BSR/ASHRAE Addendum bb to
ANSI/ASHRAE Standard 135-2012

Public Review Draft

Proposed Addendum bb to Standard 135-2012, BACnet® - A Data Communication Protocol for Building Automation and Control Networks

First Public Review (January 2015)
(Draft shows Proposed Changes to Current Standard)

This draft has been recommended for public review by the responsible project committee. To submit a comment on this proposed standard, go to the ASHRAE website at www.ashrae.org/standards-research--technology/public-review-drafts and access the online comment database. The draft is subject to modification until it is approved for publication by the Board of Directors and ANSI. Until this time, the current edition of the standard (as modified by any published addenda on the ASHRAE website) remains in effect. The current edition of any standard may be purchased from the ASHRAE Online Store at www.ashrae.org/bookstore or by calling 404-636-8400 or 1-800-727-4723 (for orders in the U.S. or Canada).

This standard is under continuous maintenance. To propose a change to the current standard, use the change submittal form available on the ASHRAE website, www.ashrae.org.

The appearance of any technical data or editorial material in this public review document does not constitute endorsement, warranty, or guaranty by ASHARE of any product, service, process, procedure, or design, and ASHRAE expressly disclaims such.

ASHRAE, 1791 Tullie Circle, NE, Atlanta GA 30329-2305

[This foreword and the "rationales" on the following pages are not part of this standard. They are merely informative and do not contain requirements necessary for conformance to the standard.]

FOREWORD

The purpose of this addendum is to present a proposed change for public review. These modifications are the result of change proposals made pursuant to the ASHRAE continuous maintenance procedures and of deliberations within Standing Standard Project Committee 135. The proposed changes are summarized below.

135-2012bb-1 Zero Config MAC Addresses for MS/TP, p. 3

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2012 and Addenda is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout. Only this new and deleted text is open to comment at this time. All other material in this addendum is provided for context only and is not open for public review comment except as it relates to the proposed changes.

135-2012*bb*-1 Zero Config MAC Addresses for MS/TP

Rationale

This addendum was created to support creation of low cost MS/TP devices that do not include hardware (LCD and buttons, DIP switch, numeric dials, jumpers, etc.), for configuring their MAC address.

[Insert new **9.4.1** Automatic Addressing of MS/TP Nodes, p. 94]

9.4.1 Automatic Addressing of MS/TP Nodes

While many MS/TP nodes will be configured with a static address through DIP switches, rotary dials, or proprietary software configuration, certain implementations may be dynamically configured through an automatic addressing mechanism.

A "zero-configuration auto-addressing mode" node is an MS/TP master node that does not begin with a preconfigured MS/TP address and chooses an MS/TP address after monitoring the frames for an unused address. The node will use the Test Request and Test Response frames for address confirmation, by including some information in the Data portion of the frame that is unique to the node. Additionally, the node will constantly monitor all MS/TP frames and will choose another MAC address if it sees another node using its chosen MAC address. Once the node chooses and confirms an MS/TP MAC address, it shall use this address in the token passing. It is required to maintain this address across device restarts and loss of power and use the address as its preferred address when it restarts. The node shall use a range of MS/TP addresses from 64 to 127 (64 addresses), which allows fixed addresses to exist from 0 to 63 (64 addresses). The node shall use a Max_Master that is fixed at 127.

[add to **9.5.2**, Variables p. 95]

9.5.2 Variables

...

ZeroConfigurationMode	*A Boolean flag set to TRUE if this node is a zero configuration master node.*
Address List	*A list, used by a ZeroConfigurationMode node, that tracks the used and available MS/TP addresses, along with which devices have passed a Token or Poll For Master frame (master nodes).*
ReceivedValidFrameNotForUs	*Boolean flag set to TRUE by the Receive State Machine if a valid frame is received that is not addressed to TS. Set to FALSE by the Master node state machine.*
ZeroConfigurationAddress	*A Master node address, chosen randomly from the Address List. A successful chosen address shall be stored in non-volatile memory across device restarts and loss of power, and shall be used during the first attempt to confirm the address. Subsequent address confirmation attempts, if needed, must use a randomly chosen address to prevent deadlock. Success is indicated by responding to a BACnet Data Expecting Reply or a BACnet Data Not Expecting Reply frame for nodes that have a BACnet Application layer.*
ZeroConfigurationUUID	*A 16 octet UUID (Universally Unique IDentifier) that conforms to RFC-4122.*

[add to **9.5.3** Parameters, p. 96]

9.5.3 Parameters

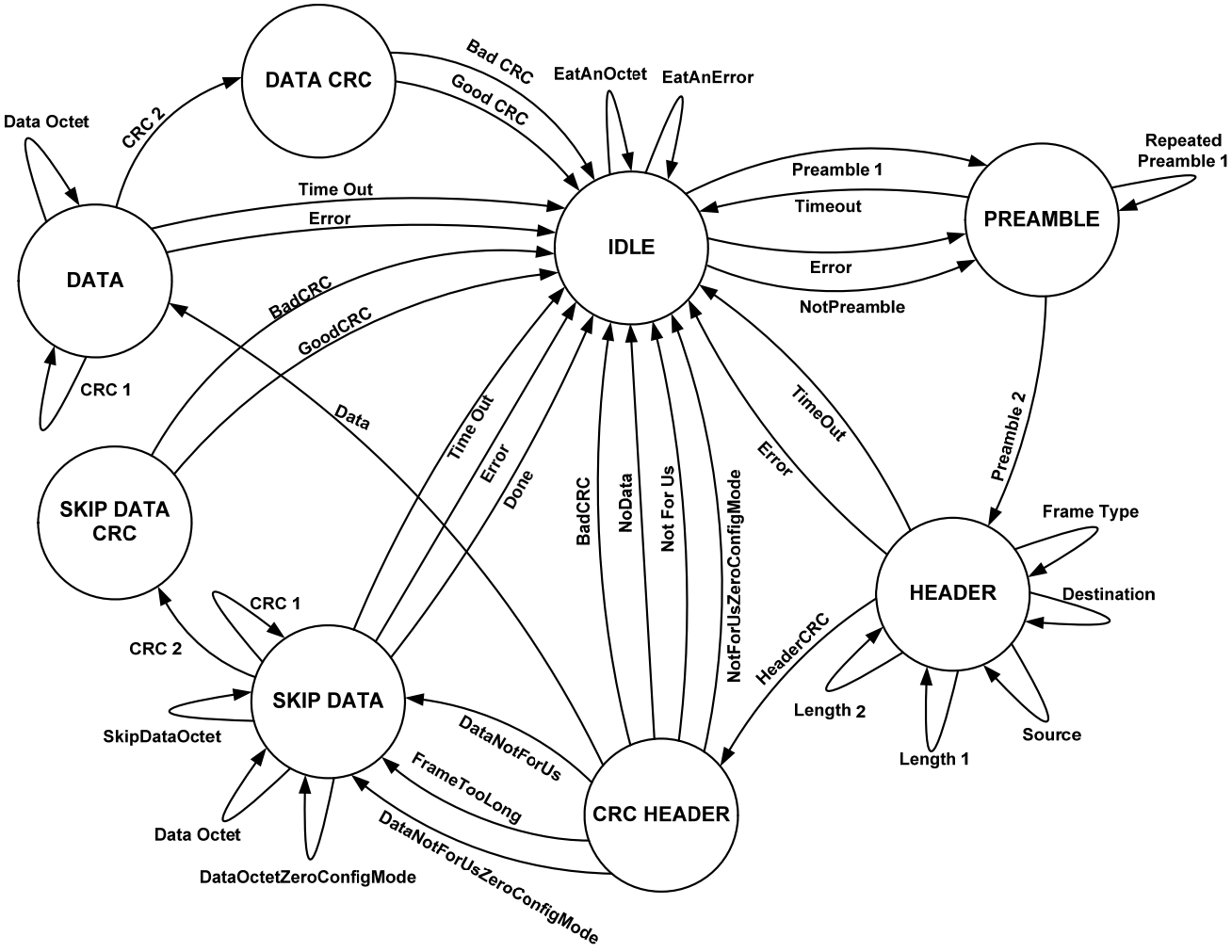
...

$T_{confirmation\_timeout}$ *The minimum time without a DataAvailable or ReceiveError event that a node must wait for a station to begin replying to a Test Request: 300 milliseconds.*

...

[Change **Figure 9-3**. Receive Frame State Machine, p. 97]

[add transition line NotForUsZeroConfigurationMode from HEADER_CRC to IDLE]
[add transition line DataNotForUsZeroConfigurationMode from HEADER_CRC to SKIP_DATA]
[add transition line DataOctetZeroConfigurationMode from SKIP_DATA to SKIP_DATA]
[add transition line SkipDataOctet from SKIP_DATA to SKIP_DATA]
[add transition line CRC1 from SKIP_DATA to SKIP_DATA]
[add transition line CRC2 from SKIP_DATA to SKIP_DATA_CRC]
[add state SKIP_DATA_CRC]
[add transition line BadCRC from SKIP_DATA_CRC to IDLE]
[add transition line GoodCRC from SKIP_DATA_CRC to IDLE]



[Change **9.5.4.4**, HEADER_CRC, p. 99]

9.5.4.4 HEADER_CRC

In the HEADER_CRC state, the node validates the CRC on the fixed message header.

BadCRC
If the value of HeaderCRC is not X '55',

then set ReceivedInvalidFrame to TRUE to indicate that an error has occurred during the reception of a frame, and enter the IDLE state to wait for the start of the next frame.

NotForUs
If the value of the HeaderCRC is X '55' and DataLength is zero and the value of DestinationAddress is not equal to either TS (this station) or 255 (broadcast) *and ZeroConfigurationMode is FALSE*,

then enter the IDLE state to wait for the start of the next frame.

NotForUsZeroConfigurationMode

If the value of the HeaderCRC is X '55' and DataLength is zero and the value of DestinationAddress is not equal to either TS (this station) or 255 (broadcast) and ZeroConfigurationMode is TRUE,

then set ReceivedValidFrameNotForUs to TRUE to indicate that a frame with no data has been received and is not for us, and enter the IDLE state to wait for the start of the next frame.

DataNotForUs
If the value of the HeaderCRC is X '55' and DataLength is not zero and the value of DestinationAddress is not equal to either TS (this station) or 255 (broadcast) *and ZeroConfigurationMode is TRUE,*

then set Index to zero and enter the SKIP_DATA state to consume the data and data CRC portions of the frame.

DataNotForUsZeroConfigurationMode

If the value of the HeaderCRC is X '55' and DataLength is not zero and the value of DestinationAddress is not equal to either TS (this station) or 255 (broadcast) and ZeroConfigurationMode is TRUE,

then set Index to zero; set DataCRC to X'FFFF'; and enter the SKIP_DATA state to receive the data and data CRC portions of the frame.

...

[change **9.5.4.7,** SKIP_DATA, p. 101]

9.5.4.7 SKIP_DATA

In the SKIP_DATA state, the node waits for the data portion of a frame to be received so that its contents can be ~~ignored.~~ *either ignored or consumed if ZeroConfigurationMode is TRUE.*

...

DataOctet
If ReceiveError is FALSE and DataAvailable is TRUE and Index is less than DataLength+1 *and ZeroConfigurationMode is TRUE*,

then set DataAvailable to FALSE; set SilenceTimer to zero; increment Index by 1; and enter the SKIP_DATA state.

Done

If ReceiveError is FALSE and DataAvailable is TRUE and Index is equal to DataLength+1 *and ZeroConfigurationMode is FALSE,*

then set DataAvailable to FALSE; set SilenceTimer to zero; and enter the IDLE state to wait for the start of the next frame.

DataOctetZeroConfigurationMode

If ReceiveError is FALSE and DataAvailable is TRUE and Index is less than DataLength+1 and Index is less than InputBufferSize and ZeroConfigurationMode is TRUE,

then set DataAvailable to FALSE; set SilenceTimer to zero; accumulate the contents of DataRegister into DataCRC; save the contents of DataRegister at InputBuffer[Index]; increment Index by 1; and enter the SKIP_DATA state.

SkipDataOctet

If ReceiveError is FALSE and DataAvailable is TRUE and Index is less than DataLength+1 and Index is greater than or equal to InputBufferSize and ZeroConfigurationMode is TRUE,

then set DataAvailable to FALSE; set SilenceTimer to zero; accumulate the contents of DataRegister into DataCRC; increment Index by 1; and enter the SKIP_DATA state.

CRC1

If ReceiveError is FALSE and DataAvailable is TRUE and Index is equal to DataLength and ZeroConfigurationMode is TRUE,

then set DataAvailable to FALSE; set SilenceTimer to zero; accumulate the contents of DataRegister into DataCRC; increment Index by 1; and enter the SKIP_DATA state.

CRC2

If ReceiveError is FALSE and DataAvailable is TRUE and Index is equal to DataLength plus 1 and ZeroConfigurationMode is TRUE,

then set DataAvailable to FALSE; set SilenceTimer to zero; accumulate the contents of DataRegister into DataCRC; and enter the SKIP_DATA_CRC state.

[add to **9.5.4**, Receive Frame Finite State Machine, p. 97]

9.5.4.X SKIP_DATA_CRC

In the SKIP_DATA_CRC state, the node validates the CRC of the message data not for us if ZeroConfigurationMode is TRUE.

BadCRC

If the value of DataCRC is not X'F0B8',

then set ReceivedInvalidFrame to TRUE to indicate that an error has occurred during the reception of a frame, and enter the IDLE state to wait for the start of the next frame.

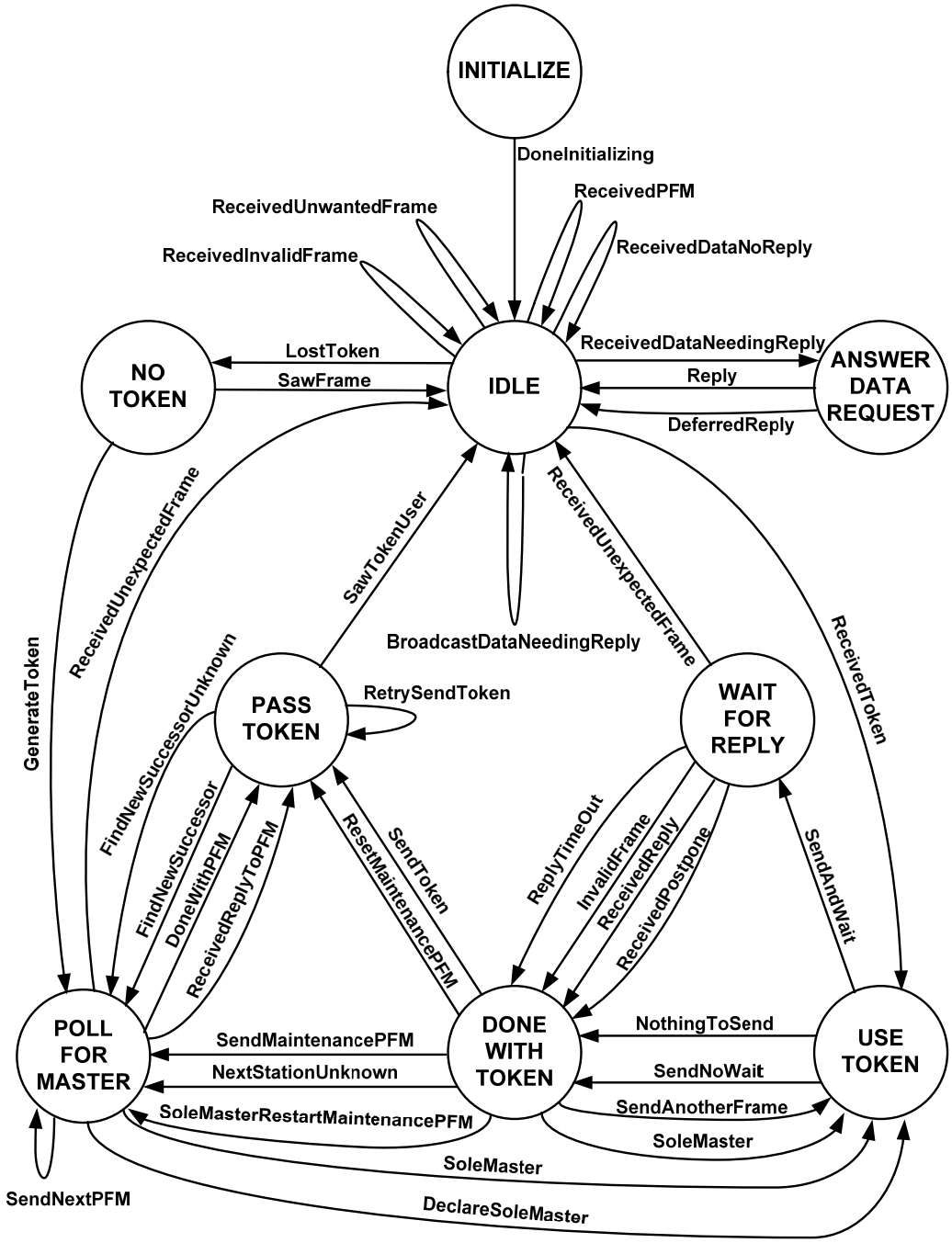
GoodCRC

If the value of DataCRC is X'F0B8',

then set ReceivedValidFrameNotForUs to TRUE to indicate the complete reception of a valid frame that is not for us, and enter the IDLE state to wait for the start of the next frame.

[change Figure 9-4. Master Node State Machine., p. 103]

[add transition InitializeZeroConfigurationMaster from INITIALIZE to ZERO_CONFIGURATION_IDLE]
[add state ZERO_CONFIGURATION_IDLE and transitions]
[add state ZERO_CONFIGURATION_PFM and transitions]
[add state ZERO_CONFIGURATION_TOKEN and transitions]
[add state ZERO_CONFIGURATION_CONFIRM and transitions]



[add **Figure 9-4b** Master Node State Machine, p. 103]

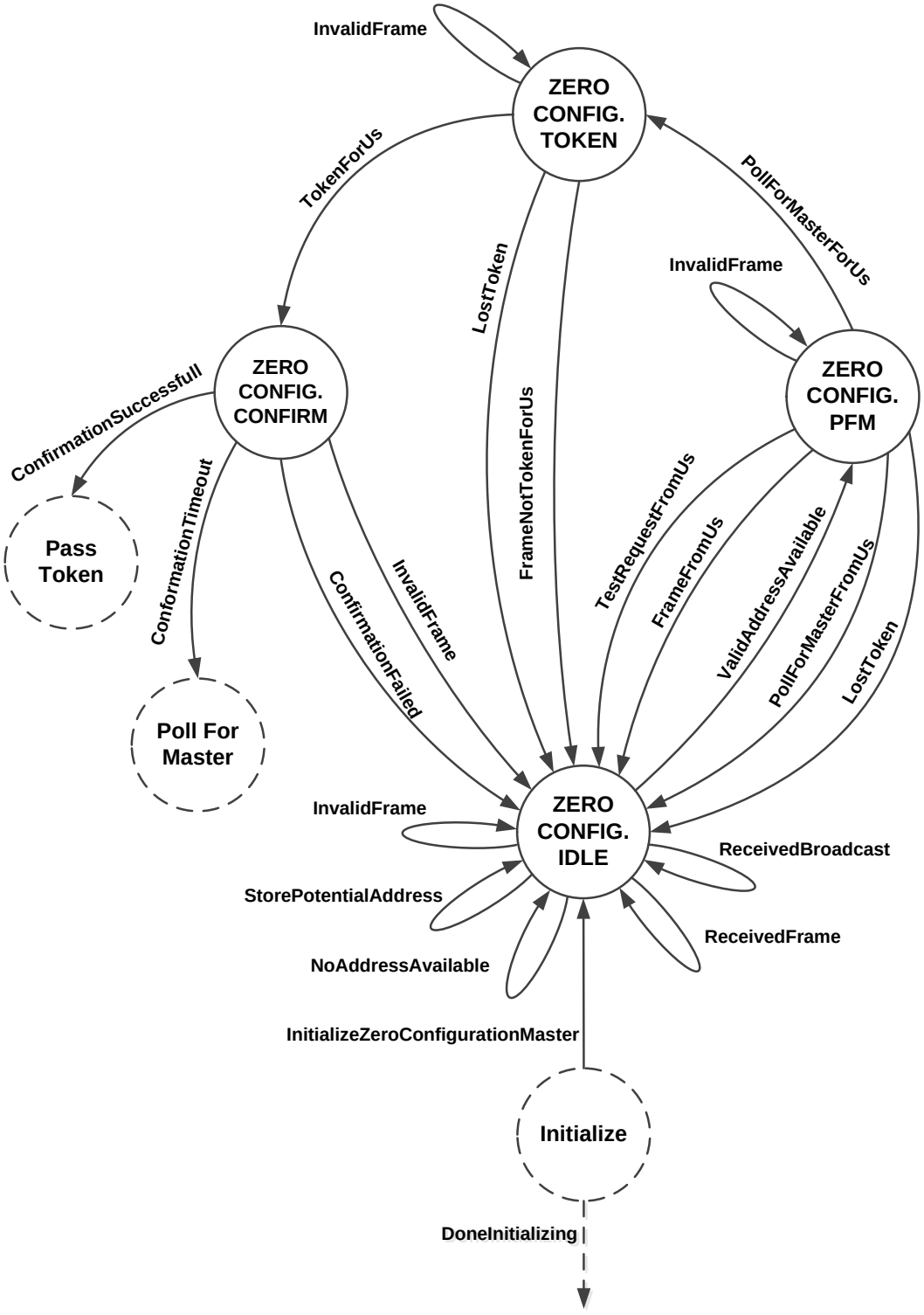


Figure 9-4b: Master Node State Machine - Zero-configuration States

[change **9.5.6.1** INITIALIZE, p. 103]

9.5.6.1 INITIALIZE

When a master node is powered up or reset, it shall unconditionally enter the INITIALIZE state.

InitializeZeroConfigurationMaster
If ZeroConfigurationMode is TRUE,

then set TS to 255, and enter the ZERO_CONFIGURATION_IDLE state.

DoneInitializing
~~Unconditionally,~~ *If ZeroConfigurationMode is FALSE,*

then set TS to the node's station address, set NS equal to TS (indicating that the next station is unknown), set PS equal to TS, set TokenCount to $\mathbf{N_{poll}}$ (thus causing a Poll For Master to be sent when this node first receives the token), set SoleMaster to FALSE, set ReceivedValidFrame and ReceivedInvalidFrame to FALSE, and enter the IDLE state.

[add to **9.5.6**, Master Node Finite State Machine, p. 102]

9.5.6.X1 ZERO_CONFIGURATION_IDLE

The ZERO_CONFIGURATION_IDLE state is entered when the node's station address is unknown (TS=255), and ZeroConfigurationMode is TRUE.

ReceivedBroadcast
If ReceivedValidFrame is TRUE, and the SourceAddress does not exist in our Address List,

then the SourceAddress is in use. Store the SourceAddress from the frame in an Address List to indicate that the address is in use. Set ReceivedValidFrame to FALSE, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

ReceivedFrame
If ReceivedValidFrameNotForUs is TRUE, and the FrameType is not Poll For Master,

then the SourceAddress is in use. Store the SourceAddress from the frame in an Address List to document that the address is in use, and note in the Address List if the frame was a Token frame (so this node can find its peer). Set ReceivedValidFrameNotForUs to FALSE, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

InvalidFrame
If ReceivedInvalidFrame is TRUE,

then an invalid frame was received. Set ReceivedInvalidFrame to FALSE, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

StorePotentialAddress
If ReceivedValidFrameNotForUs is TRUE, FrameType is equal to Poll For Master, and the DestinationAddress is not marked in the Address List as Poll For Master address,

then this address is potentially unused. Store the DestinationAddress from the frame in the Address List to

document that the address is a Poll For Master frame. Set ReceivedValidFrameNotForUs to FALSE, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

ValidAddressAvailable

If ReceivedValidFrameNotForUs is TRUE, FrameType is equal to Poll For Master, and the DestinationAddress is marked in the Address List as a Poll For Master Address,

then the Poll For Master cycle has completed for all the devices. Set TS to the ZeroConfigurationAddress, chosen from all of the unused addresses in the Address List marked as Poll For Master addresses. Set ReceivedValidFrame to FALSE, set ReceivedValidFrameNotForUs to FALSE, and enter the ZERO_CONFIGURATION_PFM state to wait for the next frame.

NoAddressAvailable

If ReceivedValidFrame is TRUE or ReceivedValidFrameNotForUs is TRUE, and the SourceAddress exists in the Address List, and there are no unused addresses in the Address List that are between 64 and 127, inclusive,

then the address monitor cycle is complete. Set ReceivedValidFrame to FALSE, set ReceivedValidFrameNotForUs to FALSE, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

LostToken

If SilenceTimer is greater than or equal to Tno_token + (127 * $\mathbf{T_{slot}}$) + (TS* $\mathbf{T_{slot}}$) (where TS is the ZeroConfigurationAddress),

then assume that the token has been lost. Set TS to the ZeroConfigurationAddress, chosen from the set of unused addresses in the Address List between 64 and 127, inclusive. Call SendFrame to transmit a Test Request frame to TS with a ZeroConfigurationUUID in the Data portion of the frame; enter the ZERO_CONFIGURATION_CONFIRM state to wait for a Test Response.

9.5.6.X2 ZERO_CONFIGURATION_PFM

The ZERO_CONFIGURATION_PFM state is entered when a node is waiting for a Poll For Master frame sent to the ZeroConfigurationAddress it has chosen, and ZeroConfigurationMode is TRUE.

TestRequestFromUs

If ReceivedValidFrame is TRUE or ReceivedValidFrameNotForUs is TRUE, and the FrameType is Test Request, and the SourceAddress is equal to TS or the DestinationAddress is equal to TS,

then another node chose our address. Clear all the flags and addresses from the Address List, set ReceivedValidFrame to FALSE, set ReceivedValidFrameNotForUs to FALSE, set TS to 255, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

PollForMasterFromUs

If ReceivedValidFrame is TRUE or ReceivedValidFrameNotForUs is TRUE, and the FrameType is Poll For Master, and the SourceAddress is equal to TS,

then another node is using our address. Clear all the flags and addresses from the Address List, set ReceivedValidFrame to FALSE, set ReceivedValidFrameNotForUs to FALSE, set TS to 255, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

PollForMasterForUs

If ReceivedValidFrame is TRUE, and the FrameType is Poll For Master, and the DestinationAddress is equal to TS,

then respond. Call SendFrame to transmit a Reply To Poll For Master frame to the node whose address is specified by SourceAddress (Source Address of the Poll); set ReceivedValidFrame to FALSE; and enter the ZERO_CONFIGURATION_TOKEN state to wait for a Token frame.

FrameFromUs

If ReceivedValidFrame is TRUE or ReceivedValidFrameNotForUs is TRUE, and the FrameType is not Test Request, and the Frame Type is not Poll For Master, and the SourceAddress is equal to TS,

then another node chose our address. Clear all the flags and addresses from the Address List, set ReceivedValidFrame to FALSE, set ReceivedValidFrameNotForUs to FALSE, set TS to 255, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

InvalidFrame

If ReceivedInvalidFrame is TRUE,

then wait for another frame. Enter the ZERO_CONFIGURATION_PFM state to wait for the Poll For Master frame addressed to us.

LostToken

If SilenceTimer is greater than or equal to Tno_token + (127 * $\mathbf{T_{slot}}$) + (TS* $\mathbf{T_{slot}}$) (where TS is the ZeroConfigurationAddress),

then assume that the token has been lost. Clear all the flags and addresses from the Address List, set ReceivedValidFrame to FALSE, set ReceivedValidFrameNotForUs to FALSE, set TS to 255, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

9.5.6.X3 ZERO_CONFIGURATION_TOKEN

The ZERO_CONFIGURATION_TOKEN state is entered when a node is waiting for a Token frame from the master to which it previously sent a Reply To Poll For Master frame, and ZeroConfigurationMode is TRUE.

TokenForUs

If ReceivedValidFrame is TRUE, and the FrameType is Token, and the DestinationAddress is equal to TS,

then confirm that we can use this address. Set ReceivedValidFrame to FALSE; Call SendFrame to transmit a Test Request frame to the node whose address is specified by SourceAddress (Source Address of the Token), and include the ZeroConfigurationUUID in the Data portion of the frame; Enter the ZERO_CONFIGURATION_CONFIRM state to wait for a Test Response.

FrameNotTokenForUs

If ReceivedValidFrame is TRUE or ReceivedValidFrameNotForUs is TRUE, and the FrameType is not Token,

then restart the process. Clear all the flags and addresses from the Address List, set ReceivedValidFrame to FALSE, set ReceivedValidFrameNotForUs to FALSE, set TS to 255, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

InvalidFrame

If ReceivedInvalidFrame is TRUE,

then wait for another frame. Enter the ZERO_CONFIGURATION_TOKEN state to wait for the Token frame addressed to us.

LostToken

If SilenceTimer is greater than or equal to Tno_token + (127 * $\mathbf{T_{slot}}$) + TS* $\mathbf{T_{slot}}$) (where TS is the

ZeroConfigurationAddress),

then assume that the token has been lost. Clear all the flags and addresses from the Address List, set ReceivedValidFrame to FALSE, set ReceivedValidFrameNotForUs to FALSE, set TS to 255, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

9.5.6.X4 ZERO_CONFIGURATION_CONFIRM

The ZERO_CONFIGURATION_CONFIRM state is entered when a node is waiting for a Test Response frame and ZeroConfigurationMode is TRUE.

ConfirmationTimeout

If SilenceTimer is greater than or equal to Tconfirmation_timeout,

then use the Address. Set PS to TS; set NS to TS (no known successor node); set RetryCount and EventCount to zero; set TokenCount to Npoll (thus causing a Poll For Master to be sent when this node first receives the token), and set SoleMaster to FALSE; and enter the POLL_FOR_MASTER state to find a new successor to TS.

InvalidFrame

If SilenceTimer is less than $\mathbf{T}_{\text{confirmation\_timeout}}$ and ReceivedInvalidFrame is TRUE,

then an invalid frame was received. Set ReceivedInvalidFrame to FALSE, Clear all the flags and addresses from the Address List, set TS to 255, and enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

ConfirmationSuccessful

If SilenceTimer is less than $\mathbf{T}_{\text{confirmation\_timeout}}$, ReceivedValidFrame is TRUE, FrameType is Test Response, and the Data sent in the Test Request match,

then use the Address. Set PS to TS; call SendFrame to transmit a Token frame to NS (found by looking at Address List entries with Token marked); set RetryCount and EventCount to zero; set TokenCount to one; and enter the PASS_TOKEN state.

ConfirmationFailed

If SilenceTimer is less than $\mathbf{T}_{\text{confirmation\_timeout}}$, ReceivedValidFrame is TRUE, and FrameType is Test Response and Data sent in the Test Request does not match, or the FrameType is not Test Response,

then start over. Set ReceivedFrame to FALSE; Clear all the flags and addresses from the Address List; Set TS to 255; Enter the ZERO_CONFIGURATION_IDLE state to wait for the next frame.

[Append to Clause **9.5.6.2**, p. 104]
[Repeat the below changes for :]
[**9.5.6.4** WAIT_FOR_REPLY, p. 105]
[**9.5.6.8** POLL_FOR_MASTER, p. 107]

9.5.6.2 IDLE

...

ReceivedValidFrameFromUs

If ReceivedValidFrameNotForUs is TRUE, and SourceAddress is equal to TS, and ZeroConfigurationMode is TRUE,

then another node is using our ZeroConfigurationAddress. Set ReceivedValidFrameNotForUs to FALSE; Clear all the flags and addresses from the Address List; Set TS to 255; Enter the ZERO_CONFIGURATION_IDLE

state to wait for the next frame.

ReceivedValidFrameNotForUs
If ReceivedValidFrameNotForUs is TRUE, and SourceAddress is not equal to TS, and ZeroConfigurationMode is TRUE,

then discard the frame. Set ReceivedValidFrameNotForUs to FALSE; Enter the IDLE state to wait for the next frame.

[Change **12.X** in Addendum 135-2012*ai*]

12.X Network Port Object

…

Table 12-X. Properties of the Network Port Object Type

Property Identifier	Property Datatype	Conformance Code
…		
Zero_Configuration_Mode_Enable	*BOOLEAN*	*O*[N]
…		

…
[N] *Required if the port is an MS/TP port and the device supports MS/TP ZeroConfigurationMode.*
…

[Insert **Clause 12.X.Y** in Addendum 135-2012*ai*]

12.X.Y Zero_Configuration_Mode_Enable

This property, of type BOOLEAN, indicates whether or not MS/TP ZeroConfigurationMode is enabled. A value of TRUE indicates that ZeroConfigurationMode is enabled, FALSE indicates it is not.

This property is required if MS/TP ZeroConfigurationMode is supported by this network port.

If this property is writable, then a successful write to this property shall set the Changes_Pending property to TRUE. A value written to this property shall become effective when the device receives a ReinitializeDevice service request with a 'Reinitialized State of Device' of ACTIVATE_CHANGES or WARMSTART.

[Change **ANNEX A**, p. 776]

...
Data Link Layer Options:

...

☐ MS/TP master (Clause 9), baud rate(s): _____
☐ *MS/TP zero configuration master (Clause 9), baud rate(s):* _____

...