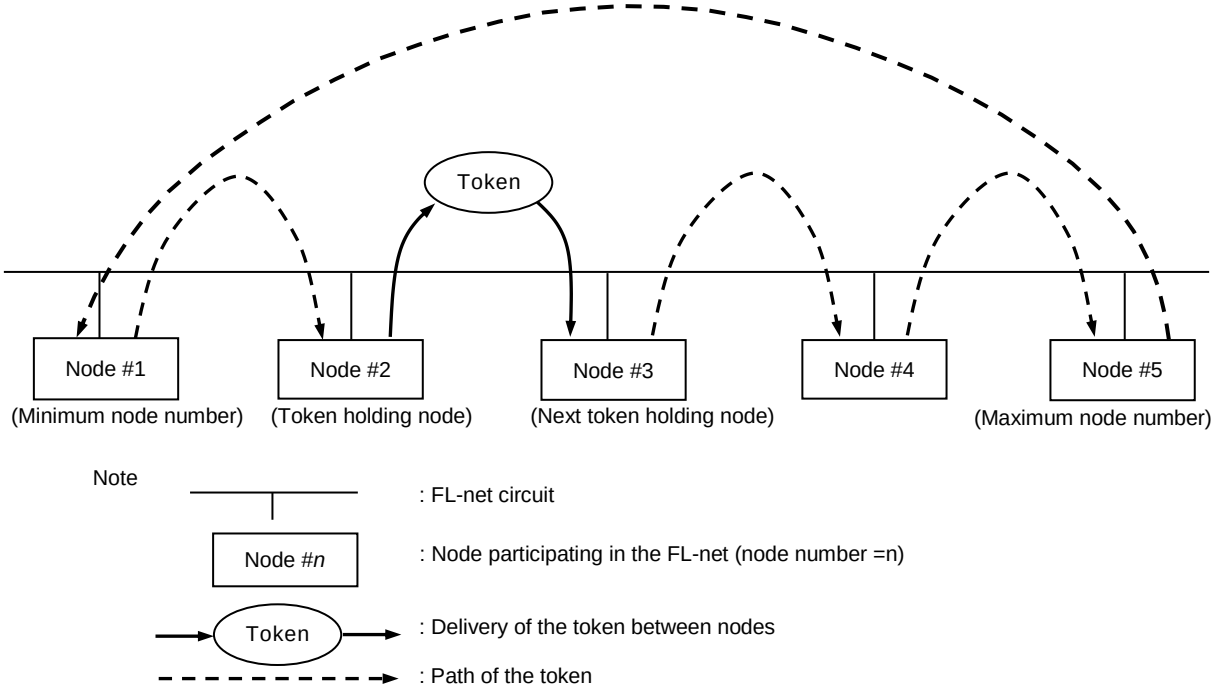


Figure 83 - The flow of a token

9.1.3 Reissuing Token

A token is reissued when both the following two monitor times timed out.

a) Monitor with the refresh cycle measurement time

Each node monitors the time after obtaining a token addressed to the self node until it obtains a next token addressed to the self node (refresh cycle measurement time).

When this time exceeds the allowable refresh cycle time, time is considerd to be out.

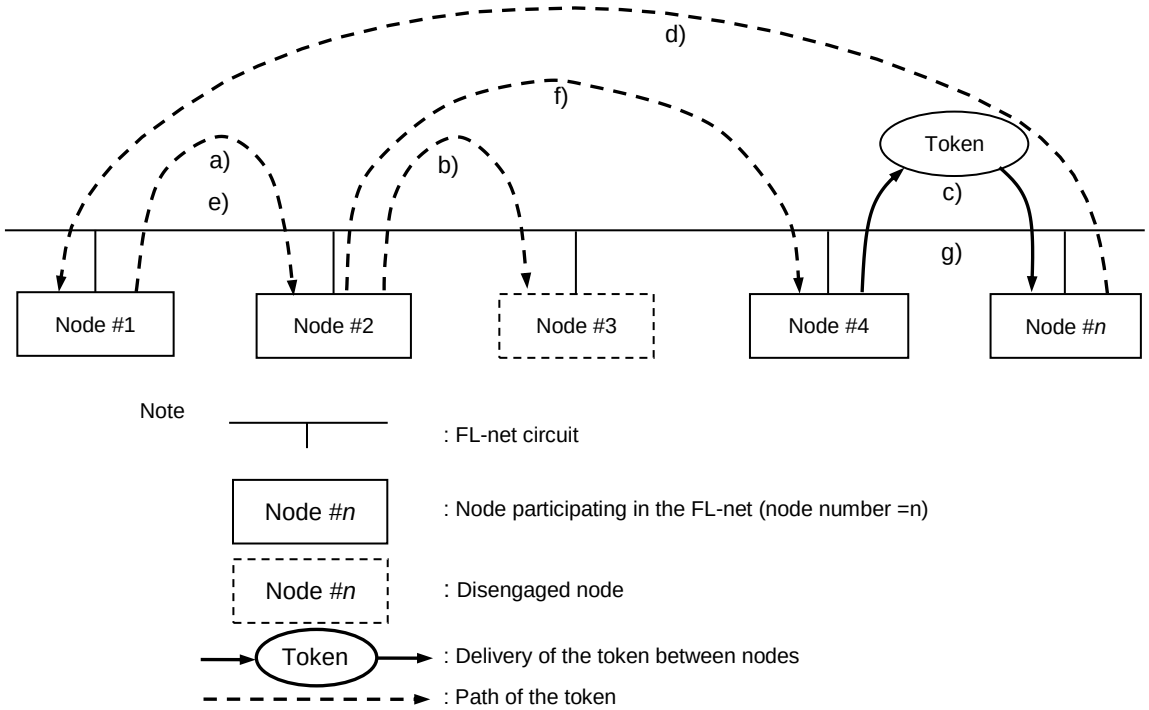
b) Monitor with the token watchdog time

Monitoring with the token watchdog time shall measure the time after a token last flown in the network.

When this time exceeds "total token watchdog time of each node from a token holding node to the self node", time is considered to be out.

When a token frame from a node not participating is received, process is performed with the token watchdog time of 255 ms. However, when it is a node which removed, the value when the node participated is used.

The flow of reissuing a token shall be as shown in Figure 84.



a) Transmit a token from the node #1 to the node #2.
b) Transmit a token from the node #2 to the node #3.
c) Since the node #3 has removed from the FL-net, a token is reissued from the node #4 to the node #n.
d) Transmit a token from the node #n to the node #1.
e) Transmit a token from the node #1 to the node #2.
f) Since the node #3 has removed, a token is transmitted from the node #2 to the node #4.
g) Transmit a token from the node #4 to the node #n. After this, repeat d) to g).

Figure 84 - The flow of reissuing a token

9.1.4 Preventing from Having Two or More Tokens

When a node received a token frame addressed to another node when it is holding a token, the node shall give priority to the one with a smaller node number. In other words, when the self node number is smaller than the node number of destination side in the received frame, the node shall discard the received token and maintain the status in which the self node is holding the token. To the contrary, when the self node number is larger than the node number of destination side in the received frame, the node shall discard the token of the self node and change from the status holding the token to the status waiting for the token.

9.2 Joining and Disengaging Node

9.2.1 Joining Node

Each node shall monitor the network until the joining token detection time of each times out when starting.

At this time, when a node does not detect a token frame, it determines to be in the network startup state, and processes a), and when a node detect a token frame, it determines to be in the in-ring startup state, and processes b).

a) When a token frame is not received (Network startup state)

The flow of the procedure is shown in a) of Figure 85 when a node starts simultaneously with another node under the network startup state, and shown in b) of Figure 85 when a node starts alone under the network startup state.

Each node shall prepare for transmitting a trigger frame if a token frame is not detected within the joining token detection time. The trigger frame is transmitted after trigger frame transmission waiting time (TrWT) (refer to b) in Figure 85). Each node does not transmit a

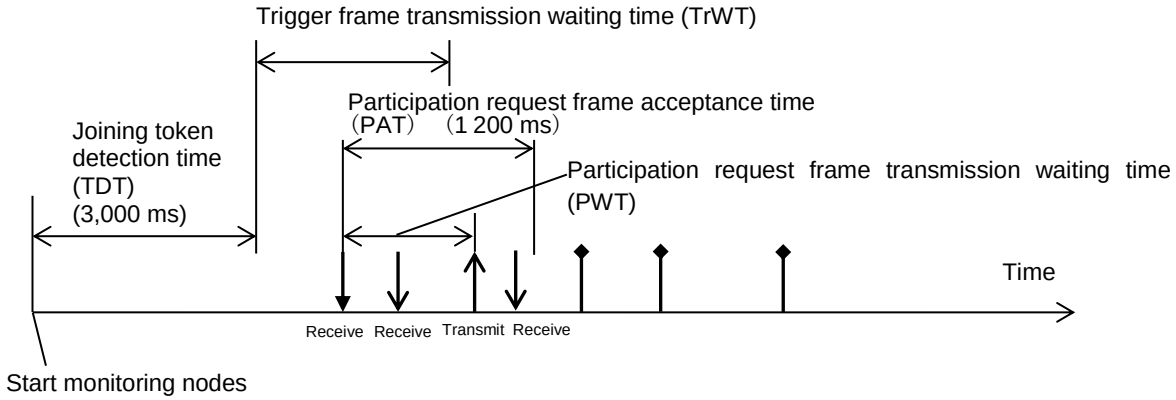
trigger frame when it received a trigger frame from another node before transmitting a trigger frame (refer to a) in Figure 85).

A node waits for all the nodes to transmit participation request frames for the participation request frame acceptance times (1,200ms) after receiving or transmitting a trigger frame, while detecting the node numbers, overlap of the common memory area, and updating participating node management information. Overlap is detected for the area of all the identified nodes.

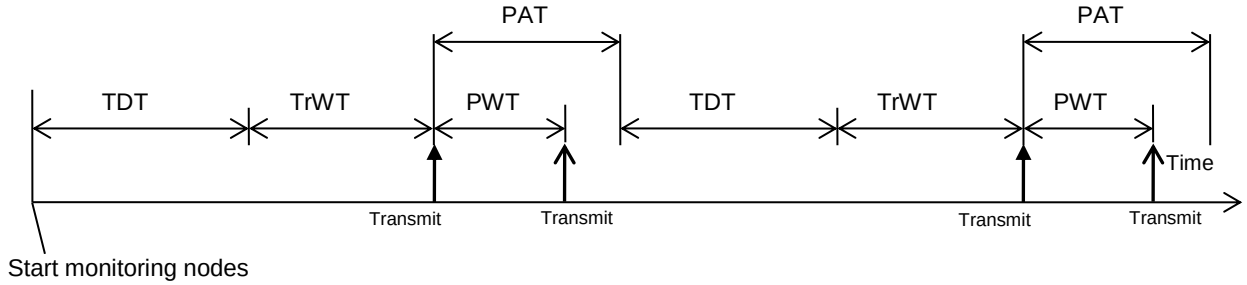
A participation request frame shall be transmitted after participation request frame transmission waiting time (PWT) has passed from receiving or transmitting a trigger frame. When the participation request frame acceptance time has passed, the node with the minimum node number shall issue a token first.

The node which identified overlap of the node number with the participation request frame of another node does not participate in the network, and perform no transmission or reception. Error information of node number overlap shall be notified to the upper layer after setting the node number overlap flag of the status information of the self node. The node which identified overlap of the address joins the network, sets the top address of the common memory area 1 and 2 and the common memory size to "0", and does not transmit cyclic data. At this time, the address overlap detection flag in the status information of the FA link shall be set, and the common memory data validity notice flag shall be reset. Overlap of the node numbers and overlap of the addresses are detected until the self node receives a token.

When no frame is received before the participation request frame acceptance time times out, these processes are repeated until a participation request frame or a token frames of other than the self node is received. When existence of other nodes has not been identified before the participation request frame acceptance time times out after retransmitting the participation request frame 3 times, the receiving waiting status flag of the status information of the self node shall be set.



a) An example when starting simultaneously with another node



b) An example when the self node alone is starting

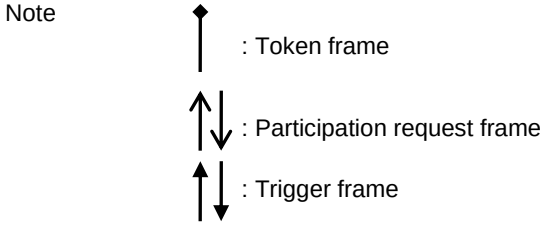


Figure 85 - Joining at the network startup state

b) When a token frame is received (in-ring startup state)

The process of joining at the in-ring startup state shall be as shown in Figure 86. When a node detects a token frame within the joining token detection time, it shall determine the link has already been established. At this time, the node refrains from transmitting the participation request frame until the token goes around 3 times after receiving a token frame to the minimum node number. The node detects overlap of the node numbers and the area in the common memory addresses with the frames received in this period, a and updates participating node management information. Overlap is detected for the areas of all the identified nodes.

In this case, when there is no overlap of a node number, the node shall transmit a participation request frame and participate in the network after the participation request frame transmission waiting time has passed. A participation request frame is transmitted whether or not a token is held.

When a node detects a overlap of a node number, it does not transmit a participation request frame and does not participate in the network. The overlap error of a node number shall be notified to the upper layer after setting the node number overlap flag of the status information

of the self node.

When a node detects a overlap of the address, it joins the network, sets the top address of the common memory area 1 and 2 and the common memory size to "0", and does not transmit cyclic data. At this time, the address overlap detection flag of the status information of FA link shall be set, and the common memory data validity notice flag shall be reset.

Detection of overlap of the node number and overlap of the address is performed when not in the network.

After joining the network with a participation request frame, when a token frame addressed to the self node is not received after waiting for 3 laps, a joining process is performed again. In addition, when a token frame is not received after transmitting a participation request frame before the three circulation waiting time passes, the joining process is performed again.

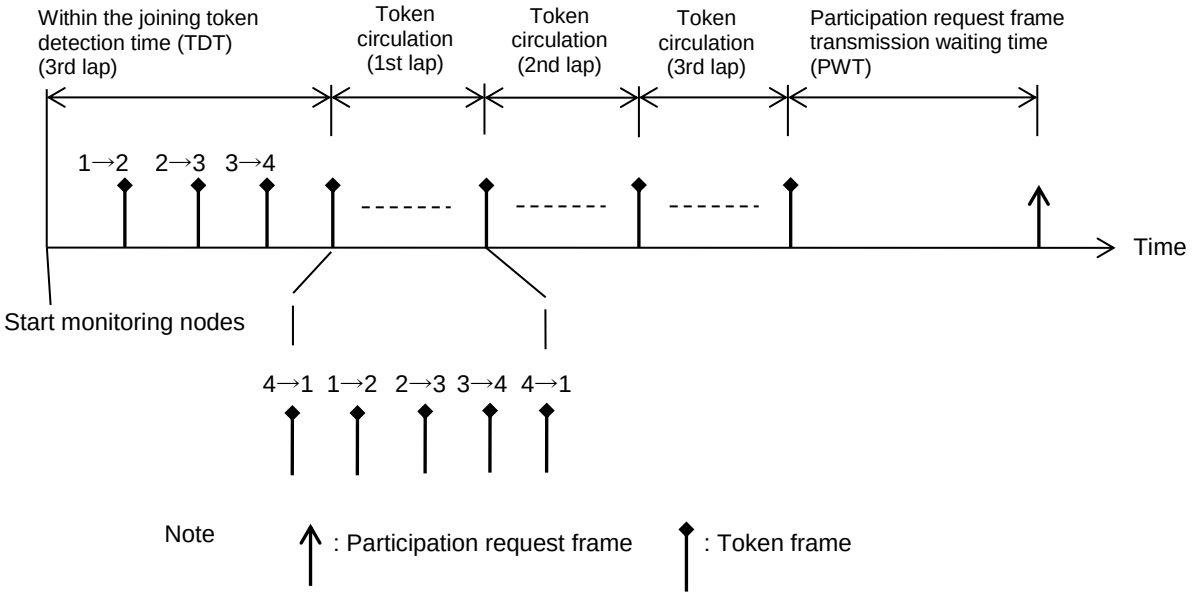


Figure 86 - Participating in the in-ring startup state

9.2.2 Node to be Joined

When a node participating in the network received a participation request frame or when the node received the last cyclic frame from an unidentified node, each node shall update participating node management information on the corresponding node.

When updating participating node management information, overlap of the area of the common memory shall be detected.

9.2.3 Management of Disengagement

Each node checks the node number of source side each time a token frame is received, and when a token frame from a certain node is not received 3 times in succession, the node shall be removed.

When a certain node is considered to have removed from the network, the status of the FA link within participating node management information shall be changed to the status in which the node removed. In addition, when other nodes removed during token circulation, and only the self node is left, the process shall shift to the joining sequence of 9.2.1.

9.2.4 Monitoring Token Addressed to the self Node

Each node monitors the token addressed to the self node. In other words, when the token addressed to the self node is not received 3 times in succession on condition that the token is normally going around in the network, the self node shall be considered to have removed. After this, the removed node shall shift to the joining sequence of 9.2.1 again.

9.3 Data Communication between Nodes

9.3.1 Types of Data Communication

The FA link supports two types of data transmission.:

— Cyclic transmission which transmits periodic data

— Message transmission which transmits aperiodic data

9.3.2 Cyclic Transmission

Cyclic transmission shall be as follows.

a) Outline

Cyclic transmission is a function for each node to notify other nodes of data in its transmission area allocated to the common memory by using broadcast.

Cyclic transmission is performed each time the node holds the token, and all the data on the common memory to be transmitted shall be transmitted.

Nodes participating in the network which does not perform cyclic data transmission are also allowed.

b) The amount of data and the number of frames

Cyclic transmission shall divide a frame into two or more frames based on the amount of data to be transmitted. Therefore, the frame structure of cyclic transmission differs depending on the amounts of data including the area 1 data and the area 2 data.

1) When the transmission data is 1024 octets or less

When the amount of transmission data including the area 1 and the area 2 does not exceed 1024 octets, the data in the area 1 and the area 2 are combined to be transmitted in one frame (refer to Figure 87).

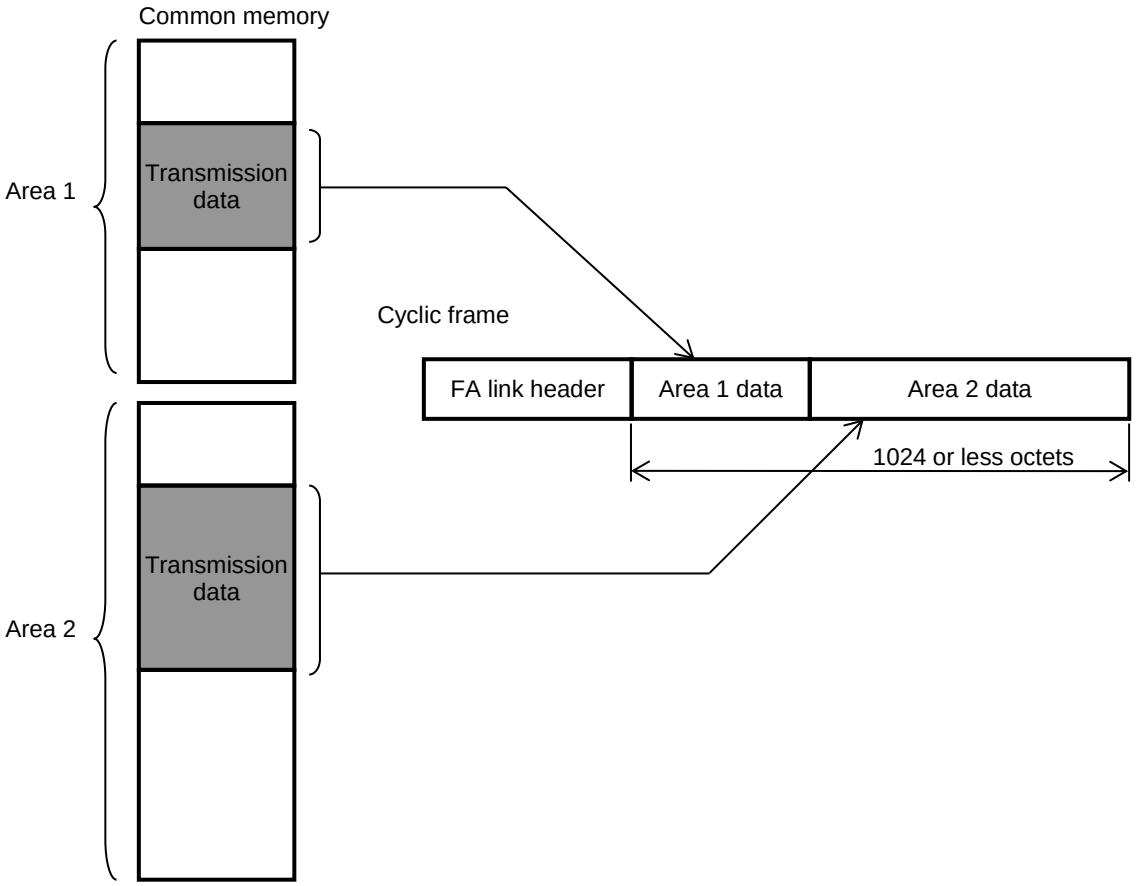


Figure 87 - Frame structure when the transmission data is 1024 or less octets

2) When the transmission data is larger than 1024 octets

When the amount of transmission data including the area 1 and the area 2 is larger than 1024 octets, the data shall be divided into two or more frames for transmission. At this time, the first cyclic frame including all the data of the area 1 and all or part of the data of the area 2 not exceeding 1024 octets in total shall be transmitted. The following frames are transmitted by 1024 octets until all the data in the transmission area is finished, and only the last frame has the data size of 1024 or less octets. An example of a frame structure when transmission data is larger than 1024 octets (the frame structure when dividing into three frames for transmission) shall be as shown in Figure 88.

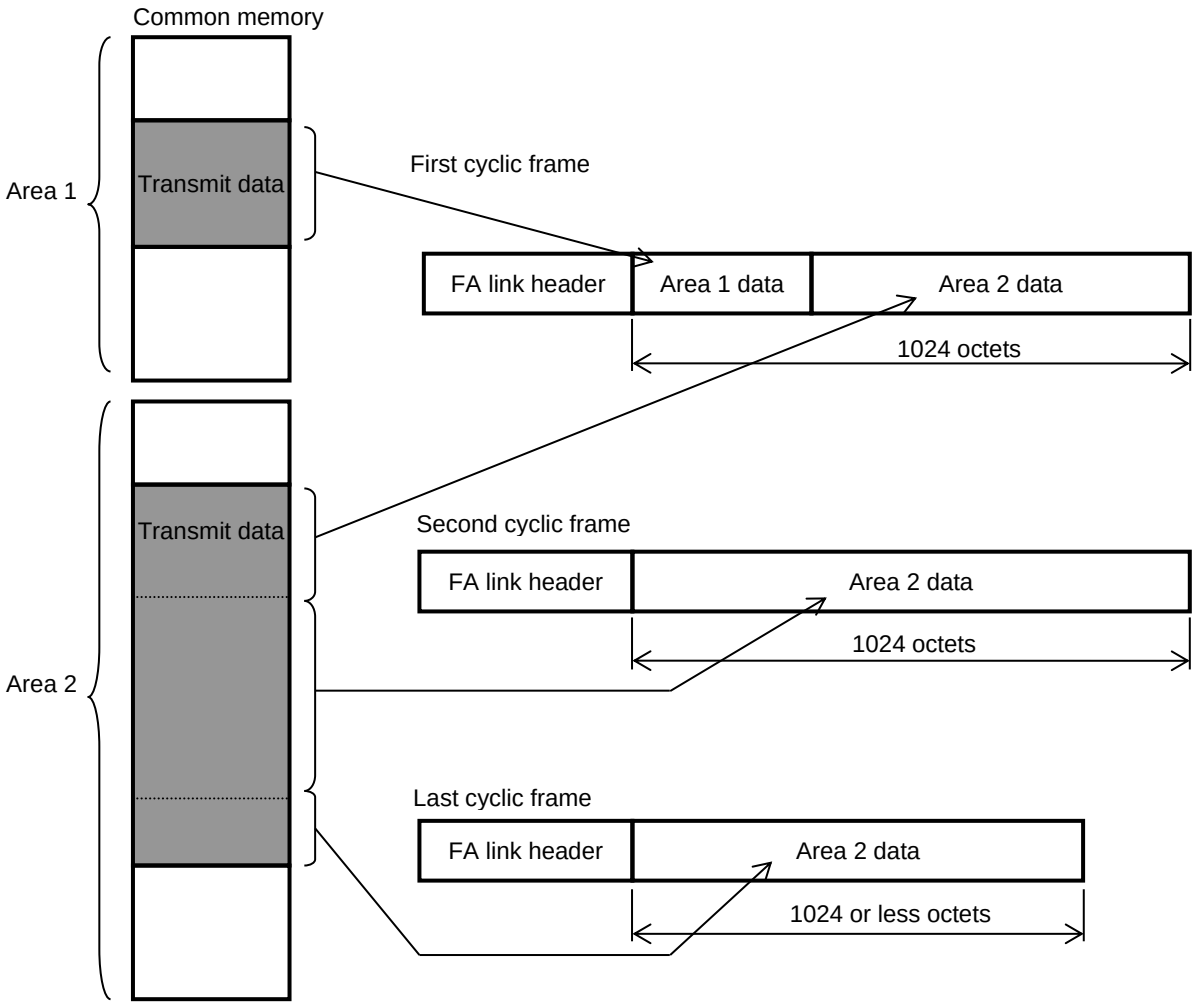


Figure 88 - Frame structure when the transmission data is larger than 1024 octets

The cyclic frames divided into two or more frames are continuously sent while the node is holding the token.

The divided cyclic frame is added with the information on TBN, CBN, and BSIZE. In addition, each cyclic frame shall be added with the top address and the size of the common memory area 1 and area 2.

The top address and the size of the area 1 and the area 2 are the same as those of each divided cyclic frames.

Division and assembly of cyclic frame shall be managed by TBN, CBN, and BSIZE.

A receiving node shall consider whether or not the cyclic frame is divided based on the information in the TBN in the header. The receiving node stores divided blocks in the buffer one by one. After all the divided cyclic frames are received, they are written in the common memory based on the initial address and the size of each of the area 1 and the area 2 in the

common memory. The node shall discard all the received data from a certain transmission node at that time when all the cyclic frames divided and sent from the node have not been received.

9.3.3 Message Transmission

Message transmission shall be as follows.

a) Outline

When a node received a token, it can transmit a maximum of 1 message frame before transmitting cyclic frames.

The amount of data which can be transmitted in one transmission shall be 1024 octets or less (excluding the header).

The message transmission function includes peer to peer transmission which specifies a destination node and 1 to n broadcast for all nodes. The 1 to 1 message transmission specifies the destination's node number as the node number of destination side, and the 1 to n message transmission sets the node number of destination side to "255" for transmission. The receiving node shall consider the message is on the 1 to n message transmission when the node number of destination side of the received message is "255".

The 1 to 1 message transmission has a function to confirm reception, and the 1 to n message transmission does not have a function to confirm reception.

Types of message transmission and the receipt confirmation function shall be shown in Table 19.

Table 19 - The list of the types of message transmission and the receipt confirmation function

Types of message	Receipt confirmation (ACK response)	Resending of message frames	Sequence number management
1 to 1 message transmission	Available	Available	Available
1 to n message transmission	Not available	Not available	Available

b) Timing of message transmission

A node can transmit a maximum of one message frame when it obtained a token before transmitting a cyclic frame based on the conditions indicated below.

1) When the refresh cycle measurement time (RMT) is 90% or less of the allowable refresh cycle time (RCT), one message frame can be transmitted.

2) When RMT is larger than 90 % of RCT and is smaller than RCT, one message frame can be transmitted on condition that a message was not transmitted last time.

3) When RMT is equal to RCT or more, a message frame is not transmitted.

c) Receipt Confirmation (ACK response)

The 1 to 1 message transmission has a function to confirm reception. This function shall be implemented with the ACK response. The receiving node returns an ACK to the transmission node in order to inform the transmission node of the message frame having been normally delivered to the receiving node. The ACK shall be transmitted in the frames for cyclic transmission. The sequence of receipt confirmation with the ACK shall be as shown in Figure 89.

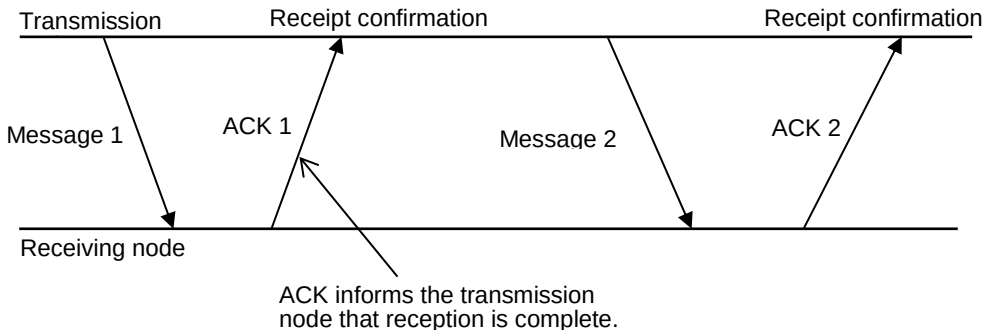


Figure 89 - The sequence of receipt confirmation with the ACK

d) Resending function

The transmission node shall resend a message frame when the status of the received ACK is abnormal after transmitting the 1 to 1 message frame and when the acknowledge waiting time (AWT) has passed and the ACK has not been received from the destination node after transmitting the message frame and before receiving a token addressed to the self node 3 times.

Resending is performed 3 times at most, and when the ACK has not been received from the destination node of the message frame after resending 3 times, failure of message transmission is informed to the upper layer.

e) Sequence Number Management

In the 1 to 1 message transmission, messages can be identified and managed by adding the sequence numbers to messages. Receiving nodes shall consider whether or not the message was resent with the sequence number.

The sequence numbers are managed with two numbers: the version of sequence number and the sequence number.

1) Management of the sequence number and the version of sequence number in transmission node

Each transmission node shall have one version of the sequence number. The version of the sequence number shall be created by each node by using time information, random numbers or random data equivalent to it every time the node starts.

After this, each node shall store and transmit the value of the version of sequence number in the header of a message frame when transmitting the message.

The sequence number is set in the header of a message frame when transmitting a message. The sequence number is initialized at start up, and is incremented from the initial value "1." The value of the sequence number is incremented from 1, but 16#FFFF_FFFF shall be followed by 1.

The sequence number shall be incremented when transmission is normally completed, in other words, when the ACK is normally received and the transmitted message frame is considered to have been normally received by the receiving node, and when the ACK did not return when the message was transmitted or when resending failed three times after resending in case the status of the ACK are abnormal.

2) Confirmation of the sequence number and the version of sequence number in the receiving node

The receiving node shall manage the version of sequence number and the sequence number for each node. Each node shall initialize all of that management information to "0" before joining the network.

The receiving node shall register the value as the version of sequence number when the

version of sequence number in the header of the received message differs from the version of sequence number of the corresponding node. At this time, the message frame is discarded and the error of the version of sequence number is returned as the status of the ACK. When the managed version of the sequence number is the initial value"0", the version of sequence number in the header of the received message is not checked and the message frame shall become valid and the value shall be registered as the version of sequence number.

Then, when the sequence number in the header of the received message is the same as the sequence number information in the corresponding node, the receiving node identifies it as a resent message, and the received data is considered to have been processed and is discarded, but when the message is 1 to 1, the status of the ACK shall be transmitted as normal reception.

When the sequence number in the header of the message differs from one managed by the receiving node, the message is identified as a new message and the message data is taken in, and the sequence number shall be registered.

9.3.4 Token and Data

The frames transmitted when the node is holding a token include the following 9 types (refer to Figure 90).

a) When there is no cyclic data or message data and all the slaves allocated to flexible master have participated, after transmitting a cyclic frame without slave data, the token frame shall be transmitted.

b) When transmitting cyclic data and when all the slaves allocated to flexible master have participated, after transmitting the cyclic frame, the token frame shall be transmitted.

c) When dividing and transmitting cyclic data and when all the slaves allocated to flexible master have participated, the cyclic frame is divided for transmission, and after transmitting the last cyclic frame, the token frame shall be transmitted.

d) When the alternate transmission order indicates a message and only message data exist, message frames are transmitted, and after transmitting cyclic frames without data, the token frame shall be transmitted.

e) When the alternate transmission order indicates a message and cyclic data and message data exist, message frames are transmitted, and after transmitting the cyclic frame, the token frame shall be transmitted.

f) When the alternate transmission order indicates a message, and divided cyclic data and message data exist, message frames are transmitted, then, cyclic data is divided for transmission and after transmitting the last cyclic frame, the token frame shall be transmitted.

g) When the alternate transmission order indicates a solicitation frame, and an unparticipated slave exists, a solicitation frame is transmitted, then, after transmitting a cyclic frame without data, the token frame shall be transmitted.

h) When the alternate transmission order indicates a solicitation frame, an unparticipated slave exists, and cyclic data exists, a solicitation frame is transmitted, after transmitting the cyclic frame, the token frame shall be transmitted.

i) When the alternate transmission order indicates a solicitation frame, an unparticipated slave exists, and divided cyclic data exists, a solicitation frame is transmitted, then, cyclic data is divided for transmission, and after transmitting the last cyclic frame, the token frame shall be transmitted.

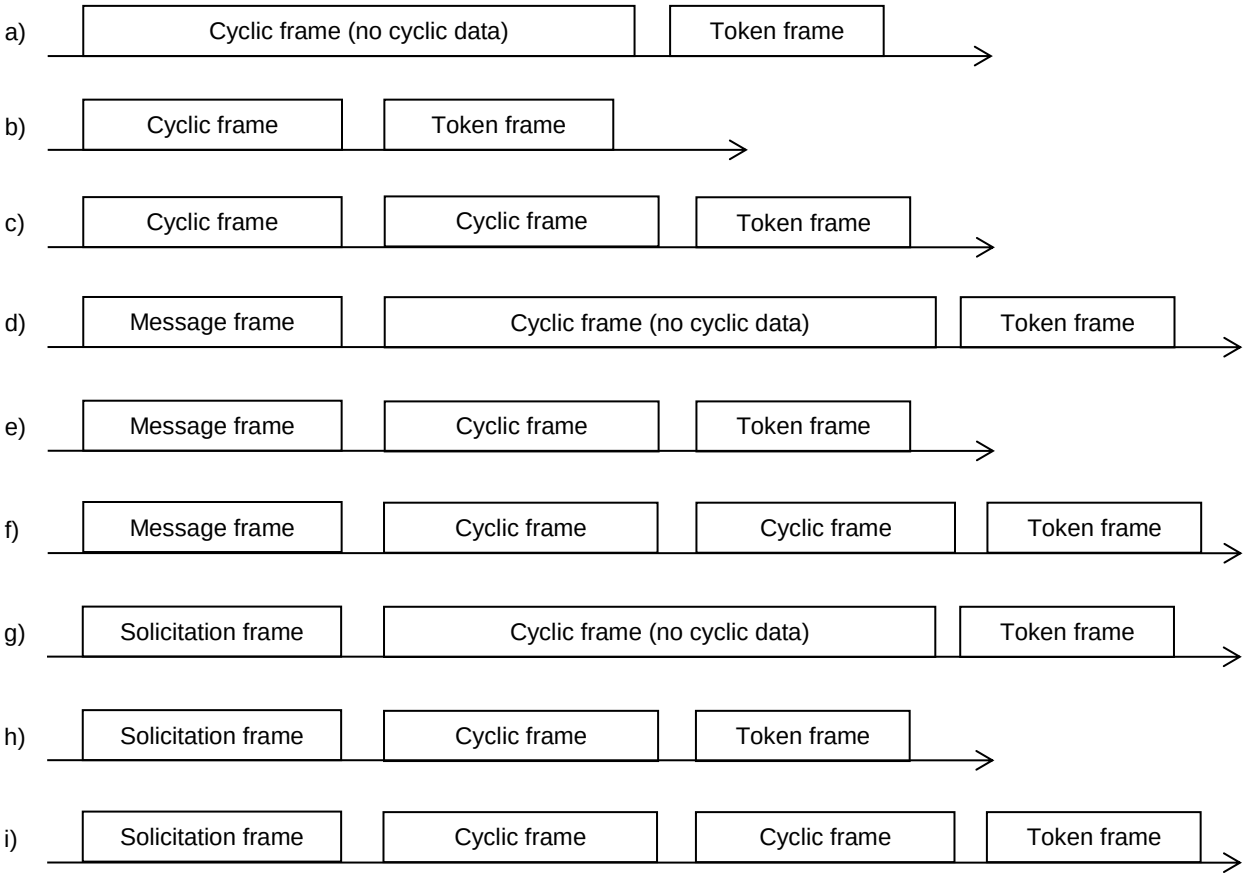


Figure 90 - Token and data

9.4 Device Level Network

9.4.1 Startup Operation

Start up operation shall be as follows.

a) Master

Start up operation of the master shall be as follows.

1) Fixed setting mode

Participates in the network with the start up procedure shown in 9.2.

2) Flexible setting mode

When all the slaves in accordance with the IO definition have not participated in the network, the solicitation frames are transmitted by dividing when holding a token.

When all other nodes are flexible slaves at start up, the token does not circulate since a link is not established. Therefore, solicitation frames are periodically batch transferred (when the participation request frame acceptance time has passed after participation request frames were transmitted).

When all the slaves in accordance with the IO definition participate in the network, transmission of the solicitation frames shall be suspended, and when a slave is detected to have been removed, transmission of the solicitation frames shall be resumed.

b) Slave

Start up operation of a slave shall be as follows.

1) Fixed setting mode

Participates in the network with the start up procedure shown in 9.2.

2) Flexible setting mode

Start up by waiting for a solicitation frame.

When the IO definition addressed to the self node is detected when receiving the solicitation frame, the slave shall set the common memory in accordance with the IO definition, and participate in the network with the start up procedure shown in 9.2. The master node number shall be identified in this process.

When a solicitation frame is received from the same master again after completing setting, only when the contents have been changed, the status for waiting for the joining token detection time begins after the area is redefined. When a solicitation frame is received from a different master, the solicitation frame shall be discarded. The node shall be reset or the power supply shall be turned on again to reflect the definition from a different master.

9.4.2 Operation while Establishing a Link

Operation while establishing a link shall be as follows.

a) Master

Operation while establishing a link of the master shall be as follows.

1) Fixed setting mode

Output to the slave shall be performed with cyclic frame transmission and input shall be received with cyclic frames from a slave.

2) Flexible setting mode

Output to the slave shall be performed with cyclic frame transmission and input shall be received with cyclic frames from a slave.

When all the slaves in accordance with the IO definition have not participated in the network, the solicitation frames shall be transmitted when holding a token while performing cyclic frame transmission. For conditions of transmitting solicitation frames, refer to 7.7.

When all the slaves participate in the network, the solicitation frames are not transmitted.

When disengagement of slaves is monitored, and there is no participating slave, the solicitation frames shall be transmitted when holding a token while performing cyclic frame transmission.

Note Monitoring of slave disengagement is considered from participating node management information in the master node. Specifically, whether the numbers and the common memory area of the node participating in the network are within the expected range is checked to monitor disengagement.

b) Slave

Operation while establishing a link of a slave shall be as follows.

1) Fixed setting mode

The output to the master shall be performed with cyclic frame transmission and the input shall be received with cyclic frames from a master. When the master has removed, the output shall be held or cleared in accordance with the slave operating instructions.

2) Flexible setting mode

The output to the master shall be performed with cyclic frame transmission and the input shall be received with cyclic frames from a master. When the master has removed, the output shall be held or cleared in accordance with the slave operating instructions.

After the master has removed, the slave waits the solicitation frames. When the master is

reconnected, the slave checks the contents of the received solicitation frame and participates in the network under the in-ring startup state.

9.4.3 Operation with the Multi Master

Operation with the multi master shall be as follows.

a) Master

Operation of the master shall be as follows when the plural masters participate in the network.

1) Flexible setting mode

The participation of the slave which the self node manages shall be confirmed with the reception the cyclic frames and the master node number of the status areas in cyclic frame from the slave.

When the master node number is other node number than the self node, the slave shall be considerd to be duplicated.

b) Slave

Operation of the slave shall be as follows when the plural masters participate in the network.

1) Flexible setting mode

Overlap check of the area of the common memory, allocation, and identification of the master station number shall be performed according to the solicitation frame received first.

When different solicitation frames are received from the same master, only when the contents have been changed, the status for waiting for the joining token detection time begins after redefinition.

When solicitation frames from a different master have been received, they shall be discarded.

Note The same master shall be distinguished in SA of the solicitation frame header format. The solicitation frame header format shall be described in 10.4.

9.4.4 Operation at Disengagement

Disengagement management shall be based on 9.2.

The master node checks a node number of source side each time it receives a token frame, and when the token frame from the same node is not received 3 times in succession, the node shall be considered to have been removed.

When the node is considered to have been removed, the status of the FA link within participating node management information shall be changed to the status in which the node has removed. When other nodes left during token circulation, and only the self node is left, the process shall shift to the joining sequence

The node handling IO input and output performs the following operation.

a) Master node

Operation when the master removes shall be as follows.

1) Flexible setting mode

When the removed node is a slave managed by the self node, the solicitation frame transmission waiting time (SFTWT) shall be started.

Solicitation frames are transmitted when the SFTWT times out.

b) Slave node

Operation when the slave node removes shall be as follows.

1) Flexible setting mode

The slave node removes by monitoring a token addressed to the self node, and the process shall shift to the joining sequence. When connection is restored, an in-progress participating sequence shall be performed. When a solicitation frame is received, no process shall be performed.

When it is reset with a power stoppage, etc., the state for waiting for a solicitation frame starts, a solicitation frame is received, and the in-progress participating sequence shall be performed.

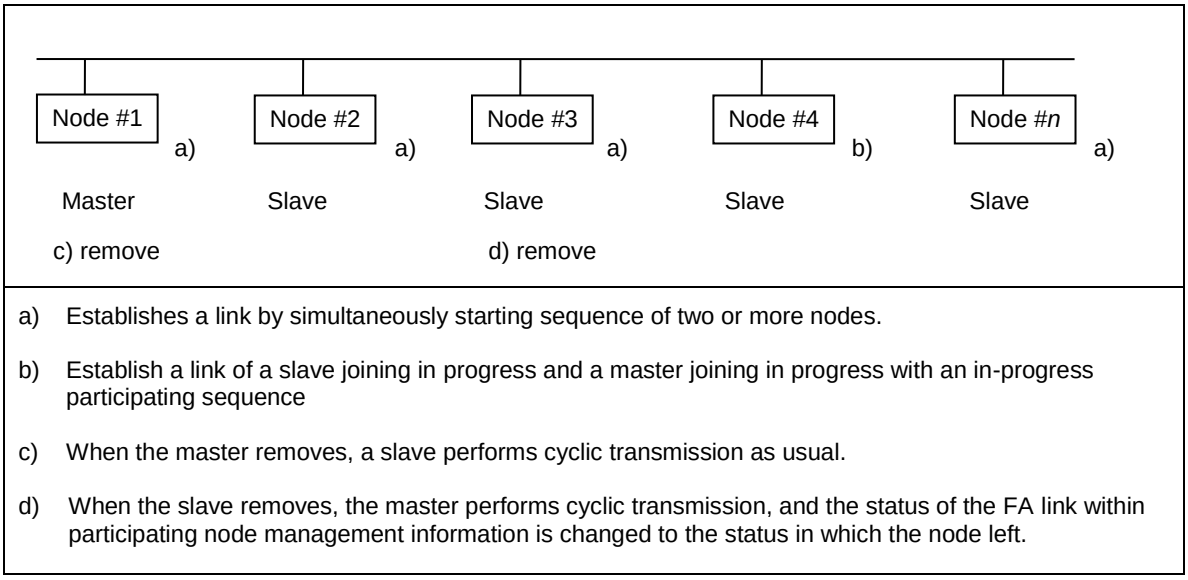
9.4.5 Link Establishment Operation

Link establishment operation is as follows.

a) Fixed setting mode

The joining sequence of the fixed setting mode is the same as that of Ver.2.00.

The flow of the joining sequence shall be shown in Figure 91.



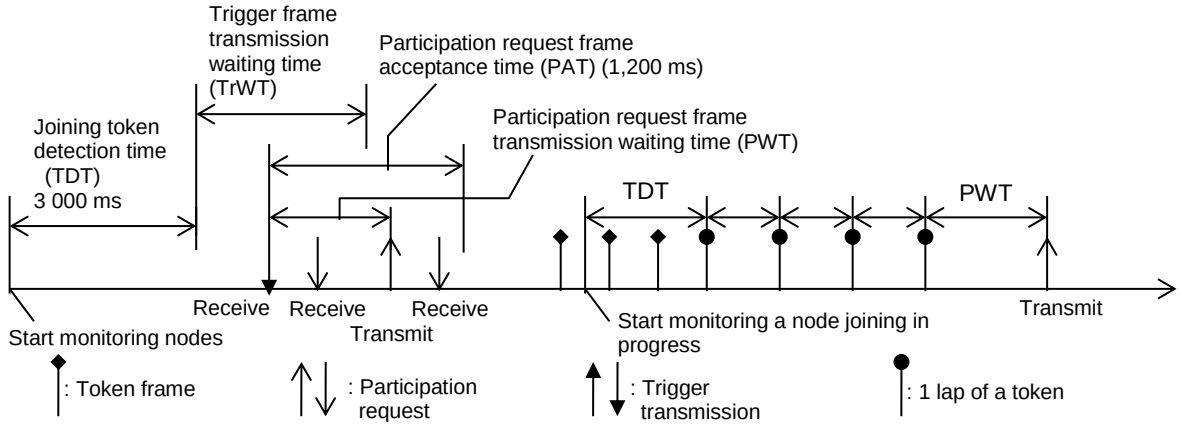
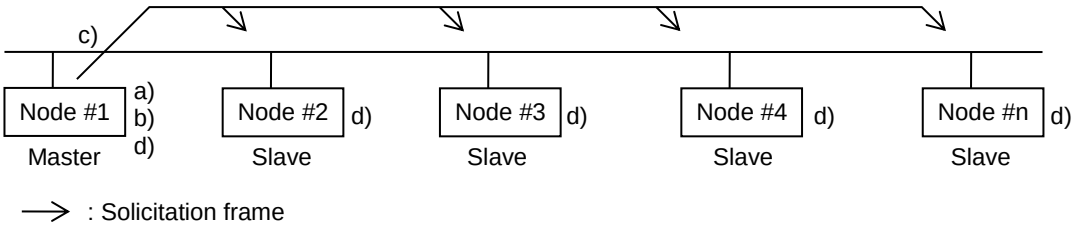


Figure 91 - Joining sequence of the fixed setting mode

b) Flexible setting mode (When there is one flexible master)

Flexible master shall transmit a solicitation frame to a slave after ending the joining sequence. Flexible slave shall perform the joining sequence after receiving the solicitation frame from flexible master. For details of the joining sequence, refer to 9.2. The flow of the joining sequence shall be as shown in Figure 92.



a) Without a token

b) The participation request frame acceptance time times out after a participation request frame is transmitted

c) Transmit a solicitation frame

d) All nodes simultaneously establish links with a link participating sequence

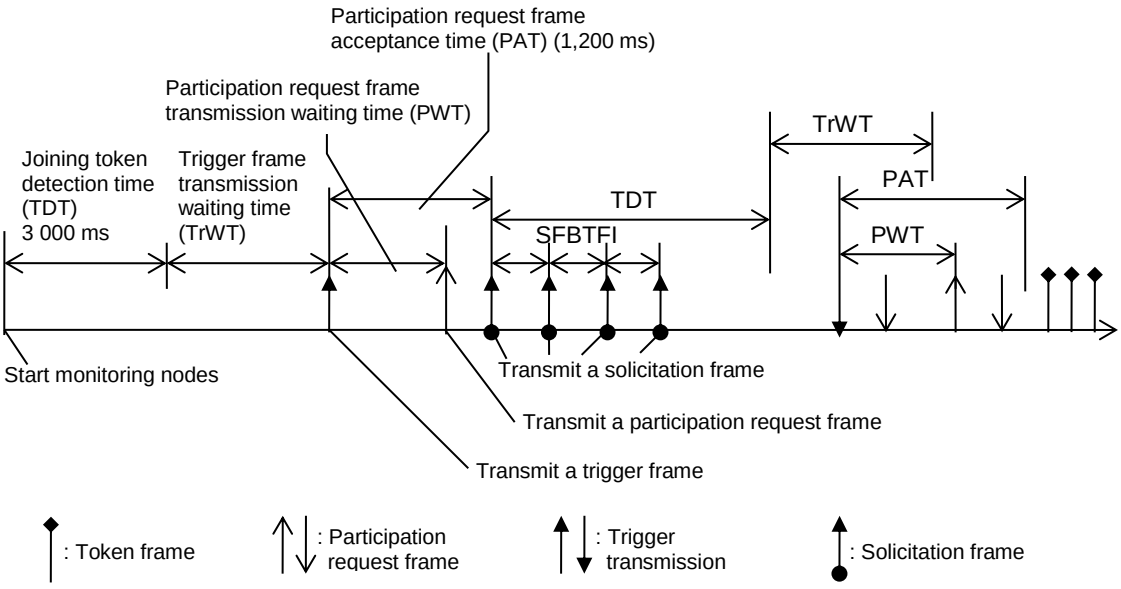
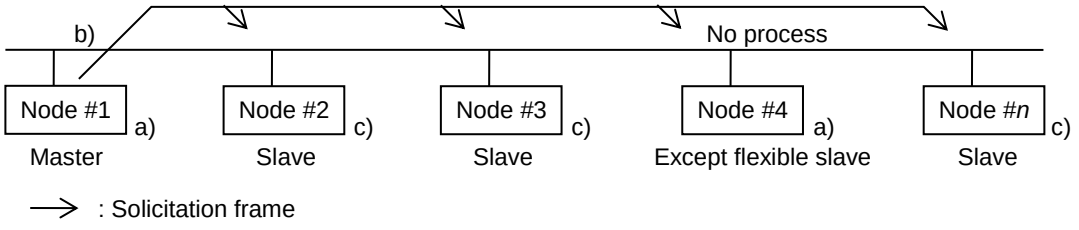


Figure 92 - The joining sequence of the flexible setting mode (when there is one flexible master)

c) Flexible setting mode (when there is one flexible master and nodes other than flexible slave coexist)

Flexible master shall transmit a solicitation frame to a slave after ending the joining sequence. Flexible slave shall perform the joining sequence after receiving the solicitation frame from flexible master. Nodes other than flexible slave shall discard solicitation frames. For details of the joining sequence, refer to 9.2. The flow of the joining sequence shall be as shown in Figure 93.



a) Establish a link with the simultaneous starting sequence of the node #1 and the node #4.

b) Transmit solicitation frames when the node #1 is holding the token. Transmit solicitation frames when the SFTWT is up until all the slaves have joined.

c) A slave establishes a link with an in-progress participating sequence.

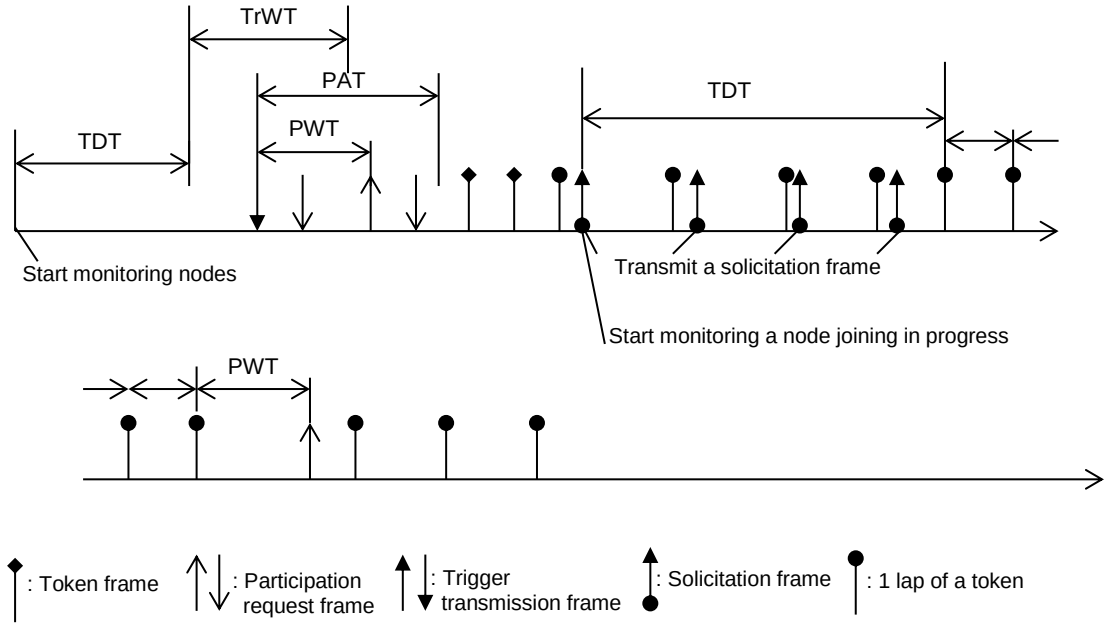
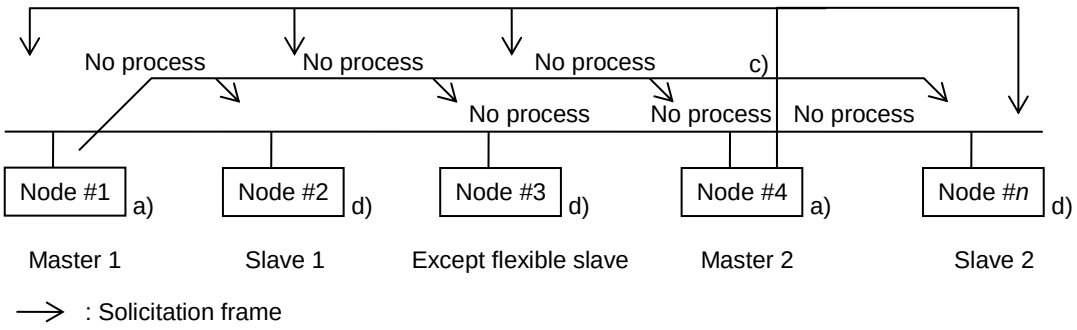


Figure 93 - The joining sequence of the flexible setting mode (when there is one flexible master and nodes other than flexible slave coexist)

d) Flexible setting mode (in the case of two or more flexible masters)

Flexible master shall transmit a solicitation frame to a slave after ending the joining sequence. Flexible slave shall perform the joining sequence after receiving the solicitation frame from flexible master. Nodes other than flexible slave shall discard solicitation frames. Flexible master shall discard solicitation frames from other flexible masters. For details of the joining sequence, refer to 9.2. The flow of the joining sequence shall be as shown in Figure 94.



a) Establish a link with a simultaneous starting sequence of the node #1, the node #3, and the node #4.

b) Transmit solicitation frames when the node #1 is holding the token. Transmit solicitation frames when the SFTWT times out until all the slaves join.

c) Transmit solicitation frames when the node #4 is holding the token. Transmit solicitation frames when the SFTWT times out until all the slaves join.

d) A slave establishes a link with an in-progress participating sequence.

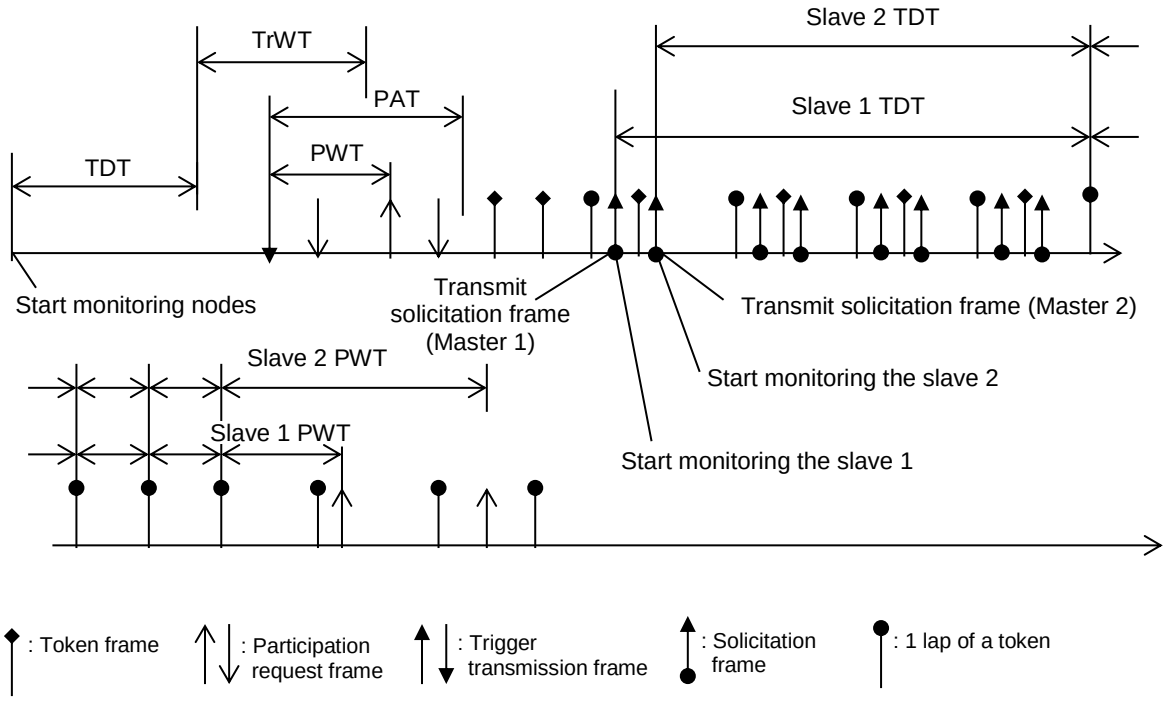
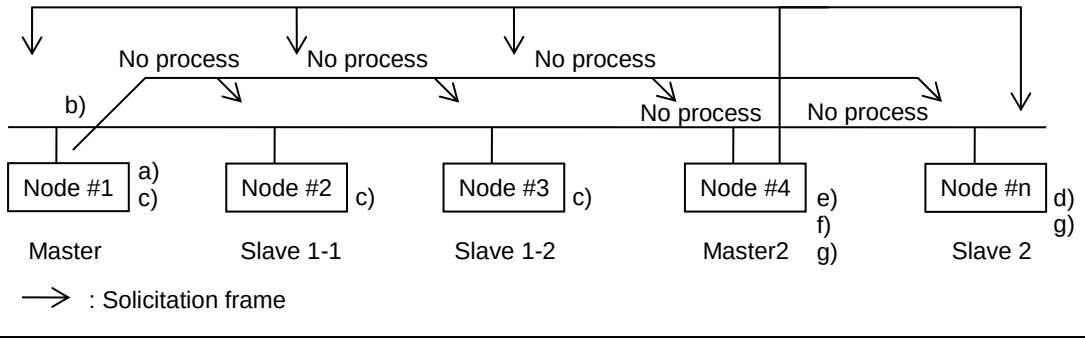


Figure 94 - The joining sequence of the flexible setting mode (when there are two or more flexible masters)

e) Flexible master and flexible slave participating in progress while establishing a link

Flexible master shall transmit solicitation frames to the slave after ending the joining sequence. Flexible slave shall perform the joining sequence after receiving the solicitation frame from flexible master.

Nodes other than flexible slave shall discard solicitation frames. For details of the joining sequence, refer to 9.2. The flow of the joining sequence shall be shown in Figure 95.



a) Without a token

b) Transmit solicitation frame when the participation request frame acceptance time times out after transmitting participation request frames.

c) The master 1, the slave 1-1, and the slave 1-2 nodes shall simultaneously establish links with a link participating sequence.

d) Connection of the slave 2 starts the status to wait for in-progress participation.

e) Establish a link with the in-progress participating sequence when the master 2 is connected.

f) Transmit solicitation frames when the master 2 is holding a token.

g) A slave shall establish a link with an in-progress participating sequence.

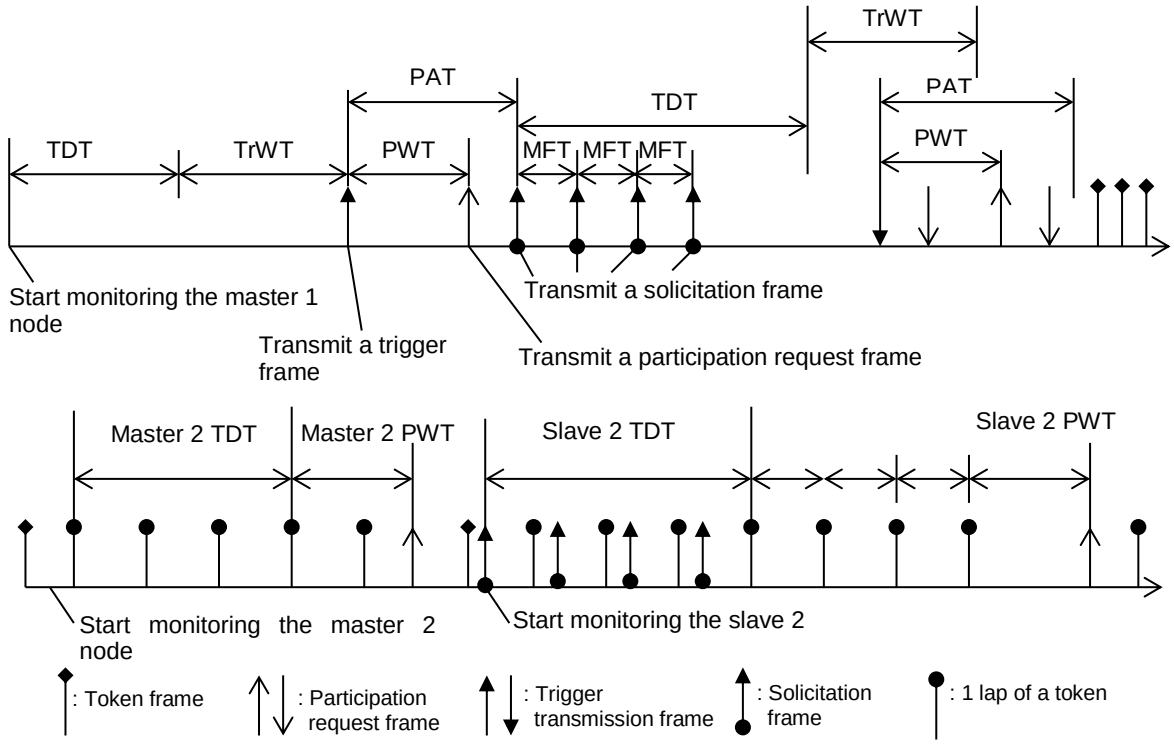


Figure 95 - Joining sequence of flexible master and flexible slave participating in progress while establishing a link

9.4.6 Operation in Abnormal Setting of IO Allocation of any Slave

Abnormalities in the IO allocation of flexible slave include two cases: when flexible master detects it due to wrong settings of flexible master with a setting tool etc. and when the slave detects it due to wrong settings with the solicitation frame from flexible master.

a) The abnormalities in IO allocation

The abnormalities in IO allocation shall be as follows.

1) Abnormal slave node number

Flexible master sets the IO allocation response to be "Abnormal slave node number", discards the IO allocation information, and do not transmit a solicitation frame when the setting value of the slave node number part is not in the range from 1 to 249 when the IO allocation setting frame is received.

Flexible slave shall discard the IO allocation information about the slave when the slave node number out of range is detected when solicitation frames are received.

2) Abnormal address

Flexible master sets the IO allocation response to be "Abnormal slave common memory address", discards the IO allocation information, and do not transmit a solicitation frame when the common memory address value of flexible slave is not in the range of the common memory area when then IO allocation setting frame is received. Flexible slave does not participate in the network when the common memory address exceeds the range when a solicitation frame is received.

3) Abnormal size

Flexible master sets the IO allocation response to be "Abnormal slave common memory size", discards the IO allocation information, and do not transmit a solicitation frame when the common memory size of flexible slave is not in the range of the common memory area when the IO allocation setting frame is received.

Flexible slave does not participate in the network when the common memory size exceeds the range when a solicitation frame is received.

b) Receiving solicitation frames

When the solicitation frame addressed to the self node is received by a node other than flexible slave, it shall be discarded.

c) Overlap of solicitation frames

When flexible slave received solicitation frames from two or more masters, the IO allocation information received first is used.

d) Double definition of a slave

When the solicitation frame of one master has two definitions of the same slave, the second IO allocation information is used.

10 FA Link Frame Format

10.1 Header Format

An FA link header has the size from 64 to 96 octets. The configuration of the FA link frame shall be as shown in Figure 96.

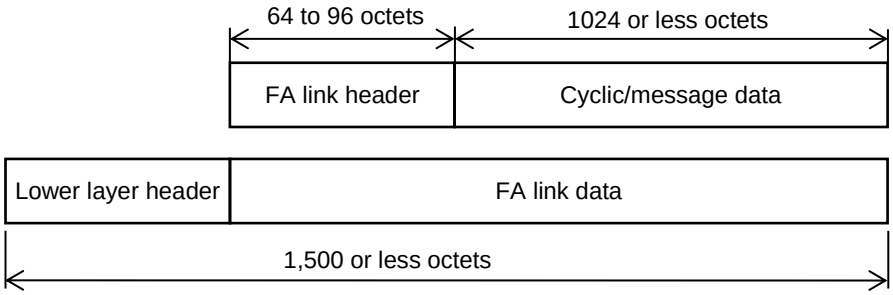


Figure 96 - The configuration of the FA link frame

The basic structure of the FA link header shall be as follows. The FA link header is added to the beginning of all the frames in the FA link protocol. In addition, the address of the header is a data position flowing in the network. The bit positions are also written in the order to flow in the network in the same manner.

10.1.1 The Basic Structure of the FA Link Header

The basic structure of the FA link header shall be shown in Figure 97, and the base items in Table 20.

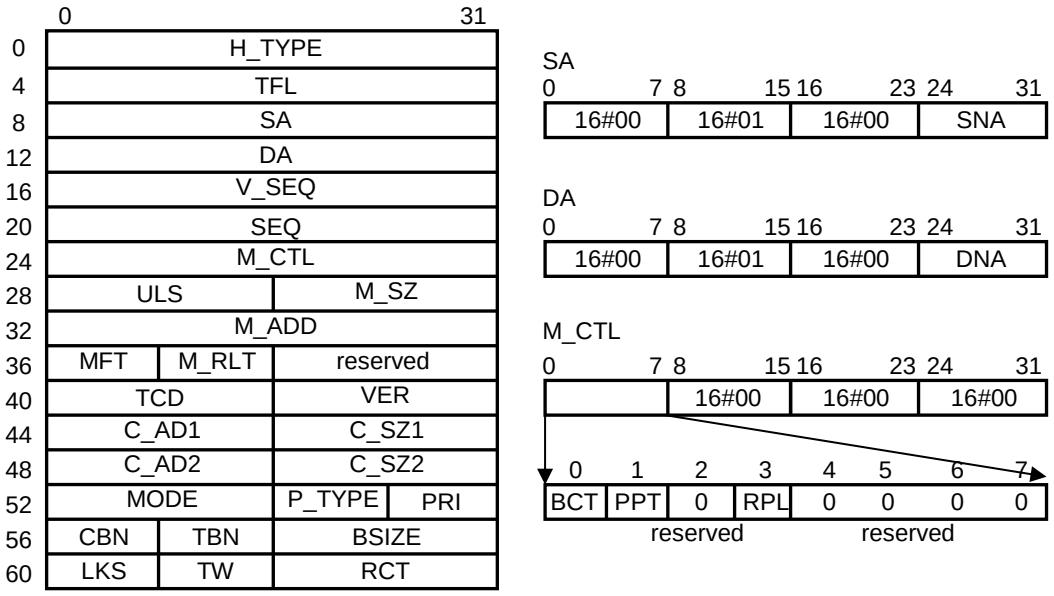


Figure 97 - Basic structure of the FA link header

Table 20 - The base items of the FA link header

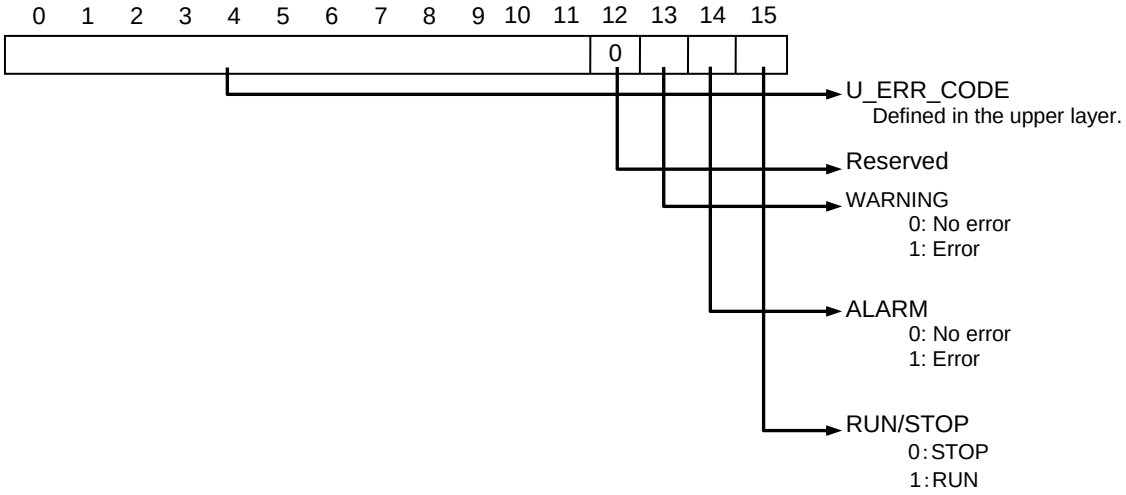
Symbol	Size (Octet)	Items	Setting value
H_TYPE	4	Header type	FACN: FL-net
TFL	4	Octet length including the header and data	Octet lengths shall be stored in the 2 octets of 16-31, and 0-15 shall be 16#0000.
SNA	1	Node number of source side	―
DNA	1	Node number of destination side	―
V_SEQ	4	Version of sequence number	―
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	―
BCT	1 bit	Broadcast transmission	―
PPT	1 bit	Peer to peer transmission	―
RPL	1 bit	Existence of ACK data	―
ULS	2	Upper layer status (RUN/STOP/ALARM/WARNING/NORMAL)	―
M_SZ	2	Message data size in virtual address space	―
M_ADD	4	Offset address of the virtual address space	―
MFT	1	Allowable minimum frame interval time	―
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	―
VER	2	Program version	- Fixed to 16#0000
C_AD1	2	Common area 1 datatop address	―
C_SZ1	2	Common memory area 1 data size	―
C_AD2	2	Common area 2 datatop address	―
C_SZ2	2	Common memory area 2 data size	―
MODE	2	FA link protocol version (major version, minor version), token mode	―
P_TYPE	1	Protocol type	- Fixed to 16#80
PRI	1	Message priority	- Fixed to 16#00
CBN	1	Current fragment block number	―
TBN	1	Total fragment block number	―
BSIZE	2	Current block size (octet)	―
LKS	1	Link status (FA link status)	―
TW	1	Token watchdog time	―
RCT	2	Allowable refresh cycle time	―

The basic structure of the FA link header shall be as follows.

a) ULS

The data of ULS indicates the upper layer status and includes 1 bit of RUN/STOP, Reserved, WARNING, ALARM, and a 12-bit code (refer to Figure 98).

The upper layer status is defined with 16 bits.

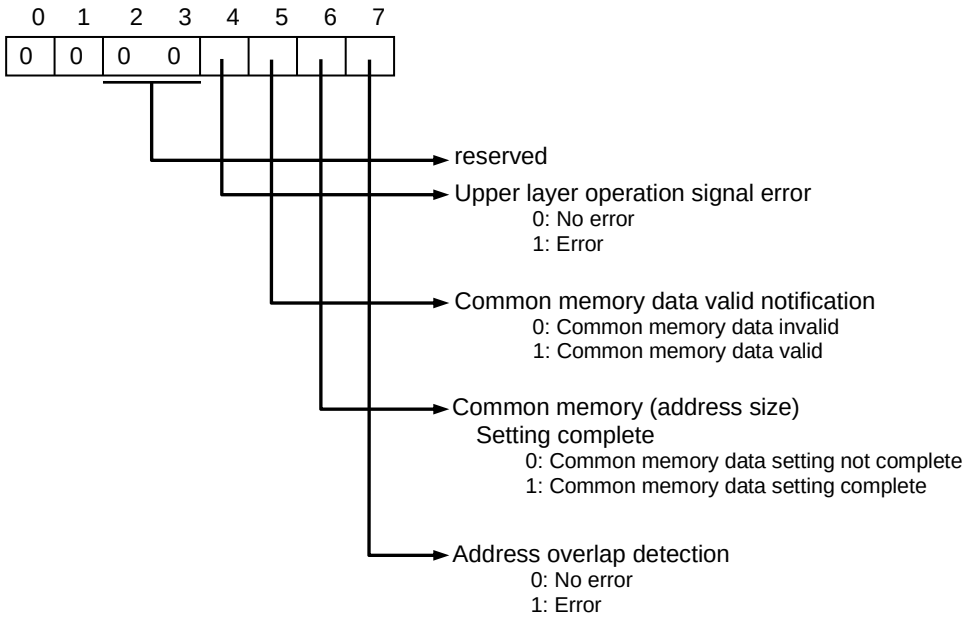


Note The definition of U_ERR_CODE (upper layer error code) is determined by the vendor. Refer to 7.3.6.

Figure 98 - Data of ULS

b) LKS

The data of LKS indicates the status of the network FA link (refer to Figure 99). The status of the link is defined with 8 bits.



Note Refer to 7.3.6.

Figure 99 - Data of LKS

c) TCD

The list of TCD shall be shown in Table 21.

Table 21 – List of TDC

	Transaction codes	Frames
1	0 to 9 999	(Reserved)
2	10 000 to 59 999	Transparent message frame
3	60 000 to 64 999	(Reserved)
4	65 000	Token frame
5	65 001	Cyclic frame
6	65 002	Participation request frame
7	65 003	Byte block read frame (request)
8	65 004	Byte block write frame (request)
9	65 005	Word block read frame (request)
10	65 006	Word block write frame (request)
11	65 007	Network parameter read frame (request)
12	65 008	Network parameter write frame (request)
13	65 009	Stop command frame (request)
14	65 010	Operation command frame (request)
15	65 011	Profile read frame (request)
16	65 012	Trigger frame
17	65 013	Log data read frame (request)
18	65 014	Log data clear frame (request)
19	65 015	Message return frame (request)
20	65 016	Vendor specific message frame (request)
21	65 017	Solicitation frame (broadcast)
22	65 018	IO allocation setting frame (request)
23	65 019	IO allocation read frame (request)
24	65 020	Start frame of token retention time measurement (request)
25	65 021	End frame of token retention time measurement (request)
26	65 022	Start frame of measurement in general purpose communication data sender log (request)
27	65 023	End frame of measurement in general purpose communication data sender log (request)
28	65 024	Configuration parameter setting frame (request)
29	65 025	Participating node management information parameter read frame (request)
30	65 026	Read frame of self node management information parameter (request)
31	65 027	Self node setting information parameter read frame (request)
32	65 028	Node reset frame (request)
33	65 029 to 65 202	(Reserved) (for future extension)
34	65 203	Byte block read frame (response)
35	65 204	Byte block write frame (response)
36	65 205	Word block read frame (response)
37	65 206	Word block write frame (response)
38	65 207	Network parameter read frame (response)
39	65 208	Network parameter write frame (response)

Table 21 – List of TDC (continued)

	Transaction codes	Frames
40	65 209	Stop command frame (response)
41	65 210	Operation command frame (response)
42	65 211	Profile read frame (response)
43	65 212	(Reserved)
44	65 213	Log data read frame (response)
45	65 214	Log data clear frame (response)
46	65 215	Message return frame (response)
47	65 216	Vendor specific message frame (response)
48	65 217	(Reserved)
49	65 218	IO allocation setting frame (response)
50	65 219	IO allocation read frame (response)
51	65 220	Start frame of token retention time measurement (response)
52	65 221	End frame of token retention time measurement (response)
53	65 222	Start frame of measurement in general purpose communication data sender log (response)
54	65 223	End frame of measurement in general purpose communication data sender log (response)
55	65 224	Configuration parameter setting frame (response)
56	65 225	Read frame of participating node management information parameter (response)
57	65 226	Read frame of self node management information parameter (response)
58	65 227	Self node setting information parameter read frame (response)
59	65 228	Node reset frame (response)
60	65 229 to 65 399	(Reserved) (for future extension)
61	65 400 to 65 535	(Reserved)

d) TFL

The data of the TFL indicates the octet length including the header and the data part before dividing each frame.

Participation request frame: Indicates the octet length of the header 16#0000_0060(96)

Trigger frame: Indicates the octet length of the header 16#0000_0060(96)

Token frame: Indicates the octet length of the header 16#0000_0040(64)

Message frame: Indicates the octet length including the header and data part

Cyclic frame: Indicates the octet length including the header and the data part before being divided [(the size of the area 1 + the size of the area 2) x 2].

TFL includes the data length of ACK.

e) M_RLT

M_RLT at a response frame shall be as follows.

1) 16#00: Normal response

2) 16#01: Abnormal response

Indicates that the parameter of the message has a certain error although the requested message is implemented. At this time, an error code shall be set to the data part. This code can be arbitrarily defined by a vendor.

3) 16#02: Unimplemented response

Indicates the processing function for the requested message is not implemented. At this time, the size of the data part shall be 0.

f) BSIZE

BSIZE indicates the data size including the header of each frame. BSIZE shall include the data length of an ACK. When it is not divided, the value of TFL and BSIZE shall be the same (as is the case with an ACK).

g) SNA

SNA shall set the node number of the source side in all the frames.

h) DNA

DNA of each frame shall set the following values:

Token frame: The node number of the destination to which to pass a token
Cyclic frame: The node number of the destination to which to pass a token
Participation request frame and trigger frame: 16#FF
1 to 1 message: Node number of destination side
1 to n message: 16#FF

i) MODE

The data of MODE indicates the FA link protocol version (the major version and the minor version), and the status of the token mode (refer to Figure 100).

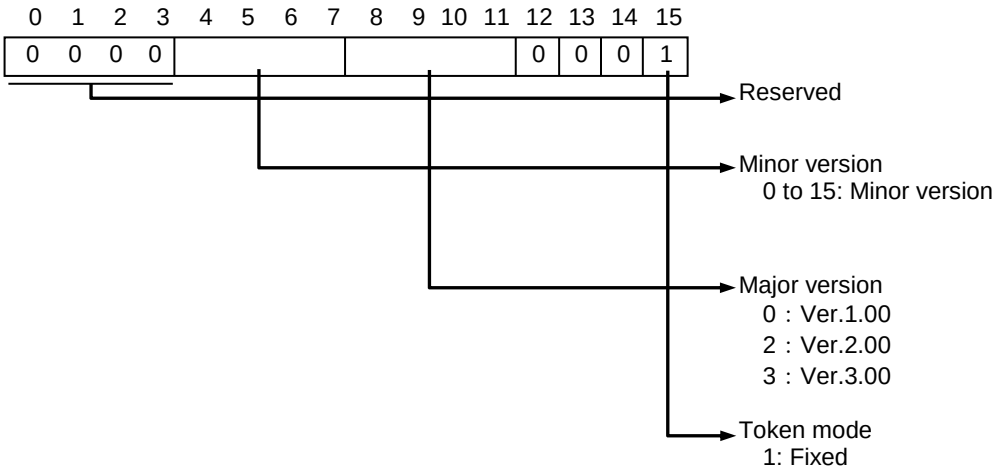


Figure 100 - Data of MODE

10.1.2 Token Header

The format of the token header shall be shown in Figure 101 and the items of the token header shall be shown in Table 22.

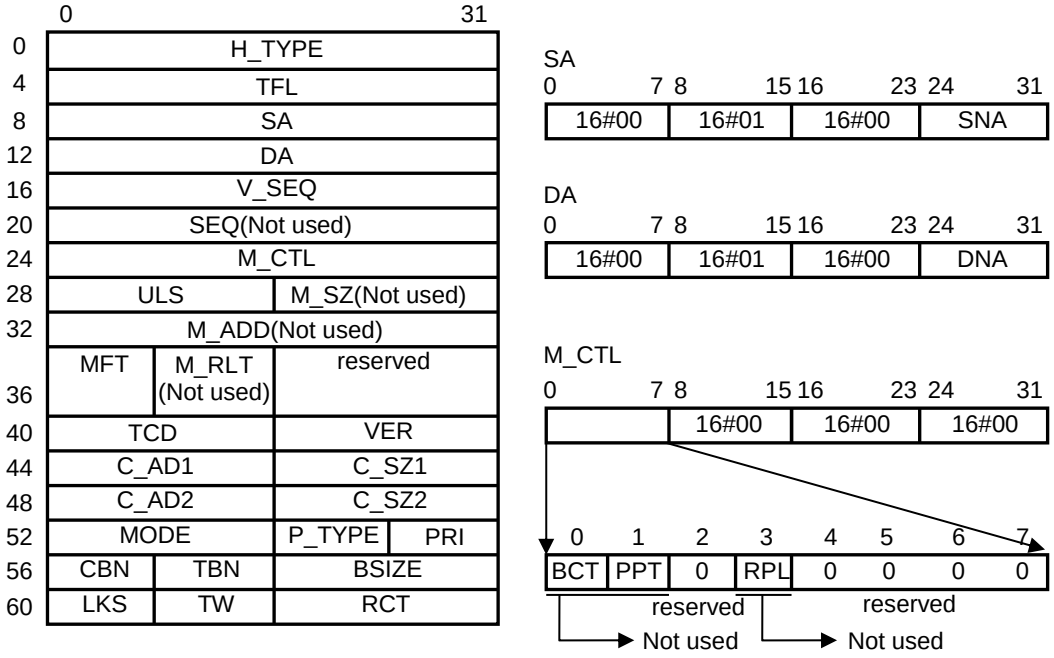


Figure 101 - The format of the token header

Table 22 - The items of a token header

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	FACN: FL-net
TFL	4	Octet length including the header and data	Fixed to 16#0000_0040
SNA	1	Node number of source side	–
DNA	1	Node number of destination side	–
V_SEQ	4	Version of sequence number	–
ULS	2	Upper layer status (RUN/STOP/ALARM/WARNING/NORMAL)	–
MFT	1	Allowable minimum frame interval time	–
TCD	2	Transaction code	Fixed to 16#FDE8
VER	2	Program version	Fixed to 16#0000
C_AD1	2	Common area 1 datatop address	–
C_SZ1	2	Common memory area 1 data size	–
C_AD2	2	Common area 2 datatop address	–
C_SZ2	2	Common memory area 2 data size	–
MODE	2	FA link protocol version (major version, minor version), token mode	–
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0040
LKS	1	Link status (FA link status)	–
TW	1	Token watchdog time	–
RCT	2	Allowable refresh cycle time	–

10.1.3 Cyclic Header

When there is an ACK to transmit, RPL flag shall be set and the ACK data to be mentioned later shall be added between the header and the cyclic data.

The format of a cyclic header shall be shown in Figure 102, and the items of the cyclic header shall be shown in Table 23.

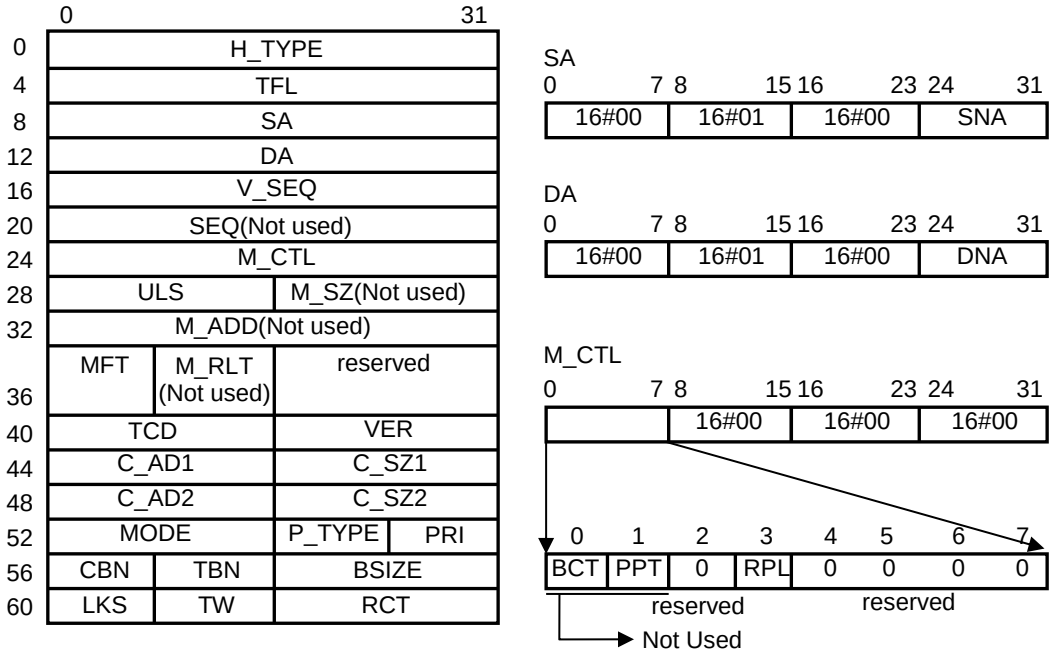


Figure 102 - The format of a cyclic header

Table 23 - The items of a cyclic header

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	FACN: FL-net
TFL	4	Octet length including the header and data	Octet lengths shall be stored in the 2 octets of 16 to 31, and 0 to 15 shall be 16#0000.
SNA	1	Node number of source side	—
DNA	1	Node number of destination side	—
V_SEQ	4	Version of sequence number	—
RPL	1 bit	Existence of ACK data	When there is ACK data: 1
ULS	2	Upper layer status (RUN/STOP/ALARM/WARNING/NORMAL)	—
MFT	1	Allowable minimum frame interval time	—
TCD	2	Transaction code	Fixed to 16#FDE9
VER	2	Program version	Fixed to 16#0000
C_AD1	2	Common area 1 datatop address	—
C_SZ1	2	Common memory area 1 data size	—
C_AD2	2	Common area 2 datatop address	—
C_SZ2	2	Common memory area 2 data size	—
MODE	2	FA link protocol version (major version, minor version), token mode	—

Table 23 - The items of a cyclic header (Continued)

Symbols	Size (Octet)	Items	Setting values
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	–
TBN	1	Total fragment block number	–
BSIZE	2	Current block size (octet)	–
LKS	1	Link status (FA link status)	–
TW	1	Token watchdog time	–
RCT	2	Allowable refresh cycle time	–

10.1.4 Message Header

The message header shall be as follows.

a) In the case of a message that uses the virtual address space

The virtual address space is used in the byte block read/write service and in the word block read/write service among the services of the message. The format of the message header using the virtual address space shall be shown in Figure 103 and the items of the message header using the virtual address space shall be shown in Table 24.

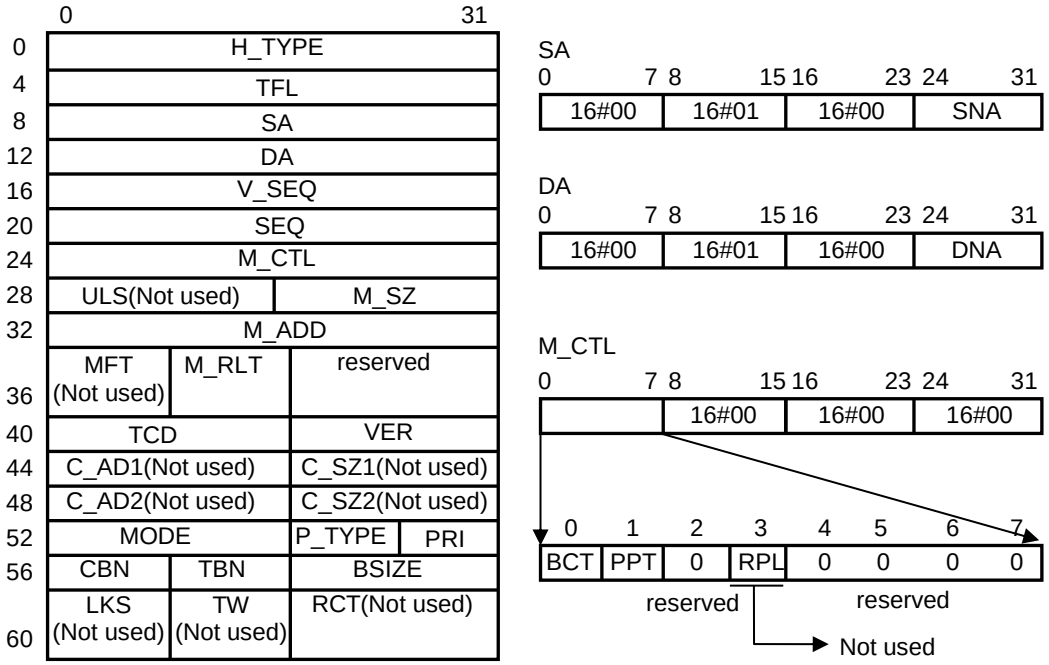


Figure 103 - The format of the message header that uses the virtual address space

Table 24 - The items of the message header that uses the virtual address space

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	FACN: FL-net
TFL	4	Octet length including the header and data	Octet lengths shall be stored in the 2 octets of 16 to 31, and 0 to 15 shall be 16#0000.
SNA	1	Node number of source side	―
DNA	1	Node number of destination side	―
V_SEQ	4	Version of sequence number	―
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	―
BCT	1 bit	Broadcast transmission	1 at 1 to n transmission
PPT	1 bit	Peer to peer transmission	1 at 1 to 1 transmission
M_SZ	2	Message data size in virtual address space	―
M_ADD	4	Offset address of the virtual address space	―
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	The header type of the FL-net
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol version (major version, minor version), token mode	―
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	―

b) In the case of a message which does not use the virtual address space

The format of a message header which does not use the virtual address space shall be shown in Figure 104 and the items of a message header which does not use the virtual address space shall be shown in Table 25.

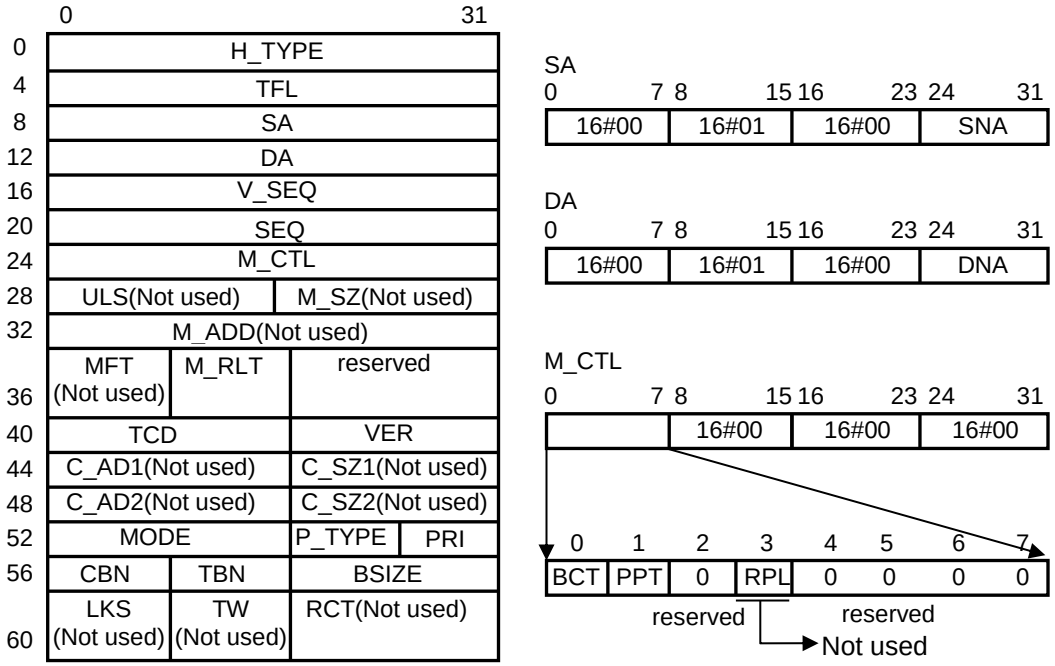


Figure 104 - The format of a message header which does not use the virtual address space

Table 25 - The items of a message header which does not use the virtual address space

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	FACN: FL-net
TFL	4	Octet length including the header and data	Octet lengths shall be stored in the 2 octets of 16 to 31, and 0 to 15 shall be 16#0000.
SNA	1	Node number of source side	—
DNA	1	Node number of destination side	—
V_SEQ	4	Version of sequence number	—
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	—
BCT	1bit	Broadcast transmission	1 at 1 to n transmission
PPT	1bit	Peer to peer transmission	1 at 1 to 1 transmission
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	The header type of the FL-net
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol version (major version, minor version), token mode	—
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	—

c) In the case of a vendor specific message

The format of a vendor specific message header shall be shown in Figure 105 and the items of a vendor specific message shall be shown in Table 26.

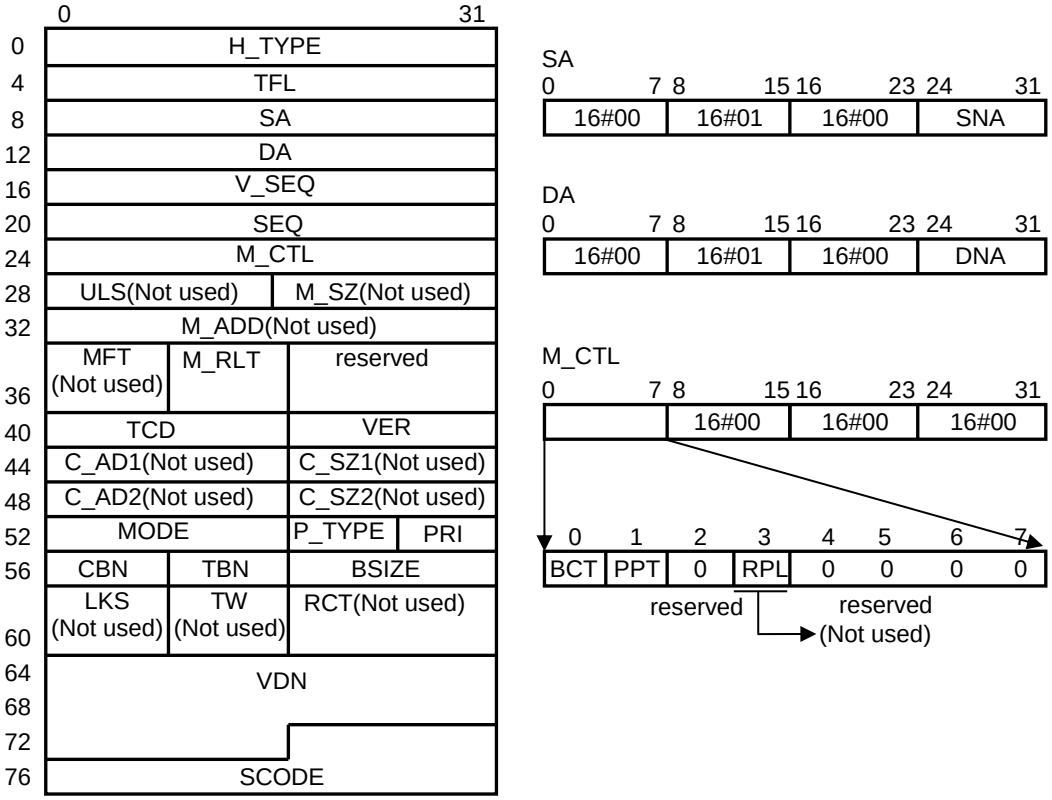


Figure 105 - The format of a vendor specific message header

Table 26 - The items of a vendor specific message header

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	FACN: FL-net
TFL	4	Octet length including the header and data	Octet lengths shall be stored in the 2 octets of 16 to 31, and 0 to 15 shall be 16#0000.
SNA	1	Node number of source side	—
DNA	1	Node number of destination side	—
V_SEQ	4	Version of sequence number	—
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	—
BCT	1bit	Broadcast transmission	1 at 1 to n transmission
PPT	1bit	Peer to peer transmission	1 at 1 to 1 transmission
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	The header type of the FL-net
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol version (major version, minor version), token mode	—
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	—
VDN	10	Vendor name	—
SCODE	6	Subcode	Code managed by a vendor

10.1.5 Participation Request Frame Header and Trigger Frame Header

The format of a participation request frame header and a trigger frame header shall be shown in Figure 106 and the items of the participation request frame header and the trigger frame header shall be shown in Table 27.

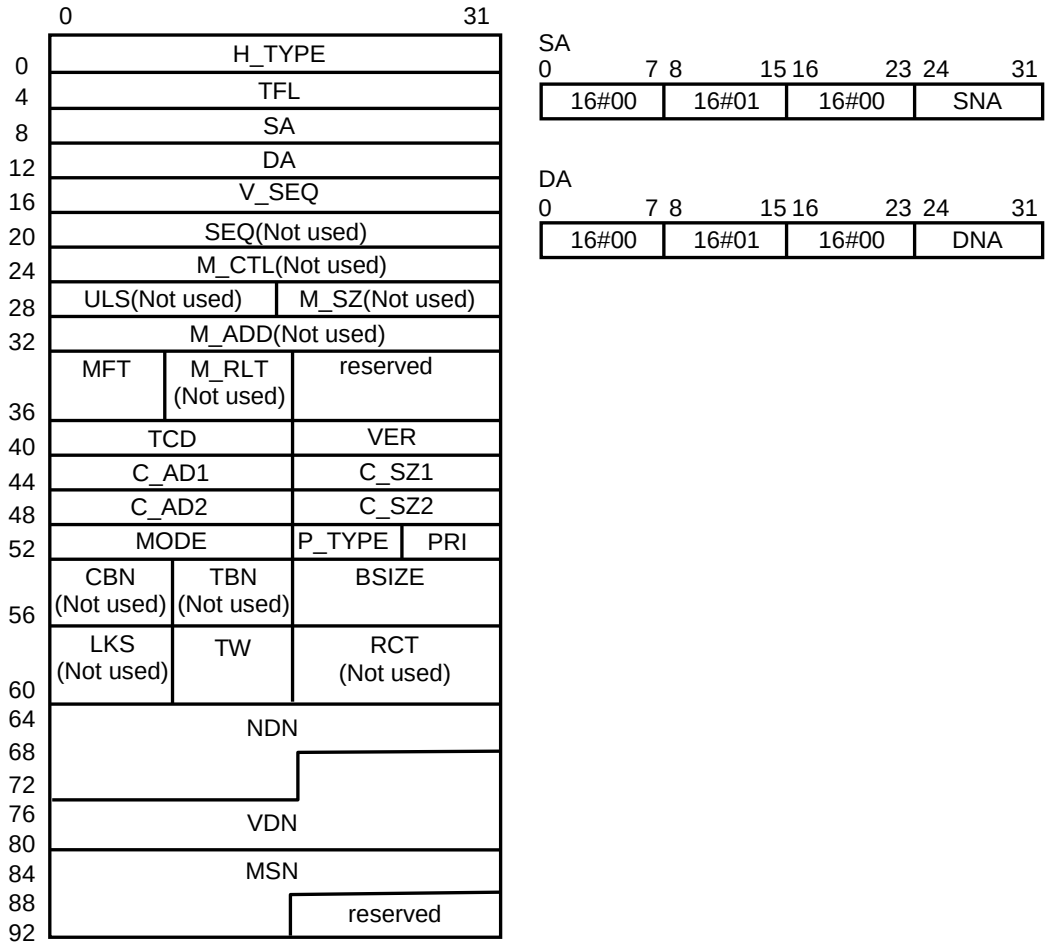


Figure 106 - The format of a participation request frame header and a trigger frame header

Table 27 - The items of a participation request frame header and a trigger frame header

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	FACN: FL-net
TFL	4	Octet length of a header	Fixed to 16#0000_0060
SNA	1	Node number of source side	—
DNA	1	Node number of destination side	—
V_SEQ	4	Version of sequence number	—
MFT	1	Allowable minimum frame interval time	—
TCD	2	Transaction code	The header type of the FL-net
VER	2	Program version	Fixed to 16#0000
C_AD1	2	Common area 1 datatop address	—
C_SZ1	2	Common memory area 1 data size	—
C_AD2	2	Common area 2 datatop address	—
C_SZ2	2	Common memory area 2 data size	—
MODE	2	FA link protocol version (major version, minor version), token mode	—
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
BSIZE	2	Current block size (octet)	Fixed to 16#0060
TW	1	Token watchdog time	—
NDN	10	Node information: Node name of node information	—
VDN	10	Node information: Vendor name	—
MSN	10	Node information: Manufacturer model name of node information	—

10.2 ACK Data

The ACK data shall be as follows.

a) Basic specification

ACK by message transmission is transmitted when performing cyclic transmission.

The ACK data which is a collection of ACK is added between the cyclic header and the cyclic data in cyclic transmission.

When the cyclic data is divided for transmission, the ACK data is attached to the last one of the divided frames for transmission. The location of the ACK data during transmission shall be as shown in Figure 107.

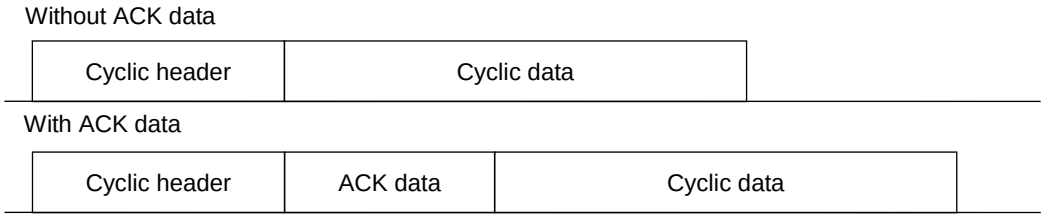
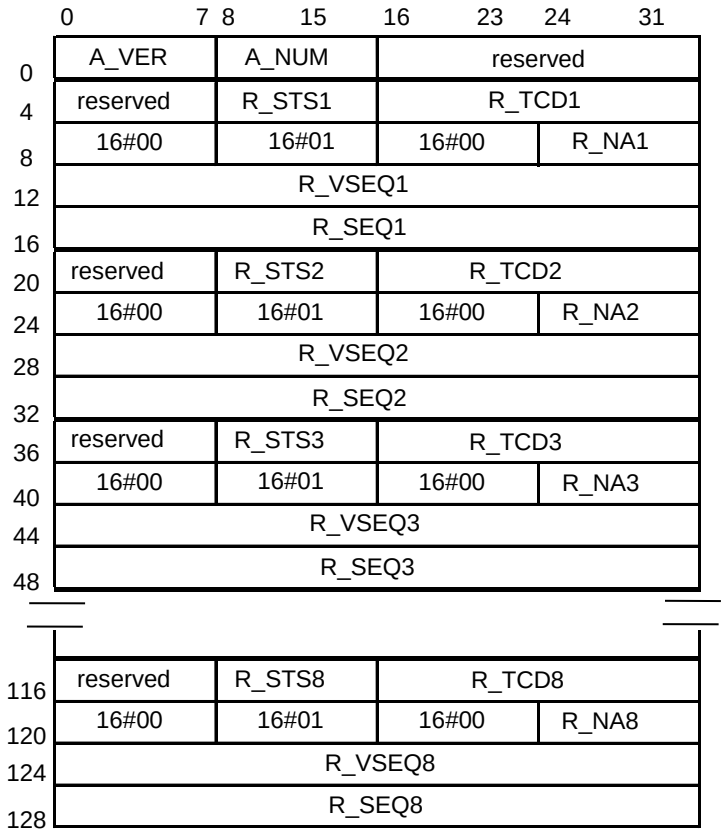


Figure 107 - The location of ACK data during transmission

The size of ACK data shall be of variable length and from 20 to 132 octets. The number of ACKs which can be transmitted in one cyclic transmission is up to 8.

b) ACK data

The format of ACK data shall be shown in Figure 108 and the ACK data items shall be shown in Table 28.



Note This figure assumes there are 8 ACKs. ACK data shall change its size by 16 octets.

Figure 108 - The format of ACK data

Table 28 - ACK data items

Symbols	Size (Octet)	Items	Setting values
A_VER	1	The version of ACK data	Fixed to 16#00
A_NUM	1	The number of ACKs (maximum of 8)	–
R_STSx	1	Returns the status flag, an error, etc. of ACK	–
R_TCDx	2	TCD of the message corresponding to ACK to be transmitted	Received TCD
R_NAx	1	The node number of destination side of ACK	–
R_VSEQx	4	The version of sequence number received	–
R_SEQx	4	The sequence number received	–

The meaning of the value of R_STSx shall be as follows.

— 16#01 indicates that a response node normally received the message.

— 16#02 indicates that the receive buffer for the messages of the response node is full.

— 16#03 indicates that initialization of the message reception process of the response node has not been finished.

— 16#05 indicates that the response node detected an error in the version of sequence number.

— 16#06 indicates that the response node detected an abnormal format.

— 16#10 indicates that there is no ACK from the request node and that the request node detected a reception timeout.

— 16#14 indicates that the request node detected the sequence number error of ACK.

— 16#15 indicates that the request node detected an error in the version of sequence number of ACK.

Notes R_STSx values of 16#10,16#14 and 16#15 shall be detected at a request node, and not used in the ACK data.

10.3 User Data

10.3.1 Cyclic Data

The format of a cyclic frame at cyclic transmission shall be as shown in Figure 109. In frames at cyclic transmission, data shall be written and transmitted in the order of the area 1 and the area 2. When there is ACK data, it shall be inserted after the cyclic header.

When the cyclic data is larger than 1024 octets, it shall be divided into two or more frames for transmission. The first one of the divided frames is written with the data in the area 1 and the area 2, and the second one and the followings are written with the data of the area 2.

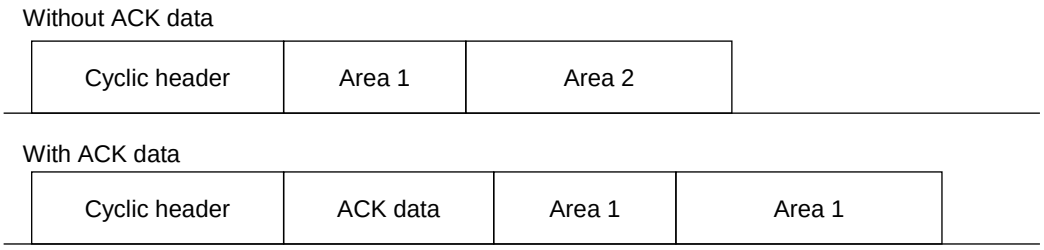


Figure 109 - The format of a cyclic frame

10.3.2 Message Data

The format of the message frame for message transmission shall be as shown in Figure 110.

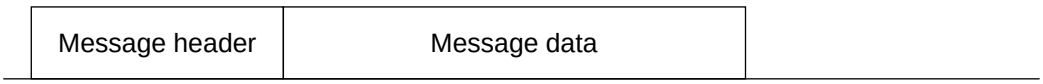


Figure 110 - The format of a message frame

Message data shall be as follows.

a) Byte block read

The data part of the byte block read shall be shown in Table 29.

Table 29 - The data part of the byte block read

Items	Contents of data part
Request	Not available
Normal response (M_RLT＝0)	Offset +0 +1 +2 — 7 … 0: Read data; Any size at the maximum of 1024 octets is acceptable.
Abnormal response (M_RLT＝1)	Offset +0 +1 +2 — 7 … 0: Error code; Error codes shall be defined by venders.; Any size at the maximum of 1024 octets is acceptable.
Unimplemented response (M_RLT＝2)	Not available

b) Byte block write

The data part of the byte block write shall be shown in Table 30.

Table 30 - The data part of byte block write

Items	Contents of data part
Request	Offset +0 +1 +2 — 7 … 0: Read data; Any size at the maximum of 1024 octets is acceptable.
Normal response (M_RLT＝0)	Not available
Abnormal response (M_RLT＝1)	Offset +0 +1 +2 — 7 … 0: Error code; Error codes shall be defined by venders.; Any size at the maximum of 1024 octets is acceptable.
Unimplemented response (M_RLT＝2)	Not available

c) Word block read

The data part of word block write shall be shown in Table 31.

Table 31 - The data part of word block read

Items	Contents of data part
Request	Not available
Normal response (M_RLT＝0)	Offset +0 / +1 / +2 — 15 … 0 Read data Any size at the maximum of 1024 octets is acceptable.
Abnormal response (M_RLT＝1)	Offset +0 / +1 / +2 — 15 … 0 Error code Error codes shall be defined by venders. Any size at the maximum of 1024 octets is acceptable.
Unimplemented response (M_RLT＝2)	Not available

d) Word block write

The data part of word block write shall be shown in Table 32.

Table 32 - The data part of word block write

Items	Contents of data part
Request	Offset +0 / +1 / +2 — 15 … 0 Read data Any size at the maximum of 1024 octets is acceptable.
Normal response (M_RLT＝0)	Not available
Abnormal response (M_RLT＝1)	Offset +0 / +1 / +2 — 15 … 0 Error code Error codes shall be defined by venders. Any size at the maximum of 1024 octets is acceptable.
Unimplemented response (M_RLT＝2)	Not available

e) Network parameter read

The data part of network parameter read shall be shown in Table 33.

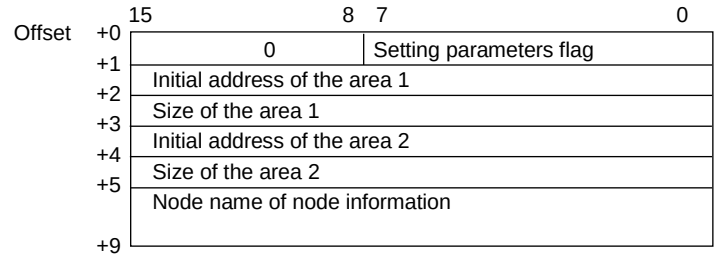
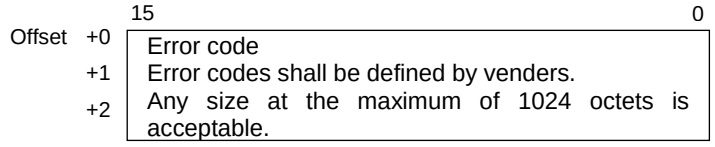
Table 33 - The data part of network parameter read

Items	Contents of data part
Request	Not available
Normal response (M_RLT＝0)	0ffset +0 15 8 7 0 +0 to +4: Node name of node information +5 to +9: Vendor name +10 to +14: Manufacturer model name of node information +15: Initial address of the area 1 +16: Size of the area 1 +17: Initial address of the area 2 +18: Size of the area 2 +19: Spare / Token monitoring timeout time +20: Spare / Allowable minimum frame interval time +21: Spare / Status of the link +22: Spare / Protocol type +23: Upper layer status +24: Allowable refresh cycle time setting value +25: Refresh cycle measurement value (current value) +26: Refresh cycle measurement value (maximum value) +27: Refresh cycle measurement value (minimum)
Abnormal response (M_RLT＝1)	Offset +0 15 0 +0 to +2: Error code Error codes shall be defined by venders. Any size at the maximum of 1024 octets is acceptable.

f) Network parameter write

The data part of the network parameter write shall be shown in Table 34.

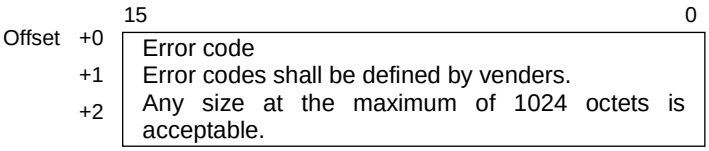
Table 34 - The data part of the network parameter write

Items	Contents of data part
Request	 Only settings of the address and the size are written when the setting parameters flag is 16#01. Only settings of the node name are written when the setting parameters flag is 16#02. Both settings of the address and settings of the node name are written when the setting parameters flag is 16#03.
Normal response (M_RLT＝0)	Not available
Abnormal response (M_RLT＝1)	
Unimplemented response (M_RLT＝2)	Not available

g) Stop command

The data part of the stop command service shall be shown in Table 35.

Table 35 - The data part of the stop command service

Items	Contents of data part
Request	Not available
Normal response (M_RLT＝0)	Not available
Abnormal response (M_RLT＝1)	
Unimplemented response (M_RLT＝2)	Not available

h) Operation command

The data part of the operation command service shall be shown in Table 36.

Table 36 - The data part of the operation command service

Items	Contents of data part
Request	Not available
Normal response (M_RLT=0)	Not available
Abnormal response (M_RLT=1)	Offset +0 / +1 / +2 (15 … 0) Error code Error codes shall be defined by venders. Any size at the maximum of 1024 octets is acceptable.
Unimplemented response (M_RLT=2)	Not available

i) Profile read

The data part of the profile read shall be shown in Table 37.

Table 37 - The data part of the profile read

Items	Contents of data part
Request	Not available
Normal response (M_RLT=0)	Offset +0 / +1 / +2 (15 … 0) Read data Data is outside the scope of this stipulation.
Abnormal response (M_RLT=1)	Offset +0 / +1 / +2 (15 … 0) Error code Error codes shall be defined by venders. Any size at the maximum of 1024 octets is acceptable.

j) Transparent message

The data part of transparent message service shall be shown in Table 38.

Table 38 - The data part of the transparent message service

Items	Contents of data part
Request	Offset +0 / +1 / +2 (15 … 0) Data of transparent message service Any size at the maximum of 1024 octets is acceptable.

k) Log data read

The data part of the log data read shall be shown in Table 39.

Table 39 - The data part of the log data read

Items	Contents of data part
Request	Not available
Normal response (M_RLT＝0)	Offset +0 to +2 (15 … 0): Read data For data, refer to Appendix B. The size is 512 octets.
Abnormal response (M_RLT＝1)	Offset +0 to +2 (15 … 0): Error code Error codes shall be defined by venders. Any size at the maximum of 1024 octets is acceptable.

l) Log data clear

The data part of the log data clear shall be shown in Table 40.

Table 40 - The data part of the log data clear

Items	Contents of data part
Request	Not available
Normal response (M_RLT＝0)	Not available
Abnormal response (M_RLT＝1)	Offset +0 to +2 (1 … 0): Error code Error codes shall be defined by venders. Any size at the maximum of 1024 octets is acceptable.

m) Message return

The data part of the message return service shall be shown in Table 41.

Table 41 - The data part of the message return service

Items	Contents of data part
Request	Offset +0 to +2 (15 … 0): Data Users can use arbitrary data. Any size at the maximum of 1024 octets is acceptable.
Normal response (M_RLT＝0)	Offset +0 to +2 (15 … 0): Data Data is the same as the request. Any size at the maximum of 1024 octets is acceptable.

n) Vendor specific message

The data part of a vendor specific message shall be shown in Table 42.

Table 42 - The data part of the vendor specific message

Items	Contents of data part
Request	Offset +0 / +1 / +2 : 15 … 0 Data Users can use arbitrary data. Any size at the maximum of 1024 octets is acceptable.
Normal response (M_RLT＝0)	Offset +0 / +1 / +2 : 15 … 0 Data Users can use arbitrary data. Any size at the maximum of 1024 octets is acceptable.
Abnormal response (M_RLT＝1)	Offset +0 / +1 / +2 : 15 … 0 Error code Error codes shall be defined by venders. Any size at the maximum of 1024 octets is acceptable.
Unimplemented response (M_RLT＝2)	Not available
Note When a vendor specific message is not implemented, and when a request for a vendor name not supported is received, an unimplemented response shall be used.	

10.4 Solicitation frame

10.4.1 Solicitation Frame Header

The solicitation frame header shall be as follows.

a) Header structure

The header structure of a solicitation frame shall be as shown in Figure 111.

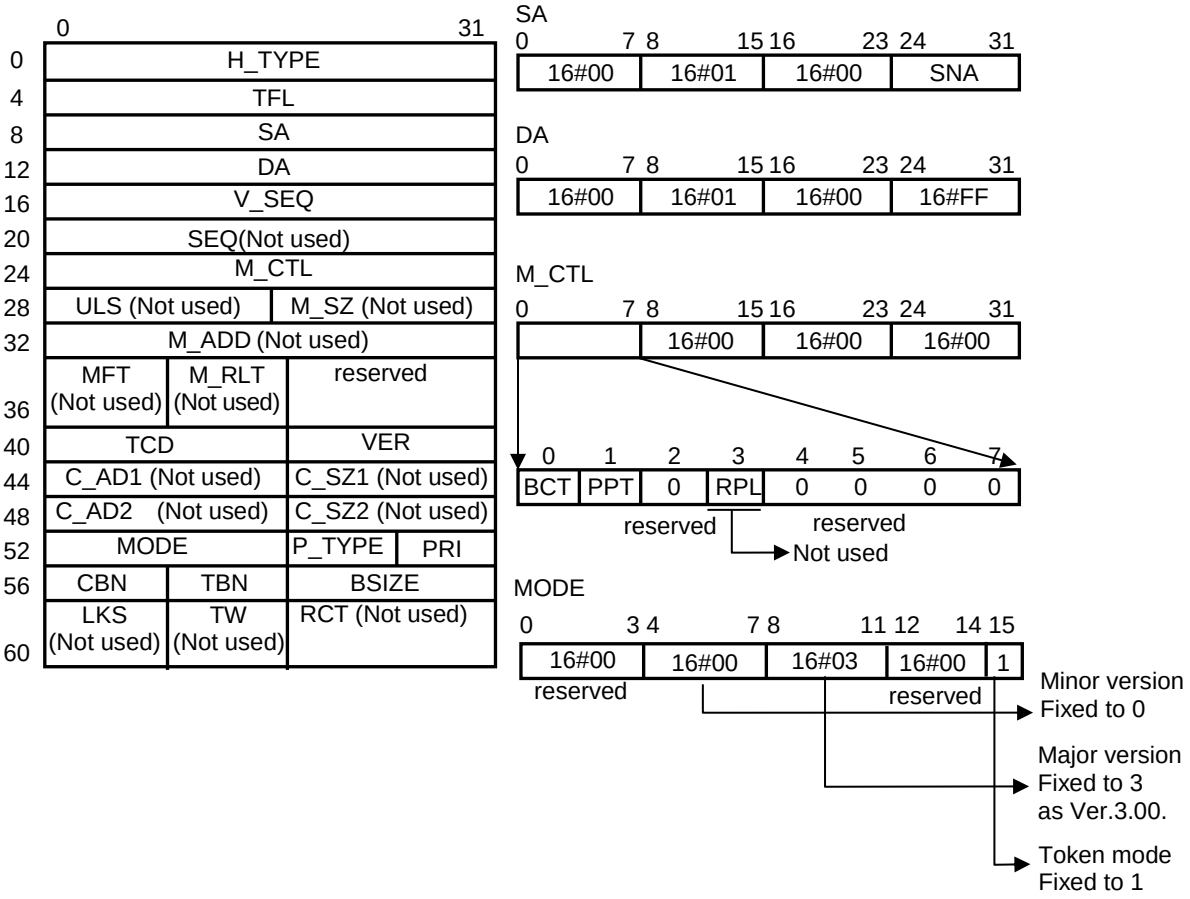


Figure 111 - The header structure of the solicitation frame

b) Details of header items

Details of the header items of a solicitation frame shall be shown in Table 43.

Table 43 - Details of header items

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	“FACN”
TFL	4	Octet length including header and data	64 to 1 088
SNA	1	Node number of source side	–
DNA	1	Node number of destination side	255
V_SEQ	4	Version of sequence number	–
BCT	1 bit	Broadcast transmission	1
PPT	1 bit	Peer to peer transmission	0
TCD	2	Transaction code	65 017
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol, token mode	–
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0000

10.4.2 The Data Part Format of the Solicitation Frame

Solicitation frames shall be transmitted from the master to a target slave and define the common memory allocation in the flexible setting mode. As shown in Figure 112, the data part of the solicitation frame shall set the slave node number, fixed settings and flexible settings identification flags, IO data area, and the status areas (the solicitation frames only include request data, and no response).

The data size of a solicitation frame is 8 words per node.

One transmission shall be up to 64 nodes (1024 octets) according to the defined transmission frame size.

When more than 64 nodes are included, solicitation data shall be transmitted with a number of solicitation frames (maximum of 4).

Offset (Decimal) (octet)	0 15	16 31	
0	Slave node number	Input data address	One set of slave information
4	Input data size	Output data address	
8	Output data size	Input status address	
12	Output status address	Lock ID	
16 to 28			One set of slave information
29 to 44			One set of slave information
. .			

Note Data is stored in a little-endian octet order.

Figure 112 - The data part format of the solicitation frame

a) Details of the slave node number part

It is an area to specify the node number of a slave. The most significant bit specifies whether it is fixed mode or flexible mode. Details of the slave node number part shall be as shown in Figure 113.

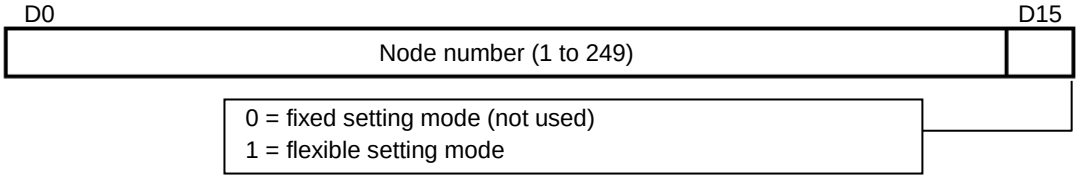


Figure 113 - Details of the slave node number part

b) Details of the address part

It is an area to specify the word address of the common memory of a slave. The most significant bit specifies the area 1 or the area 2. Details of the address part shall be as shown in Figure 114.

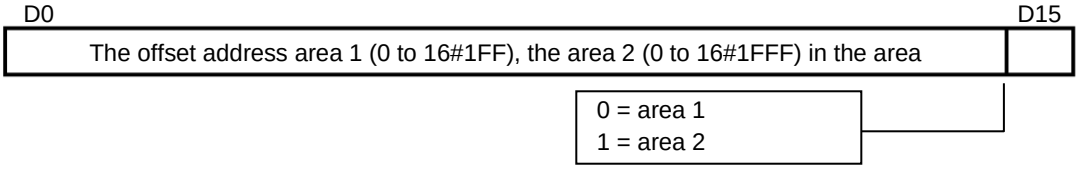


Figure 114 - Details of the address part

c) Details of the data size part

Details of the data size part shall be as shown in Figure 115.

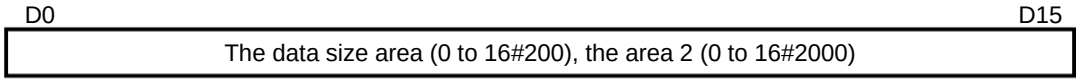


Figure 115 - Details of the data size part

It is an area to specify the word size of the common memory of a slave.

Note The input status size and output status size are fixed to 16 words and shall not be set. When extra status data area is required, it shall be allocated in the IO area.

d) Details of the Lock ID part

Details of the Lock ID part shall be as shown in Figure 116.

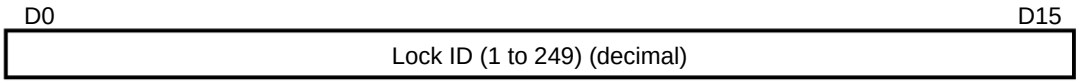


Figure 116 - Details of the Lock ID part

The last 1 octetof the IP address of the master is allocated.

Example: Address: 192.168.0.1 ⟶ Lock ID: 1

Address: 192.168.0.100 ⟶ Lock ID: 100

10.5 IO Allocation Setting Frame

10.5.1 The Header Format of the IO Allocation Setting Frame

The frame header format of the IO allocation shall be as follows.

a) Header structure

The header structure of the IO allocation setting frame shall be as shown in Figure 117.

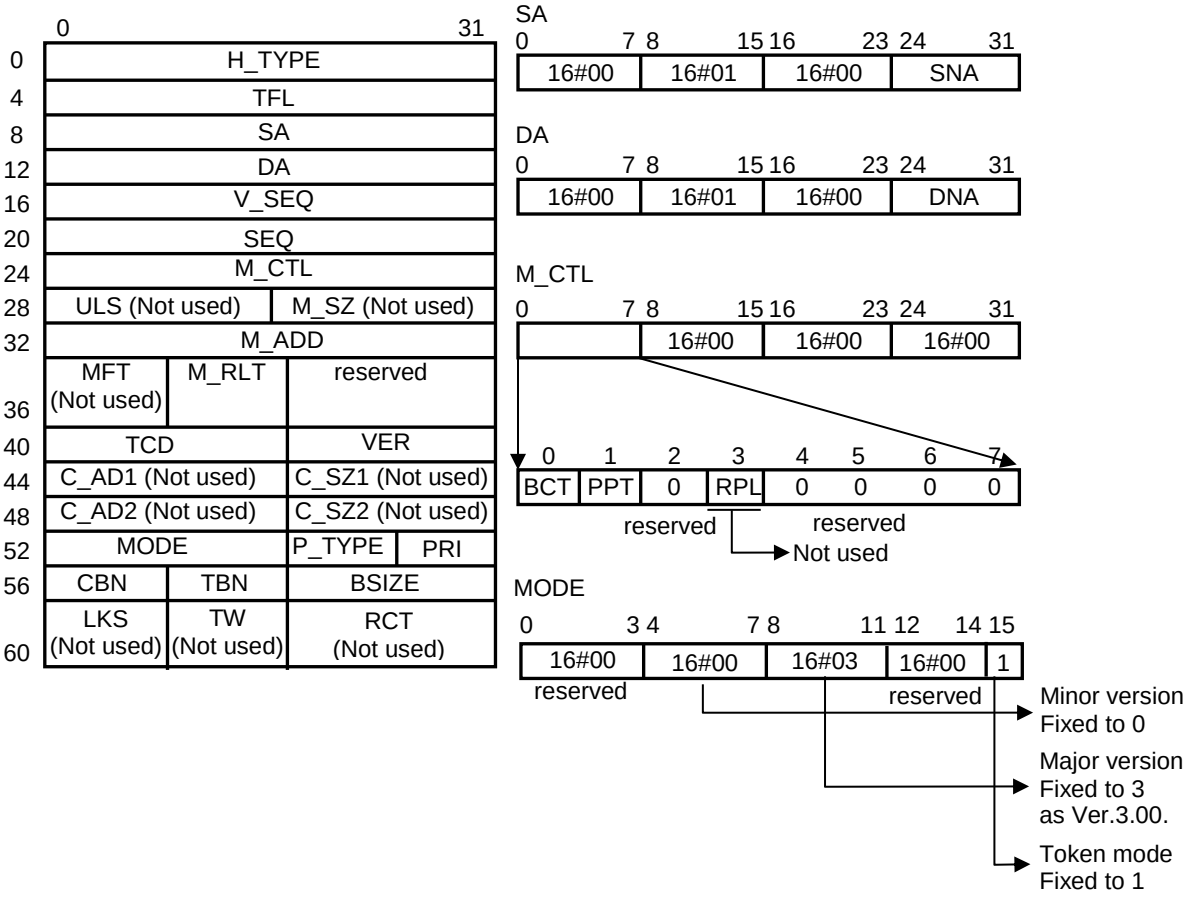


Figure 117 - The header structure of the IO allocation setting frame

b) Details of header items

Details of the header items shall be shown in Table 44.

Table 44 - Details of the header items

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	“FACN”
TFL	4	Octet length including header and data	64 to 88 (decimal)
SNA	1	Node number of source side	1 to 254 (decimal)
DNA	1	Node number of destination side	1 to 254 (decimal)
V_SEQ	4	Version of sequence number	Fixed to 16#0000_0000
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	16#0000_0001 to 16#FFFF_FFFF
BCT	1 bit	Broadcast transmission	0
PPT	1 bit	Peer to peer transmission	1
M_RLT	1	Message result (Normal, Abnormal)	Response 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	65 018, 65 218(decimal)
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol, token mode	—
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0000

10.5.2 The Format of the Data Part of the IO Allocation Setting Frame

The format of the data part of the IO allocation setting frame shall be as follows.

a) The data part upon request

The data part upon request shall be as shown in Figure 118.

Request

Offset (Decimal) (octet)	0 15	16 31
0	Setting No. (n-th)	Number of control slaves (total)
4	Lock ID	Lock timeout setting value
8	Slave node number	Input data address
12	Input data size	Output data address
16	Output data size	Input status address
20	Output status address	Reserve

Slave information

Note 1 Data is stored in a little-endian octet order.
Note 2 IO allocation from the setting tool to flexible master shall be set node by node and when there are two or more slave nodes, the settings shall be repeated for the number of nodes.
Note 3 The number of control slaves: the total number of slaves managed by the master.
Note 4 Setting No.: Indicates the position in sequence of the setting information over the number of control slaves.
Note 5 Lock ID: In case there are two or more information on the slave to be set up, when the first setting No. is sent, any ID added to the response data from the command server of flexible master of the responder shall be added and set from the second time. When there are two or more setting tools which access the same master, this ID is used for exclusion control of IO allocation from other setting tools.
Note 6 Lock timeout setting value: Set when the setting No. = 1.
The node regards this value as lock time timeout period of IO allocation.
The setting ranges shall be from 1 to 65,535 (seconds).
- The setting No. indicates the position in sequence of the setting in the (total) number of the control slaves.
- When the setting No. = 1, the master shall start a new setting and discard the old one.
- When the same setting is received, the latest shall be overwritten.
- It is consider to be the end of setting when the number of the control slave and the setting No. are the same.
- When all are to be deleted, the number of the control slaves and the setting No. shall be set to 0.

Note 7 When the setting No. = 16#FFFF and the number of control slaves = 16#FFFF, it functions as a command to cancel the lock of IO allocation.

Figure 118 - The data part upon receipt (receipt of request from a partner)

1) Details of the slave node number part

Details of the slave node number part shall be as shown in Figure 119.

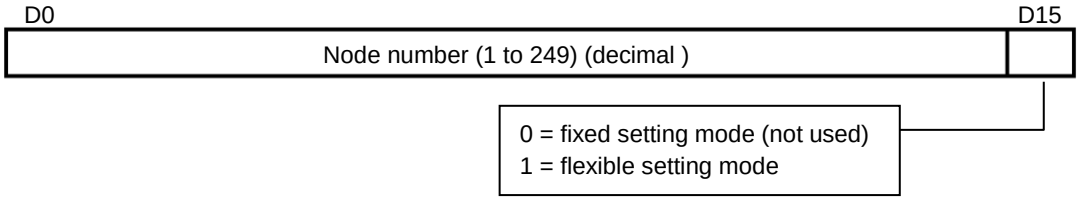


Figure 119 - Details of the slave node number part

It is an area to specify the node number of a slave. The most significant bit specifies whether it is fixed or flexible.

2) Details of the address part

Details of the address part shall be as shown in Figure 120.

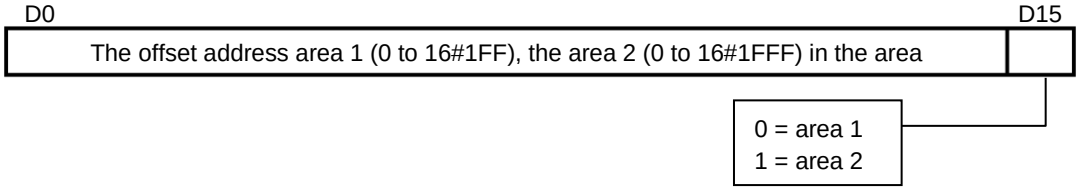


Figure 120 - Details of the address part

It is an area to specify the word address of the common memory of a slave. The most significant bit specifies the areas 1 and 2.

3) Details of the input and output data size part

Details of the input and output data size part shall be as shown in Figure 121.

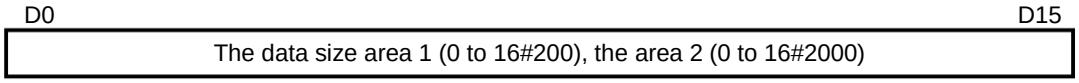


Figure 121 - Details of the input and output data size part

It is an area to specify the word size of the common memory of a slave.

Note The input status size and output status size are fixed to 16 words and shall not be set.

When extra status data area is required, it shall be allocated in the IO area.

b) The data part upon response

The data part upon response shall be as shown in Figure 122.

Response

Offset (Decimal) (octet)	0 15	16 31
0	Result (M_RLT)	Information
4	Lock ID (response)	Reserve

Result :=0 Normal
:=1 Continue
:=2 Abnormal
Information: Result = 2 the factor of the abnormal condition
1 = Abnormal slave node number
2 = Abnormal slave common memory address
3 = Abnormal slave common memory size
4 = Exclusive control locked
5 = Exclusive lock time timed out
Lock ID: Any IDs returned from the command server.
This ID shall be added to the second and the following requested data part.

Note When the Result = "0 normal" and "1 Continue", the message result of the header (M_RLT) = 0: normal.
When the result = "2 Abnormal", the message result of the header (M_RLT) = 1: abnormal.

Figure 122 - The data part when transmitting a response (response to a request received from the partner)

10.6 IO Allocation and Read Frame

10.6.1 The Header Format of the IO Allocation and Read Frame

The header format of the IO allocation read frame shall be as follows.

a) Header structure

The IO allocation shall be read from the master node.

Since the size is for one node, the setting No. shall be set upon request. The header structure of the IO allocation read frame shall be as shown in Figure 123.

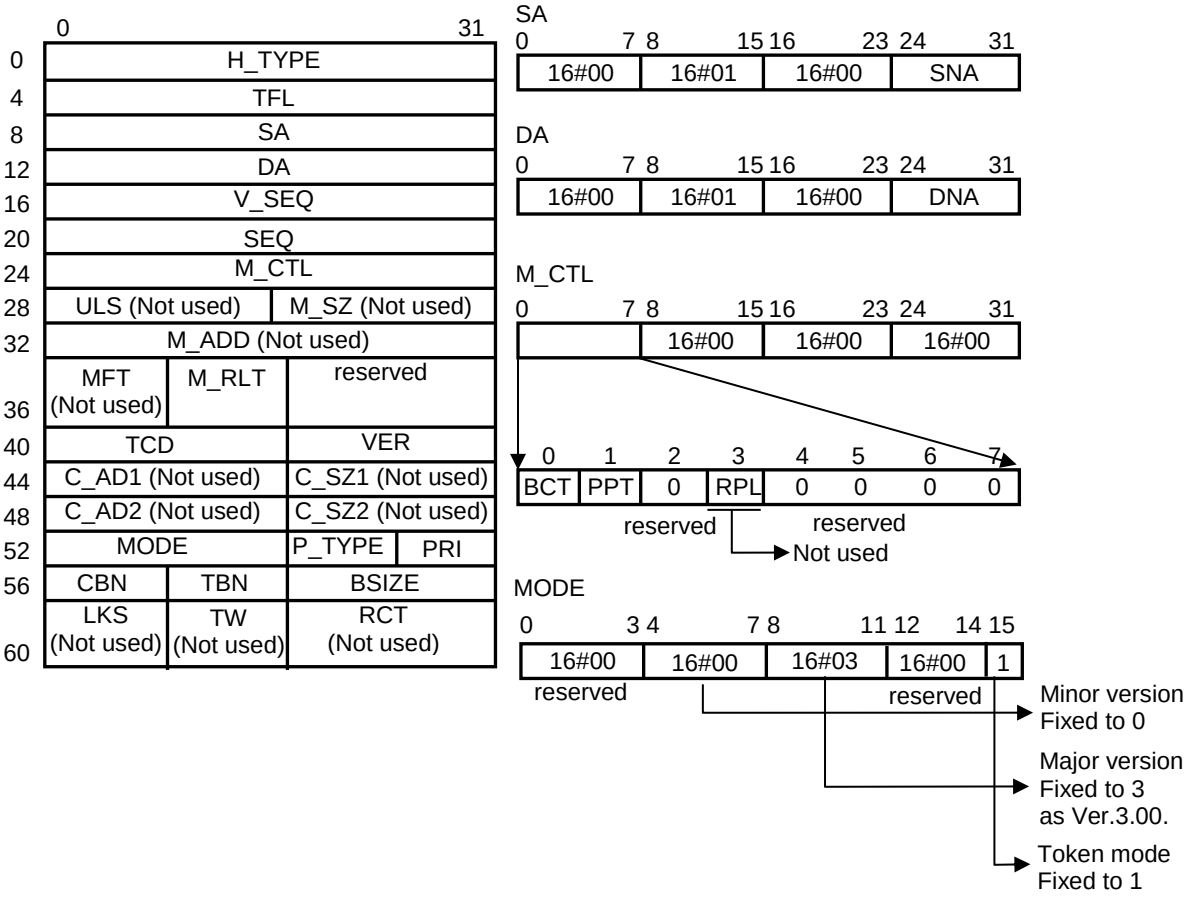


Figure 123 - The header structure of the IO allocation read frame

b) Details of header items

Details of the header items shall be shown in Table 45.

Table 45 - Details of the header items

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	“FACN”
TFL	4	Octet length including header and data	64 to 84 (decimal)
SNA	1	Node number of source side	1 to 254 (decimal)
DNA	1	Node number of destination side	1 to 254 (decimal)
V_SEQ	4	Version of sequence number	Fixed to 16#0000_0000
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	16#0000_0001 to 16#FFFF_FFFF
BCT	1 bit	Broadcast transmission	0
PPT	1 bit	Peer to peer transmission	1
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	65 019, 65 219(decimal)
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol, token mode	–
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0000

10.6.2 The Format of the IO Allocation and Read Frame Data Part

The format of the IO allocation read frame data part shall be as follows.

a) The data part upon request

The data part upon request shall be as shown in Figure 124.

Request	Offset (Decimal) (octet)	0 – 15	16 – 31
	0	Setting No. m (1 to 248) (decimal)	

Figure 124 - The data part upon request

b) The data part upon response

The data part upon response shall be as shown in Figure 125.

Response

Offset (Decimal) (octet)	0 15	16 31
0	Setting No.	Number of control slaves
4	Slave node number	Input data address
8	Input data size	Output data address
12	Output data size	Input status address
16	Output status address	Reserve

Slave information when the setting No. = m

Note 1 Data is stored in a little-endian octet order.
Note 2 The setting No. indicates the setting No. upon IO allocation.
Note 3 When the setting No. of requested data is in abnormal range (other than 1 to 248), the message result of the header (M_RLT) = 1 (abnormal).
Note 4 When there is no slave information, the number of control slaves and the setting No. of response data shall be 0, and the message result of the header (M_RLT) = 0 (normal).
Note 5 When the request of No. which is not set is received, the number of control slaves and the setting No. shall be 0, and the message result of the header (M_RLT) = 0 (normal).

Figure 125 - The data part upon response

1) Details of the slave node number part

Details of the slave node number part shall be as shown in Figure 126.

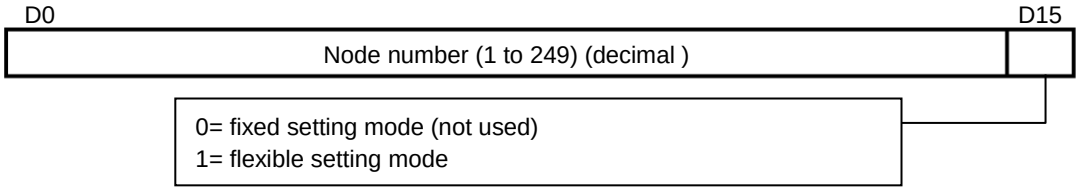


Figure 126 - Details of the slave node number part

It is an area to specify the node number of a slave. The most significant bit specifies whether it is fixed or flexible.

2) Details of the address part

Details of the address part shall be as shown in Figure 127.

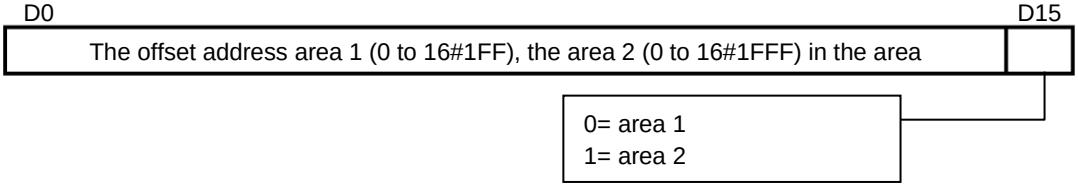


Figure 127 - Details of the address part

It is an area to specify the word address of the common memory of a slave. The most significant bit specifies the areas 1 and 2.

3) Details of the input and output data size part

Details of the input and output data size part shall be as shown in Figure 128.

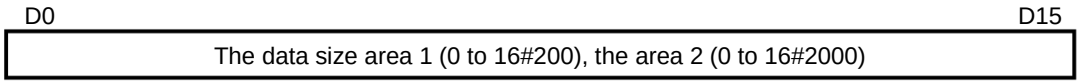


Figure 128 - Details of the input and output data size part

It is an area to specify the word size of the common memory of a slave.

Note The input status size and output status size are fixed to 16 words and shall not be set. When extra status data area is required, it shall be allocated in the IO area.

10.7 Token Retention Time Measuring Flame

10.7.1 The Header Format of the Token Retention Time Measuring Flame

The header format of the token retention time measuring flame shall be as follows.

a) Header structure

The header structure of the token retention time measuring flame shall be as shown in Figure 129.

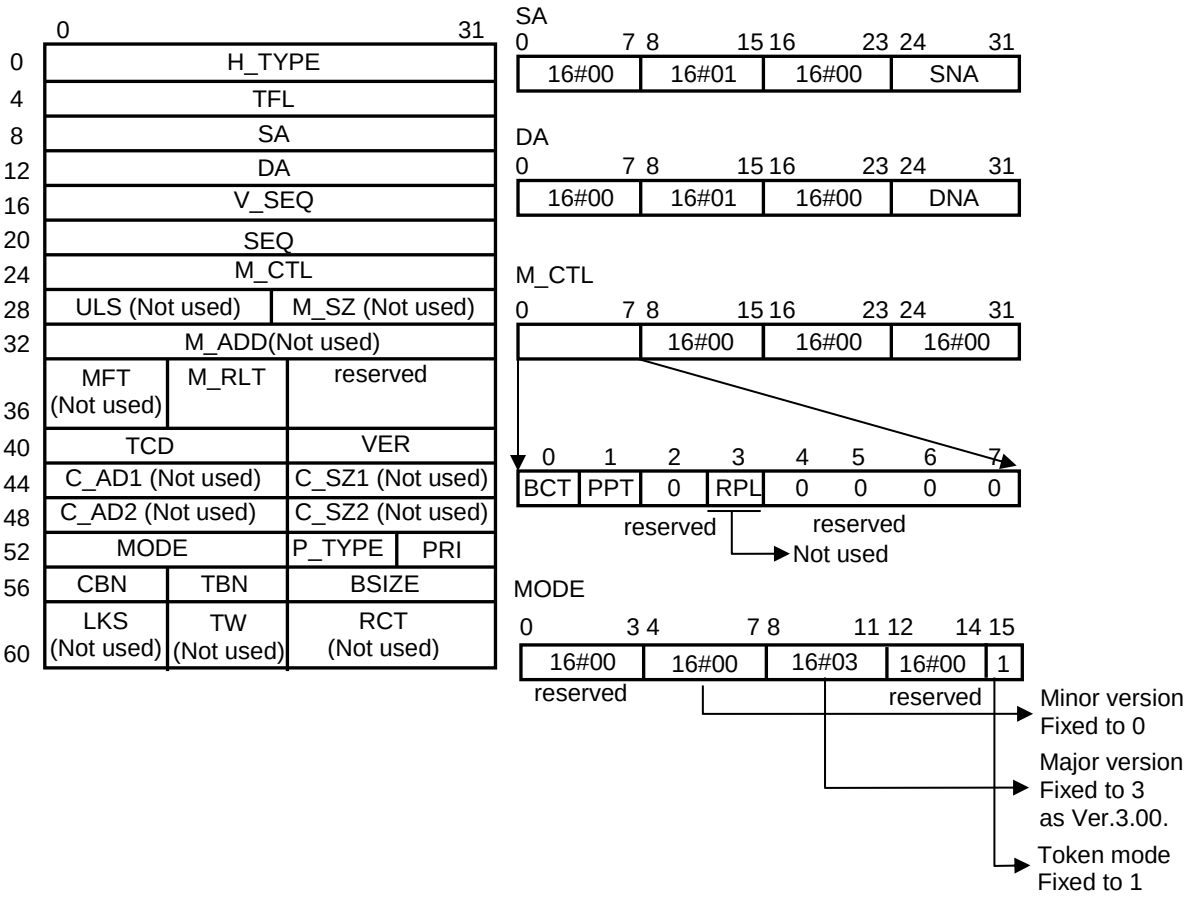


Figure 129 - The header structure of the token retention time measuring flame

b) Details of the header items

Details of the header items of the token retention time measuring flame shall be shown in Table 46.

Table 46 - Details of the header items

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	“FACN”
TFL	4	Octet length including header and data	
SA	4	Node number of source side	—
DA	4	Node number of destination side	—
V_SEQ	4	Version of sequence number	—
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	—
BCT	1 bit	Broadcast transmission	0
PPT	1 bit	Peer to peer transmission	1
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	65 020, 65 021, 65 220, 65 221 (decimal)
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol, token mode	—
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0000

10.7.2 The Format of the Data Part of the Token Retention Time Measuring Flame

The response data item added as a response to a token retention time measurement terminating request shall be shown in Table 47.

Table 47 - Token retention time measurement Response data

Offset (Decimal) (octet)	0 31
0	Number of the events of discarding token
4	Time when the event of discarding token was detected last
8	Number of the events of reissuing token
12	Time when reissuance of token occurred last
16	The number of the token retention timeouts
20	Time when the token retention timeout event occurred last
24	The number of the token watchdog timeouts
28	Time when the token watchdog timeout event occurred last
32	Maximum token retention time
36	Minimum token retention time
40	The time when the maximum token retention time was detected
44	Measuring time for token retention time
48	The number of tokens while measuring the token retention time
52	Detection time for the maximum refresh cycle
56	Number of the events of receiving cyclic frame
60	Number of errors in receiving cyclic frame
64	Detection time for error in receiving cyclic frame
68	Number of errors in receiving message frame
72	Detection time for error in receiving message frame
Note Data is stored in a little-endian octet order.	

10.8 The Message Frame of General Purpose Communication Data Sender Log

10.8.1 The Header Format of the Message Frame of the General Purpose Communication Data Sender Log

The header format of the message frame of the general purpose communication data sender log shall be as follows.

a) Header structure

The Header structure of the message frame of the general purpose communication data sender log shall be as shown in Figure 130.

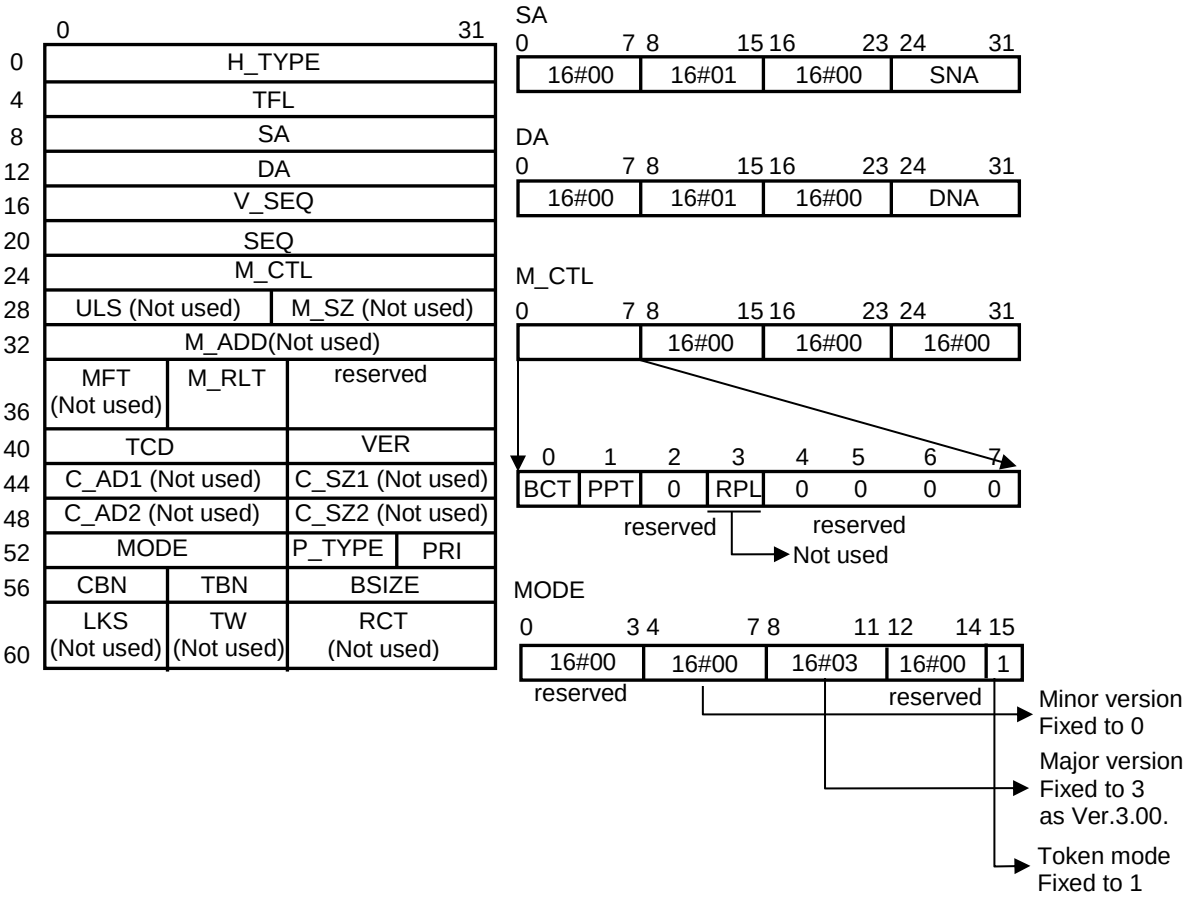


Figure 130 - The header structure of the message frame of the general purpose communication data sender log

b) Details of header items

Details of the header items of the message frame of the general purpose communication data sender log shall be shown in Table 48.

Table 48 - Details of the header items

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	“FACN”
TFL	4	Octet length including header and data	
SA	4	Node number of source side	—
DA	4	Node number of destination side	—
V_SEQ	4	Version of sequence number	—
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	—
BCT	1 bit	Broadcast transmission	0
PPT	1 bit	Peer to peer transmission	1
M_RLT	1	Message result (Normal, Abnormal)	0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	65 022, 65 023, 65 222, 65 223
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol, token mode	—
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0000

10.8.2 The Format of the Message Frame Data Part of the General Purpose Communication Data Sender Log

The response data item added as a response to the general purpose communication data sender log measurement terminating request shall be shown in Table 49.

Table 49 - The response data for the general purpose communication data sender log measurement

Offset (Decimal) (octet)	0 31
0	Measuring time for general purpose communication data sender log
4	IP 1
8	IP 1 receive counter
12	IP 2
16	IP 2 receive counter
20	IP 3
24	IP 3 receive counter
28	IP 4
32	IP 4 receive counter
36	IP 5
40	IP 5 receive counter
44	IP 6
48	IP 6 receive counter
52	IP 7
56	IP 7 receive counter
60	IP 8
64	IP 8 receive counter
68	IP 9
72	IP 9 receive counter
76	IP 10
80	IP 10 receive counter
Note Data is stored in a little-endian octet order.	

10.9 Configuration Parameter Setting Frame

This frame is to set configuration parameter of the node selected by the user, such as the address and the size of the common memory area 1 and 2, the node name, the token watchdog time, and the allowable minimum frame interval time.

Since some modules do not need to set the common memory addresses such as a fixed setting module, the users shall select the setting items.

When the configuration parameter except the node name is changed, the re-joining operation to the network shall be performed.

10.9.1 The Header Format of the Configuration Parameter Setting Frame

The header format of the configuration parameter setting frame shall be as follows.

a) Header structure

The header structure of the configuration parameter setting frame shall be as shown in Figure 131.

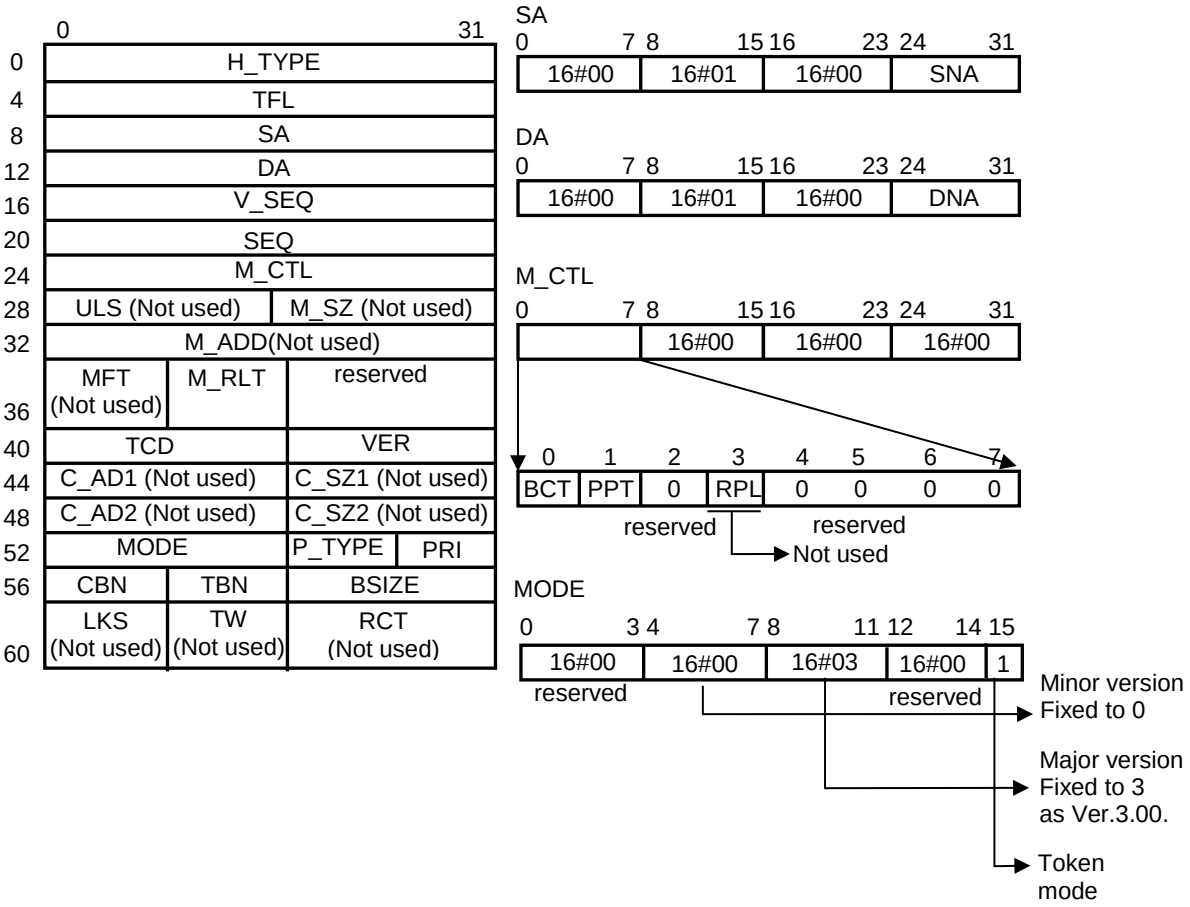


Figure 131 - The header structure of the configuration parameter setting frame

b) Details of header items

Details of the header items of the configuration parameter setting frame shall be shown in Table 50.

Table 50 - Details of the header items

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	“FACN”
TFL	4	Octet length including header and data	64 to 92 (decimal)
SNA	1	Node number of source side	1 to 254 (decimal)
DNA	1	Node number of destination side	1 to 254 (decimal)
V_SEQ	4	Version of sequence number	Fixed to 16#0000_0000
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	16#0000_0001 to 16#FFFF_FFFF
BCT	1 bit	Broadcast transmission	0
PPT	1 bit	Peer to peer transmission	1
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	65 024, 65 224 (decimal)
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol, token mode	–
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0000

10.9.2 The Format of the Data Part of the Configuration Parameter Setting Frame

The parameter shall be configured in a setting request as shown in Figure 132, and the configured contents shall be returned as the response data to confirm the setting.

Each parameter of the setting item can be selected to be updated or not updated with the update flag.

The format of the data part of the configuration parameter setting frame shall be as follows.

a) The data part upon request

The data part upon request shall be as shown in Figure 132.

Request

Offset (Decimal) (octet)	0 7 8 15 16 23 24 31			
0	Update flag		Reserve	
4	Node name			
8	Node name			
12	Node name		Reserve	
16	Address of the common memory 1		Size of the common memory 1	
20	Address of the common memory 2		Size of the common memory 2	
24	Token watchdog time	Reserve	Allowable minimum frame interval time	Reserve

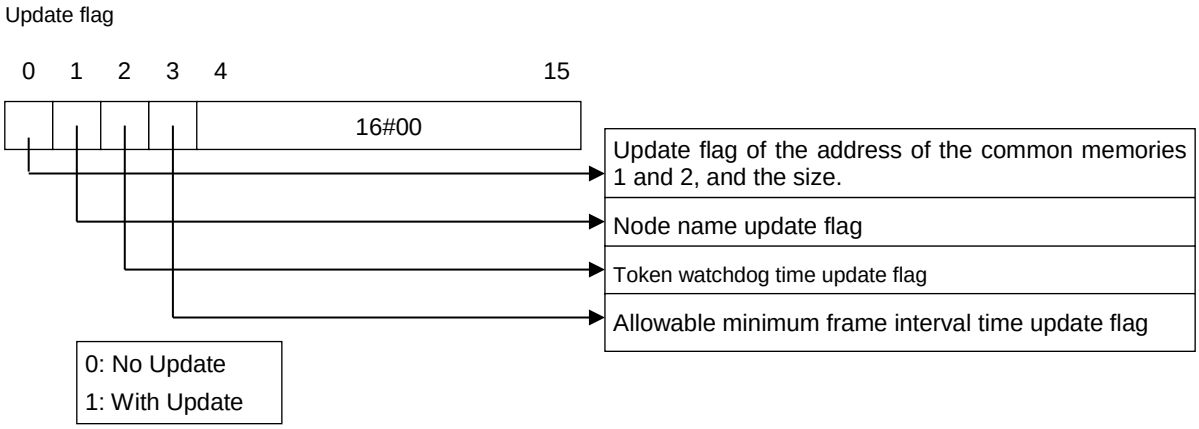


Figure 132 - The data part upon request

b) The data part upon response

The data part upon response shall be as shown in Figure 133.

Response

Offset (Decimal) (octet)	0 7 8 15 16 23 24 31			
0	Node name			
4	Node name			
8	Node name		Reserve	
12	Address of the common memory 1		Size of the common memory 1	
16	Address of the common memory 2		Size of the common memory 2	
20	Token watchdog time	Reserve	Allowable minimum frame interval time	Reserve

Note Data is stored in a little-endian octet order.

Figure 133 - The data part upon response

Details of the data part shall be described in Figure 134.

Items	Setting range	Parameter unit	Notes
Address of the common memory 1	16#0000 to 16#01FF	Word	
Size of the common memory 1	16#0000 to 16#0200	Word	
Address of the common memory 2	16#0000 to 16#1FFF	Word	
Size of the common memory 2	16#0000 to 16#2000	Word	
Node name	Maximum of 10 characters (Rests are filled with Null)	ASCII characters	
Token watchdog time	1 to 255 (decimal)	mSec (1/1 000 seconds)	(Seconds)
Allowable minimum frame interval time	0 to 50 (decimal)	100 µSec (1/1 000 000 seconds)	ex. 50＝5 000 µSec

Note When the parameter outside the setting range is requested, the message result of the header (M_RLT) =1: Abnormal.

Figure 134 - Details of the data part

10.10 Participating Node Management Information Parameter Read Frame

Participating node management information parameter read reads the status information (participating node management information parameter) of each node participating in the network.

10.10.1 The Header Format of the Participating Node Management Information Parameter Read Frame

The header format of the participating node management information parameter read frame shall be as follows.

a) Header structure

The header structure of the participating node management information parameter read frame shall be as shown in Figure 135.

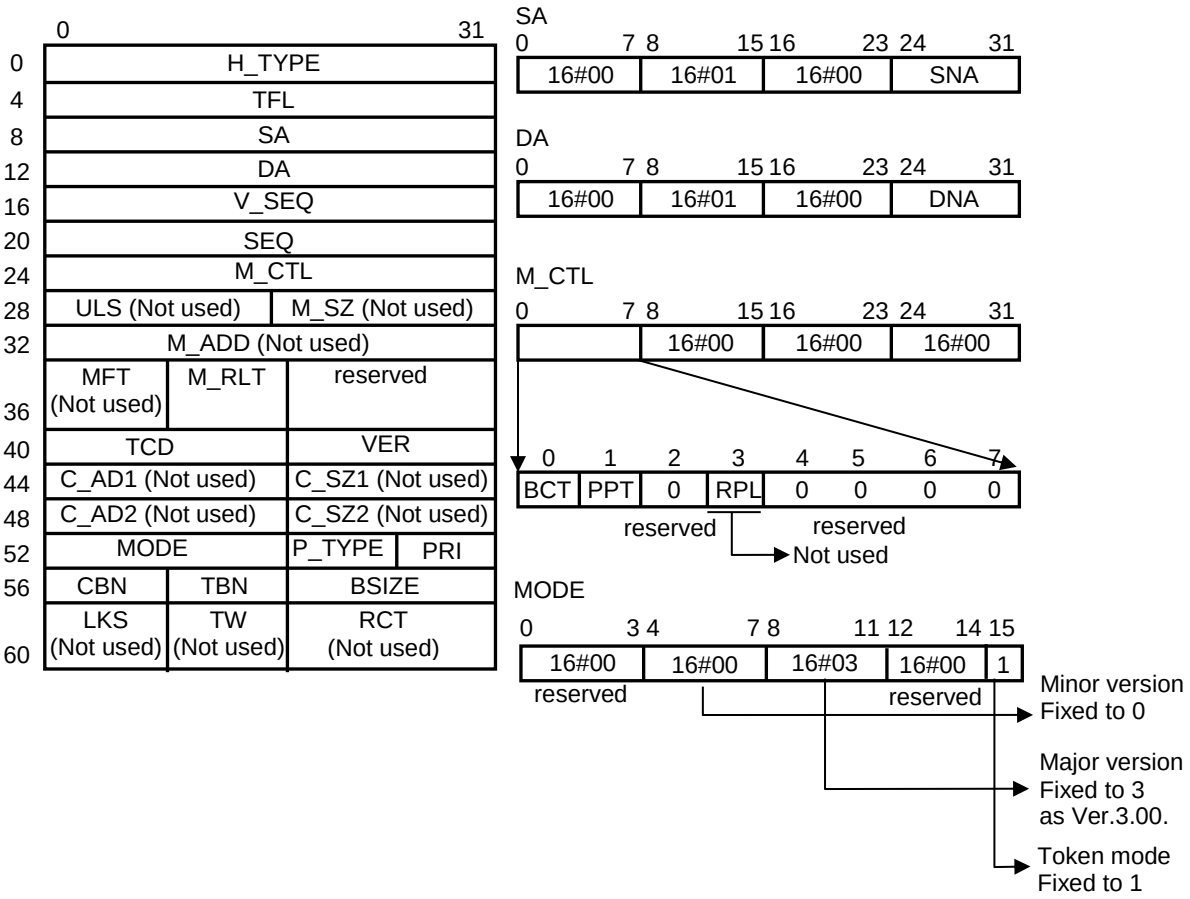


Figure 135 - The header structure of the participating node management information parameter read frame

b) Details of header items

Details of the header items of the participating node management information parameter read frame shall be shown in Table 51.

Table 51 - Details of the header items

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	“FACN”
TFL	4	Octet length including header and data	64 to 84 (decimal)
SNA	1	Node number of source side	1 to 254 (decimal)
DNA	1	Node number of destination side	1 to 254 (decimal)
V_SEQ	4	Version of sequence number	Fixed to 16#0000_0000
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	16#0000_0001 to 16#FFFF_FFFF
BCT	1 bit	Broadcast transmission	0
PPT	1 bit	Peer to peer transmission	1
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	65 025, 65 225 (decimal)
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol, token mode	–
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0000

10.10.2 The Format of the Data Part of the Participating Node Management Information Parameter Read Frame

The format of the data part of the participating node management information parameter read frame shall be as follows.

a) The data part upon request

The data part upon request shall be as shown in Figure 136.

Request

Offset (Decimal)(octet)	0 15	16 31
0	Node number	Reserve

Figure 136 - The data part upon request

b) The data part upon response

The data part upon response shall be as shown in Figure 137.

Response

Offset (Decimal) (octet)	0 15	16 31
0	Node number	Upper layer status
4	Common memory area 1 top address	Common memory area 1 data size
8	Common memory area 2 top address	Common memory area 2 data size
12	Allowable refresh cycle time	Token watchdog time
16	Allowable minimum frame interval time	FA link status

One node

Note Data is stored in a little-endian octet order.

Figure 137 - Data part upon response

Details of the participating node management information parameter shall be as follows.

1) Node number (2 octets): 1 to 254

2) Upper layer status (2 octets): Operating information, error information, etc.

3) Common memory area 1 top address (2 octets): Word address (16#0000 to 16#01FF)

4) Common memory area 1 data size (2 octets): Size (16#0000 to 16#0200)

5) Common memory area 2 top address (2 octets): Word address (16#0000 to 16#1FFF)

6) Common memory area 2 data size (2 octets): Size (16#0000 to 16#2000)

7) Allowable refresh cycle time (2 octets): 0 to 65,535: by 1 ms

8) Token watchdog time (2 octets): 1 to 255: by 1 ms

9) Allowable minimum frame interval time (2 octets): 0 to 50: by 100 μs

10) State of FA link (2 octets): participation and disengagement information, etc. (refer to Figure 138)

Note When requested data is received with parameters other than the node number from 1 to 254, the message result of the header (M_RLT) = 1: Abnormal.

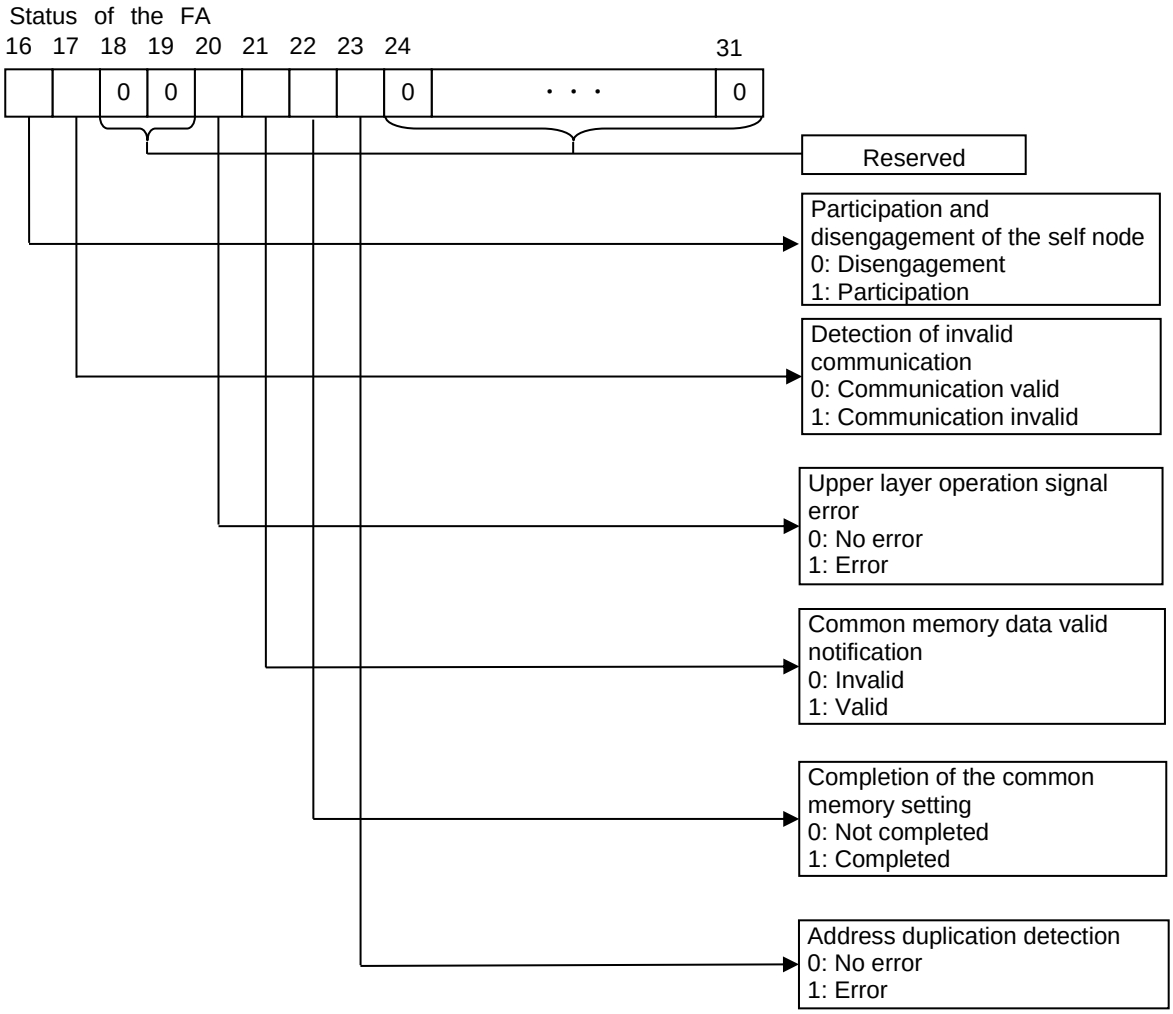


Figure 138 - Details of the FA link status

10.11 Read frame of self node management information parameter

10.11.1 Outline

Self node management information parameter read reads data about settings and operation of the self node which was set by the upper layer of the FA link layer or the network parameter write service when starting the node.

10.11.2 The header format of the self node management information parameter read frame

The header format of the read frame of self node management information parameter shall be as follows.

a) Header structure

The header structure of the read frame of self node management information parameter shall be as shown in Figure 139.

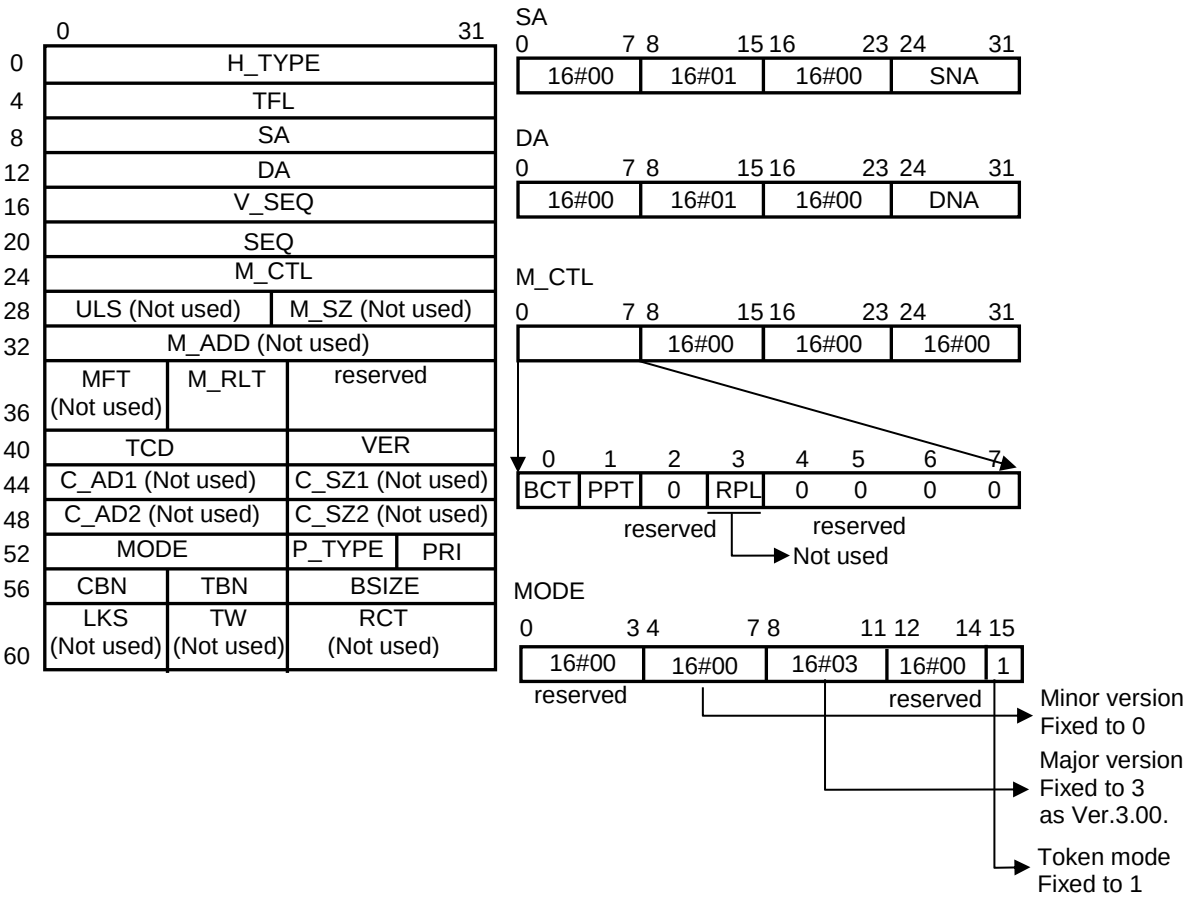


Figure 139 - The header structure of the read frame of self node management information parameter

b) Details of header items

Details of the header item of the read frame of self node management information parameter shall be shown in Table 52.

Table 52 - Details of the header items

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	“FACN”
TFL	4	Octet length including header and data	64 to 128 (decimal)
SNA	1	Node number of source side	1 to 254 (decimal)
DNA	1	Node number of destination side	1 to 254 (decimal)
V_SEQ	4	Version of sequence number	Fixed to 16#0000_0000
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	16#0000_0001 to 16#FFFF_FFFF
BCT	1 bit	Broadcast transmission	0
PPT	1 bit	Peer to peer transmission	1
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	65 026, 65 226 (decimal)
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol, token mode	–
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0000

10.11.3 The Format of the Data Part of the read frame of self node management information parameter

The format of the data part of the read frame of self node management information parameter shall be as follows.

a) The data part upon request

The data part upon request shall be as shown in Figure 140.

Request	Not available

Figure 140 - The data part upon request

b) Data part upon response

The data part upon response shall be as shown in Figure 141.

Response

Offset (Decimal)(octet)	0 15	16 31
0	Node number	Reserve
4	Common memory area 1 top address	Common memory area 1 data size
8	Common memory area 2 top address	Common memory area 2 data size
12	Upper layer status	Token watchdog time
16	Allowable minimum frame interval time	Reserve
20	Vendor code	
24	Vendor code	
28	Vendor code	Reserve
32	Manufacturer model name	
36	Manufacturer model name	
40	Manufacturer model name	Reserve
44	Node name	
48	Node name	
52	Node name	Reserve
56	Protocol type	Status of the FA link
60	Status of the self node	Identification of the self node class

Note Data is stored in a little-endian octet order.

Figure 141 - The data part upon response

Details of the self node management information parameters shall be as follows.

1) Node number (2 octets): 1 to 254

2) Common memory area 1 top address (2 octets): Word address (16#0000 to 16#01FF)

3) Common memory area 1 data size (2 octets): Size (16#0000 to 16#0200)

4) Common memory area 2 top address (2 octets): Word address (16#0000 to 16#1FFF)

5) Common memory area 2 data size (2 octets): Size (16#0000 to 16#2000)

6) Upper layer status (2 octets): Operating information, error information, etc. (refer to Figure 142)

7) Token watchdog time (2 octets): 1 to 255: by 1 ms

8) Allowable minimum frame interval time (2 octets): 0 to 50: by 100 μs

9) Vendor code (10 octets): the name of the vendor (ASCII characters)

10) Manufacturer model name (10 octets): the Manufacturer model name of the node(ASCII characters)

11) Node name of node information (10 octets): the name of the node set by the user (ASCII characters)

12) Protocol type (2 octets): fixed to 16#0080

13) Status of the FA link (2 octets): participation, disengagement, etc. (refer to Figure 143)

14) Status of the self node (2 octets): overlap detection etc. of the node numbers (refer to Figure 144)

15) Identification of the self node class (2 octets): Indicates the node class of the self node (refer to Table 53)

Table 53 - Table of Node class Identification Values

Node class name		Value	Note
Ver.2.00	Ver.2.00	1	Since Ver.2.00 node does not support the command server, it does not return a response.
	Ver.2.00	2	
Ver.3.00	Controller + flexible master	3	For details, refer to the product category of Clause 5.
	Flexible master	4	
	Fixed master	5	
	Flexible slave	6	
	Fixed slave	7	

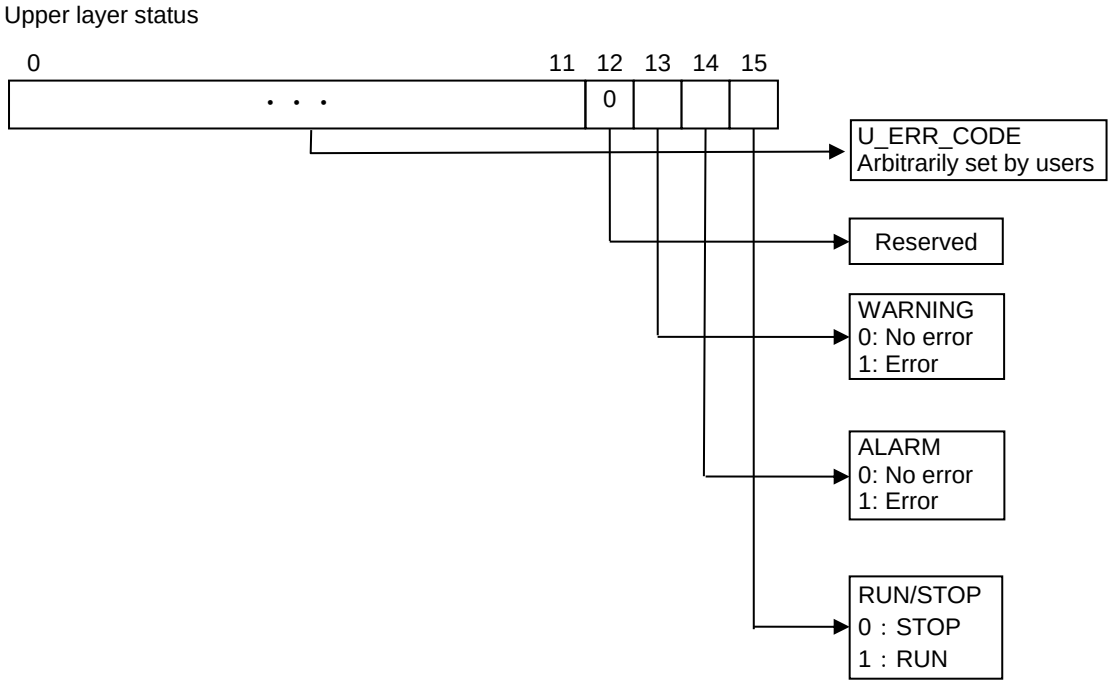


Figure 142 - Details of the upper layer status

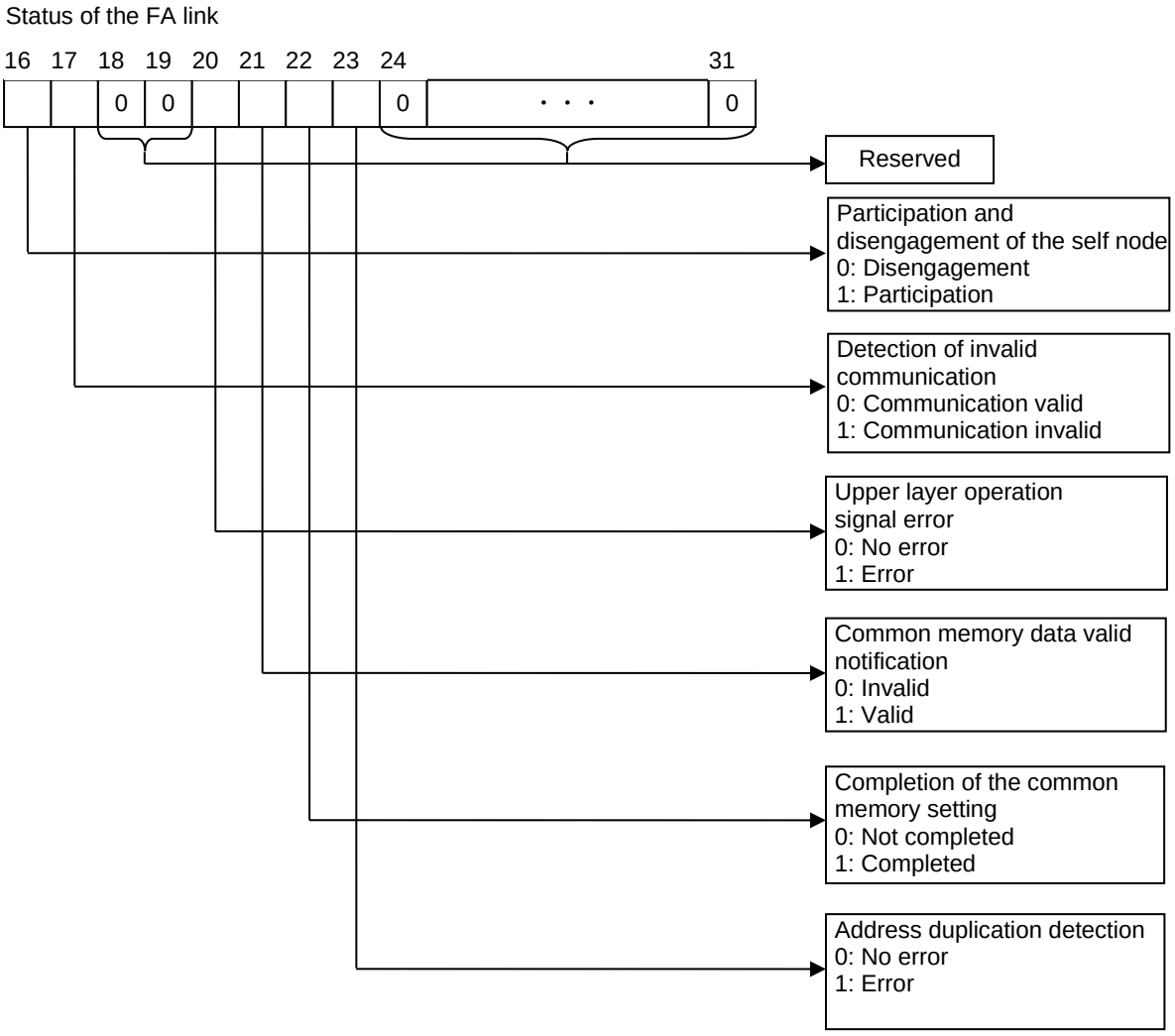


Figure 143 - Details of the FA link status

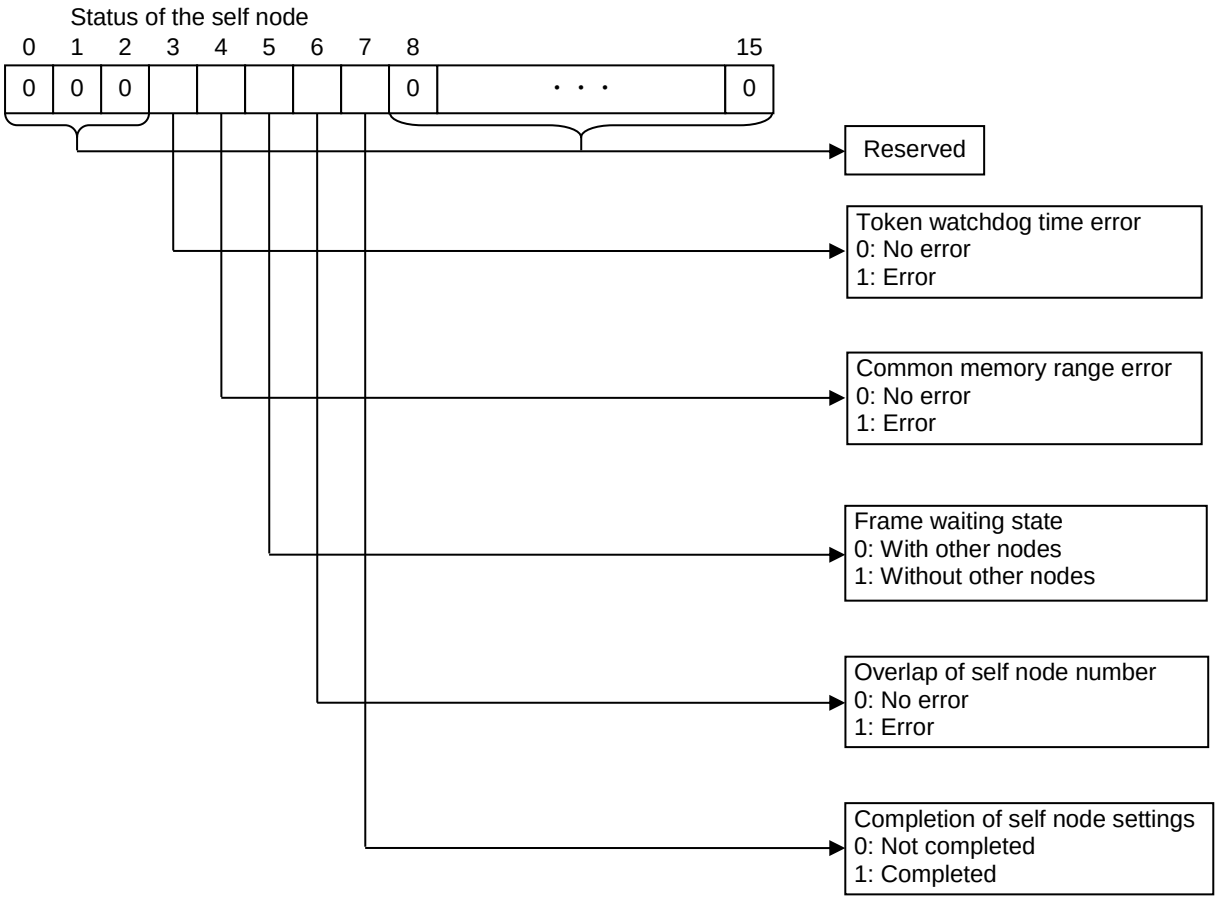


Figure 144 - Details of the self node status

10.12 Read Frame of Self node Configuration Information Parameter

Read node setting information of the node selected by the user.

Values set up by configuration parameter setting can be directly read, which is not the node information flowing in the FL-net network. It is used to confirm parameters set in the node.

10.12.1 The Header Format of the Read Frame of Self node Configuration Information Parameter

The header format of the self node setting information parameter read frame shall be as follows.

a) Header structure

The header structure of the self node setting information parameter read frame shall be as shown in Figure 145.

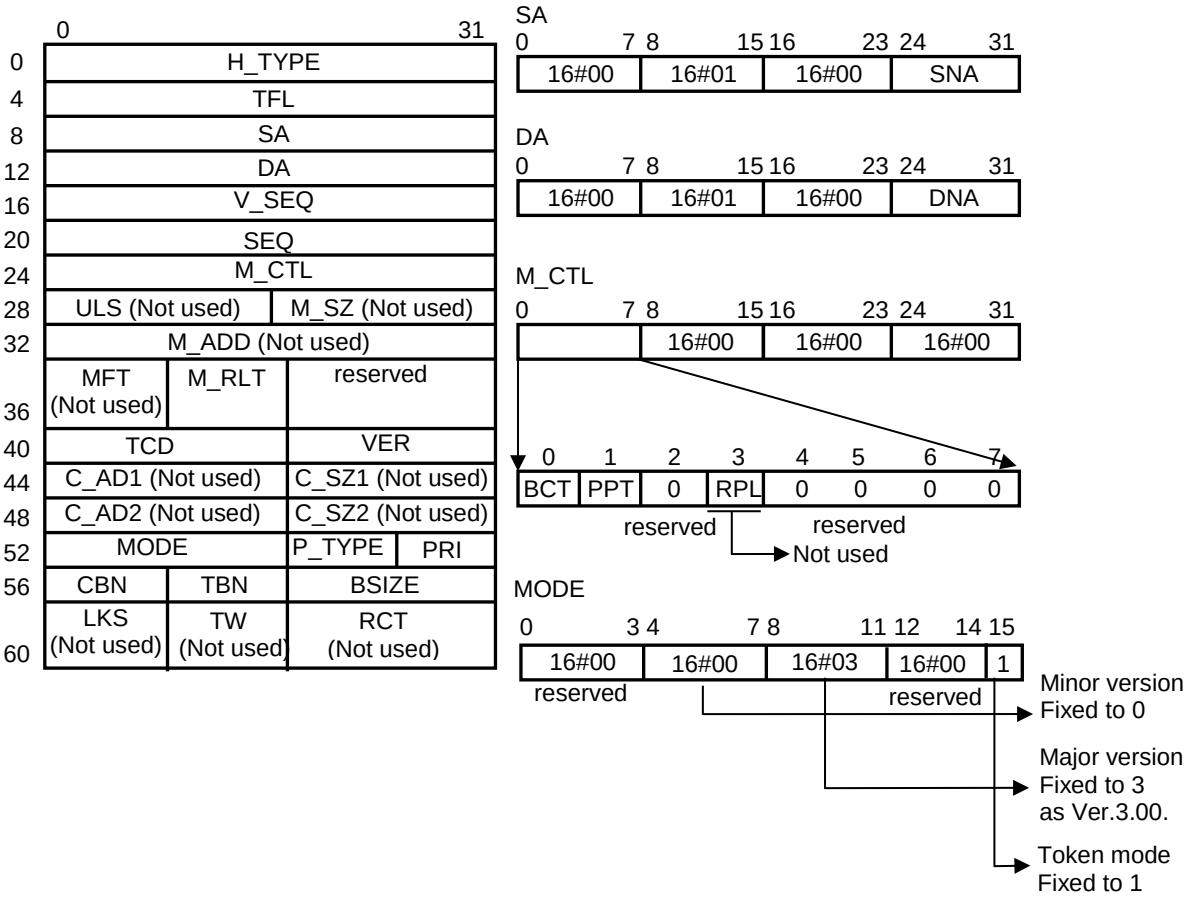


Figure 145 - The header structure of the self node setting information parameter read frame

b) Details of header items

Details of the header items of the self node setting information parameter read frame shall be shown in Table 54.

Table 54 - Details of the header items

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	“FACN”
TFL	4	Octet length including header and data	64 to 88 (decimal)
SNA	1	Node number of source side	1 to 254
DNA	1	Node number of destination side	1 to 254
V_SEQ	4	Version of sequence number	Fixed to 16#0000_0000
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	16#0000_0001 to 16#FFFF_FFFF
BCT	1 bit	Broadcast transmission	0
PPT	1 bit	Peer to peer transmission	1
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	65 027, 65 227
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol, token mode	Refer to Figure 100.
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0000

10.12.2 The Format of the Data Part of the Read Frame of Self node Configuration Information Parameter

In response to a request, a response shall be returned with setting information added to the data part. The format of the data part of the self node setting information parameter read frame shall be as follows.

a) The data part upon request

The data part upon request shall be as shown in Figure 146.

Request	Not available

Figure 146 - The data part upon request

b) Data part upon response

The data part upon response shall be as shown in Figure 147.

Response

Offset (Decimal) (octet)	0 15	16 31
0	Address of the common memory 1	Size of the common memory 1
4	Address of the common memory 2	Size of the common memory 2
8	Token watchdog time	Allowable minimum frame interval time
12	Node name	
16	Node name	
20	Node name	Reserve

Note Data is stored in a little-endian octet order.

Figure 147 - The data part upon response

10.13 Node Reset Frame

A node reset frame is used when a node is reset not by physically shutting power off but by remotely performing it.

Reset operation is performed by resetting the FA link layer and the operation is notified to application when performed.

10.13.1 The Header Format of the Node Reset Frame

The header format of the node reset frame shall be as follows.

a) Header structure

The header structure of the node reset frame shall be as shown in Figure 148.

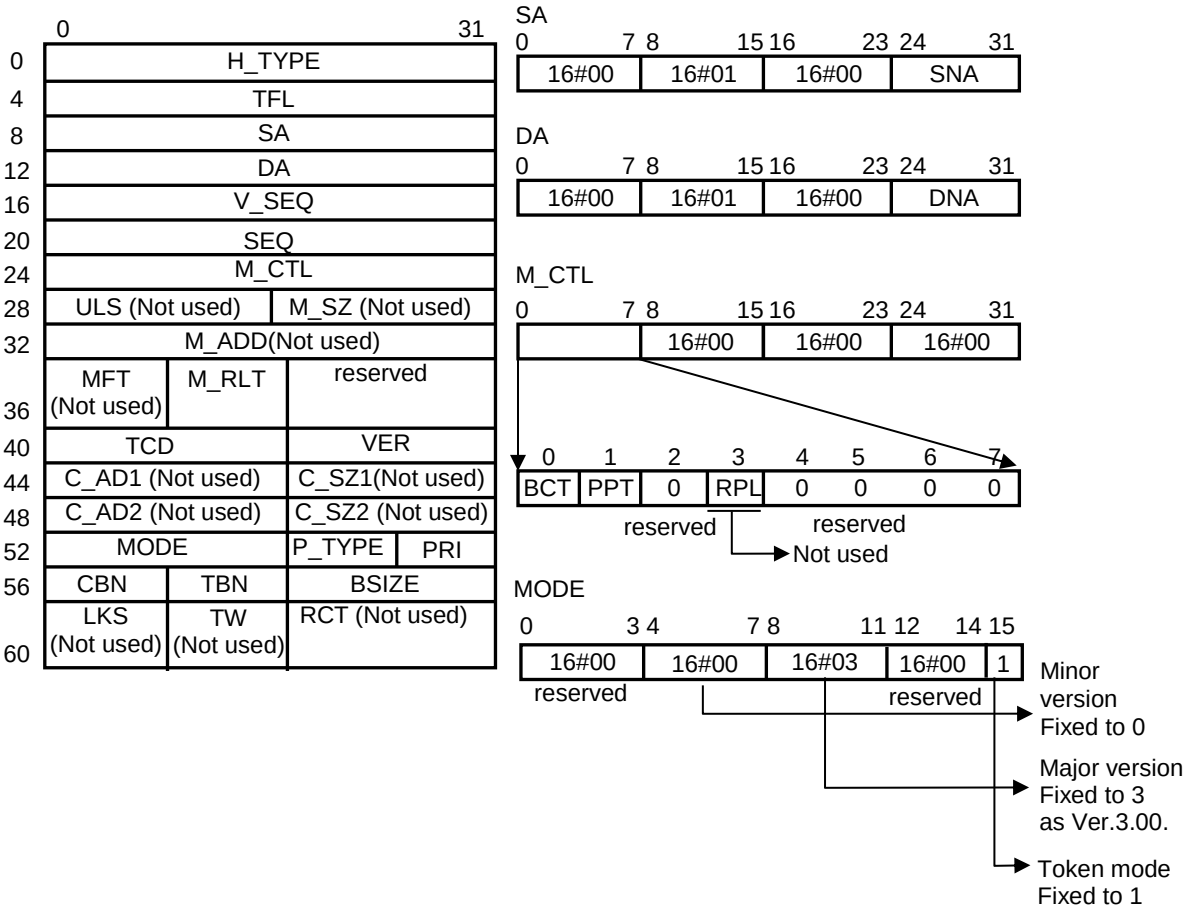


Figure 148 - The header structure of the node reset frame

b) Details of header items

Details of the header items of the node reset frame shall be shown in Table 55.

Table 55 - Details of the header items

Symbols	Size (Octet)	Items	Setting values
H_TYPE	4	Header type	“FACN”
TFL	4	Octet length including header and data	64 (decimal)
SNA	1	Node number of source side	1 to 254
DNA	1	Node number of destination side	1 to 254
V_SEQ	4	Version of sequence number	Fixed to 16#0000_0000
SEQ	4	Sequence number (16#FFFF_FFFF) is followed by (16#0000_0001). Sequence number 0 shall not be used.	16#0000_0001～16#FFFF_FFFF
BCT	1 bit	Broadcast transmission	0
PPT	1 bit	Peer to peer transmission	1
M_RLT	1	Message result (Normal, Abnormal)	Response message 0: Normal, 1: Abnormal, 2: Not implemented
TCD	2	Transaction code	65 028, 65 228
VER	2	Program version	Fixed to 16#0000
MODE	2	FA link protocol, token mode	–
P_TYPE	1	Protocol type	Fixed to 16#80
PRI	1	Message priority	Fixed to 16#00
CBN	1	Current fragment block number	Fixed to 16#01
TBN	1	Total fragment block number	Fixed to 16#01
BSIZE	2	Current block size (octet)	Fixed to 16#0000

10.13.2 The Format of the Data Part of the Node Reset Frame

The format of the data part of the node reset frame shall be as follows.

a) The data part upon request

The data part upon request shall be as shown in Figure 149.

Request	Not available

Figure 149 - The data part upon request

b) Data part upon response

The data part upon response shall be as shown in Figure 150.

Respons	Not available

Figure 150 - The data part upon response

Note Operation when receiving a node reset frame When a node received a node reset frame, it resets the self node after returning a response (refer to Figure 151).

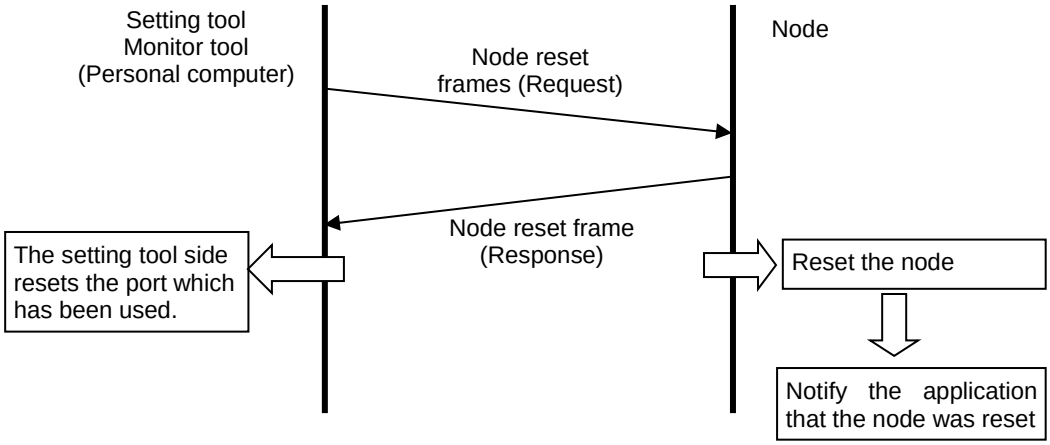


Figure 151 - The receiving sequence of the node reset frame

Appendix A
(Reference)
The State Transition Diagram of the FA Link Protocol

A.1 The State Transition of The Overall Protocol

A.1.1 State Definition

The state definition shall be as follows.

a) Initialization request waiting state

The status for waiting for the upper layer to set initialization information required for participation to the network, such as a self node number.

b) Joining token detection time waiting state

The status for judging whether or not the network link is established.

If a token frame is received even once within this period of time, the network shall be considered to be in the valid linking state.

c) Trigger frame transmission waiting time waiting state

The state for waiting for transmitting a trigger frame when a node starts.

d) Participation request frame acceptance time waiting state

The status for establishing network information with a participation request frame in the participation request acceptance time synchronized with a trigger when newly joining.

e) Three circulation waiting state

The status for collecting network information with the last cyclic frames when joining in progress.

f) Participation request frame transmission waiting state

The status for waiting for transmission of a participation request frame after collection of network information is completed when joining in progress.

g) Token waiting state

The status for not holding a token while participating in the network. The node waits for a token addressed to the self node to be received, and monitors other nodes. Existence of slaves is checked while it is a master.

h) Token retention state

The status after receiving the token frame addressed to the self node before transmitting a token frame addressed to the next node. A solicitation frame shall be transmitted when there are not enough slaves while it is a master. In addition, Existence of slaves is checked.

Note When it is in the joining token detection time waiting state and it is not in the valid linking state, the trigger frame transmission waiting time waiting state shall shift to the participation request frame acceptance time waiting state. This trigger frame transmission waiting time waiting state or the participation request frame acceptance time waiting state shall be called a "network startup state". The valid linking state is identified and the three circulation waiting state shall shift to the participation request frame transmission waiting state. This three circulation waiting state or the participation request frame transmission waiting state shall be called an "in-ring startup state".

i) Waiting for receiving a solicitation frame state

The status for waiting for receiving a solicitation frame when the slave in the flexible setting mode starts in the device level network function.

j) Solicitation frame batch transmission state

The status for transmitting a solicitation frame when the self node is participating to the network while in the network startup state.

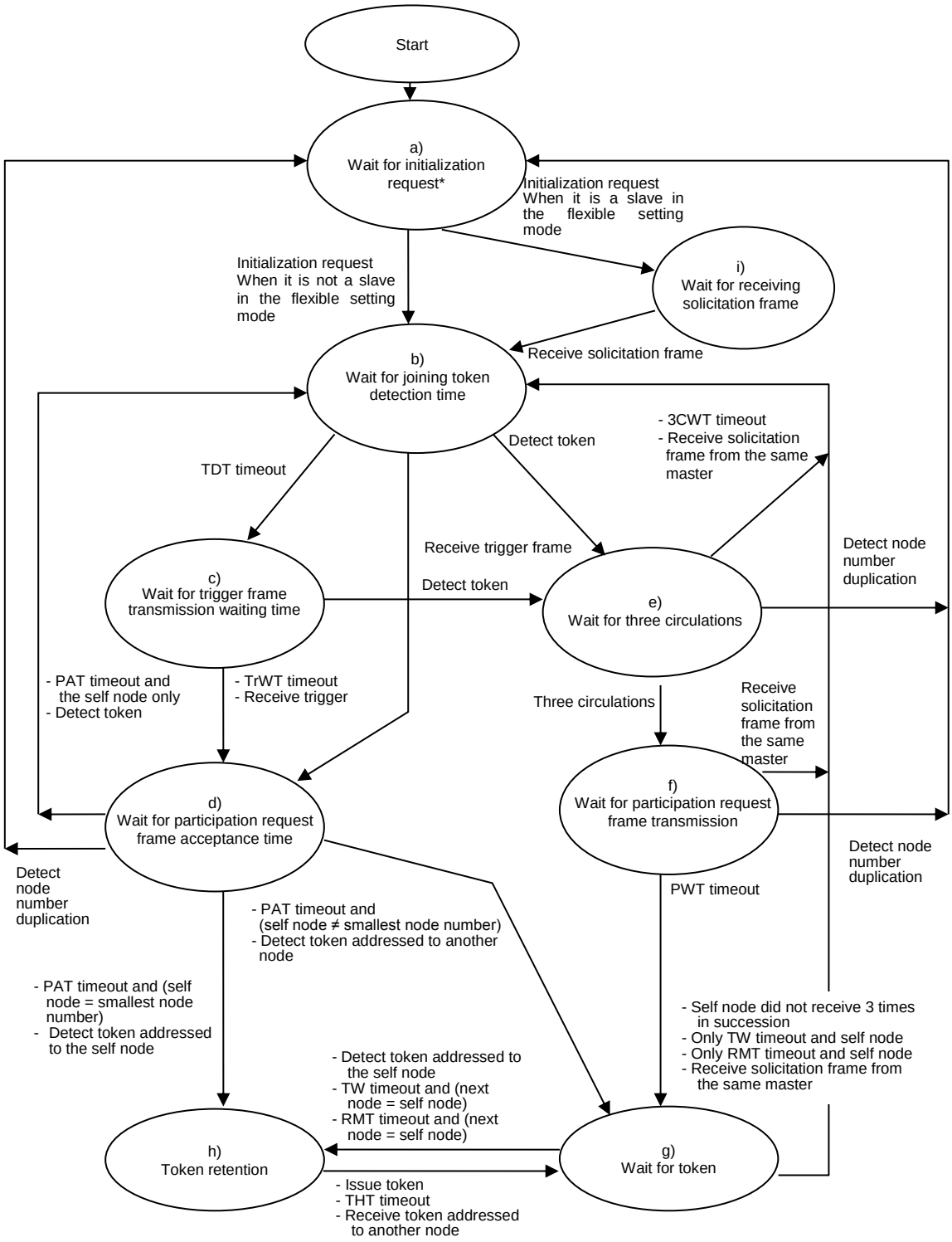
k) Transmitting a solicitation frame state

The status in the valid linking state for transmitting a solicitation frame by solicitation frame transmission waiting time being timed out in the token retention status. It shifts to this status when there are not enough slaves while it is a master.

A.1.2 State Transition Diagram

A.1.2.1 The State Transition Diagram of the Overall Protocol (slave)

The state transition diagram (slave) of the overall FA link protocol shall be as shown in Figure A.1.

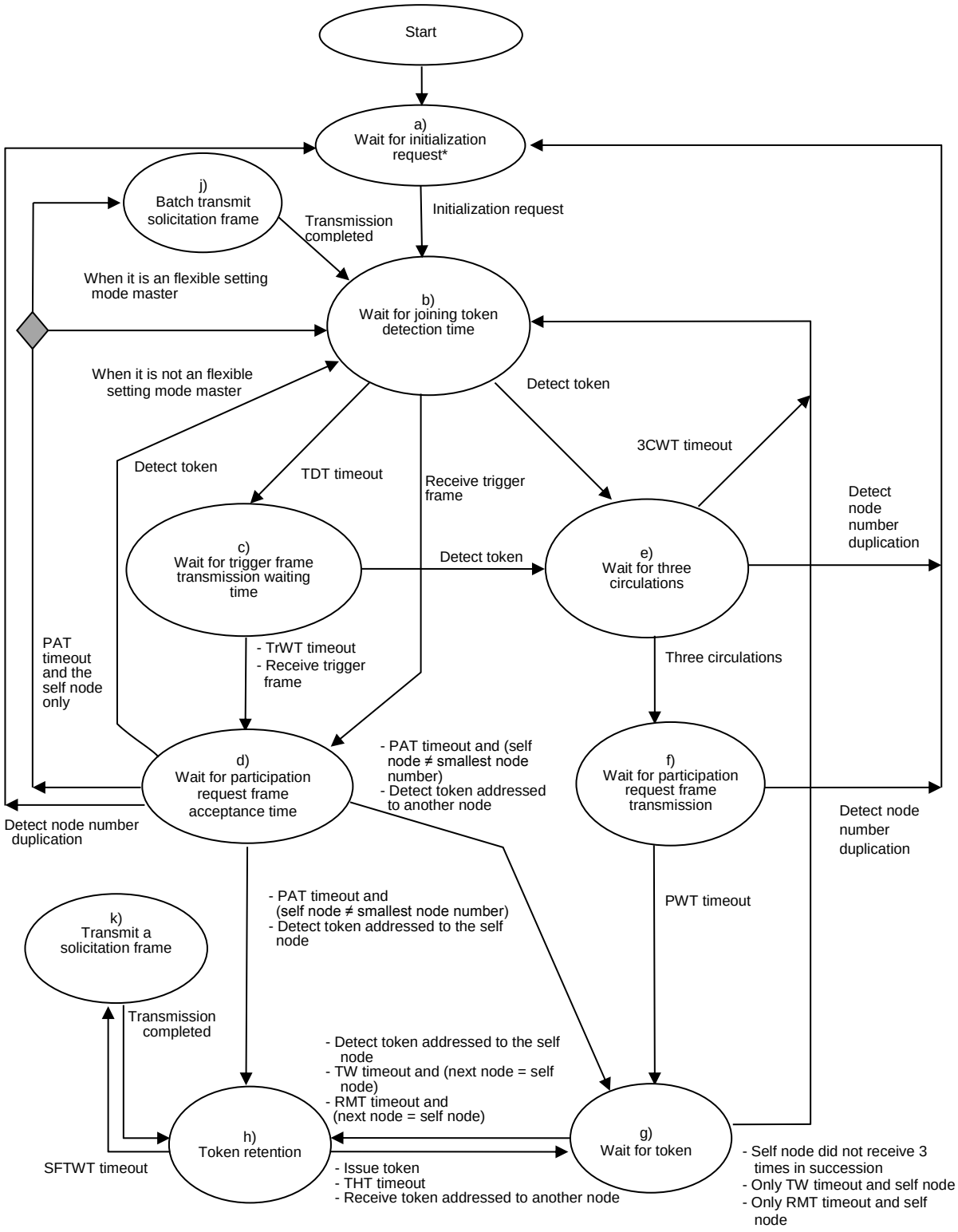


Note* Shift to "a) Wait for initialization request" when the initialization request event is generated.

Figure A.1 - The state transition diagram of the overall protocol (Slave)

A.1.2.2 The State Transition Diagram of the Overall Protocol (master and Ver.2.00 node)

The state transition diagram of the overall FA link protocol (master and Ver.2.00) shall be as shown in Figure A.2.



Note* Shift to "a) Wait for initialization request" when an initialization request event is generated.

Figure A.2 - The state transition diagram of the overall protocol (master and Ver.2.00 node)

A.1.3 State Transition Table

The state transition table of the overall protocol shall be shown in Table A.2. How to interpret the state transition table shall be shown in Table A.1.

Table A.1 - How to interpret the state transition table

State	Event
	Event name
State name	Contents of process
(State number)	Destination of shift
Note Destination of shift is indicated by "=> (State number)".	

Table A.2 - FA link status transition table

State	Event		
	Initialization request from the upper layer	Reception of participation request frame	Reception of trigger
Wait for initialization from the upper layer	Start joining token detection Tm	No process	No process
a)	When it is not an flexible setting slave => b) When it is an flexible setting slave => i)	=> a)	=> a)
Detect joining token	Start joining token detection Tm	No process	- Stop joining token Tm - Start participating transmission Tm - Start participating acceptance Tm
b)	When it is not an flexible setting slave => b) When it is an flexible setting slave => i)	=> b)	=> d)
Wait for trigger frame transmission waiting time	Start joining token detection Tm	No process	- Stop trigger Tm - Start participating transmission Tm - Start participating acceptance Tm
c)	When it is not an flexible setting slave => b) When it is an flexible setting slave => i)	=> c)	=> d)
Wait for participation request frame acceptance time	Start joining token detection Tm	- Update participating TBL - Confirm address overlap - Confirm node number overlap	No process
d)	When it is not an flexible setting slave => b) When it is an flexible setting slave => i)	Node number overlap confirmation is abnormal => a) Other cases => d)	=> d)
Wait for three circulations	Start joining token detection Tm	- Update participating TBL - Confirm address overlap - Confirm node number overlap	No process
e)	When it is not an flexible setting slave => b) When it is an flexible setting slave => i)	Node number overlap confirmation is abnormal => a) Other cases => e)	=> e)
Wait for participation request frame transmission	Start joining token detection Tm	- Update participating TBL - Confirm address overlap - Confirm node number overlap	No process
f)	When it is not an flexible setting slave => b) When it is an flexible setting slave => i)	Node number overlap confirmation is abnormal => a) Other cases => f)	=> f)
Wait for token	Start joining token detection Tm	Update participating TBL	No process
g)	When it is not an flexible setting slave => b) When it is an flexible setting slave => i)	=> g)	=> g)
Token retention	No process	Update participating TBL	No process
h)	=> h)	=> h)	=> h)
Wait for receiving solicitation frame	Start joining token detection Tm	No process	No process
i)	When it is not an flexible setting slave => b) When it is an flexible setting slave => i)	=> i)	=> i)
Batch transmit solicitation frame	1) When it is not an flexible setting slave Start joining token detection Tm 2) When it is an flexible setting slave No process	No process	No process
j)	When 1) => b) When 2) => i)	=> j)	=> j)
Transmit a solicitation frame	Start joining token detection Tm	No process	No process
k)	When it is not an flexible setting slave => b) When it is an flexible setting slave => i)	=> k)	=> k)

Table A.2 - FA link status transition table (Continued)

State	Event		
	Reception of token addressed to another node	Reception of token addressed to the self node	Reception of cyclic frame (TBN≠CBN)
Wait for initialization from the upper layer	No process	No process	No process
a)	=> a)	=> a)	=> a)
Detect joining token	Start three circulation waiting Tm	Start three circulation waiting Tm	No process
b)	=> e)	=> e)	=> b)
Wait for trigger frame transmission waiting time	Start three circulation waiting Tm	Start three circulation waiting Tm	No process
c)	=> e)	=> e)	=> c)
Wait for participation request frame acceptance time	1) after transmitting participation request frames Start token watchdog Tm 2) before transmitting participation request frames Start joining token Tm	1) after transmitting participation request frames - Start token watchdog Tm - Start solicitation frame transmission waiting Tm 2) before transmitting participation request frames Start joining token Tm	No process
d)	When 1) => g) When 2) => b)	When 1) => h) When 2) => b)	=> d)
Wait for three circulations	Confirm node number overlap	Confirm node number overlap	No process
e)	three circulations of the smallest node token => f) node number overlap confirmation is abnormal => a) before three circulations of the smallest node token => e)	three circulations of the smallest node token => f) node number overlap confirmation is abnormal => a) before three circulations of the smallest node token => e)	=> e)
Wait for participation request frame transmission	Confirm node number overlap	Confirm node number overlap	No process
f)	Node number overlap confirmation is abnormal => a) Node number overlap confirmation is normal => f)	Node number overlap confirmation is abnormal => a) Node number overlap confirmation is normal => f)	=> f)
Wait for token	- Start token watchdog Tm - Watch another node leaving from the network - Watch a token addressed to the self node - Start solicitation frame transmission waiting Tm	- Start token retention Tm - Start allowable minimum frame interval Tm - Start refresh cycle measurement Tm - Start solicitation frame transmission waiting Tm	Cyclic process
g)	Self node did not receive 3 times in succession => b) Other cases => g)	=> h)	=> g)
Token retention	1) Self node number > node number of recept token - Discard self node token - Stop token retention Tm 2) Self node number < node number of recept token - Ignore recept token	No process	No process
h)	When 1) => g) When 2) => h)	=> h)	=> h)
Wait for receiving solicitation frame	No process	No process	No process
i)	=> i)	=> i)	=> i)
Batch transmit solicitation frame	No process	No process	No process
j)	=> j)	=> j)	=> j)
Transmit a solicitation frame	No process	No process	No process
k)	=> k)	=> k)	=> k)

Table A.2 - FA link status transition table (Continued)

State	Event		
	Reception of cyclic frame (TBN=CBN)	Reception of message frame	Joining token detection timeout
Wait for initialization from the upper layer	No process	No process	–
a)	=> a)	=> a)	
Detect joining token	No process	No process	Start trigger Tm
b)	=> b)	=> b)	=> c)
Wait for trigger frame transmission waiting time	No process	No process	–
c)	=> c)	=> c)	
Wait for participation request frame acceptance time	- Update participating TBL - Confirm address overlap - Confirm node number overlap	No process	–
d)	Node number overlap confirmation is abnormal => a) Other cases => d)	=> d)	
Wait for three circulations	- Update participating TBL - Confirm address overlap - Confirm node number overlap	No process	–
e)	Node number overlap confirmation is abnormal => a) Other cases => e)	=> e)	
Wait for participation request frame transmission	- Update participating TBL - Confirm address overlap - Confirm node number overlap	No process	–
f)	Node number overlap confirmation is abnormal => a) Other cases => f)	=> f)	
Wait for token	- Update participating TBL - Cyclic process	Message process	–
g)	=> g)	=> g)	
Token retention	No process	No process	–
h)	=> h)	=> h)	
Wait for receiving solicitation frame	No process	No process	–
i)	=> i)	=> i)	
Batch transmit solicitation frame	No process	No process	–
j)	=> j)	=> j)	
Transmit a solicitation frame	No process	No process	–
k)	=> k)	=> k)	

Table A.2 - FA link status transition table (Continued)

State	Event		
	Token watchdog timeout	Refresh cycle timeout	participating transmission timeout
Wait for initialization from the upper layer a)	–	–	–
Detect joining token b)	–	–	–
Wait for trigger frame transmission waiting time c)	–	–	–
Wait for participation request frame acceptance time d)	–	–	Transmit participation request frame => d)
Wait for three circulations e)	–	–	–
Wait for participation request frame transmission f)	–	–	- Transmit participation request frame - Start token watchdog Tm - Start solicitation frame transmission waiting Tm => g)
Wait for token g)	1) next node = self node Start token retention Tm 2) Only the self node in the network Start joining token detection Tm When 1) => h) When 2) => b)	1) next node = self node Start token retention Tm 2) Only the self node in the network Start joining token detection Tm When 1) => h) When 2) => b)	–
Token retention h)	–	–	–
Wait for receiving solicitation frame i)	–	–	–
Batch transmit solicitation frame j)	–	–	–
Transmit a solicitation frame k)	–	–	–

Table A.2 - FA link status transition table (Continued)

State	Event		
	Transmission of trigger timeout	Participation request acceptance timeout(PAT timeout)	Three circulation waiting timeout
Wait for initialization from the upper layer a)	–	–	–
Detect joining token b)	–	–	–
Wait for trigger frame transmission waiting time c)	- Transmit trigger - Start participating transmission Tm - Start participating acceptance Tm => d)	–	–
Wait for participation request frame acceptance time d)	–	Confirm the smallest node number [a)] 1) self node = smallset node Start token retention Tm 2) self node ≠smallest node Start token watchdog Tm Start solicitation frame transmission waiting Tm 3) the self node alone and it is not a flexible setting master Start joining token detection Tm 4) the self node alone and it is a flexible setting master Start solicitation frame transmission waiting Tm When 1) => h) When 2) => g) When 3) => b) When 4) => j)	–
Wait for three circulations e)	–	–	Start joining token detection Tm => b)
Wait for participation request frame transmission f)	–	–	–
Wait for token g)	–	–	–
Token retention h)	–	–	–
Wait for receiving solicitation frame i)	–	–	–
Batch transmit solicitation frame j)	–	–	–
Transmit a solicitation frame k)	–	–	–

Table A.2 - FA link status transition table (Continued)

State	Event		
	Minimum frame interval timeout	Token retention timeout	Solicitation frame batch transmission frame interval timeout (SFBTFI)
Wait for initialization from the upper layer a)	–	–	–
Detect joining token b)	–	–	–
Wait for trigger frame transmission waiting time c)	–	–	–
Wait for participation request frame acceptance time d)	–	–	–
Wait for three circulations e)	–	–	–
Wait for participation request frame transmission f)	–	–	–
Wait for token g)	–	–	–
Token retention h)	1) Last transmission data - Transmit token frame to next node - Stop token retention Tm 2) Not last transmission data - Transmit a frame - Start minimum frame interval Tm	Start token watchdog Tm	–
	When 1) => g) When 2) => h)	=> g)	
Wait for receiving solicitation frame i)	–	–	–
Batch transmit solicitation frame j)	–	–	Batch transmit solicitation frame[e)] (continued from divided transmission with batch transmission)
			=> j)
Transmit a solicitation frame k)	–	–	–

Table A.2 - FA link status transition table (Continued)

State	Event		
	Solicitation frame transmission waiting time timeout (SFTWT)	Reception of solicitation frames	Transmission of solicitation frame completed
Wait for initialization from the upper layer a)	–	No process => a)	–
Detect joining token b)	–	Reallocate area when it is a slave[b)] => b)	–
Wait for trigger frame transmission waiting time c)	–	–	–
Wait for participation request frame acceptance time d)	–	–	–
Wait for three circulations e)	–	Reallocate area when it is a slave[b)] => b)	–
Wait for participation request frame transmission f)	–	Reallocate area when it is a slave[b)] => b)	–
Wait for token g)	–	Reallocate area when it is a slave[b)] => b)	–
Token retention h)	When the self node is an flexible setting master[d)] -Check slaves -Transmit divided solicitation frame => k)	–	–
Wait for receiving solicitation frame i)	–	IO registration Master node status area set => b)	–
Batch transmit solicitation frame j)	–	–	1) Transmission of all solicitation frames completed[c)] No process 2) Transmission of all solicitation frames is not completed[c)] Start solicitation frame batch transmission frame interval Tm When 1) => b) When 2) => j)
Transmit a solicitation frame k)	–	–	Start solicitation frame transmission waiting Tm => h)

Table A.2 - FA link status transition table (Continued)

Notes	"No process" exists as a status, but no process shall be performed. "-" does not exist as a status, and no process shall be performed.
Note a)	- When the self node equal the smallest node, start the token retention Tm(THT) and shift to h). - When the self node different the smallest node, start the token watchdog Tm(TW) and shift to g). - When the self node alone and it is not a flexible setting master, start the joining token detection Tm(TDT) and shift to b). - When the self node alone and it is a flexible setting master, start solicitation frame batch transmission frame interval(SFBTFI) and shift to j).
b)	When it is a slave node, it shifts when receiving the solicitation frame from its master (the same master) or when the Lock ID of the solicitation frame matches. When it is not the self master and the solicitation frame whose Lock ID does not match is received, no process is performed.
c)	When the solicitation frame batch transmission is in progress and divided transmission of frames is in progress, the next divided frame shall be transmitted. When a divided frame is the last divided frame, shift to b) after transmission.
d)	As for the condition to shift to the solicitation frame transmission state k), when the slave is not established and the refresh measurement time (RMT) is not more than 90% of the allowable refresh cycle time (RCT) while holding a token, it shifts to k) alternately with message transmission and one solicitation frame shall be transmitted. When RMT is larger than 90% of RCT and smaller than RCT, it shifts to k) on condition that the message or the solicitation frame was not transmitted last time, and one solicitation frame shall be transmitted. When RMT is not less than RCT, it does not shift to k), and the solicitation frame shall not be transmitted.

A.2 The State Transition of Message Transmission

A.2.1 State Definition

The state definition shall be as follows.

a) Message transmission request waiting state

The status for waiting for a request to send a message from the upper layer.

b) Message transmission waiting state

The status after receiving a message transmission request from the upper layer and before it obtains a token.

c) ACK reception waiting state

The status after message is transmitted before the message acknowledge waiting time (AWT) times out

A.2.2 State Transition Diagram

The state transition diagram of message transmission shall be as shown in Figure A.3.

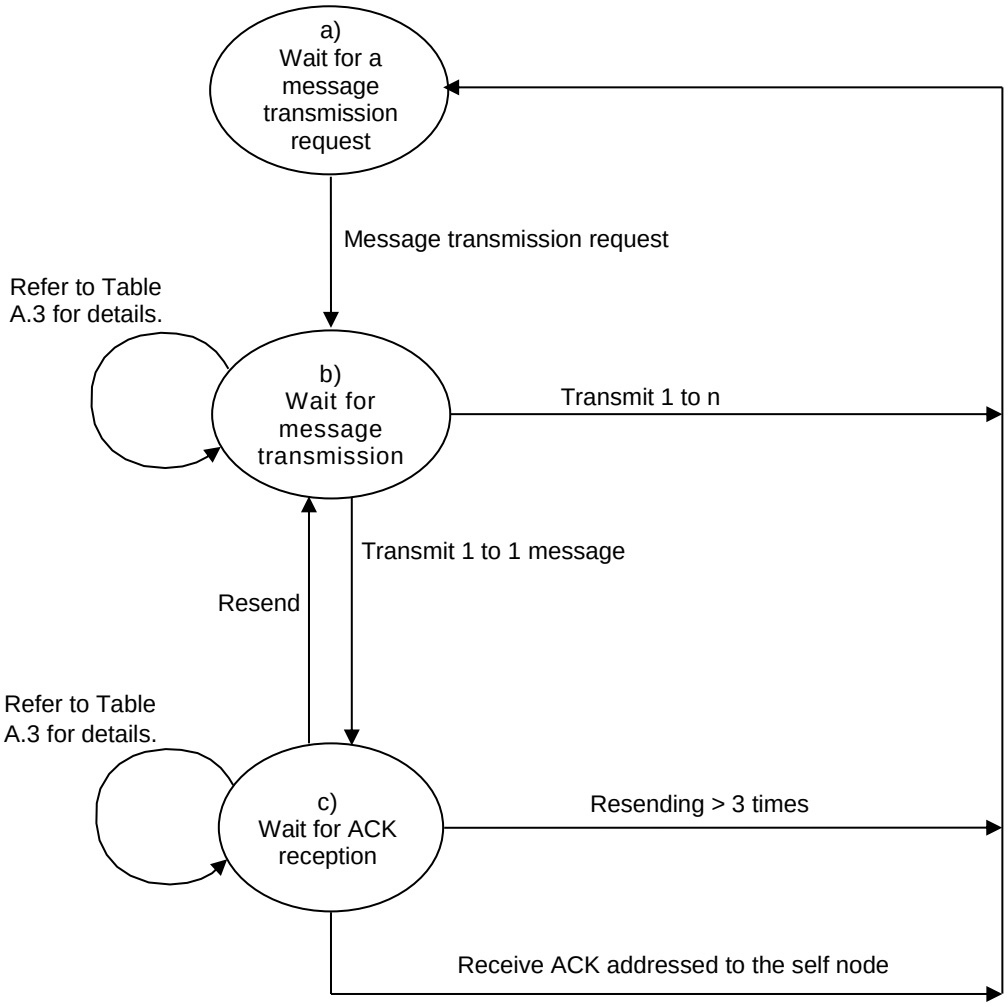


Figure A.3 - The state transition of message transmission

A.2.3 State Transition Table

The state transition table when transmitting a message shall be shown in Table A.3.

Table A.3 - The state transition table when transmitting a message

State	Event			
	Message transmission request from the upper layer	Receive ACK addressed to the self node	AWT timeout	Obtain token
Wait for message transmission request from the upper layer a)	Message transmission request => b)	No process => a)	No process => a)	No process => a)
Wait for a message transmission waiting time b)		No process => b)	No process => b)	1) When X[a)] - Transmit message - Start the waiting for ACK reception timer 2) No process when Y[b)] When 1) and 1 to n => a) When 1) and 1 to 1 => c) When 2) =>b)
Wait for ACK reception c)		- Confirm the state - Notify the upper layer of completion => a)	1) No process shall be performed when not having three circulations 2) No process shall be performed when there are three or more circulations and the number of resending is less than 3 times 3) Abnormality is notified to the upper layer when there are three or more circulations and the number of resending is 3 times or more When 1) => c) When 2) => b) When 3) => a)	1) No process shall be performed when it is less than ACK waiting time 2) No process shall be performed when the ACK waiting time timed out, there are 3 or more circulations, and the number of resending is less than 3 times. 3) No process shall be performed when the ACK waiting time timed out and there are less than 3 circulations 4) Abnormality is notified to the upper layer when ACK waiting time timed out, there are 3 or more circulations, and the number of resending is 3 or more times When 1) => c) When 2) => b) When 3) => c) When 4) => a)

Note [a)] X: When any one of the following conditions is satisfied.
1) RMT≦RCT×0.9
2) RCT × 0.9 < RMT < RCT and no message was transmitted last time

[b)] Y: When any one of the following conditions is satisfied.
1) RMT≧RCT
2) RCT × 0.9 < RMT < RCT and a message was transmitted last time

A.3 State Transition of Message Reception

A.3.1 State Definition

The state definition shall be as follows.

a) Initial setting from the upper layer waiting state

The state for waiting for an initial setting from the upper layer.

b) Message reception waiting state

The state after setting the initial setting from the upper layer and before receiving a message addressed to the self node.

c) ACK transmission waiting state

The state after receiving a message and before obtaining a token to transmit ACK.

A.3.2 State Transition Diagram

The state transition diagram of message reception shall be as shown in Figure A.4.

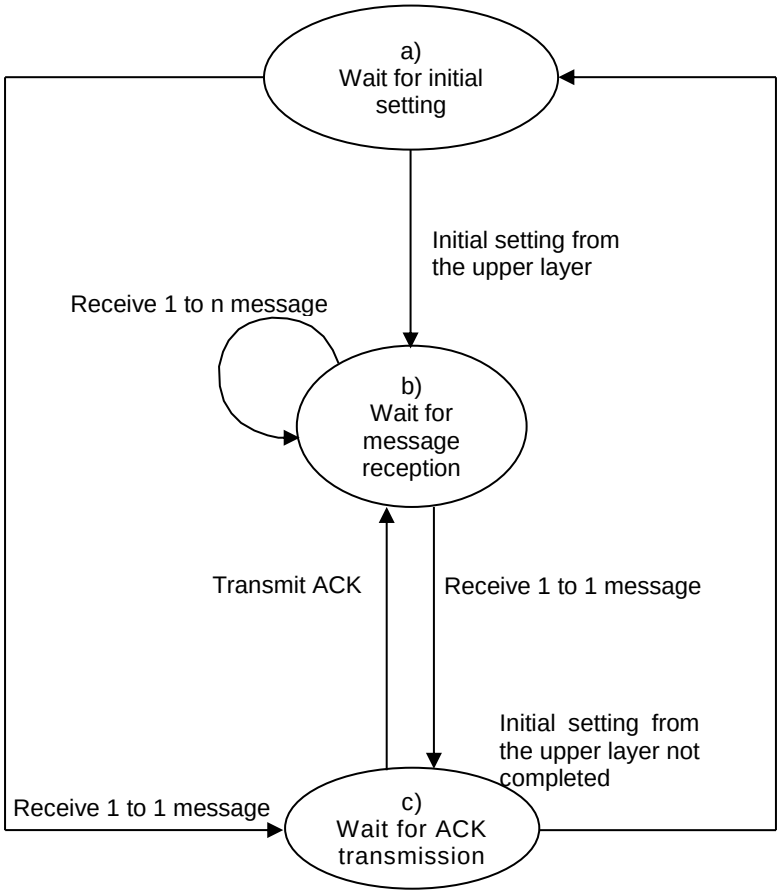


Figure A.4 - The state transition of message reception

A.3.3 State Transition Table

The state transition table when receiving a message shall be shown in Table A.4.

Table A.4 - The state transition table when receiving a message

State	Event		
	Initial setting from the upper layer	Receive a message addressed to the self node	Obtain token
Wait for initial setting from the upper layer a)	Clear message buffer => b)	Discard message Request ACK (3) => c)	No process => a)
Wait for message reception b)		Sequence number confirmation (= new) 1 to 1 message 1) When the message reception buffer is full - Request ACK (02) - Discard message 2) When the format is abnormal - Request ACK (06) - Discard message 3) Version of sequence number confirmation - Request ACK (05) - Discard message 4) Other than the above - Request ACK (01) - Notify to the upper layer. - 1 to n message 5) When the message reception buffer is full - Discard message 6) When the format is abnormal - Discard message 7) Version of sequence number confirmation - Discard message 8) Other than the above - Notify to the upper layer. - Sequence number confirmation (= the same message as the last time) 9) 1 to 1 message - Request ACK (01) - Discard message 10) 1 to n message - Discard message 1) to 4), 9) => c) 5) to 8), 10) => b)	No process => b)
Wait for ACK transmission c)			Transmit ACK Initial setting not completed => a) Initial setting completed => b)

Appendix B
(Reference)
Log data

Log data include required items and optional. Optional items can be freely decided by vendors, but items to be implemented shall be published. The contents of the log include the followings.

a) Transmission and reception related items

— The total number of transmission at the socket part

— The total number of errors of transmission at the socket part

— The total number of reception at the socket part

— The total number of reception errors at the socket part

b) Lower layer related items

— The number of transmission timeout

— The number of discarded frames

— The number of receiving CRC error frames

— The number of receiving abnormal size frames

c) Cyclic transmission related items

— Number of errors in receiving cyclic frame

d) Message transmission related items

— The number of times message transmission resending was over

— Number of errors in receiving message frame

— The number of resending messages

e) ACK related items

— The number of ACK errors

f) Token related items

— The number of detection of duplicated token

— Number of the events of discarding token

— Number of the events of reissuing token

g) State related items

— The number of frame waiting state

— The number of joining

— The number of self node disconnection

— The number of self node disconnection by skipped token

— The number of recognizing disconnection of other nodes

Log information and a location shall be shown in Table B.1.

Each data is 4-octet data. Offset is indicated by 4 octets.

Table B.1 - Log information and location

Offset	Name	Description	Implementation
0	The total number of transmission at the socket part	Transmission and reception related data	Required
4	The total number of errors of transmission at the socket part	Transmission and reception related data	Required
8	The number of Ethernet transmission errors	Transmission and reception related data	Optional
12	–	Transmission and reception related data (Reserved)	–
16	–	Transmission and reception related data (Reserved)	–
20	–	Transmission and reception related data (Reserved)	–
24	The total number of reception at the socket part	Transmission and reception related data	Required
28	The total number of reception errors at the socket part	Transmission and reception related data	Required
32	The number of Ethernet reception errors	Transmission and reception related data	Optional
36	–	Transmission and reception related data (Reserved)	–
40	–	Transmission and reception related data (Reserved)	–
44	–	Transmission and reception related data (Reserved)	–
48	The number of token transmission	Frame type related data	Optional
52	The number of cyclic frame transmission	Frame type related data	Optional
56	The number of 1 to 1 message transmission	Frame type related data	Optional
60	The number of 1 to n message transmission	Frame type related data	Optional
64	–	Frame type related data (Reserved)	–
68	–	Frame type related data (Reserved)	–
72	The number of times tokens were received	Frame type related data	Optional
76	Number of the events of receiving cyclic frame	Frame type related data	Optional
80	The number of times 1 to 1 messages were received	Frame type related data	Optional
84	The number of times 1 to n message were received	Frame type related data	Optional
88	–	Frame type related data (Reserved)	–
92	–	Frame type related data (Reserved)	–
96	Number of errors in receiving cyclic frame	Cyclic transmission related data	Required
100	The number of time of cyclic address size errors	Cyclic transmission related data	Optional
104	The number of cyclic CBN errors	Cyclic transmission related data	Optional
108	The number of cyclic TBN errors	Cyclic transmission related data	Optional
112	The number of cyclic BSIZE errors	Cyclic transmission related data	Optional
116	Detection time for error in receiving cyclic frame[a)]	Time when a cyclic transmission receiving error was detected	Optional
120	–	Cyclic transmission related data (Reserved)	–
124	–	Cyclic transmission related data (Reserved)	–
128	–	Cyclic transmission related data (Reserved)	–
132	–	Cyclic transmission related data (Reserved)	–
136	–	Cyclic transmission related data (Reserved)	–
140	–	Cyclic transmission related data (Reserved)	–
144	The number of resending messages	Message transmission related data	Required
148	The number of times message transmission resending was over	Message transmission related data	Required
152	Detection time for the maximum refresh cycle[a)]	Time when the largest value was detected while measuring the refresh cycle	Optional

Table B.1 - Log information and location (Continued)

Offset	Name	Description	Implementation
156	–	Message transmission related data (Reserved)	–
160	–	Message transmission related data (Reserved)	–
164	–	Message transmission related data (Reserved)	–
168	Number of errors in receiving message frame[a)]	Message transmission related data	Required
172	The number of message sequence number errors	Message transmission related data	Optional
176	The number of recognizing resent message sequence numbers	Message transmission related data	Optional
180	Detection time for error in receiving message frame[b)]	The time when a message transmission and reception error was detected	Optional
184	–	Message transmission related data (Reserved)	–
188	–	Message transmission related data (Reserved)	–
192	The number of ACK errors	ACK related data	Required
196	The number of ACK version of sequence number errors	ACK related data	Optional
200	The number of ACK sequence number errors	ACK related data	Optional
204	The number of ACK node number errors	ACK related data	Optional
208	The number of ACK TCD errors	ACK related data	Optional
212	–	ACK related data (Reserved)	–
216	–	ACK related data (Reserved)	–
220	–	ACK related data (Reserved)	–
224	–	ACK related data (Reserved)	–
228	–	ACK related data (Reserved)	–
232	–	ACK related data (Reserved)	–
236	–	ACK related data (Reserved)	–
240	The number of detection of duplicated token	Token related data	Required
244	Number of the events of discarding token	Token related data	Required
248	Number of the events of reissuing token	Token related data	Required
252	Time when the event of discarding token was detected last[a)]	The time when the number of times tokens were discarded was detected	Optional
256	Time when reissuance of token occurred last [a)]	The time when the number of times tokens were reissued was detected	Optional
260	Time when the token retention timeout event occurred last [a)]	The time when the number of token retention timeout was counted up	Optional
264	Number of the events of token retention timeout	Token related data	Optional
268	Number of the events of token watchdog timeout	Token related data	Optional
272	Time when the token watchdog timeout event occurred last[a)]	The time when the number of token watchdog timeout was counted up	Optional
276	Maximum token retention time [a)]	The maximum token retention time measured after receiving a token retention time measurement start and before it received the stop	Optional
280	Minimum token retention time[a)]	Minimum token retention time measured after receiving a token retention time measurement start and before it received the stop	Optional
284	Detection time for the maximum token retention time[a)]	The time when the maximum token retention time was updated	Optional
288	Total operation time	Operation state of node, date on joining and disengagement	Optional
292	The number of frame waiting state	Operation state of node, date on joining and disengagement	Required
296	The number of joining	Operation state of node, date on joining and disengagement	Required
300	The number of self node disconnection	Operation state of node, date on joining and disengagement	Required
304	The number of self node disconnection by skipped token	Operation state of node, date on joining and disengagement	Required

Table B.1 - Log information and location (Continued)

Offset	Name	Description	Implementation
308	The number of recognizing disconnection of other nodes	Operation state of node, date on joining and disengagement	Required
312	Measuring time for token retention time[a)]	The time from when the token retention time measurement started to when it was terminated	Optional
316	Number of tokens while measuring the token retention time[a)]	The number of tokens received during token retention time measurement	Optional
320	–	(Reserved)	–
324	–	(Reserved)	–
328	–	(Reserved)	–
332	Measuring time for general purpose communication data sender log[b)]	The time from when the general purpose communication data sender log measurement started to when it was terminated	Optional
336	List of participation confirmation node	Participation confirmation node related data	Optional
340	List of participation confirmation node	Participation confirmation node related data	Optional
344	List of participation confirmation node	Participation confirmation node related data	Optional
348	List of participation confirmation node	Participation confirmation node related data	Optional
352	List of participation confirmation node	Participation confirmation node related data	Optional
356	List of participation confirmation node	Participation confirmation node related data	Optional
360	List of participation confirmation node	Participation confirmation node related data	Optional
364	List of participation confirmation node	Participation confirmation node related data	Optional
368	IP 1 [b)]	The IP address of the general purpose communication data sender log 1	Optional
372	IP 1 receive counter[b)]	The number of reception from the IP of the general purpose communication data sender log 1	Optional
376	IP 2 [b)]	The IP address of the general purpose communication data sender log 2	Optional
380	IP 2 receive counter[b)]	The number of reception from the IP of the general purpose communication data sender log 2	Optional
384	IP 3 [b)]	The IP address of the general purpose communication data sender log 3	Optional
388	IP 3 receive counter[b)]	The number of reception from the IP of the general purpose communication data sender log 3	Optional
392	IP 4 [b)]	The IP address of the general purpose communication data sender log 4	Optional
396	IP 4 receive counter[b)]	The number of reception from the IP of the general purpose communication data sender log 4	Optional
400	IP 5 [b)]	The IP address of the general purpose communication data sender log 5	Optional
404	IP 5 receive counter[b)]	The number of reception from the IP of the general purpose communication data sender log 5	Optional
408	IP 6 [b)]	The IP address of the general purpose communication data sender log 6	Optional
412	IP 6 receive counter[b)]	The number of reception from the IP of the general purpose communication data sender log 6	Optional
416	IP 7 [b)]	The IP address of the general purpose communication data sender log 7	Optional
420	IP 7 receive counter[b)]	The number of reception from the IP of the general purpose communication data sender log 7	Optional
424	IP 8 [b)]	The IP address of the general purpose communication data sender log 8	Optional
428	IP 8 receive counter[b)]	The number of reception from the IP of the general purpose communication data sender log 8	Optional
432	IP 9 [b)]	The IP address of the general purpose communication data sender log 9	Optional
436	IP 9 receive counter[b)]	The number of reception from the IP of the general purpose communication data sender log 9	Optional
440	IP 10 [b)]	The IP address of the general purpose communication data sender log 10	Optional
444	IP 10 receive counter[b)]	The number of reception from the IP of the general purpose communication data sender log 10	Optional

Table B.1 - Log information and location (Continued)

Offset	Name	Description	Implementation
448	–	Vendor definition possible area related data (Reserved)	–
452	–	Vendor definition possible area related data (Reserved)	–
456	–	Vendor definition possible area related data (Reserved)	–
460	–	Vendor definition possible area related data (Reserved)	–
464	–	Vendor definition possible area related data (Reserved)	–
468	–	Vendor definition possible area related data (Reserved)	–
472	–	Vendor definition possible area related data (Reserved)	–
476	–	Vendor definition possible area related data (Reserved)	–
480	–	Vendor definition possible area related data (Reserved)	–
484	–	Vendor definition possible area related data (Reserved)	–
488	–	Vendor definition possible area related data (Reserved)	–
492	–	Vendor definition possible area related data (Reserved)	–
496	–	Vendor definition possible area related data (Reserved)	–
500	–	Vendor definition possible area related data (Reserved)	–
504	–	Vendor definition possible area related data (Reserved)	–
508	–	Vendor definition possible area related data (Reserved)	–
Notes [a] Response data for token retention time measurement [b] Response data for general purpose communication data sender log measurement			

Appendix C
(Reference)
Port Number List

Table C.1 - Port number list

Functoins		Protocol types	Port numbers
FA link protocol	Receive token frames Receive cyclic frames	UDP	55 000
	Receive message frames	UDP	55 001
	Receive trigger frames Receive participation request frames Receive solicitation frames	UDP	55 002
	Transmit FA link protocol	UDP	55 003
Device level network	Transmit solicitation frames	UDP	55 003
General purpose command server	Receive general purpose command	TCP/UDP	55 004
	Transmit general purpose command	TCP/UDP	55 004