Specification

Z-Wave and Z-Wave Long Range Network Layer Specification

Version:	3.0
Description:	This specification defines the Network (NWK) layer for ITU-T G.9959 compliant transceivers and Z-Wave long Range transceivers, which enables network operations and routing on Z-Wave networks. Implementations claiming compliance with this specification can be used with an application layer to certify Z-Wave and Z-Wave long range products.
Written By:	CSWG
Date:	2023/05/26
Reviewed By:	CSWG
Restrictions:	Public

Approved by:
Z-Wave Alliance Board of Directors

THIS SPECIFICATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE ALLIANCE, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS SPECIFICATION.

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
0.1	2020.07.01	CSWG	ALL	Z-Wave Network Layer v.0.7 initial document
0.3	2020.09.01	CSWG	ALL	Z-Wave Long Range network layer added
0.5	2020.09.24	CSWG	ALL	Implemented review comments
0.7	2020.09.28	CSWG	Frontpage, header and footer	Made document ready for Z-Wave alliance donation
0.7.1	2020.11.23	CSWG	6.3.1.2 4.3.2.13 6.3 All	Added speed bits to the LR Node Information Frame Added description of Command Class field in New Node Registered command Fixed LR command class numbers in frame figures Fixed references to alliance documents Added neighbour discovery to figures Figure 4.80 and Figure 4.90 Replaced “slave node” with “end node”
0.7.2	2021.02.26	CSWG	References 6.3.1.2.1.1 6.5.1 6.4	Fixed reference names Fixed Table reference Removed reference to undefined constant Changed description of *nwkLRExcludeRequestForeignNetBackOff* to fit Long Range frame names
0.8	2021.03.01	CSWG	None	Initial review by TC done, updating revision to 0.8
0.9	2021.03.29	CSWG	Frontpage	Cleanup for IPR review
1.0	2021.08.27	ZWA Board		Approved for Publication
1.5	2021.10.19	Sensative AB Silicon Labs	4.5.4.3 6.5.4.1 4.5.4.2 4.3.2.13.2	Updated SmartStart behavior for battery powered up end devices Added missing figure 4.68 Clarified the use of New Node Registered when removing a node
1.7	2022.04.19	CSWG	Frontpage	Reviewed by CSWG, Updated version to 1.7 for TC review
1.9	2022.05.04	CSWG	Frontpage	Reviewed by TC, Updated version to 1.9 for IPR review
2.5	2023.03.02	CSWG	Frontpage, header and footer 4.3.2.1 4.3.2.5 4.3.2.11 4.3.2.13 4.5.9.4 4.5.4.3.1.2 6.5.4.1.1.2 4.3.2.12.2	Updated version and dates Updated the Routing text in the Node Information Frame command Added optional Wake Up Time field to the Get Nodes In Range command Updated Status text in the Network Transfer End command Fixed missing Speed Extension field in the New Node Registered command Added clarification on FL nodes optionally finding other FL nodes in the neighbor discovery process Updated description of INIF transmission to match previous change to SmartStart triggering Updated description of the use of Assign Return Route to the SUC nodeID
2.9	2023-0526	CSWG	Revision	Document approved by CSWG and TC
3.0	2023-7-20	ZWA BoD		Approved by the Z-Wave Alliance Board of Directors

Table of Contents

Table of Figures

1 ABBREVIATIONS

Abbreviation	Explanation
AL	Always Listening
APL	Application Layer
DLPDU	Data Link Protocol Data Unit
FL	Frequently Listening
ISM	(unlicensed) Industrial Scientific and Medical
MAC	Medium Access Control
MPDU	MAC Protocol Data Unit
NIB	Network Information Base
NIF	Node Information Frame Command. Refer to 4.3.2.1
NLDE	Network Layer Data Entity
NL	Non-Listening
NPDU	Network Layer Protocol Data Unit
NLME	Network Layer Management Entity
NSDU	Network Service Data Unit
NWI	Network Wide Inclusion
NWK	Network Layer
NLDE-SAP	Network Layer Data Entity - Service Access Point
NLME-SAP	Network Layer Management Entity - Service Access Point
MLDE-SAP	MAC Layer Data Entity - Service Access Point
MLME-SAP	MAC Layer Management Entity - Service Access Point
OSI	Open System Interconnection
PLDE-SAP	Physical Layer Data Entity – Service Access Point
PLME-SAP	Physical Layer Management Entity – Service Access Point
PHY	Physical layer
S2	Security 2 Command Class. Refer to [TECC]
S2 DSK	Security 2 Device Specific Key. Refer to [TECC]
SAP	Service Access Point
SAR	Segmentation and Reassembly
SIS	SUC ID Server
SUC	Static Update Controller

2 INTRODUCTION

2.1 Audience

All Z-Wave Alliance members

2.2 Z-Wave technology overview

Z-Wave is a wireless mesh protocol oriented to the residential control and automation market but also suitable for light commercial applications. The Z-Wave technology offers a simple yet reliable method to wirelessly control lights, door locks, thermostats and a range of systems in residential and commercial environments. The Z-Wave protocol works in the unlicensed industrial, scientific, and medical (ISM) bands. The specific frequency band varies from region to region and the frequency bands are defined in [G.9959].

It is known that any wireless radio network may suffer from frequent frame drops due to fading effects, spurious noise and reflections. To combat such conditions, the Z-Wave protocol provides a low-level retransmission approach. In addition, the Z-Wave protocol employs a network layer mesh routing protocol to extend networks beyond what is possible in direct range.

2.3 Z-Wave Long Range technology overview

Z-Wave Long Range provides an extended range version of the Z-Wave technology, targeting deployments over a kilometer radius, suitable in both indoors and outdoors areas. All Z-Wave applications can run using either the Z-Wave or the Z-Wave Long Range PHY/MAC.

The Z-Wave Long Range protocol does not use any mesh routing and employs only direct range communication.

The Z-Wave Long Range protocol also operate in the unlicensed industrial, scientific, and medical (ISM) bands. The specific frequency band varies from region to region and the frequency bands are defined in [LR_PHY], [LR_MAC].

2.4 Network layer specification

This specification presents the Z-Wave and Z-Wave Long Range *Network Layer* definitions that will be implemented by devices operating in Z-Wave and Z-Wave Long Range networks. The Network Layer features in relation to the OSI reference model and other layers is presented in section 3.1 and 5.1.

2.5 Glossary

The key words **shall**, **should** and **may** are formally used to indicate requirement levels in this document:

- **Shall:**
  This word indicates that the definition is an absolute requirement of the specification.

- **Should:**
  This word indicates that there may exist valid reasons in particular circumstances to ignore an item, but the full implications must be understood and carefully weighed before choosing a different course.

- **May**:
  This word indicates that an item is truly optional. One vendor may choose to include the item because a marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item.
  An implementation which does not include an option **shall** be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include an option **shall** be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

The use of the key words **shall**, **should** and **may** is compliant with the requirement levels used in ITU recommendations.

3 Z-WAVE PROTOCOL OVERVIEW

The Z-Wave protocol is a low bandwidth half duplex protocol designed for reliable wireless communication in a low-cost control network. The main purpose of the protocol is to enable short message transportation in a reliable manner. The Z-Wave protocol is not designed to transfer a large amount of data or any kind of streaming or timing critical data.

3.1 The Z-Wave protocol stack architecture

The Open System Interconnection (OSI) reference model is a representation system for characterizing and standardizing the functions of a communication system in terms of abstraction layers. This allows us to describe similar communication functionalities into logical layers. The 7 layers of the OSI model are regarded by many as an idealized model; too abstract and fine-grained for most real-world protocols. It is however useful to refer to the OSI model when describing a given communication protocol framework. With respect to that, the Z-Wave protocol stack would be described using the model as shown in Figure 3.1. Note that the Z-Wave application layer consists of the OSI stack layers knows as transport, session, presentation and application.

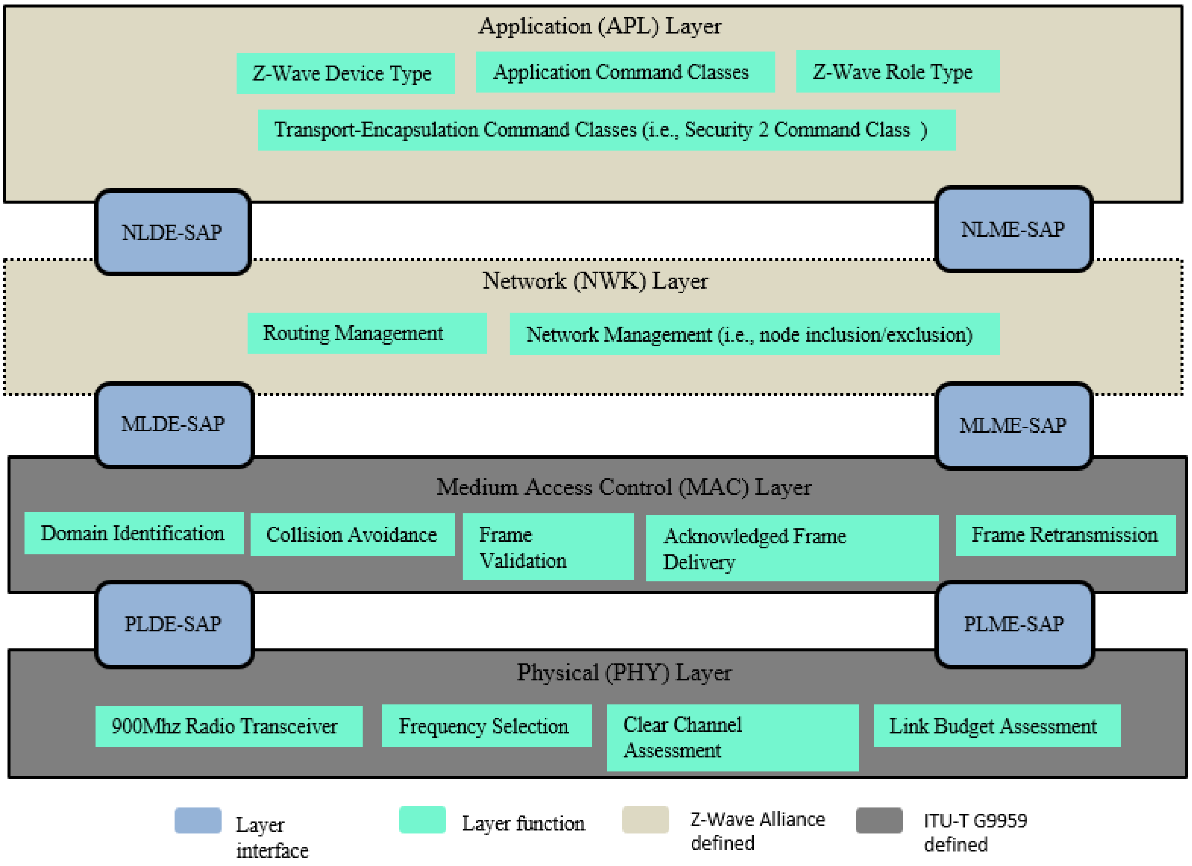


Figure 3.1 Z Wave protocol stack architecture

As depicted in Figure 3.1, the Z-Wave protocol stack is made up of OSI layers where each layer performs set of services for the upper layer. Each layer has two main interfaces to facilitate the communication with upper layers through a Service Access Point (SAP). The interfaces are described as a data entity and management entity that provide a data transmission service and all other services, respectively.

[G.9959] defines the physical and medium access control layers.

- The physical layer offers a data flow control between the MAC and PHY layers and adds PHY-related management headers. The PHY layer is responsible for activation and deactivation of the radio transceiver, data transmission and reception, frequency selection, clear channel assessments, and the link budget assessment of received frames.
- The MAC layer defines the Z-Wave data transfer model and frame structure. During a Z-Wave frame transmission, the MAC layer takes the payload data from higher layers and construct the MAC data payload (MPDU) and the MPDU header. The header comprises addresses, frame control and frame length information. The frame control field is about 16 bits in length and contains information about the frame type and other control flags that can be used by higher layer.

On the foundation of those two lower layers, the Z-Wave alliance defines the Network layer (NWK) and application layers.

The Z-Wave Network Layer (NWK) defines a multi-hop routing protocol, that is employed by Z-Wave nodes to extend their communication range. It means that the Z-Wave nodes can therefore send frames to nodes that are not in direct radio communication range. Besides, the Z-Wave NWK layer is responsible for network formation (i.e., inclusion/exclusion of nodes to/from a network) and its maintenance. The Z-Wave NWK layer manages the network establishment using command frames known as the Z-Wave Protocol Command Class (described in section 4.3). These Z-Wave NWK commands are designed for network formation specific purposes.

NWK:002E.1 The Z-Wave application layer is responsible for building applications using dedicated Command Classes, (defined in [ACC], [MCC], [TECC], [NPCC]).In order to be certifiable, applications **shall** comply with Z-Wave device types defined in [DT] and [DTV2]. Finally, the applications layer is also responsible for providing some network management functionalities using the NWK interface (for details, refer to [RT]).

This specification defines NWK layer. The upper and lower layers are outside the scope of this specification.

3.2 Network Layer reference model

The Network Layer (NWK) provides an interface between the application layer and the MAC layer. The NWK layer relies on services provided by the MAC layer and offers services to higher layers though the Network Layer Data Entity (NLDE) and Network Layer Management Entity (NLME) service point interfaces. The NLME provides management service interface where the NWK layer management functionalities can be invoked. The NLME is responsible for maintaining a Network Information Base (NIB) that contains the routing information of the network. Figure 3.2 illustrates the components and interface of NWK layer.

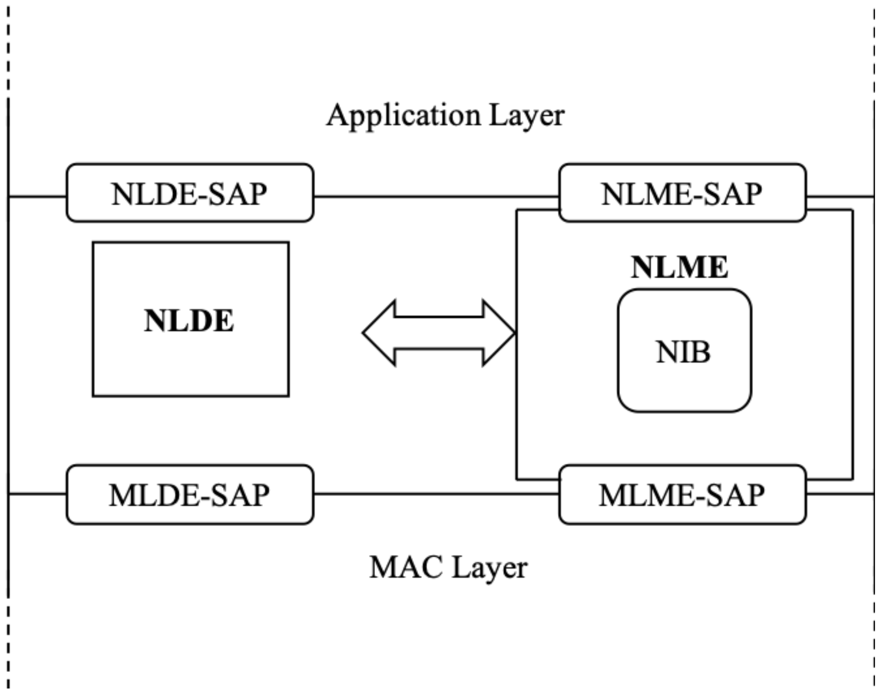


Figure 3.2 The Network layer reference model

NWK:0022.1 The Z-Wave NWK layer **shall** provide two services to the Application layer that are accessed through two SAPs:

- The data service, accessed through NLDE-SAP, and
- The network management service accessed through the NLME-SAP.

The detailed description of the Z-Wave NWK functional model is presented in chapter 4 .

3.3 Z-Wave definitions

3.3.1 Z-Wave network topology basic principles

The following is a summary of the basic Network topology principles established by [G.9959]:

1. Groups of nodes are divided into domains:
   - The division of physical nodes into domains is logical. Domains may fully or partially overlap each other's radio frequency ranges.
   - The Z-Wave Network Layer supports up to 2^{32} domains.
   - Each domain is identified by a unique **HomeID**.
   - Management of different domains in the same physical media is handled by individual domain masters.
2. The domain is a set of nodes connected to the same medium:
   - One node in the domain operates as a domain master, known as the Primary Controller.
   - Each domain may contain up to *232* nodes (including the domain master).
   - Each node in the domain is identified by a **NodeID** that is unique within the actual domain.
   - Nodes of the same domain can communicate with each other either directly or via other nodes in the same domain.
3. Nodes of different G.9959 domains:
   - The Z-Wave Network Layer provides connectivity within one domain.
In some cases, frames from a foreign domain are repeated into the current domain. Inter-domain communication is beyond the scope of this specification
4. The network is self-healing:
   - Nodes may autonomously establish new routes on demand.
Full mesh routing is supported. There is no requirement for star or tree network topologies.

3.3.2 Controller and end nodes

The Z-Wave network layer defines two networking node types: controller and end.

The controller nodes are responsible for setting up and maintaining the Z-Wave network. They can include or exclude nodes and they are aware for the network topology. This allows controllers to determine the possible routes between any two nodes in the network. Controllers can exchange network topology with each other.

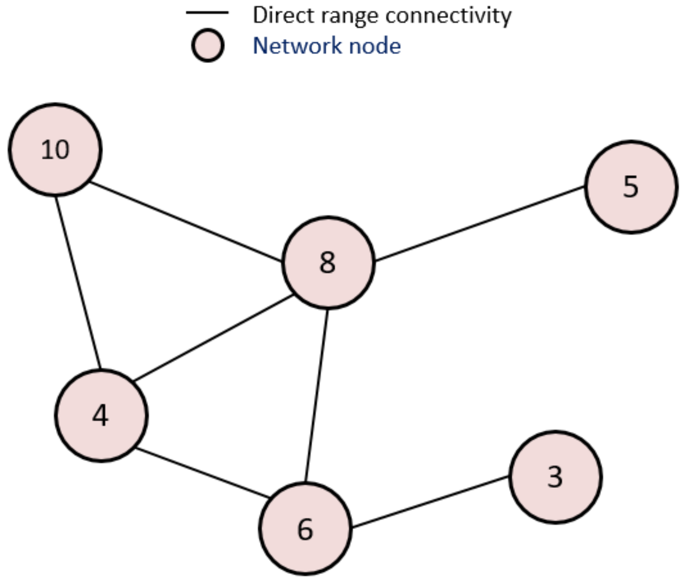
End nodes can only be added or removed from a network by a controller, they do not calculate routes, and they simply rely on route information provided by the controllers. Note that end nodes can send commands to other nodes and "control" other nodes functionalities at the application level.

Both controller and end nodes can participate in mesh routing. It enables nodes within a network to communicate with each other even when they are out of direct communication range.

3.3.3 Network topology

In this specification, network topology refers to the list of nodes present in a network as well the list of direct range neighbours for each node.

Figure 3.3 illustrates the concept of network topology.



Topology data:

Node	Neighbours
10	4, 8
4	10, 8, 6
6	4, 8, 3
8	10, 4, 6, 5
3	6
5	8

Figure 3.3 Network topology example

3.3.4 Z-Wave controller roles

A Z-Wave controller is a node that has the capability to provide network management functionalities such as adding/removing nodes to/from a network and distributing network topology to other controllers. The NWK layer defines several controller roles in a network:

3.3.4.1 Primary Controller

The Primary Controller is by default the controller that starts the network. The Primary Controller can always be used to set up and maintain a network. It can add/remove nodes and knows the network topology.

There can be only one Primary Controller in a network.

The Primary Controller may offer additional services to other controllers:

A Primary Controller can handover the Primary Controller (and/or SUC/SIS) role to another controller.

3.3.4.2 Secondary Controller

All controllers that are not the Primary Controller are Secondary Controllers.

If the Primary Controller does not have any SUC/SIS capability, a Secondary Controller cannot include nodes, exclude nodes, or receive updated network topology automatically.

If a SUC is present in the network, a Secondary Controller can request updated network topology at any time.

3.3.4.3 SUC Controller

A Static Update Controller (SUC) is the controller that has the responsibility to keep the network topology and distribute it to other controllers, on demand.

There can be only one SUC in a network. Both Primary Controllers and Secondary Controllers can be SUC.

3.3.4.4 SIS Controller

A SUC ID Server (SIS) Controller is a controller that has both the SUC Controller role and the Primary Controller role. In addition, it provides the SIS functionality. The SIS functionality consists in the ability to reserve NodeIDs to other controllers for enabling them to include and exclude nodes.

3.3.4.5 Inclusion Controllers

If the Primary Controller is the SIS, Secondary Controllers in the same network become Inclusion Controllers. Inclusion Controllers are secondary controllers that can include and exclude nodes on behalf of the SIS.

3.3.5 Node operation modes

Z-Wave nodes may operate in three different receiving modes.

3.3.5.1 Always Listening (AL)

AL nodes’ RF receiver is always on and these nodes participate in mesh routing by repeating Routed NPDUs and Explore NPDUs.

3.3.5.2 Frequently Listening (FL)

FL nodes' RF receiver is turned off most of the time. The RF receiver is turned on at regular intervals for a short duration to listen the Wake Up Beams. AL nodes can reach FL nodes by issuing a Wake Up Beam prior to issuing commands to the FL nodes.

FL nodes do not participate in routing and do not repeat Routed NPDUs and Explore NPDUs.

[G.9959] defines two possible Wake Up intervals FL nodes: 250ms and 1000ms, and three channel configurations.
When operating with a channel configuration 1 and 2, some NWK commands/frames indicate which setting to use (250ms or 1000ms).
When operating with channel configuration 3, the 1000ms setting is always used in frames/commands, despite using fragmented beams.

For more details, refer to [G.9959].

3.3.5.3 Non-Listening (NL)

NL nodes cannot be awakened by another node. They wake up and transmit frames at fixed configured intervals to a single NodeID destination.

NL nodes reporting can be configured using the Wake Up Command Class. For more details, refer to [MCC].

3.3.6 Network addressing

Z-Wave supports the following types of addressing:

- Singlecast
- Multicast
- Broadcast

The type of addressing and its frame format are defined in the MPDU Header (refer to [G.9959]).

NWK:0001.1 NWK:0002.1 In this specification, some commands **shall not** be sent using multicast addressing and **shall** be ignored if received via multicast addressing. In these cases, the Z-Wave Multicast frame and the broadcast NodeID (0xFF) **shall** both be considered multicast addressing methods.

4 Z-WAVE NETWORK LAYER SPECIFICATION

4.1 General Description

The Z-Wave Network Layer provides transmission services for higher layers using services provided by the MAC layer. The Z-Wave NWK layer services include:

- multi-hop routing.
- route discovery.
- routing acknowledgment.
- managing the information that enables communication to FL nodes.
- network formation and maintenance.

The Z-Wave NWK accommodates delivery of the application messages across multiple hops.

4.1.1 Z-Wave NWK Layer overview

The network layer is required to provide functionality to ensure correct operation of the [G.9959] MAC sub-layer, and it provide a suitable service interface to the application layer. To interface with the application layer, the network layer conceptually includes two service entities that provide the necessary functionalities. These service entities are the data service and the management service. The Z-Wave NWK layer data entity (NLDE) provides the data transmission service via its associated SAP (NLDE-SAP), and the Z-Wave NWK layer management entity (NLME) provides the management service via its associated SAP (NLME-SAP). The NLME utilizes the NLDE to achieve some of its management tasks and it also maintains a database known as the Network Information Base (NIB) that contains information regarding the network topology.

4.1.2 Network Layer Data Entity (NLDE)

NWK:0003.1 The NLDE **shall** provide a data service for the application layer to transport Data Link Protocol Data Unit (DLPDU) to a destination located in the same network.

[G.9959] introduces the DLPDU, and it must contain the application data as depicted Figure 4.1.

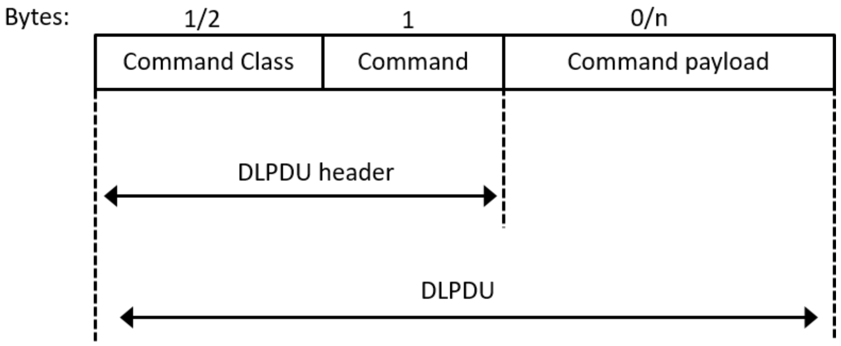


Figure 4.1 DLPDU format

NWK:0004.1 The NLDE **shall** provide the following services:

NWK:0005.1

- **Generation of NPDUs (Network Protocol Data Unit)**: The NLDE **shall** be capable to generate appropriate NPDUs from application data.

NWK:0006.1

- **Routing**: The NLDE **shall** be able to transmit an NPDU to the appropriate node that will either be a repeater along the route or the destination.

4.1.2.1 Network Layer Management Entity (NLME)

NWK:0007.1 NWK:0008.1 The NLME **shall** provide a management service to leverage the network's routing capabilities. The NLME **shall** provide the following services:

- **Route Discovery**: this is the ability to discover a valid route to a destination for subsequent messages.
- **Reception Control**: this is the ability to ensure that a routed NPDU can be delivered correctly, including communication control information for FL nodes.
- **Routing:** this is the ability to leverage AL nodes in the network as repeaters to carry a message to a destination that is not in direct range.
- **Network Inclusion:** This is the ability to join or create a network.
- **Network Exclusion:** This is the ability to leave a network.
- **Network Maintenance:** This is the ability to monitor the health of a network and create a neighbour topology and a routing table.

4.2 Frame format

The Z-Wave Network layer routing provides mesh networking functionality allowing to deliver messages through repeater nodes.

The Network layer also provides route discovery, using special frames called Explore Frames.

NWK:0009.1 When the network layer receives data units from upper layers, it **shall** add routing information and forward the received data to the MAC layer.

4.2.1 NPDU formats

This clause specifies the format of the NPDU. Each NPDU consists of the following components:

- An optional NPDU Header, comprising routing information
- NSDU (Network Service Data Unit), encapsulating frame specific data payload.

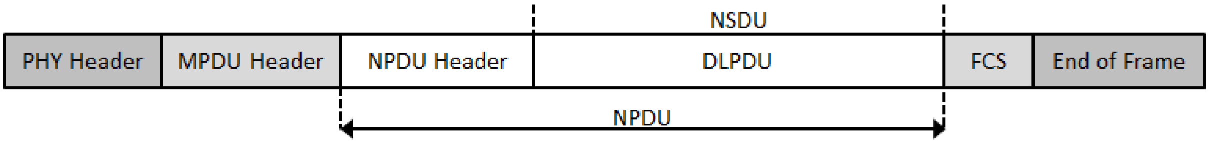


Figure 4.2 General NPDU format

[G.9959] introduces a *Frame Control* field in the MPDU Header. This field dictates the format of the NPDU. The *Frame Control* field format is shown for information in Figure 4.3 and Figure 4.4,

Bit Byte	7	6	5	4	3	2	1	0
5	Routed	Ack Req	Low power	Speed modified	Header type			
6	Reserved	Beaming info		Reserved	Sequence number			

Figure 4.3 MDPU Header Frame Control field (Channel configuration 1,2)

Bit Byte	7	6	5	4	3	2	1	0
5	Ack Req	Low power	Reserved		Header type			
6	Reserved	Beaming info			Reserved			

Figure 4.4 MDPU Header Frame Control field (Channel configuration 3)

This document specifies two types of network layer frames:

1. Routed NPDUs
2. Explore NPDUs

Table 4.1 shows the MPDU Frame Control configuration for a Routed NPDU.

Table 4.1 Routed NPDU configuration

	Routed	Header Type
Channel Configuration 1,2	1	0x01 (Singlecast MPDU)
Channel Configuration 3	N/A	0x08 (Routed MPDU)

The Routed NPDU **shall** use the format outlined in Figure 4.5.

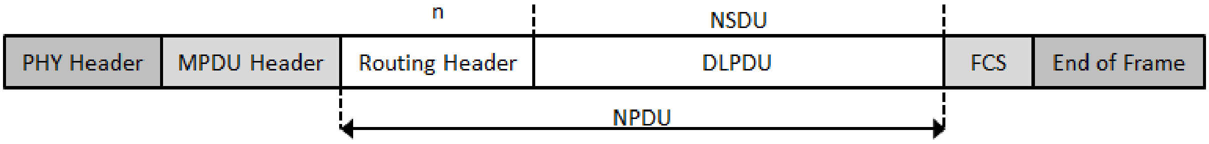


Figure 4.5 Routed NPDU format

Table 4.2 shows the MPDU Frame Control configuration for an Explore NPDU.

Table 4.2 Explore NPDU configuration

Header Type
0x05 (Explore MPDU)

In this case, the NPDU **shall** use the format outlined in Figure 4.6

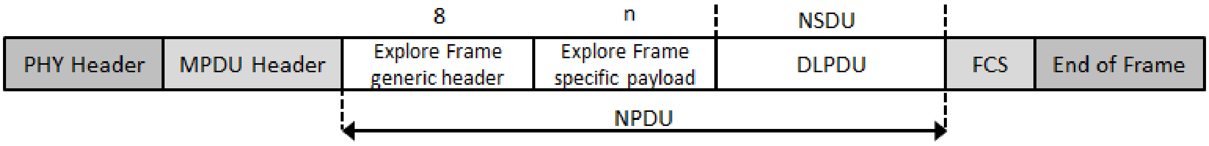


Figure 4.6 Explore NPDU format

4.2.2 Routed NPDUs

Routed NPDUs are used to transmit a frame when the sending node knows a route to the destination.
NWK:000A.1 Routed NPDUs **shall** use Singlecast addressing.

4.2.2.1 General routing header format

Routed NPDUs use a routing header that can be described in a general manner according to Figure 4.7 for channel configuration 1,2 and Figure 4.8 for channel configuration 3.

<table>
<tr><th>Octet</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="2"><b>1</b></td><td colspan="3">Reserved</td><td>Routed Speed Modified</td><td rowspan="2">Extended Header</td><td rowspan="2">R-Err</td><td rowspan="2">R-Ack</td><td rowspan="2">Direction</td></tr>
<tr><td colspan="4">Failed Hop <i>(R-Err = 1)</i></td></tr>
<tr><td><b>2</b></td><td colspan="4">Repeaters</td><td colspan="4">Hops</td></tr>
<tr><td><b>3</b></td><td colspan="8">Repeater 0</td></tr>
<tr><td><b>4</b></td><td colspan="8">Repeater 1</td></tr>
<tr><td><b>5</b></td><td colspan="8">Repeater 2</td></tr>
<tr><td><b>6</b></td><td colspan="8">Repeater 3</td></tr>
<tr><td><b>7</b></td><td colspan="4">Extended Header Body Length</td><td colspan="4">Extended Header Type</td></tr>
<tr><td><b>8</b></td><td colspan="8">Extended header body 1</td></tr>
<tr><td><b>…</b></td><td colspan="8">…</td></tr>
<tr><td><b>8 + N</b></td><td colspan="8">Extended header body N</td></tr>
</table>

Figure 4.7 General Routing Header Format, channel configuration 1,2

<table>
<tr><th>Octet</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="2"><b>1</b></td><td colspan="3">Reserved</td><td>Routed Speed Modified</td><td rowspan="2">Extended Header</td><td rowspan="2">R-Err</td><td rowspan="2">R-Ack</td><td rowspan="2">Direction</td></tr>
<tr><td colspan="4">Failed Hop <i>(R-Err = 1)</i></td></tr>
<tr><td><b>2</b></td><td colspan="4">Repeaters</td><td colspan="4">Hops</td></tr>
<tr><td><b>3</b></td><td colspan="8">Repeater 0</td></tr>
<tr><td><b>4</b></td><td colspan="8">Repeater 1</td></tr>
<tr><td><b>5</b></td><td colspan="8">Repeater 2</td></tr>
<tr><td><b>6</b></td><td colspan="8">Repeater 3</td></tr>
<tr><td><b>7</b></td><td colspan="8">Destination Wake Up</td></tr>
<tr><td><b>8</b></td><td colspan="4">Extended Header Body Length</td><td colspan="4">Extended Header Type</td></tr>
<tr><td><b>9</b></td><td colspan="8">Extended header body 1</td></tr>
<tr><td><b>…</b></td><td colspan="8">…</td></tr>
<tr><td><b>9 + N</b></td><td colspan="8">Extended header body N</td></tr>
</table>

Figure 4.8 - General Routing Header Format, channel configuration 3

4.2.2.1.1 Routed Speed Modified (1 bit) / Failed Hop (4 bits)

If the *R-Err* field is set to 0:

NWK:000B.1 This field **shall** be parsed as Routed Speed Modified (1 bit).

This field is used to indicate if the current frame is a retransmission for which the link speed has been downgraded.

For channel configuration 1,2:

NWK:000C.1 This field **may** be set to 1 to indicate that this is a retransmission at lower speed and the destination should not reuse this route as main route.

NWK:000D.1 The “Speed Modified” subfield from the MPDU Frame Control [G.9959] **shall** be ignored and this field **shall** be used instead when a routing header is present.

For channel configuration 3:

NWK:000E.1 This field is reserved. It **shall** be set to 0 by a sending node and ignored by a receiving node.

If the *R-Err* field is set to 1:

NWK:000F.1 This field **shall** be parsed as Failed Hop (4 bits).

NWK:0010.1 This field **shall** indicate the Hop (repeater number) that did not get an acknowledgement for the routed frame.

4.2.2.1.2 Extended Header (1 bit)

This field indicates if the current Routing Header contains an Extended Header.

NWK:0011.1 The value 0 **shall** indicate that the Routing Header does not contain any Extended Header field and stops after the last *Repeater* field.

NWK:0012.1 The value 1 **shall** indicate that the Routing Header contains an Extended Header and includes the *Extended Header Type* and *Extended Header Body* fields.

4.2.2.1.3 R-Err (1 bit)

This field is used to indicate that a routed frame could not be delivered to the next hop or destination NodeID.

NWK:0013.1 The value 0 **shall** indicate no transmission error.

NWK:0014.1 The value 1 **shall** indicate that a routed frame was not delivered to the next Hop.

4.2.2.1.4 R-Ack (1 bit)

This field is used to indicate that a routed frame was delivered correctly to the destination NodeID.

NWK:0015.1 The value 0 **shall** indicate that a routed frame was not (yet) acknowledged by the destination NodeID.

NWK:0016.1 The value 1 **shall** indicate that a routed frame was acknowledged by the destination NodeID.

4.2.2.1.5 Direction (1 bit)

The *Direction* field indicates the direction of the frame.

NWK:0017.1 A node sending out a routed frame **shall** set this field to 0 (outgoing routed frame).

NWK:0018.1 A node replying with a Routed Error/Routed Acknowledgement **shall** set this field to 1. (reply to an outgoing Routed Frame)

4.2.2.1.6 Repeaters (4 bits)

This field is used to indicate the total number of repeaters for the routed frame.

NWK:0019.1 This field **shall** be in the range 1…4.

4.2.2.1.7 Hops (4 bits)

This field is used to indicate the progress of the frame through the route. The Hops value represents the index of the next repeater that **shall** repeat (or receive) the routed frame. NWK:001A.1

NWK:001B.1 The source NodeID **shall** set this field to 0 when sending a routed frame and each repeater **shall** increment this field when they forward the frame if the *Direction* field is set to 0.

NWK:001C.1 Each repeater **shall** decrement this field when they forward the frame if the *Direction* field is set to 1.

NWK:001D.1 In general, the value **shall** be set to the next repeater in charge of repeating the frame. (i.e. the value 0x00 indicates that repeater 0 **shall** repeat this frame, value 0x01 indicates that repeater 1 **shall** repeat this frame, etc.).

The last repeater will set this field to the total number of repeaters + 1.

NWK:001E.1 When a routed frame returns to the source NodeID (e.g. a Routed Ack / Error), the Repeater 0 node **shall** set this field to 0x0F.

4.2.2.1.8 Repeater 0 (8 bits)

This field is used to indicate the NodeID of the first repeater for this route.

4.2.2.1.9 Repeater 1 (8 bits)

This field is used to indicate the NodeID of the second repeater for this route.

NWK:001F.1 This field **shall** be omitted if the *Repeaters* field value is less than 2.

4.2.2.1.10 Repeater 2 (8 bits)

This field is used to indicate the NodeID of the third repeater for this route.

NWK:0020.1 This field **shall** be omitted if the *Repeaters* field value is less than 3.

4.2.2.1.11 Repeater 3 (8 bits)

This field is used to indicate the NodeID of the fourth repeater for this route.

NWK:0021.1 This field **shall** be omitted if the *Repeaters* field value is less than 4.

4.2.2.1.12 *Destination* Wake Up (8 bits)

For channel configuration 1,2:

NWK:0023.1 This field **shall** be omitted.

For channel configuration 3:

This field is used to indicate if the destination of the frame is an FL node and requires beaming prior to delivering the frame.

NWK:0024.1 The value 0x00 **shall** indicate that the destination is an AL node and can be forwarded the frame immediately.
NWK:0025.1 The value 0x02 **shall** indicate that the destination requires a Fragmented Beam prior to deliver a frame.

4.2.2.1.13 Extended Header Body Length (4 bits)

This field is used to indicate the length in bytes of the *Extended header Body* field.

4.2.2.1.14 Extended Header Type (4 bits)

NWK:0026.1 This field is used to indicate the type of the extended header. It **shall** be encoded according to Table 4.3.

Table 4.3 Extended Header Type encoding

Value	Description
0x00	Destination Wake Up Type
0x01	Incoming Routed RSSI Type
0x02..0x07	*Reserved*

4.2.2.2 Extended headers

4.2.2.2.1 Destination Wake Up Type

This extended header is used to carry beaming information relative to the source NodeID and destination NodeID for the actual route.

When using this extension:

- NWK:0027.1 The *Extended Header Body Length* field **shall** be set to 1
- NWK:0028.1 The *Extended Header Type* field **shall** be set to 0.

NWK:0029.1 The Extended Header Body **shall** be a bitmask encoded according to Figure 4.9

7	6	5	4	3	2	1	0
Reserved	Destination beaming 1000ms	Destination beaming 250ms	Source beaming 1000ms	Source beaming 250ms	Reserved		

Figure 4.9 Destination Wake Up Extension encoding

4.2.2.2.1.1 Destination beaming 1000ms (1 bit)

If this bit is set to 1, it indicates if the destination NodeID requires a Long Continuous Beam before the frame can be delivered.

4.2.2.2.1.2 Destination beaming 250ms (1 bit)

If this bit is set to 1, it indicates if the destination NodeID requires a short Continuous Beam before the frame can be delivered.

4.2.2.2.1.3 Source beaming 250ms (1 bit)

NWK:002A.1 This field is obsoleted. It **shall** be set to 0 by a sending node and ignored by a receiving node.

4.2.2.2.1.4 Source beaming 1000ms (1 bit)

NWK:002B.1 This field is obsoleted. It **shall** be set to 0 by a sending node and ignored by a receiving node.

4.2.2.2.2 Incoming Routed RSSI Type

This extension is used to report RSSI levels when returning a Routed Acknowledgment frame.

When using this extension:

- NWK:002C.1 The *Extended Header Body Length* field **shall** be set to 4
- NWK:002D.1 The *Extended Header Type* field **shall** be set to 1.

NWK:002F.1 The *Extended Header Body* field **shall** be encoded as shown in Figure 4.10.

7	6	5	4	3	2	1	0
Repeater 0 RSSI							
Repeater 1 RSSI							
Repeater 2 RSSI							
Repeater 3 RSSI							

Figure 4.10 Incoming Routed RSSI Type extension format

NWK:0030.1 The length of this extension **shall** always be 4 bytes, even if fewer repeaters are involved in routing the frame.

NWK:0031.1 A node returning a Routed Acknowledgement **shall** set these fields to 0x7F.

4.2.2.2.2.1 Repeater 0 RSSI (8 bits)

This field indicates the RSSI with which the routed frame was received from Repeater 1 (or the destination NodeID).

4.2.2.2.2.2 Repeater 1 RSSI (8 bits)

This field indicates the RSSI with which the routed frame was received from Repeater 2 (or the destination NodeID).

4.2.2.2.2.3 Repeater 2 RSSI (8 bits)

This field indicates the RSSI with which the routed frame was received from Repeater 3 (or the destination NodeID).

4.2.2.2.2.4 Repeater 3 RSSI (8 bits)

This field indicates the RSSI with which the routed frame was received from the destination NodeID.

NWK:0032.1 Each repeater **shall** update its respective RSSI byte with the RSSI value it measured when receiving the repeated frame.

NWK:0033.1 Each RSSI value **shall** be encoded using signed representation and **shall** be according to Table 4.4.

Table 4.4 - Repeater RSSI field encoding

Values		Description
Hexadecimal	Decimal	
0x80..0xFF	-128..-1	This value represents RSSI in dBm
0x00..0x7C	0..124	This value represents RSSI in dBm
0x7D	125	RSSI is below sensitivity and could not be measured
0x7E	126	Radio saturated, RSSI could not be measured
0x7F	127	RSSI measurement unavailable

4.2.2.3 Routing Frames

4.2.2.3.1 Routed frame

Table 4.5 shows the allowable frame sub-field configuration for a Routed frame. Note that all frames with this configuration are Routed frames.

Table 4.5 Routed frame sub-field configuration

R-Ack	R-Err	Direction
0	0	0

NWK:0034.1 A routed frame **may** comprise a DLPDU data payload.

NWK:0035.1 If transmitting using Channel Configuration 1,2 to a FL node destination, a sending node **shall** include the Destination Wake Up Extension if beaming is required prior to delivering the frame.

NWK:0036.1 A node sending a Routed frame **shall not** include the Incoming Routed RSSI Extension.

4.2.2.3.2 Routed Acknowledgment frame

Table 4.6 shows the allowable frame sub-field configuration for a Routed Acknowledgement frame. Note that all frames with this configuration are Routed Acknowledgement frames.

Table 4.6 Routed Acknowledgement frame sub-field configuration

R-Ack	R-Err	Direction
1	0	1

NWK:0037.1 A Routed Acknowledgement frame **shall not** comprise a DLPDU data payload.

NWK:0038.1 A node returning a Routed Acknowledgement **shall not** include the Destination Wake Up Extension.

NWK:0039.1 A node returning a Routed Acknowledgement **should** include the Incoming Routed RSSI Extension. If using the Incoming Routed RSSI Extension, a sending node **shall** set the repeater 0..4 values to 0x7F.

4.2.2.3.3 Routed Error frame

Table 4.7 shows the allowable frame sub-field configuration for a Routed Error frame. Note that all frames with this configuration are Routed Error frames

Table 4.7 Routed Error frame sub-field configuration

R-Ack	R-Err	Direction
0	1	1

NWK:003A.1 A Routed Error frame **shall not** comprise a DLPDU data payload.

NWK:003B.1 A node returning a Routed Error **shall not** include the Destination Wake Up Extension.

NWK:003C.1 A node returning a Routed Error **shall not** include the Incoming Routed RSSI Extension.

4.2.3 Explore NPDUs

Explore NPDUs are used to transmit a frame when the sending node does not know a route to the destination.

4.2.3.1 Explore frame general header format

Explore NPDUs use a general header outlines in Figure 4.11.

Octet	7	6	5	4	3	2	1	0
1	Version			Command				
2	Reserved					Stop	Direction	Source Routed
3	Session Tx Random Interval							
4	TTL				Repeater Count			
5	Repeater 0							
6	Repeater 1							
7	Repeater 2							
8	Repeater 3							

Figure 4.11 Explore frame general header

4.2.3.1.1 Version (3 bits)

NWK:003D.1 The *Version* field indicates the type of explore frame being sent. A sending node **shall** set this field to 0x01. All other values are reserved.

4.2.3.1.2 Command (5 bits)

NWK:003E.1 The *Command* field indicates the type of Explore Frame being sent. It **shall** be encoded according to Table 4.8.

Table 4.8 Explore Frame Header::Command field encoding

Value	Description
0x00	Normal Explore Frame: This frame is used to find a new route to a destination. It may carry an application payload.
0x01	Inclusion Request Explore Frame: This frame is used during inclusion scenarios and will be repeated across networks.
0x02	Search Result Explore Frame: This frame is used to confirm the reception of a Normal Explore Frame.
0x03..0x1F	*Reserved*

4.2.3.1.3 Direction (1 bit)

The *Direction* field indicates the direction of the frame.

NWK:003F.1 A node sending out an Explore frame **shall** set this field to 0 (outgoing frame).

NWK:0040.1 A node returning an Explore Frame in response to another **shall** set this field to 1.
NWK:0041.1 The value 0 **shall** indicate that Repeater 0 shall be the first to repeat the frame.
NWK:0042.1 The value 1 **shall** indicate that the last repeater in the list **shall** be the first to repeat the frame.

4.2.3.1.4 Source Routed (1 bit)

NWK:0043.1 The *Source Routed* field indicates if the Explore Frame contains a valid route that repeaters **shall** follow or if the repeaters **shall** add themselves when they forward the frame.

NWK:0044.1 The value 0 **shall** indicate that the initial repeater list is zeroed out and each node receiving the frame will add itself into the repeater list and repeat the frame.

NWK:0045.1 The value 1 **shall** indicate that the initial repeater list is defined and **shall** be respected by repeaters.

4.2.3.1.5 Stop (1 bit)

The *Stop* field is used to stop all current Explore Frame repeating. This is used to ensure that a frame can be sent without excessive interference from Explore Frame repeating.

NWK:0046.1 The *Stop* field indicates that AL nodes **shall** abort repeating any queued Explore Frame.

A queued Explore Frame is an Explore Frame that has been queued for repeating but is waiting the Session Tx Random Interval to expire before being sent.

4.2.3.1.6 Session Tx Random Interval (8 bits)

NWK:0047.1 The *Session Tx Random Interval* field **shall** indicate the back off time for each repeater, i.e. a delay to apply before repeating an Explore Frame.

NWK:0048.1 This field **shall** be ignored if the *Command* field is set to 0x02 (Explore Command Search Result), in this case, the Explore Frame **shall** be repeated immediately.

NWK:0049.1 For an outgoing Explore Frame, repeater **shall** apply a back off in ms given by:

$$TxRandomInterval = \begin{cases} 50 + 200 \times (R-1) + 2 \times Rand(S), & R > 0 \\ [0..S], & R = 0 \end{cases}$$

With:

- R the Repeater Number (hop)
- S the Session Tx Random Interval
- Rand(x) a function returning a random number between 0 and x

4.2.3.1.7 TTL (4 bits)

The *TTL* field indicates how many more repeaters can the frame go through. A node sending a new Explore Frame **shall** set this field to 4 and each repeater **shall** decrement this field. NWK:004A.1

NWK:004B.1 A node repeating an Explore frame **shall**:

- decrement this field if the *Direction* field is set to 0
- increment this field if the *Direction* field is set to 1.

NWK:004C.1 If the *TTL* field is set to 0x00 and the *Direction* field is set to 0, a receiving node **shall not** repeat the Explore Frame.

4.2.3.1.8 Repeater Count (4 bits)

This field is used to indicate the total number of repeaters for the Explore Frame.

NWK:004D.1 This field **shall** be in the range 1..4.

4.2.3.1.9 Repeater 0 (8 bits)

The *Repeater 0* field is used to indicate the NodeID of the first repeater for this Explore Frame.

The value 0 **shall** indicate that the Explore Frame has not been repeated yet. NWK:004E.1

4.2.3.1.10 Repeater 1 (8 bits)

The *Repeater 1* field is used to indicate the NodeID of the second repeater for this Explore Frame.

NWK:004F.1 The value 0 **shall** indicate that the Explore Frame has not been repeated by a second repeater yet.

4.2.3.1.11 Repeater 2 (8 bits)

The *Repeater 2* field is used to indicate the NodeID of the third repeater for this Explore Frame.

NWK:0050.1 The value 0 **shall** indicate that the Explore Frame has not been repeated by a third repeater yet.

4.2.3.1.12 Repeater 3 (8 bits)

The *Repeater 3* field is used to indicate the NodeID of the fourth repeater for this Explore Frame.

NWK:0051.1 The value 0 **shall** indicate that the Explore Frame has not been repeated by a fourth repeater yet.

4.2.3.2 Normal Explore Frame

Table 4.9 shows the allowable frame sub-field configuration for Normal Explore Frames. Note that all frames with the *Command* sub-field set to 0x00 are Normal Explore Frames.

Table 4.9 Normal Explore frame subfield configuration

Command	Direction	Source Routed	Stop	Session Tx Random Interval
0x00	0	0	0	> 0

NWK:0052.1 A sending node **should** set the Session Tx Random Interval subfield to *nwkRecommendedSessionTxRandomInterval*.

4.2.3.2.1 Frame format

The Normal Explore Frame has no explore frame specific payload. Its format is shown in Figure 4.12.

Octets	7	6	5	4	3	2	1	0
8	General Explore Frame Header *(Command = 0x00). Refer to* Figure 4.11.							

Figure 4.12 Normal Explore Frame format

4.2.3.2.2 Fields description

There is no Command Specific Payload for a Normal Explore Frame, it will only comprise the Explore Frame general header described in 4.2.3.1 Explore frame general header format.

4.2.3.2.3 Additional payload

NWK:0053.1 A Normal Explore Frame **may** carry a DLPDU payload. Refer to Figure 4.6.

4.2.3.3 Inclusion Request Explore Frame

Table 4.10 shows the allowable frame sub-field configuration for Inclusion Request Explore Frames. Note that all frames with the *Command* sub-field set to 0x01 are Inclusion Request Explore frames.

Table 4.10 Inclusion Request Explore frame subfield configuration

Command	Direction	Source Routed	Stop	Session Tx Random Interval
0x01	0	0	0	>0

NWK:0054.1 A sending node **should** set the Session Tx Random Interval subfield to *nwkRecommendedSessionTxRandomInterval*.

4.2.3.3.1 Frame format

NWK:0055.1 The Inclusion Request Explore Frame format **shall** comply with Figure 4.13.

Octets	7	6	5	4	3	2	1	0
8	General Explore Frame Header *(Command = 0x01). Refer to* Figure 4.11.							
1	Network HomeID 1							
1	Network HomeID 2							
1	Network HomeID 3							
1	Network HomeID 4							

Figure 4.13 Inclusion Request Explore Frame format

4.2.3.3.2 Fields description

4.2.3.3.2.1 Network HomeID (4 bytes)

This field is used to indicate the HomeID of the repeating nodes.

NWK:0056.1 This field **shall** be set to 0 by a node sending an Inclusion Request Explore Frame.

NWK:0057.1 A node repeating an Inclusion Request Explore Frame **shall** set this field to its HomeID.

This is the HomeID of the node repeating the frame. When NWI Mode is enabled, Inclusion Request Explore Frames are repeated even if they belong to another HomeID.

4.2.3.3.3 Additional payload

NWK:0058.1 An Inclusion Request Explore Frame **shall** carry a DLPDU using the Z-Wave Protocol Command Class and one of the following commands:

- Node Information Frame Command
- SmartStart Prime Command
- SmartStart Inclusion Request Command
- SmartStart Included Node Information Command

NWK:0059.1

An Inclusion Request Explore Frame **shall not** carry any other commands than the list mentioned above. This is illustrated in Figure 4.14.

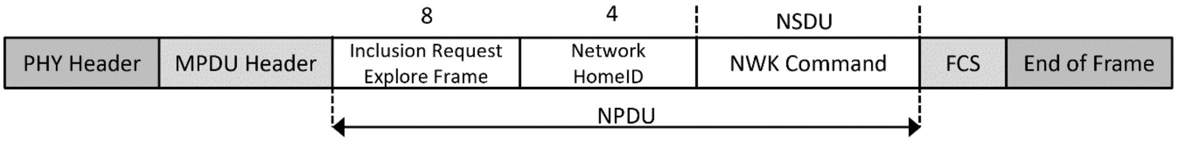


Figure 4.14 Inclusion Request Explore Frame with Z-Wave NWK command

4.2.3.4 Search Result Explore Frame

Table 4.11 shows the allowable frame sub-field configuration for Search Result Explore Frames. Note that all frames with the *Command* sub-field set to 0x02 are Search Result Explore frames.

Table 4.11 Search Result Explore frame subfield configuration

Command	Direction	Source Routed	Stop	Session Tx Random Interval
0x02	1	1	1	0x00

4.2.3.4.1 Frame format

NWK:005A.1 The Search Result Explore Frame format **shall** comply with Figure 4.15.

Octets	7	6	5	4	3	2	1	0
8	General Explore Frame Header *(Command = 0x02). Refer to* Figure 4.11.							
1	NodeID							
1	Frame Handle							
1	TTL Result				Repeater Count Result			
1	Repeater 0 Result							
1	Repeater 1 Result							
1	Repeater 2 Result							
1	Repeater 3 Result							

Figure 4.15 Search Result Explore Frame format

4.2.3.4.2 Fields description

4.2.3.4.2.1 NodeID (8 bits)

The *NodeID* field is used to indicate the sending NodeID of the Normal Explore Frame for which the Search Result Explore frame is returned.

4.2.3.4.2.2 Frame Handle (8 bits)

NWK:005B.1 The *Frame Handle* field **shall** contain the Sequence number found in the MPDU Header of the Normal Explore Frame for which the Search result Explore Frame is returned.

4.2.3.4.2.3 TTL result (4 bits)

The *TTL Result* field contains the value of the *TTL* field as it was received by the destination NodeID in the Normal Explore Frame that triggered the Search Result Explore Frame to be returned.

4.2.3.4.2.4 Repeater Count Result (4 bits)

The *Repeater Count Result* field contains the value of the *Repeater Count* field as it was received by the destination NodeID in the Normal Explore Frame that triggered the Search Result Explore Frame to be returned.

4.2.3.4.2.5 Repeater 0 Result (8 bits)

The *Repeater 0 Result* field contains the value of the *Repeater 0* field as it was received by the destination NodeID in the Normal Explore Frame that triggered the Search Result Explore Frame to be returned.

4.2.3.4.2.6 Repeater 1 Result (8 bits)

The *Repeater 1 Result* field contains the value of the *Repeater 1* field as it was received by the destination NodeID in the Normal Explore Frame that triggered the Search Result Explore Frame to be returned.

4.2.3.4.2.7 Repeater 2 Result (8 bits)

The *Repeater 2 Result* field contains the value of the *Repeater 2* field as it was received by the destination NodeID in the Normal Explore Frame that triggered the Search Result Explore Frame to be returned.

4.2.3.4.2.8 Repeater 3 Result (8 bits)

The *Repeater 3 Result* field contains the value of the *Repeater 3* field as it was received by the destination NodeID in the Normal Explore Frame that triggered the Search Result Explore Frame to be returned.

4.2.3.4.3 Additional payload

NWK:005C.1 A Search Result Explore Frame **shall not** carry a DLPDU. This is illustrated in Figure 4.16.

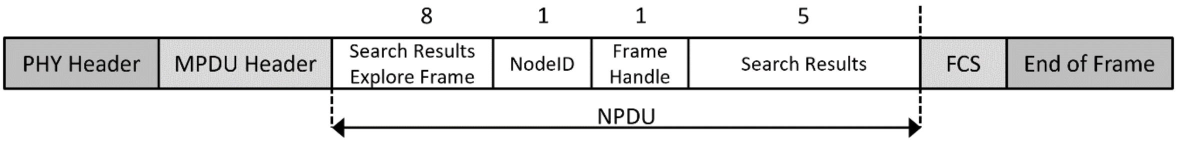


Figure 4.16 Search Result Explore frame format

4.3 Command frames

The commands defined by the Z-Wave NWK layer are categorized in Command Classes. These Command Classes are listed in Table 4.12.

Table 4.12 Z-Wave NWK Layer Command Classes

PHY/MAC	Command Class Identifier	Command Class Name	Reference
Z-Wave	0x00	No Operation Command Class	4.3.1
Z-Wave	0x01	Z-Wave Protocol Command Class	4.3.2

These Command Classes are designed for Z-Wave network formation and maintenance functionalities. The following sections illustrates how the Network Layer Management (NLME) **shall** build the individual commands for transmission.

NWK:005D.1

NWK:0204.1 During the transmission of each of these commands, the NLME **shall** construct the network layer protocol data (NPDU) part of the frame as illustrated in Figure 4.17.

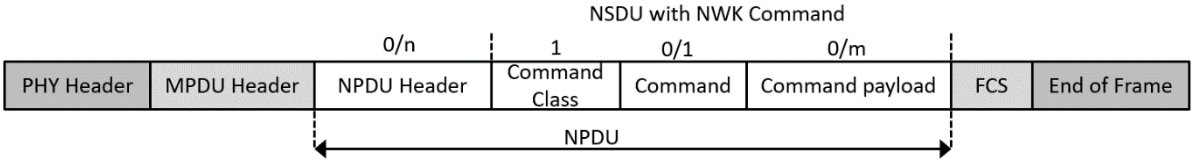


Figure 4.17The Z-Wave Network Layer Command Frame Format

The command format is identical to the DLPDU format defined in [G.9959], comprising one Command Class byte, one (optional) Command byte and an optional variable length Command Payload.

NWK:005E.1 NWK Commands **shall not** use any segmentation or encryption. No application payload **shall** be added from upper layers. These are illustrated in Figure 4.18.

NWK:005F.2 Command Classes in the range 0x00..0x1F s**hall** be considered as NWK Command Classes (both for Z-Wave and Z-Wave Long Range).

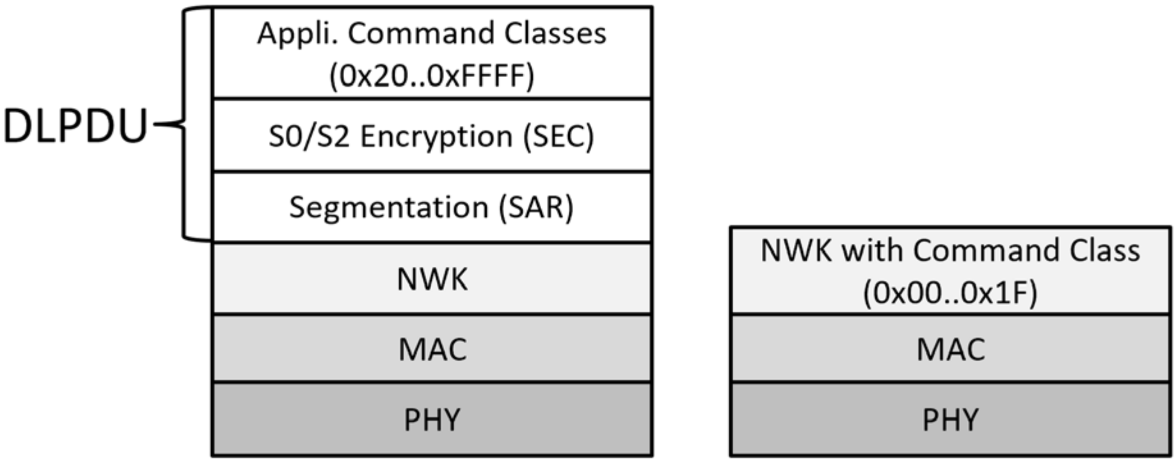

DLPDU
Appli. Command Classes (0x20..0xFFFF)
S0/S2 Encryption (SEC)
Segmentation (SAR)
NWK
MAC
PHY
NWK with Command Class (0x00..0x1F)
MAC
PHY

Figure 4.18 Z-Wave and Z-Wave Long Range NWK Command Encapsulation

4.3.1 No Operation Command Class

This Command Class is a special command frame that is used by a Z-Wave network manager entity to test the availability of a node in a network.

NWK:0060.1 The No Operation Command Class (NOP) **shall** be supported by all Z-Wave nodes. This command class **shall not** have any associated commands or payload. Therefore, the network layer protocol data **shall** only be composed of a *Command Class* byte with no payload during the transmission this Command Class. This is shown in Figure 4.19.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NO_OPERATION (0x00)							

Figure 4.19 NOP Command Class format

NWK:0061.1 This Command Class **may** be used to verifying if an excluded node is still part of the network. This Command **may** also be used on application level e.g. checking if a SUC/SIS is reachable from a new node in the network.

4.3.2 Z-Wave Protocol Command Class

NWK:0062.1 This Command Class is used for setup and maintenance of networks. The Z-Wave Protocol Command Class **shall** be supported by all nodes. Some commands are used only for certain nodes types or network roles (e.g. end nodes or controllers); and in this case, it will be indicated in the command themselves.

The Command frames defined by this Command Class are listed in Table 4.13.

Table 4.13 Z-Wave Protocol Command Class Commands

Command Frame Identifier	Command Name	Reference
0x01	Node Information Frame Command	4.3.2.1
0x02	Request Node Information Frame Command	4.3.2.2
0x03	Assign IDs Command	4.3.2.3
0x04	Find Nodes in Range Command	4.3.2.4
0x05	Get Nodes in Range Command	4.3.2.5
0x06	Range Info Command	4.3.2.6
0x07	Command Complete Command	4.3.2.7
0x08	Transfer Presentation Command	4.3.2.8
0x09	Transfer Node Information Command	4.3.2.9
0x0A	Transfer Range Information Command	4.3.2.10
0x0B	Transfer End Command	4.3.2.11
0x0C	Assign Return Route Command	4.3.2.12
0x0D	New Node Registered Command	4.3.2.13
0x0E	New Range Registered Command	4.3.2.14
0x0F	Transfer New Primary Controller Complete Command	4.3.2.15
Ox10	Automatic Controller Update Start Command	4.3.2.16
0x11	SUC Node ID Command	4.3.2.17
0x12	Set SUC Command	4.3.2.18
0x13	Set SUC ACK Command	4.3.2.19
0x14	Assign SUC Return Route Command	4.3.2.20
0x15	Static Route Request Command	4.3.2.21
0x16	Lost Command	4.3.2.22
0x17	Accept Lost Command	4.3.2.23
0x18	NOP Power Command	4.3.2.24
0x19	Reserve Node IDs Command	4.3.2.25
0x1A	Reserved IDs Command	4.3.2.26

Command Frame Identifier	Command Name	Reference
0x1B	Reserved	N/A
0x1C	Reserved	N/A
0x1D	Reserved	N/A
0x1E	Reserved	N/A
0x1F	Nodes Exist Command	4.3.2.27
0x20	Nodes Exist Reply Command	4.3.2.28
0x21	Reserved	N/A
0x22	Set NWI Mode Command	4.3.2.29
0x23	Exclude Request Command	4.3.2.30
0x24	Assign Return Route Priority Command	4.3.2.31
0x25	Assign SUC Return Route Priority Command	4.3.2.32
0x26	SmartStart Included Node Information Command	4.3.2.33
0x27	SmartStart Prime Command	4.3.2.34
0x28	SmartStart Inclusion Request Command	4.3.2.35

4.3.2.1 Node Information Frame Command

The Node Information Frame command is used to advertise the capabilities of the sending node.

4.3.2.1.1 Frame format

NWK:0063.1 The Node Information Frame Command **shall** be formatted as illustrated in Figure 4.20.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_ZWAVE (0x01)</td></tr>
<tr><td colspan="8">Command = ZWAVE_CMD_NODE_INFO (0x01)</td></tr>
<tr><td>Listening</td><td>Routing</td><td colspan="3">Supported speed</td><td colspan="3">Protocol version</td></tr>
<tr><td>Optional functionality</td><td>Sensor 1000ms</td><td>Sensor 250ms</td><td>Beam capability</td><td>Routing End Node</td><td>Specific Device</td><td>Controller</td><td>Security</td></tr>
<tr><td colspan="5">Reserved</td><td colspan="3">Speed Extension</td></tr>
<tr><td colspan="8">Basic Device Type</td></tr>
<tr><td colspan="8">Generic Device Class</td></tr>
<tr><td colspan="8">Specific Device Class</td></tr>
<tr><td colspan="8">Command Class 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Command Class N</td></tr>
</table>

Figure 4.20 Node Information Frame Command format

4.3.2.1.1.1 Protocol Version (3 bits)

NWK:0064.1 This field is used to advertise the version of the protocol that is implemented by the sending node. The Protocol Version field **shall** be formatted as indicated in Table 4.14.

Table 4.14 Protocol version bit description

Values	Description
0	Reserved
1	Z Wave version 2.0
2	Z-Wave version ZDK 5.0x, ZDK 4.2x
3	Z-Wave version ZDK 4.5x and ZDK 6.0x
4..7	*Reserved*

4.3.2.1.1.2 Supported Speed (3 bits)

NWK:0065.1 The *Supported Speed* field indicates the transmission data rate supported by the sending node. This field **shall** be treated as a bitmask and **shall** have at least one speed bits set as depicted in Table 4.15.

Table 4.15 Maximum Speed supported by the node

Bit	Description
3	9.6 kbits/second is supported
4	40 kbits/second is supported
5	*Reserved*

NWK:0066.1 If this field is set to 0, (no bit is set to 1), a receiving node **shall** assume that the maximum supported speed by the sending node is 9.6 kbits/second.

NWK:0067.1 The bits set in this field **shall** comply with channel configuration requirements from [G.9959]

4.3.2.1.1.3 Speed Extension (3 bits)

NWK:0068.1 The *Speed Extension* field is used to describe the if the node support additional data rate to the ones advertised in the *Supported Speed* field. This field **shall** be treated as a bitmask and **shall** be encoded according to Table 4.16.

NWK:0069.1 The bits set in this field **shall** comply with channel configuration requirements from [G.9959]

Table 4.16 Node Information Frame Command Speed extension description

Bit	Description
0	100 kbits/second is supported
1	*Reserved*
2	*Reserved*

4.3.2.1.1.4 Routing (1 bit)

The *Routing* field is used to indicate if the node can repeat Routed NPDUs and Explore NDPUs.

NWK:006A.2 This field **shall** be set if the network layer supports being a repeater. If the network layer does not support being a repeater this field **shall** be set to 0.

NWK:006B.1 Controller nodes trying to calculate routes **shall** assume that nodes with both the *Routing* field set to 1 and the *Listening* field set to 1 will repeat frames.

4.3.2.1.1.5 Listening (1 bit)

The *Listening* field is used to advertise if the node is always listening or not.

NWK:006C.1 AL nodes **shall** be set this field to 1.
NWK:006D.1 FL and NL nodes **shall** set this field to 0.

4.3.2.1.1.6 Security (1 bit)

NWK:006E.1 The *Security* field **shall** be set to 1 if the sending node supports secure communication. The *Security* field **shall** be set to 0 if the node does not support secure communication.

NWK:006F.1 Nodes supporting the S0 or S2 Command Classes **shall** set this field to 1. Refer to [TECC].

4.3.2.1.1.7 Controller (1 bit)

NWK:0070.1 The *Controller* field **shall** be set to 1 if the sending node is a controller, and to 0 if the sending node is an end node.

NWK:0071.1 If this field is set to 1, the Node Information Frame Command **shall** include a *Basic Device Type* field.

NWK:0072.1 If this field is set to 0, the *Routing End Node* field **shall** be set to 1.

4.3.2.1.1.8 Specific Device (1 bit)

NWK:0073.1 This bit field **shall** set to 1 when the node information frame contains specific device type field.

4.3.2.1.1.9 Routing End Node (1 bit)

NWK:0074.1 The *Routing End Node* field **shall** be set to 1 if the sending node is an end node.

NWK:0075.1 If this field is set to 1, the Node Information Frame Command **shall not** include a Basic Device Type field.

NWK:0076.1 If this field is set to 0, the *Controller* bit field **shall** be set to 1.

4.3.2.1.1.10 Beam capability (1 bit)

This field is used to indicate if the sending node can issue Wake Up beams.

NWK:0077.1 The value 0 **shall** indicate that the sending node cannot wake up FL nodes.
The value 1 **shall** indicate that the sending node can wake up FL nodes.

4.3.2.1.1.11 Sensor 250ms (1 bit)

This bit is set to 1 when the node is a FLiRS node and wakes up every 250ms to check if a wakeup beam is present.

4.3.2.1.1.12 Sensor 1000ms (1 bit)

This bit is set to 1 when the node is a FLiRS node and wakes up every 1000ms to check if a wakeup beam is present.

4.3.2.1.1.13 Optional Func. (1 bit)

The *Optional Functionality* field is used to indicate that this node supports additional application Command Classes to the mandatory minimum for the advertised Generic/Specific Device Class defined in [DC].

NWK:0078.1 Sending nodes **should** set this field to 1.

4.3.2.1.1.14 Basic Device Type (8 bits)

NWK:0079.1 The *Basic Device Type* field **shall** be included in the Node Information Frame Command if the *Controller* field is set to 1.

NWK:007A.1 This field **shall** be omitted if the *Controller* field is set to 0.

NWK:007B.1 This field **shall** be encoded according to Table 4.17

Table 4.17 Node Information Frame Command - Basic Device Type encoding

Value	Description
0x00	*Reserved*
0x01	The node is a portable controller, it may change location regularly.
0x02	The node is a static controller, it is not supposed to change location
0x02..0xFF	*Reserved*

4.3.2.1.1.15 Generic device class (8 bits)

NWK:007C.1 The *Generic Device Class* field contains an identifier that identifies what Generic Device Class this node is part of and **shall** be set by the application.

For a detailed description of all available Generic Device Classes, refer to [DC] for Z-Wave devices, [DT] for Z-Wave Plus devices, and [DTV2] Z-Wave Plus v2 devices.

4.3.2.1.1.16 Specific device class (8 bits)

NWK:007D.1 The *Specific Device Class* field specifies what Specific Device Class is implemented by the node, and this **shall** be set by the application.

For a detailed description of all available Specific Device Classes, refer to [DC] for Z-Wave devices, [DT] for Z-Wave Plus devices, and [DTV2] Z-Wave Plus v2 devices.

4.3.2.1.1.17 Command class (N bytes)

This field is used to advertise the list of Command Classes (Refer to [ACC], [MCC], [TECC] and [NPCC]) supported by the sending node using non-secure communication.

NWK:007E.1 The *Command Class* field **shall not** be longer than 35 bytes.

NWK:007F.1 The field **shall** advertise the list of Command Classes that the node supports. Command Classes advertised in this field **shall** be in the range 0x21..0xFFFF. Command Classes in the range 0x00..0x20 s**hall not** be advertised in this field.

4.3.2.1.2 When generated

NWK:0080.1 When generated, the fields **shall** be configured as follow:

NWK:0081.1
- Protocol Version field **shall** be set to 0x03

4.3.2.1.3 Effect on receipt

On receipt of this command, a receiving node is notified of the sender's Z-Wave capabilities and non-secure supported Command Classes.

NWK:0082.1 If the network layer does not expect to receive a Node Information Frame Command (i.e. not in add mode or did not issue a Request Node Information Frame Command), the command (or its data) **shall** be forwarded to the upper protocol layer (application).

4.3.2.2 Request Node Information Frame Command

This command is used to request a node to return a Node Information Frame Command.

4.3.2.2.1 Frame format

The Request Node Information Frame Command **shall** be formatted as illustrated in Figure 4.21. NWK:0083.1

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_REQUEST_NODE_INFO (0x02)							

Figure 4.21 Request Node Information Frame Command format

4.3.2.2.2 When generated

This command **shall** be sent using singlecast addressing and **shall not** be sent using multicast addressing. NWK:0084.1

4.3.2.2.3 Effect on receipt

On receipt of this command, a receiving node **shall** return a Node Information Frame Command in response. A receiving node **shall not** return a response if this command is received via multicast addressing. NWK:0085.1

4.3.2.3 Assign IDs Command

This command is used to assign NodeID and HomeID to the receiving node.

4.3.2.3.1 Frame format

NWK:0086.1 The Assign IDs Command **shall** be formatted as illustrated in Figure 4.22.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_ASSIGN_IDS (0x03)							
NodeID							
HomeID 1							
HomeID 2							
HomeID 3							
HomeID 4							

Figure 4.22 Assign IDs Command format

4.3.2.3.1.1 NodeID (8 bits)

The *NodeID* field is used to assign or unassign a NodeID to a node.

NWK:0087.1 The value 0x00 **shall** indicate the node that it **shall** be excluded from the network.
Values in the range 0x01..0xE8 **shall** indicate the new NodeID assigned to the receiving node.

4.3.2.3.1.2 HomeID (4 bytes)

The *HomeID* field is used to assign or unassign the HomeID to a node.

NWK:0088.1 The value 0x00000000 **shall** indicate the node that it **shall** be excluded from the network and assign itself a random HomeID or NWI HomeID based on its S2 DSK.
Other values **shall** indicate the new HomeID assigned to the receiving node.

4.3.2.3.2 When generated

NWK:0089.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using multicast addressing.

4.3.2.3.3 Effect on receipt

NWK:008A.1 On receipt of this command, a receiving node **shall** update its HomeID and NodeID if and only if it is currently in Learn Mode.

NWK:008B.1 A receiving node **shall** ignore the command if it is received via multicast addressing.
A receiving node **shall** ignore this command if is not in learn mode.

NWK:008C.1 A receiving node **shall** ignore this command if the NodeID field is set to a value greater than 0xE8.

4.3.2.4 Find Nodes in Range Command

This command is used to request the receiving node to find the nodes in direct range.

4.3.2.4.1 Frame format

NWK:008D.1 The Find Nodes in Range Command **shall** be formatted as illustrated in Figure 4.23

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_ZWAVE (0x01)</td></tr>
<tr><td colspan="8">Command = ZWAVE_CMD_FIND_NODES_IN_RANGE (0x04)</td></tr>
<tr><td>Speed Present</td><td colspan="2">Reserved</td><td colspan="5">Bitmask length</td></tr>
<tr><td colspan="8">Node bitmask 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Node bitmask N</td></tr>
<tr><td colspan="8">Wake Up Time (optional Rx)</td></tr>
<tr><td colspan="5">Reserved</td><td colspan="3">Data rate (optional Rx)</td></tr>
</table>

Figure 4.23 Find Nodes in Range Command format

4.3.2.4.1.1 Reserved (2 fields)

NWK:008E.1 Reserved fields **shall** be set to 0 by a sending node and **shall** be ignored by a receiving node.

4.3.2.4.1.2 Speed Present (1 bit)

This field is used to advertise if this command frame contains the data rate to be used during neighbour discovery.

NWK:008F.1 This field **shall** be set to 0 if the data rate option is not defined.

NWK:0090.1 This field **shall** be set to 1 if the data rate to be used is defined.

4.3.2.4.1.3 Bitmask Length (5 bits)

This field is used to advertise the length in bytes of the *Node bitmask* field.

4.3.2.4.1.4 Node bitmask (N bytes)

NWK:0091.1 This field is used to advertise the neighbouring nodes ID mask that the receiving node **shall** try to reach in direct range during the range test.

NWK:0092.1 The length of this field in bytes **shall** be according to the *Bitmask Length* field value.

NWK:0093.1 This field **shall** be treated as a bit mask, and LSB in Node bitmask 1 represents Node 1. This field **shall** be encoded as follow:

- Bit 0 in Node bitmask 1 indicates the NodeID 1
- Bit 1 in Node bitmask 1 indicates the NodeID 2
- ...

The bit value 0 **shall** be used to advertise that the corresponding node should not try to find the node. The bit value 1 **shall** be used to advertise that the corresponding node should try to find the node. NWK:0094.1

The first byte of this field **shall** represent node 1..8. NWK:0095.1

4.3.2.4.1.5 Wake Up Time (optional Rx) (8 bits)

This field is used to indicate the Wake Up beam duration that **shall** be used for waking up the all the destination nodes present in the bitmask. NWK:0096.1

All nodes present in the Node Bitmask field **shall** have the same Wake Up Time setting. NWK:0097.1

Table 4.18 Wake Up Time

Values	Description
0x00	No Wakeup beam needed (AL nodes)
0x01	1000ms wake up beam
0x02	250ms wake up beam
0x03..0xFF	*Reserved*

Note: This field **shall** be present in the frame when transmitting the Find Node in Range command, and a receiving node **shall** verify if the field is present in the frame when parsing the frame. NWK:0098.1

If this field is not present in the frame, a receiving node **shall** use the Wakeup Time value 0x00. NWK:0099.1

4.3.2.4.1.6 Data rate (optional Rx) (3 bits)

The data rate to be used during neighbour discovery. Speed present bit **shall** be set to 1 when this field is included. This field value **shall** be encoded according to Table 4.19. NWK:009A.1

Table 4.19 Find Node in Range data rate encoding

Value	Speed
0x01	9.6 kbps
0x02	40 kbps
0x03	100 kbps
0x04..0x07	*Reserved*

This field **shall** be present in the frame when a node transmits the Find Nodes in Range command, and a receiving node **shall** verify if the field is present in the frame when parsing the frame. NWK:009B.1

If this field is not present in the frame, a receiving node **shall** use the Data Rate value 0x01. NWK:009C.1

4.3.2.4.2 When generated

NWK:009D.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using multicast addressing.

NWK:009E.1 This command **shall not** be sent with the *Node Bitmask* field set to 0.

4.3.2.4.3 Effect on receipt

NWK:009F.1 On receipt of this command, a receiving node **shall** issue a NOP Power Command to all NodeIDs indicated by the *Node Bitmask* field and subsequently return a Command Complete Command to the node that initiated the range test.

NWK:00A0.1 All Z-Wave nodes **shall** keep a bitmask record of the last range test.

NWK:00A1.1 A receiving node **shall** ignore the command if it is received via multicast addressing.

This procedure is depicted in Figure 4.24

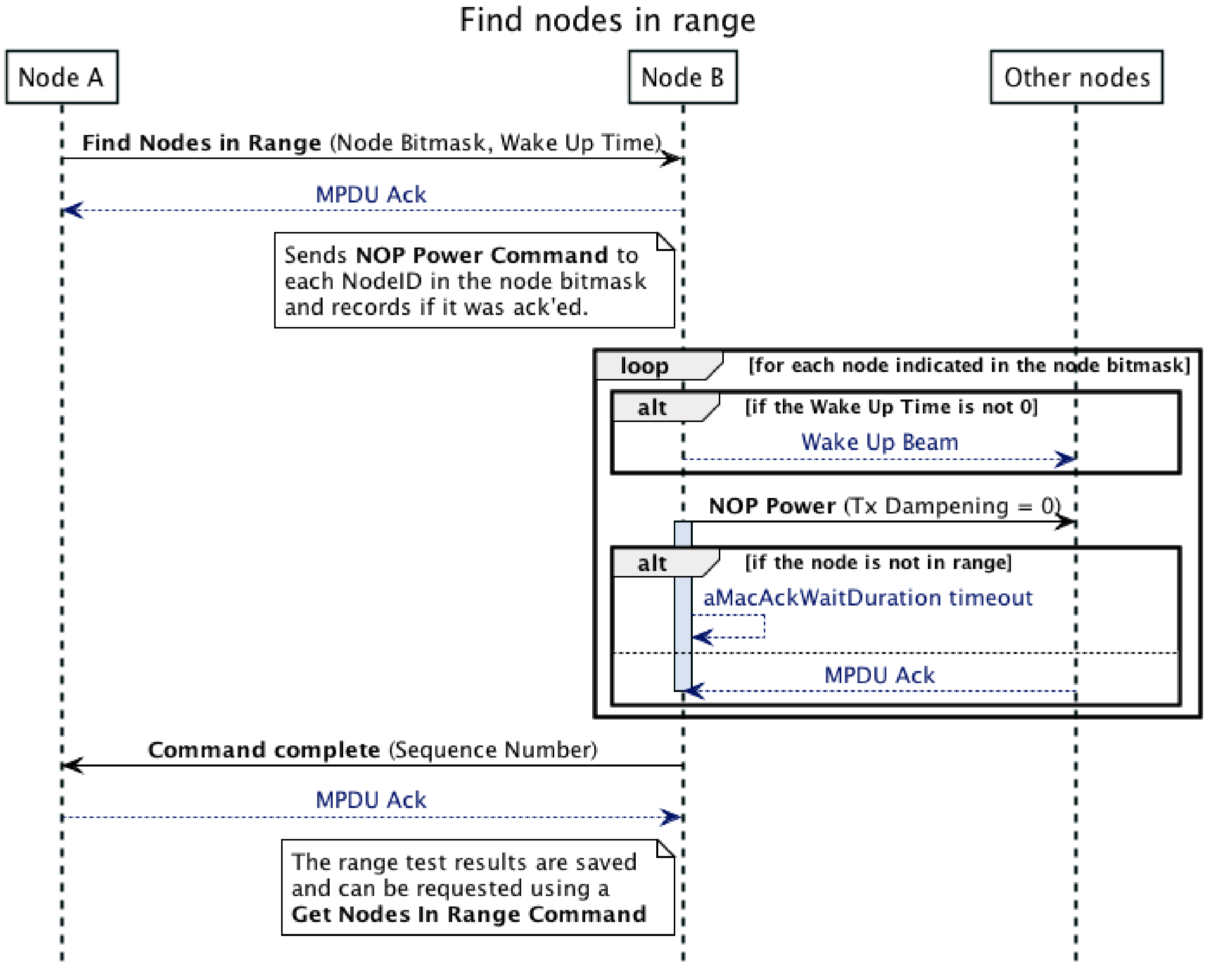

Find nodes in range
Node A
Node B
Other nodes
Find Nodes in Range (Node Bitmask, Wake Up Time)
MPDU Ack
Sends NOP Power Command to each NodeID in the node bitmask and records if it was ack'ed.
loop
[for each node indicated in the node bitmask]
alt
[if the Wake Up Time is not 0]
Wake Up Beam
NOP Power (Tx Dampening = 0)
alt
[if the node is not in range]
aMacAckWaitDuration timeout
MPDU Ack
Command complete (Sequence Number)
MPDU Ack
The range test results are saved and can be requested using a Get Nodes In Range Command

Figure 4.24 Range test procedure

4.3.2.5 Get Nodes in Range Command

The Get Nodes in Range Command is used to request the list of direct range neighbours detected with the last range test.

4.3.2.5.1 Frame format

NWK:00A2.1 The Get Nodes in Range Command **shall** be formatted as illustrated in Figure 4.25.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_GET_NODES_IN_RANGE (0x05)							
Wake Up Time (optional)							

Figure 4.25 Get Nodes in Range Command format

4.3.2.5.2 Wake Up Time (optional)

NWK:00A6.1 This field **shall** be identical to the field of the same name in the Find Nodes In Range command in section 4.3.2.4.

4.3.2.5.3 When generated

NWK:00A3.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using multicast addressing.

4.3.2.5.4 Effect on receipt

NWK:00A4.1 On receipt of this command, a receiving node **shall** return a Range Info Command.

NWK:00A5.1 A receiving node **shall not** return a response if this command is received via multicast addressing.

4.3.2.6 Range Info Command

The Range Info Command is used to advertise the list of direct range neighbours detected with the last range test.

4.3.2.6.1 Frame format

NWK:00A6.1 The Range Info Command **shall** be formatted as illustrated in Figure 4.26.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_RANGE_INFO (0x06)							
Reserved			Bitmask Length				
Node Bitmask 1							
...							
Node Bitmask N							
Wakeup Time (optional Rx)							

Figure 4.26 Range Info Command format

4.3.2.6.1.1 Reserved (3 bits)

Unused and must be set to zero

4.3.2.6.1.2 Bitmask Length (5 bits)

This field advertise the length in bytes of the Node bitmask field.

4.3.2.6.1.3 Node bitmask (N bytes)

This field used to advertise the neighbouring nodes ID mask that describes the list of nodes in the sending node vicinity.

NWK:00A7.1 The length of this field in bytes **shall** be according to the *Bitmask Length* field value.

NWK:00A8.1 This field **shall** be treated as a bit mask, and LSB in Node bitmask 1 represents Node 1. This field **shall** be encoded as follow:

- Bit 0 in Node bitmask 1 indicates the NodeID 1
- Bit 1 in Node bitmask 1 indicates the NodeID 2
- ...

NWK:00A9.1 The bit value 0 **shall** be used to advertise that the corresponding node is not a neighbouring node that the sending node can communicate with in direct range. The bit value 1 **shall** be used to advertise that the corresponding node is a neighbouring node.

NWK:00AA.1 The first byte of this field **shall** represent node 1...8.

4.3.2.6.1.4 Wake up Time (optional Rx) (8 bits)

This field specifies the wakeup beam duration that was specified in the Find Nodes In Range frame.

NWK:00AB.1 Note: This field **shall** be present in the frame when transmitting the Range Info Command, and a receiving node **shall** verify if the field is present in the frame when parsing the frame.

NWK:00AC.1 If this field is not present in the frame, a receiving node **shall** assume a Wakeup Time value 0x00.

4.3.2.6.2 When generated

NWK:00AD.1 The node bitmask field **shall** contain the result of the last range test initiated by the Find Nodes in Range Command.

NWK:00AE.1 If no range test was initiated for the current network, the *Node Bitmask* field **shall** be set to 0x00.

4.3.2.6.3 Effect on receipt

On receipt of this command, a receiving node is notified of a list of nodes in direct range for the sending node.

NWK:00AF.1 A receiving node **shall not** return a response if this command is received via multicast addressing.

4.3.2.7 Command Complete Command

This command is used to advertise the completion of a given task (i.e., range test, controller replication, …) by the sending node.

4.3.2.7.1 Frame format

NWK:00B0.1 The Command Complete Command **shall** be formatted as illustrated in Figure 4.27.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_CMD_COMPLETE (0x07)							
Sequence Number							

Figure 4.27 Command Complete Command format

4.3.2.7.1.1 Sequence Number (8 bits)

This field is used for duplicate detection for transmissions sequences that require multiple Command Complete Commands.

NWK:00B1.1 If this field is set to 0x00, it **shall** indicate that the Sequence Number is not in use and it s**hall** be ignored by the receiving node.

NWK:00B2.1 If this field is in the range 0x01..0xFF, it **shall** indicate a unique sequence number for the current operation.

4.3.2.7.2 When generated

NWK:00B3.1 When generated, the sending node **shall** be ready to receive more commands immediately after sending the command. If the sending node needs to be unavailable for a short time, this command **shall** be generated only when the node is available again.

NWK:00B4.1 This command **shall not** be sent via multicast addressing.

NWK:00B5.1 If used, the *Sequence Number* field value **shall** be incremented at each new transmission of the Command Complete Command.

4.3.2.7.3 Effect on receipt

On receipt of this command, a receiving node is notified of the sender's has completed a requested operation (optionally identified by the *Sequence Number* field).

NWK:00B6.1 A receiving node **shall** ignore this command if it is received via multicast addressing

4.3.2.8 Transfer Presentation Command

The Transfer Presentation Command is used by controller nodes to indicate that they are trying to initiate network management functions, such as:

- Add new nodes in the network
- Remove nodes from a network
- Perform a controller replication

4.3.2.8.1 Frame format

NWK:00B7.1 The Transfer Presentation Command **shall** be formatted as illustrated in Figure 4.28.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_TRANSFER_PRESENTATION (0x08)							
Option							

Figure 4.28 Transfer Presentation Command format

4.3.2.8.1.1 Option (8 bits)

The *Option* field is used to indicate the intent of the sending node. It **shall** be treated as a bitmask and **shall** be encoded according to Table 4.20. NWK:00B8.1

Table 4.20 Transfer Presentation Command - Option field encoding

Bit	Bitmask	Description
0	0x01	If set to 1, the sending controller supports Network Wide Inclusion (NWI). Refer to 4.5.4.2
1	0x02	If set to 1, the sending controller is trying to exclude a node
2	0x04	If set to 1, the sending controller is trying to include a node
3..7	*0x08..0xF0*	*These bits are reserved and **shall** be set to 0 by a sending node*

4.3.2.8.2 When generated

NWK:00B9.1 When generated, the sending node **shall** be ready to include or exclude a node from the network.

NWK:00BA.1 This command **shall** be sent to the broadcast destination (NodeID 0xFF).

NWK:00BB.1 The *Option* field bit 0 **shall** always be set to 1 by a sending node.

4.3.2.8.3 Effect on receipt

NWK:00BC.1 On receipt of this command, a receiving node in learn mode **shall** return a Node Information Frame Command to the sending node if the Option field matches the intent of the receiving node.

NWK:00BD.1 If the receiving node is not in learn mode or has already started an inclusion/exclusion procedure, it **shall not** return a Node Information Frame Command.

NWK:00BE.1 If the receiving node current learn mode does not match the include/exclude options in the command, it **shall not** return a Node Information Frame Command.

NWK:00BF.1 If the receiving node is in SmartStart Learn Mode, it **shall not** return a Node Information Frame Command.

4.3.2.9 Transfer Node Information Command

This command is used for transferring a node’s Node Information Frame Command data from one controller to another.

4.3.2.9.1 Frame format

NWK:00C0.1 The Transfer Node Information Command **shall** be formatted as illustrated in Figure 4.29.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_TRANSFER_NODE_INFO (0x09)							
Sequence Number							
NodeID							
Listening	Routing	Maximum Data Rate			Protocol version		
Optional functionality	Sensor 1000ms	Sensor 250ms	Beam capability	Routing End Node	Specific Device	Controller	Security
Reserved					Speed Extension		
Basic Device Type							
Generic Device Class							
Specific Device Class							

Figure 4.29 Transfer Node Information Command format

NWK:00C1.1 All fields not described below **shall** be identical to the fields in the Node Information Frame Command.

4.3.2.9.1.1 Sequence number (8 bits)

NWK:00C2.1 A sending node **shall** specify a unique sequence number starting from a random value. Each new message **shall** carry an increment of the value carried in the previous singlecast command.

NWK:00C3.1 A receiving node **shall** use this field for singlecast duplicate detection.

4.3.2.9.1.2 NodeID (8 bits)

This field is used to advertise the NodeID of the node to which belongs the node information data carried in this command.

4.3.2.9.2 When generated

NWK:00C4.1 When generated, the Node Information Frame Command fields **shall** be set to identical values as reported by the NodeID indicated in the *NodeID* field in its Node Information Frame Command.

NWK:00C5.1 This command **shall** be sent as part of the inclusion of a controller as specified in section 4.5.7.2 and 4.5.9.3.

4.3.2.9.3 Effect on receipt

NWK:00C6.1 This command **shall** be ignored if it is received using multicast addressing.

NWK:00C7.1 This command **shall** be ignored if the receiving node is an end node.

NWK:00C8.1 This command **shall** be ignored if the receiving node is a controller node and it is received outside a controller replication or inclusion process. Refer to sections 4.5.7.2 and 4.5.9.3 for details.

NWK:00C9.1 On receipt of this command, a controller in learn mode **shall** update its internal node table with the node information received in this command.

4.3.2.10 Transfer Range Information Command

This command is used to transfer a node's range test results from one controller to another.

4.3.2.10.1 Frame format

NWK:00CA.1 The Transfer Range Information Command **shall** be formatted as illustrated in Figure 4.30.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_TRANSFER_RANGE_INFO (0x0A)							
Sequence number							
NodeID							
Bitmask Length							
Node Bitmask 1							
…							
Node Bitmask N							

Figure 4.30 Transfer Range Information Command

NWK:00CB.1 All fields not described below **shall** be identical to the fields in the Range Info Command.

4.3.2.10.1.1 Sequence number (8 bits)

A sending node **shall** specify a unique sequence number starting from a random value. Each new message **shall** carry an increment of the value carried in the previous singlecast command. NWK:00CC.1

NWK:00CD.1 A receiving node **shall** use this field for singlecast duplicate detection.

4.3.2.10.1.2 NodeID (8 bits)

This field is used to advertise the NodeID of the node to which belongs the range test results carried in this command.

4.3.2.10.2 When generated

NWK:00CE.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using multicast addressing.

NWK:00CF.1 This command **shall** be sent as part of a controller replication or controller inclusion process. Refer to sections 4.5.7.2 and 4.5.9.3 for details.

4.3.2.10.3 Effect on receipt

NWK:00D0.1 This command **shall** be ignored if it is received using multicast addressing.

NWK:00D1.1 This command **shall** be ignored if the receiving node is an end node.

NWK:00D2.1 This command **shall** be ignored if the receiving node is a controller and it is received outside a controller replication process.

NWK:00D3.1 On receipt of this command, a controller in learn mode **shall** update its internal routing table with the range information received in this command.

4.3.2.11 Transfer End Command

This command is used to advertise the end of the current network operation, which can be:

- Static route request
- Automatic controller update
- Controller replication

4.3.2.11.1 Frame format

NWK:00D4.1 The Transfer End Command **shall** be formatted as illustrated in Figure 4.31.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_TRANSFER_END (0x0B)							
Status							

Figure 4.31 Transfer End Command format

4.3.2.11.1.1 Status (8 bits)

NWK:00D5.2 This field is used to notify the Z-Wave network information transfer status. This field value **shall** be encoded as shown in Table 4.21.

Table 4.21 Z-Wave Network Transfer Status

Values	Description
0x00	Transfer failed: An unspecified error occurred during the ongoing operation. The receiving node **may** try again immediately.
0x01	Transfer OK: The operation was carried out successfully.
0x02	Transfer update done: The transfer of data to the SUC was successful..
0x03	Transfer update aborted: The operation was aborted, for example due to another higher priority operation. The receiving node **shall** wait *nwkMinNetworkOperationBackOffTime* before trying the operation again.
0x04	Transfer update wait: The sending node is busy carrying another operation. The receiving node **shall** wait *nwkMinNetworkOperationBackOffTime* before trying the operation again.
0x05	Transfer update disabled: The requested operation is not available or not provided by the sending node. The receiving node **should not** initiate the same operation again.
0x06	Transfer update overflow: An overflow error occurred during the ongoing operation. The receiving node **may** try the operation again.

4.3.2.11.2 When generated

NWK:00D6.1 When generated, this command **shall** indicate that the current network management operation is completed and **shall** advertise its status.

NWK:00D7.1 This command **shall** not be sent outside network management operations.

4.3.2.11.3 Effect on receipt

On receipt of this command, the receiving node can conclude that the sending node has completed the requested actions and is ready to receive new commands or initiate new network management operations.

4.3.2.12 Assign Return Route Command

This command is used to advertise the assigned route to the receiving node.

4.3.2.12.1 Frame format

NWK:00D8.1 The Assign Return Route Command **shall** be formatted as illustrated in Figure 4.32.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_ZWAVE (0x01)</td></tr>
<tr><td colspan="8">Command = ZWAVE_CMD_ASSIGN_RETURN_ROUTE (0x0C)</td></tr>
<tr><td colspan="8">NodeID</td></tr>
<tr><td colspan="4">Route Number</td><td colspan="4">Number of Hops</td></tr>
<tr><td colspan="8">Repeater 0</td></tr>
<tr><td colspan="8">Repeater 1</td></tr>
<tr><td colspan="8">Repeater 2</td></tr>
<tr><td colspan="8">Repeater 3</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="3">Destination speed</td><td colspan="2">Dest. Wake up</td><td>Reserved</td></tr>
</table>

Figure 4.32 Assign Return Route Command format

4.3.2.12.1.1 Node ID (8 bits)

This field is used to advertise the NodeID of the destination for the assigned route.

The NodeID field may be different from the NodeID of the sending node.

4.3.2.12.1.2 Number of hops (4 bits)

This field is used to indicate the number of repeaters present in the frame.

NWK:00D9.1 The value 0 **shall** indicate that the route to the NodeID destination **shall** be a direct range transmission.

NWK:00DA.1 Values in the range 1..4 **shall** indicate the number of repeaters used in the route.

4.3.2.12.1.3 Route Number (3 bits)

NWK:00DB.1 This field is used to indicate a number of index to assign to the indicate route. The first assigned route **shall** have route number 0.

4.3.2.12.1.4 Repeater (4 bytes)

This field is used to indicate the list of repeaters to be used in the route.

4.3.2.12.1.5 Destination Wake Up (2 bits)

NWK:00DC.1 This field indicates the destination node Wake Up time capability. The field **shall** be encoded as shown in Table 4.22.

Table 4.22 Assign Return Route Command::Destination Wake Up encoding

Value	Description
0	No Wake Up (AL node)
1	Wake Up in 250 ms interval (FL Node)
2	Wake Up in 1000 ms interval (FL node)
3	*Reserved*

4.3.2.12.1.6 Destination Speed (3 bits)

NWK:00DD.1 This bit field specifies the data rate capability of the destination node. This field **shall** encodes as shown in Table 4.23.

Table 4.23 Assign Return Route::Destination Speed encoding

Value	Description
0	Reserved
1	9.6 kbits/s
2	40 kbits/s
3..7	*Reserved*

4.3.2.12.2 When generated

NWK:00DE.1 This command **shall not** be sent to controller nodes (i.e. nodes that set the *Controller* field to 1 in their Node Information Frame Command).

NWK:00DF.1 This command **shall not** be sent for the SUC NodeID unless the return route is set as part of an application level association. The Assign SUC Return Route Command **shall** be used instead if the return route is set by the protocol.

4.3.2.12.3 Effect on receipt

NWK:00E0.1 On receipt of this command, an end node that supports storing return routes **shall** store the specified route for use when transmitting frames to the destination specified in the frame.

4.3.2.13 New Node Registered Command

This command is used to notify the SIS controller that an inclusion controller has included a new node in the Z-Wave network.

4.3.2.13.1 Frame format

NWK:00E1.2 The New Node Registered Command **shall** be formatted as illustrated in Figure 4.33.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_NEW_NODE_REGISTERED (0x0D)							
NodeID							
Listening	Routing	Maximum speed			Protocol version		
Optional functionality	Sensor 1000ms	Sensor 250ms	Beam capability	Routing End Node	Specific Device	Controller	Security
Reserved					Speed Extension		
Basic Device Type							
Generic Device Class							
Specific Device Class							
Command Class 1							
…							
Command Class N							

Figure 4.33 New Node Registered Command format

NWK:00E2.1 All fields not described below **shall** be identical to the Node Information Frame Command fields described in section 4.3.2.1.

4.3.2.13.1.1 NodeID (8 bits)

This field is used to advertise the NodeID that has been assigned to the new node.

*4.3.2.13.1.2 Command Class 1-n (n*8 bits)*

This field is used to advertise the supported Command Classes in the included node. This field should contain the Command Classes received in the Node Information frame if available when sendiong this frame.

4.3.2.13.2 When generated

NWK:00E3.1 When generated, except for the *NodeID* field, the fields values in this command **shall** contain the values reported by the included node's Node Information Frame Command during its inclusion.

NWK:00E4.2 If generated for a node removed from the network, the *Generic Device Class* field **shall** be set to 0 in this command. All other fields after the NodeID **may** be set to zero.

NWK:00E5.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using multicast addressing.

4.3.2.13.3 Effect on receipt

On receipt of this command, a controller node is notified of a new node added in the network. This command **shall** be ignored by end nodes. NWK:00E6.1

NWK:00E7.1 This command **shall** be ignored if it was received using multicast addressing.

4.3.2.14 New Range Registered Command

This command is used to advertise the range test results that a controller has performed when including a new node in the Z-Wave network.

4.3.2.14.1 Frame format

NWK:00E8.1 The New Range Registered Command **shall** be formatted as illustrated in Figure 4.34.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_NEW_RANGE_REGISTERED (0x0E)							
NodeID							
Reserved			Neighbour Nodes Bitmask Length				
Neighbour Nodes Bitmask 1 (LSB)							
…							
Neighbour Nodes Bitmask N (MSB)							

Figure 4.34 New Range Registered Command format

4.3.2.14.1.1 NodeID (8 bits)

NWK:00E9.1 This field **shall** indicate the NodeID for which the neighbour nodes (nodes in range) are advertised in the *Neighbour Nodes Bitmask* field.

4.3.2.14.1.2 Neighbour Nodes Bitmask Length (5 bits)

NWK:00EA.1 This field **shall** advertise the length in bytes of the *Neighbour Nodes Bitmask* field. This field **should** be set to the minimum value allowing to advertise all neighbour nodes for the advertised NodeID.

4.3.2.14.1.3 Neighbour Nodes Bitmask (N bytes)

NWK:00EB.1 This field **shall** indicate the NodeIDs that are reported as direct range neighbours by the last range test for the actual NodeID.

NWK:00EC.1 The length of this field in bytes **shall** be according to the *Neighbour Nodes Bitmask Length* field value.

NWK:00ED.1 This field **shall** be treated as a bitmask, and the first bit in the LSB represents NodeID 1. This field **shall** be encoded as follow:

- Bit 0 in Node bitmask 1 **shall** represent the NodeID 1
- Bit 1 in Node bitmask 1 **shall** represent the NodeID 2
- ...

NWK:00EF.1 The bit value 0 **shall** be used to advertise that the corresponding NodeID is not a neighbour for the actual NodeID (advertised in the *NodeID* field).
The bit value 1 **shall** be used to advertise that the corresponding NodeID is a neighbour for the actual NodeID (advertised in the *NodeID* field).

4.3.2.14.2 When generated

NWK:00F0.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using multicast addressing.

NWK:00F1.1 This command **shall** only be sent as part of the following procedures:

- 4.5.8.1 Add new nodes on behalf of the SIS
- 4.5.9.1 Automatic Controller Update

4.3.2.14.3 Effect on receipt

On receipt of this command, the receiving node is notified of the new direct range topology around the NodeID advertised in the command.

NWK:00F2.1 End nodes **shall** ignore this command.

NWK:00F3.1 This command **shall** be ignored if it was received using multicast addressing.

4.3.2.15 Transfer New Primary Controller Complete Command

This command is used to complete the Primary Controller role transfer procedure to another controller node.

4.3.2.15.1 Frame format

NWK:00F4.1 The Transfer New Primary Controller Complete Command **shall** be formatted as illustrated in Figure 4.35.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_TRANSFER_NEW_PRIMARY_COMPLETE (0x0F)							
Controller Type							

Figure 4.35 Transfer New Primary Controller Complete Command format

4.3.2.15.1.1 Controller Type (8 bits)

This field is used to indicate the type of the controller.

NWK:00F5.1 A sending node **shall** set this field to the same value as the *Generic Device Clas*s field in its Node Information Frame Command.

4.3.2.15.2 When generated

The Transfer New Primary Controller Complete Command is generated as part of the Primary Controller Shift procedure. Refer to 4.5.7.2 Primary Controller shift for details.

4.3.2.15.3 Effect on receipt

On receipt of this command, the receiving node is notified it has now the Primary Controller role in the network.

NWK:00EE.1 If the receiving node is in learn mode, it **shall** assume the Primary Controller role and **may** use Primary Controller functionalities. Refer to 4.5.7 Controller roles for details.

NWK:00F6.1 If the receiving node is not in learn mode, it **shall** ignore this command.

4.3.2.16 Automatic Controller Update Start Command

This command is used to request transfer of the network topology information from the SUC/SIS controller to another controller in a network.

4.3.2.16.1 Frame format

NWK:00F7.1 The Automatic Controller Update Start Command **shall** be formatted as illustrated in Figure 4.36.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_AUTOMATIC_CONTROLLER_UPDATE_START (0x10)							

Figure 4.36 Automatic Controller Update Start Command format

4.3.2.16.2 When generated

NWK:00F8.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using multicast addressing. This command **should** only be sent to the SUC/SIS controller.

NWK:00F9.1 This command **shall** be sent from an Inclusion Controller in a network with a SUC/SIS controller before starting to add a new node to the network.

Refer to 4.5.8.1 Add new nodes on behalf of the SIS.

NWK:00FA.1 This command **may** be sent periodically by an inclusion controller to ensure that its network topology is updated. It **should not** be more than once a day.

4.3.2.16.3 Effect on receipt

NWK:00FB.1 If the receiving node is a SUC/SIS controller, it **shall** send topology updates to the sending node using New Node Registered Commands and New Range Registered Commands. Refer to section 4.5.9.1 Automatic Controller Update for details.

NWK:00FC.1 If the receiving node is not a SUC/SIS controller, it **should** return a SUC Node ID command to the sending node to inform the correct Node ID of the SUC/SIS.

NWK:00FD.1 This command **shall** be ignored if it was received using multicast addressing.

4.3.2.17 SUC Node ID Command

This command is used to advertise the Node ID of the SUC and its capabilities.

4.3.2.17.1 Frame format

NWK:00FE.1 The SUC Node ID Command **shall** be formatted as illustrated in Figure 4.37.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_SUC_NODE_ID (0x11)							
NodeID							
SUC Capabilities (optional Rx)							

Figure 4.37 SUC Node ID Command format

4.3.2.17.1.1 NodeID (8 bits)

This field is used to advertise the NodeID of the SUC Controller.

NWK:00FF.1 The value 0x00 **shall** indicate that there is no SUC in the network.
Values in the range 1..232 **shall** indicate the NodeID of the SUC Controller for this network.

4.3.2.17.1.2 SUC Capabilities (optional Rx) (8 bits)

NWK:0100.1 This field is used to indicate the capabilities of the SUC node. It **shall** be treated as a bitmask and **shall** be encoded according to Table 4.24.

Table 4.24 SUC Node ID Command - SUC Capabilities encoding

Bit	Description
0	If this bit is set to 1, it **shall** indicate that the Node ID server is running. (SIS)
1..7	*Reserved*

NWK:0101.1 Older implementations **may** omit this field from the command. If this field is not present in the frame, the value 0x00 **shall** be assumed.

4.3.2.17.2 When generated

NWK:0102.1 A sending node **shall not** send this command to the NodeID indicated in the *NodeID* field.

4.3.2.17.3 Effect on receipt

On receipt of this command, the receiving node is notified of the SUC's identity and its capabilities (SUC only or with SIS functionality).

4.3.2.18 Set SUC Command

This command is used by a Primary Controller to grant the SUC Role to another controller for the current network.

4.3.2.18.1 Frame format

NWK:0103.1 The Set SUC Command **shall** be formatted as illustrated in Figure 4.38.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_SET_SUC(0x12)							
State							
SUC Capabilities							

Figure 4.38 Set SUC Command format

4.3.2.18.1.1 State (8 bits)

NWK:0104.1 This field is used to advertise the state of the controller functionality. The field value **shall** be set to 0x01 to enable the SUC functionality.

NWK:0105.1 All other values are reserved and **shall** by ignored by receiving nodes.

4.3.2.18.1.2 SUC Capabilities (8 bits)

NWK:0106.1 This field is used to specify which service the SUC should be running. It **shall** be treated as a bitmask and encoded according to Table 4.24.

4.3.2.18.2 When generated

NWK:0107.1 The *SUC Capabilities* field **shall** be set to 0x01.

NWK:0108.1 This command **shall** only be sent by a node that has the Primary Controller role. This command **shall not** be transmitted to end nodes and **shall** only be sent to controllers.

4.3.2.18.3 Effect on receipt

NWK:0109.1 On receipt of this command, the receiving node **shall** assume the SUC role if it has the capabilities.

NWK:010A.1 A receiving node **shall** return a Set SUC ACK Command.

4.3.2.19 Set SUC ACK Command

This command is used to respond to a Set SUC Command .

4.3.2.19.1 Frame format

NWK:010B.1 The Set SUC ACK Command **shall** be formatted as illustrated in Figure 4.39.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_SET_SUC_ACK(0x13)							
Result							
SUC Capabilities							

Figure 4.39 Set SUC ACK Command format

4.3.2.19.1.1 Result (8 bits)

This field is used to advertise a status of taking the SUC role for the network.

NWK:010C.1 If the sending node has accepted to take the SUC role, this field **shall** be set to 0x01.
If the sending node has not accepted or cannot take the SUC role, this field **shall** be set to 0x00.

4.3.2.19.1.2 SUC Capabilities (8 bits)

NWK:010D.1 This field is used to indicate the capabilities of the SUC node. It **shall** be treated as a bitmask and **shall** be encoded according to Table 4.24.

4.3.2.19.2 When generated

The SUC Node ID Command is generated in response to the Set SUC Command .

4.3.2.19.3 Effect on receipt

On receipt of this command, the node that initiated a SUC role transfer is notified of the operation’s result together with the capabilities of the SUC node.

4.3.2.20 Assign SUC Return Route Command

This command is used to notify the receiving node about the route on how to reach the SUC.

4.3.2.20.1 Frame format

NWK:010E.1 The Assign SUC Return Route Command **shall** be formatted as illustrated in Figure 4.40.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_ZWAVE (0x01)</td></tr>
<tr><td colspan="8">Command = ZWAVE_CMD_ASSIGN_SUC_RETURN_ROUTE (0x14)</td></tr>
<tr><td colspan="8">NodeID</td></tr>
<tr><td colspan="4">Route Number</td><td colspan="4">Number of Hops</td></tr>
<tr><td colspan="8">Repeater 1</td></tr>
<tr><td colspan="8">Repeater 2</td></tr>
<tr><td colspan="8">Repeater 3</td></tr>
<tr><td colspan="8">Repeater 4</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="3">Destination speed</td><td colspan="2">Dest. wakeup</td><td>Reserved</td></tr>
</table>

Figure 4.40 Assign SUC Return Route Command format

The fields from this command are identical to the fields in the Assign Return Route Command.

4.3.2.20.2 When generated

NWK:010F.1 The *NodeID* field **shall** be set to the SUC NodeID.

NWK:0110.1 This command **shall** be sent by a controller when it includes an end node and a SUC/SIS is present in the network. Refer to 4.5.4.1 and 4.5.4.2 for details.

4.3.2.20.3 Effect on receipt

NWK:0111.1 On receipt of this command, the node **should** update its route for the SUC NodeID and learns which NodeID is the SUC in the network.

4.3.2.21 Static Route Request Command

This command is used to request static routes for one or more NodeID destinations.

4.3.2.21.1 Frame format

NWK:0112.1 The Static Route Request Command **shall** be formatted as illustrated in Figure 4.41.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_STATIC_ROUTE_REQUEST(0x15)							
Destination NodeID 1							
Destination NodeID 2							
Destination NodeID 3							
Destination NodeID 4							
Destination NodeID 5							

Figure 4.41 Static Route Request Command format

4.3.2.21.1.1 Destination NodeID (5 bytes)

This field is used to indicate a list of destination for which return routes are needed.

NWK:0113.1 For each byte, a value in the range 1..232 **shall** indicate a NodeID for which a return route is requested.

NWK:0114.1 For each byte, the value 0x00 **shall** indicate that the byte is unused.

4.3.2.21.2 When generated

NWK:0115.1 This command **should not** be sent. Refer to 4.5.1.1 Assigning return routes.

NWK:0116.1 If used, this command **shall** be sent to the SUC/SIS.

4.3.2.21.3 Effect on receipt

NWK:0117.1 On receipt of this command, a controller node **shall** respond with Assign Return Route Commands to the sending node. Routes **shall** be assigned for all the *Destination NodeID*s in the command. For details, refer to 4.5.9.5 End node route request.

NWK:0118.1 An end node **shall** ignore this command.

4.3.2.22 Lost Command

This command is used to request help from another node in the network to locate the SUC/SIS in the network

4.3.2.22.1 Frame format

NWK:0119.1 The Lost Command **shall** be formatted as illustrated in Figure 4.42.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_LOST (0x16)							
NodeID							

Figure 4.42 Lost Command format

4.3.2.22.1.1 NodeID (8 bits)

The node ID of the node that originally requested help.

4.3.2.22.2 When generated

NWK:011A.1 This command is obsoleted and **shall** not be transmitted.

4.3.2.22.3 Effect on receipt

NWK:011B.1 On receipt of this command a Non SUC/SIS node **shall** forward this command to the SUC/SIS if it knows the SUC/SIS NodeID. This command **shall** be ignored in a network without SUC/SIS.

NWK:011C.1 On receipt of this command, a SUC/SIS controller **should** send an Accept Lost Command, perform a Neighbour Discovery (refer to 4.5.9.4 Neighbour Discovery / Range test), issue an Assign SUC Return Route Command and Transfer End Command to the lost end node. This process is illustrated in Figure 4.43.

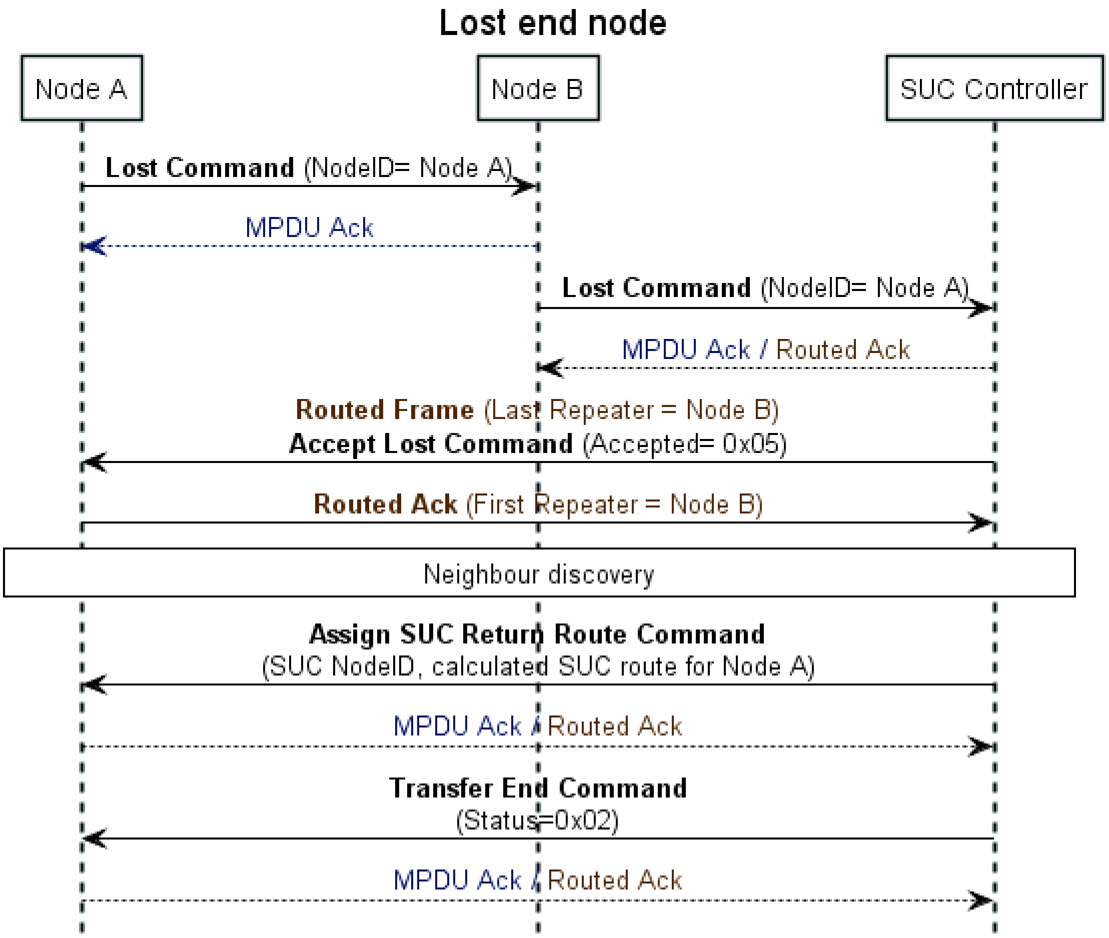

Lost end node
Node A
Node B
SUC Controller
Lost Command (NodeID= Node A)
MPDU Ack
Lost Command (NodeID= Node A)
MPDU Ack / Routed Ack
Routed Frame (Last Repeater = Node B)
Accept Lost Command (Accepted= 0x05)
Routed Ack (First Repeater = Node B)
Neighbour discovery
Assign SUC Return Route Command
(SUC NodeID, calculated SUC route for Node A)
MPDU Ack / Routed Ack
Transfer End Command
(Status=0x02)
MPDU Ack / Routed Ack

Figure 4.43 Lost end node recovery process

4.3.2.23 Accept Lost Command

The Accept Lost Command is used to indicate that the SUC has accepted to help a lost node.

4.3.2.23.1 Frame format

NWK:011D.1 The Accept Lost Command **shall** be formatted as illustrated in Figure 4.44.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_ACCEPT_LOST (0x17)							
Accepted							

Figure 4.44 Accept Lost Command format

4.3.2.23.1.1 Accepted (8 bits)

This field is used to indicate if the SUC node has accepted to help the lost end node.

NWK:011E.1 The value 0x04 **shall** indicate that the sending node has refused to help the lost end node
The value 0x05 **shall** indicate that the sending node has accepted to help the lost end node.

NWK:011F.1 All other values are reserved. Reserved values **shall not** be used by a sending node and **shall** be ignored by a receiving node.

4.3.2.23.2 When generated

NWK:0120.1 When generated, a SUC Controller **shall** perform a Neighbour Discovery (refer to 4.5.9.4 Neighbour Discovery / Range test), issue an Assign SUC Return Route Command and Transfer End Command to the lost end node. This process is illustrated in Figure 4.43.

NWK:0121.1 This command **shall not** be sent by a node that does not have the SUC role in the network.

4.3.2.23.3 Effect on receipt

On receipt of this command, lost end nodes are informed that the lost recovery process has started.

This process is illustrated in Figure 4.43.

4.3.2.24 NOP Power Command

This command is used to verify if a node is in direct range and used for Neighbour Discovery.

4.3.2.24.1 Frame format

NWK:0122.1 The NOP Power Command **shall** be formatted as illustrated in Figure 4.45.

7	6	5		4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)								
Command = ZWAVE_CMD_NOP_POWER (0x18)								
Tx Power register (Obsolete)								
Tx Power Dampening								

Figure 4.45 NOP Power Command format

4.3.2.24.1.1 Tx Power Register (Obsolete) (8 bits)

NWK:0123.1 This field **shall not** be used when sending this command.

NWK:0124.1 If no *Tx Power Dampening* field is present in the command, this field **should** be converted to dampening according to Table 4.25.

Table 4.25 NOP Power - Tx Power Register encoding

Value	Power dampening
0xf0	No dampening of Tx power
0xc8	-1 dBm
0xa7	-2 dBm
0x91	-3 dBm
0x77	-4 dBm
0x67	-5 dBm
0x60	-6 dBm
0x46	-7 dBm
0x38	-8 dBm
0x35	-9 dBm
0x32	-10 dBm
0x30	-11 dBm
0x24	-12 dBm
0x22	-13 dBm

0x20	-14 dBm
All other values	No dampening of Tx power

4.3.2.24.1.2 Tx Power Dampening (8 bits)

NWK:0125.1 This field is used to indicate which transmit power to use for sending a MPDU Ack to this frame. This field **shall** be encoded according to Table 4.26

Table 4.26 NOP Power Tx Power Dampening encoding

Value (decimal)	Power dampening
0	No dampening of Tx power
1..14	-1 dBm..-14 dBm Tx dampening **shall** be applied by the receiving node for transmitting the MPDU Ack to this command.
14..255	*Reserved*

4.3.2.24.2 When generated

NWK:0126.1 When generated, the *Tx Power Register* field **shall** be set to 0x00.

NWK:0127.1 The NOP Power Command **shall** be sent in direct range only, it **shall not** be sent in a Routed NPDU or an Explore NPDU.

NWK:0128.1 A sending node **shall** apply the Tx Power Dampening indicated by the *Tx Power Dampening* field for sending this command.

4.3.2.24.3 Effect on receipt

NWK:0129.1 On receipt of this command, the MPDU Ack frame **should** be transmitted with a Tx power that is lowered according to the parameters in the frame. The dampening should be done from the default Tx power configured in the receiving node.

4.3.2.25 Reserve Node IDs Command

This command is used by an inclusion controller to request NodeID(s) to the SIS that can be used to add new nodes in the network.

4.3.2.25.1 Frame format

NWK:012A.1 The Reserved Node IDs Command **shall** be formatted as illustrated in Figure 4.46.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_RESERVE_NODE_IDS(0x19)							
Number of Node IDs							

Figure 4.46 Reserve Node IDs Command format

4.3.2.25.1.1 Number of Node IDs (bytes)

This field is used to indicate the number of requested NodeIDs.

NWK:012B.1 This field **shall** be in the range 1..10. This field **should** be set to 1 to avoid exhausting the NodeID pool from the SIS.

4.3.2.25.2 When generated

NWK:012C.1 The Reserved Node IDs Command **shall** be sent to the SUC Node ID. It **shall** only be sent if the SUC has Server ID Capabilities (SIS).

NWK:012D.1 This command **shall** only be sent by Inclusion Controller nodes.

NWK:012E.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using multicast addressing.

4.3.2.25.3 Effect on receipt

NWK:012F.1 On receipt of this command, the SIS controller **shall** return a Reserved IDs Command in response. A receiving node **shall not** return a response if this command is received via multicast addressing.

NWK:0130.1 All nodes that are not SIS controller in the network **shall** ignore this command.

4.3.2.26 Reserved IDs Command

This command is used to advertise the granted NodeIDs to an Inclusion Controller.

4.3.2.26.1 Frame format

NWK:0131.1 The Reserved IDs Command **shall** be formatted as illustrated in Figure 4.47.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_RESERVED_IDS (0x1A)							
Number of IDs							
Reserved NodeID 1							
…							
Reserved NodeID N							

Figure 4.47 Reserved IDs Command format

4.3.2.26.1.1 Number of IDs (8 bits)

This field is used to advertise the number of granted NodeIDs.

NWK:0132.1 This field **shall** be in the range 0..10.

4.3.2.26.1.2 Reserved NodeID (N bytes)

This field is used to advertise a list of granted NodeIDs.

NWK:0133.1 Each byte **shall** represent a granted NodeID that the Inclusion Controller can use for Network Inclusion.

NWK:0134.1 The length of this field **shall** be according to the value of the *Number of IDs* field.

4.3.2.26.2 When generated

NWK:0135.1 The SIS node **should** provide as many reserved IDs as requested by the Inclusion Controller node in the Reserved IDs Command .

NWK:0136.1 A SIS node **may** refuse to provide reserved NodeIDs.

NWK:0137.1 A node without the SIS role in the network **shall not** send this command.

4.3.2.26.3 Effect on receipt

On receipt of this command, an inclusion controller is notified of the NodeIDs that have been reserved for it. These NodeID values can be used to include new nodes.

4.3.2.27 Nodes Exist Command

The Nodes Exist Command is used to advertise a list of nodes currently present in the network.

4.3.2.27.1 Frame format

NWK:0138.1 The Node Exist Command **shall** be formatted as illustrated in Figure 4.48.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_ZWAVE (0x01)</td></tr>
<tr><td colspan="8">Command = ZWAVE_CMD_NODES_EXIST (0x1F)</td></tr>
<tr><td colspan="8">Node Mask Type</td></tr>
<tr><td colspan="8">Node Mask Length</td></tr>
<tr><td colspan="8">Node Mask 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Node Mask N</td></tr>
</table>

Figure 4.48 Node Exist Command format

4.3.2.27.1.1 Node Mask Type (8 bits)

This field is used to advertise the type of nodes that the Node Mask advertises in this command.

NWK:0139.1 The value 0 **shall** indicate that all existing nodes present in the network are advertised in the *Node Mask* field.

All other values are reserved.

4.3.2.27.1.2 Node Mask Length (8 bits)

This field is used to indicate the length of the *Node Mask* field in bytes.

4.3.2.27.1.3 Node Mask (N bytes)

This field is used to advertise the list of nodes present in the network for the type advertised by the *Node Mask Type* field.

NWK:013A.1 The length of this field in bytes **shall** be according to the *Node Mask Length* field value.

4.3.2.27.2 When generated

NWK:013B.1 When generated, this command **shall** contain the list of nodes matching the advertised type. Nodes without the SIS controller role **shall not** send this command.

This command is generated as part of the Automatic Controller Update process. Refer to 4.5.9.1 Automatic Controller Update.

4.3.2.27.3 Effect on receipt

NWK:013C.1 On receipt of this command, a receiving controller node **shall** return a Nodes Exist Reply Command with the *Node Mask Type* field set to the value received in this command.

NWK:013D.1 If this command is received as part of an Automatic Controller Update process, a receiving Inclusion Controller **shall** remove non-existing nodes from its network topology.

NWK:013E.1 If this command is received outside an Automatic Controller Update process, a receiving node **should not** remove non/existing nodes from its network topology.

4.3.2.28 Nodes Exist Reply Command

The Nodes Exist Command is used to advertise if a Node Exist Command has been understood.

4.3.2.28.1 Frame format

NWK:0162.1 The Node Exist Reply Command **shall** be formatted as illustrated in Figure 4.49.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_NODES_EXIST_REPLY(0x20)							
Node Mask Type							
Status							

Figure 4.49 Node Exist Reply Command format

4.3.2.28.1.1 Node Mask Type (8 bits)

This field is used to advertise the type of nodes that the Node Mask advertises in this command.

NWK:013F.1 This field **shall** be set to the same value as the Nodes Exist Command Reserved field that triggered this command to be sent.

4.3.2.28.1.2 Status (8 bits)

This field is used to advertise the status of the Nodes Exist Command execution.

NWK:0140.1 The value 0 **shall** indicate that the *Node Mask Type* field value is not known by the node or the Node Mask data has not been used to update the node list of the current network.

NWK:0141.1 The value 1 **shall** indicate that the Node Mask Type field value is known by the node and the Node Mask data has been used to update the node list of the current network.

All other values are reserved.

4.3.2.28.2 When generated

NWK:0142.1 This command is generated in response of the Nodes Exist Command. A sending node **shall** advertise if its network node list has been updated with the *Status* Field.

4.3.2.28.3 Effect on receipt

On receipt of this command, a controller node is notified if the Nodes Exist Command has been understood by the sending node.

4.3.2.29 Set NWI Mode Command

The Set NWI Mode Command is used to instruct AL nodes in the network to repeat Inclusion Request Explore Frames and Routed frames sent on foreign HomeIDs.

4.3.2.29.1 Frame format

NWK:0143.1 The Set NWI Mode Command **shall** be formatted as illustrated in Figure 4.50

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_SET_NWI_MODE (0x22)							
Mode							
Timeout							

Figure 4.50 Set NWI Mode Command format

4.3.2.29.1.1 Mode (8 bits)

The *Mode* field is used to indicate if the NWI mode must be enabled or disabled.

NWK:0144.1 The value 0x00 **shall** indicate that the receiving node **shall** disable NWI mode and **shall not** repeat Inclusion Request Explore Frames.

NWK:0145.1 The value 0x01 **shall** indicate that the receiving node **shall** enable NWI mode and **shall** repeat Inclusion Request Explore Frames.

4.3.2.29.1.2 Timeout (8 bits)

The *Timeout* field is used to indicate how long the NWI mode must be enabled.

NWK:0146.1 The *Timeout* field **shall** be ignored if the *Mode* field is set to 0x00.

NWK:0147.1 The value 0x00 **shall** indicate the node to use a default timeout *nwkNWIModeDefaultTimeout*.

NWK:0148.1 Values in the range 0x01..0xFF **shall** indicate the number of minutes that the receiving node shall wait before disabling NWI mode.

4.3.2.29.2 When generated

NWK:0149.1 This command **shall** be transmitted using a Normal Explore Frame. Refer to 4.2.3.2.

4.3.2.29.3 Effect on receipt

NWK:014A.1 On receipt of this command, AL nodes **shall** activate or deactivate NWI mode according to the fields present in this command.

For details about NWI mode, refer to 4.5.1.3 General routing requirements

4.3.2.30 Exclude Request Command

The Exclude Request Command is used by a node looking to be excluded from its current network.

4.3.2.30.1 Frame Format

NWK:014B.1 The Exclude Request Command **shall** be formatted as illustrated in Figure 4.51

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_EXCLUDE_REQUEST (0x23)							
Listening	Routing	Maximum speed			Protocol version		
Optional functionality	Sensor 1000ms	Sensor 250ms	Beam capability	Routing End Node	Specific Device	Controller	Security
Reserved					Speed Extension		
Basic Device Type							
Generic Device Class							
Specific Device Class							
Command Class 1							
…							
Command Class N							

Figure 4.51 Exclude Request Command format

For fields description, refer to 4.3.2.1 Node Information Frame Command.

4.3.2.30.2 When generated

NWK:014C.1 This command **shall** be transmitted using a Normal Explore Frame. Refer to 4.2.3.2.

NWK:014D.1 This command **shall** be sent to the broadcast destination (NodeID 0xFF).

NWK:014E.1 The fields values set in the Exclude Request Command **shall** be identical to the fields set by the sending node in its Node Information Frame Command.

The Exclude Request Command shall be sent only if send sending node is in Learn Mode Exclusion and supports NWE. Refer to 4.5.5.3 Network Wide Exclusion (NWE) for details.

4.3.2.30.3 Effect on receipt

NWK:014F.1 On receipt of this command, a controller node currently trying to remove a node **shall** return a Assign IDs Command with the *HomeID* and *NodeID* fields set to 0 to the sending node.

NWK:0150.1 Controllers nodes not trying to exclude a node **shall** ignore this command.

Refer to 4.5.5.3 Network Wide Exclusion (NWE) for details.

4.3.2.31 Assign Return Route Priority Command

The Assign Return Route Priority Command is used to assign the priority route to a NodeID destination.

4.3.2.31.1 Frame format

NWK:0151.1 The Assign Return Route Priority Command **shall** be formatted as illustrated in Figure 4.52

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_ASSIGN_RETURN_ROUTE_PRIORITY(0x24)							
NodeID							
Route Number							

Figure 4.52 Assign Return Route Priority Command format

4.3.2.31.1.1 NodeID (8 bits)

This field is used to indicate the destination NodeID for the priority route.

4.3.2.31.1.2 Route Number (8 bits)

NWK:0152.1 This field is used to indicate the Route Number that **shall** be used as priority route for the NodeID.

Route Numbers are set with the Assign Return Route Command.

4.3.2.31.2 When generated

NWK:0153.1 A node sending this command **shall** have issued an Assign Return Route Command with the Route Number prior to sending this command.

NWK:0154.1 End nodes **shall not** send this command.

4.3.2.31.3 Effect on receipt

NWK:0155.1 On receipt of this command, an end node **shall** update its priority route to the route number indicated in this command.

NWK:0156.1 This command **shall** be ignored if the Route Number does not match any defined route.

NWK:0157.1 Controller nodes **shall** ignore this command.

NWK:0158.1 End nodes receiving this command with a valid route number **shall** always use that route as the first routing attempt when starting a transmission. For details about priority routes, refer to 4.5.1.2 Priority route.

4.3.2.32 Assign SUC Return Route Priority Command

The Assign SUC Return Route Priority Command is used to assign the priority route to the SUC.

4.3.2.32.1 Frame format

NWK:0161.1 The Assign SUC Return Route Priority Command **shall** be formatted as illustrated in Figure 4.53

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_ASSIGN_SUC_RETURN_ROUTE_PRIORITY (0x25)							
NodeID							
Route Number							

Figure 4.53 Assign SUC Return Route Priority Command format

The fields from this command are identical to the fields in the Assign Return Route Priority Command.

4.3.2.32.2 When generated

NWK:0159.1 A node sending this command **shall** have issued an Assign SUC Return Route Command with the Route Number prior to sending this command.

NWK:015A.1 The *NodeID* field of this command **shall** be set to the NodeID of the SUC.

NWK:015B.1 End nodes **shall not** send this command.

4.3.2.32.3 Effect on receipt

NWK:015C.1 On receipt of this command, an end node **shall** update its priority route to the route number indicated in this command for the SUC.

NWK:015D.1 This command **shall** be ignored if the Route Number does not match any defined route for the SUC.

NWK:015E.1 Controller nodes **shall** ignore this command.

NWK:015F.1 End nodes receiving this command with a valid route number **shall** always use that route as the first routing attempt when starting a transmission to the SUC/SIS. For details about priority routes, refer to 4.5.1.2 Priority route

4.3.2.33 SmartStart Included Node Information Command

The SmartStart Included Node Info Frame command is used by nodes to notify a controller that it was just powered up and is already part of a network.

4.3.2.33.1 Frame format

NWK:0160.1 The SmartStart Included Node Information Frame Command **shall** be formatted as illustrated in Figure 4.54.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_INCLUDED_NODE_INFO (0x26)							
NWI HomeID 1							
NWI HomeID 2							
NWI HomeID 3							
NWI HomeID 4							

Figure 4.54 SmartStart Included Node Information Frame Command format

4.3.2.33.1.1 NWI HomeID (4 bytes)

NWK:0163.1 The *NWI HomeID* field is used to uniquely identify the sending node. This field **shall** be constructed from the S2 DSK of the node as follow:

- NWK:0164.1 NWI HomeID 1..4 **shall** match byte 9..12 of the S2 DSK.
  Additionally:
- NWK:0165.1 Bits 7 and 6 of the NWI HomeID 1 **shall** be set to 1.
- NWK:0166.1 Bit 0 of the NWI HomeID 4 byte **shall** be set to 0.

This is illustrated in Figure 4.55

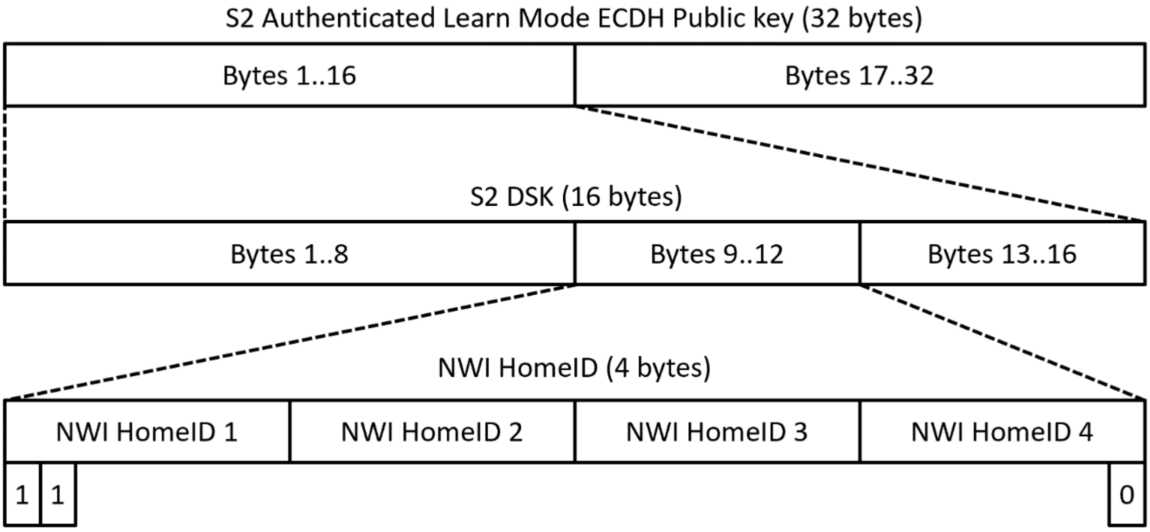


Figure 4.55 SmartStart NWI HomeID construction

4.3.2.33.2 When generated

NWK:0167.1 When generated, this command **shall** be transmitted using a Inclusion Request Explore Frame. Refer to 4.2.3.3.

NWK:0168.1 This command **shall** be sent to the broadcast destination (NodeID 0xFF).

4.3.2.33.3 Effect on receipt

On receipt of this command, a controller node trying to perform a SmartStart inclusion of a node whose S2 DSK matches the NWI HomeID field of this command **should** indicate to the application layer that the node to be included is currently included in another network and needs to be removed from the foreign network before it can be included.

4.3.2.34 SmartStart Prime Command

The Smart Prime command is used to notify SmartStart including controllers that a node is about to make an inclusion request.

4.3.2.34.1 Frame format

NWK:0169.1 The SmartStart Prime Command **shall** be formatted as illustrated in Figure 4.56

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_SMARTSTART_PRIME (0x27)							
Listening	Routing	Maximum speed			Protocol version		
Optional functionality	Sensor 1000ms	Sensor 250ms	Beam capability	Routing End Node	Specific Device	Controller	Security
Reserved					Speed Extension		
Basic Device Type							
Generic Device Class							
Specific Device Class							
Command Class 1							
…							
Command Class N							

Figure 4.56 SmartStart Prime Command format

NWK:016A.1 All fields configuration **shall** be identical to the Node Information Frame Command described in section 4.3.2.1.

4.3.2.34.2 When generated

NWK:016B.1 When generated, this command **shall** be transmitted using a Inclusion Request Explore Frame. Refer to 4.2.3.3.

NWK:016C.1 This command **shall** be sent to the broadcast destination (NodeID 0xFF).
This command **shall** be sent on the *NWI HomeID* HomeID and **shall not** be sent on the currently assigned HomeID. Refer to 4.3.2.33.1.1 NWI HomeID (4 bytes) for details.

NWK:016D.1 The sending node **shall** subsequently send a SmartStart Inclusion Request Command after *nwkSmartStartInclusionRequestDuration* seconds.

NWK:016E.1 Nodes not operating in AL mode **may** return to sleep between issuing the SmartStart Prime Command and the SmartStart Inclusion Request Command.

4.3.2.34.3 Effect on receipt

On receipt of this command, a controller node that intends to include any node using SmartStart shall verify if the Network HomeID of the Inclusion Request Explore Frame. header matches the NWI HomeID of any of the DSKs present in its SmartStart list.

NWK:016F.1 If it finds a match, the controller node **shall** enter SmartStart Inclusion and attempt to include the node when it issues a SmartStart Inclusion Request Command.

NWK:0170.1 In the unlikely event of several DSK matches for the received NWI HomeID, the controller node **shall** enter SmartStart Inclusion alternating between the possible DSK candidates.

4.3.2.35 SmartStart Inclusion Request Command

The SmartStart Inclusion Request Command is used to request to initiate a SmartStart inclusion.

4.3.2.35.1 Frame format

NWK:0171.1 The SmartStart Inclusion Request Command **shall** be formatted as illustrated in Figure 4.57

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE (0x01)							
Command = ZWAVE_CMD_SMARTSTART_INCLUDE (0x28)							
Listening	Routing	Maximum speed			Protocol version		
Optional functionality	Sensor 1000ms	Sensor 250ms	Beam capability	Routing End Node	Specific Device	Controller	Security
Reserved					Speed Extension		
Basic Device Type							
Generic Device Class							
Specific Device Class							
Command Class 1							
…							
Command Class N							

Figure 4.57 SmartStart Inclusion Request Command format

NWK:0172.1 All fields configuration **shall** be identical to the Node Information Frame Command described in section 4.3.2.1.

4.3.2.35.2 When generated

NWK:0173.1 When generated, this command **shall** be transmitted using a Inclusion Request Explore Frame. Refer to 4.2.3.3.

NWK:0174.1 This command **shall** be sent to the broadcast destination (NodeID 0xFF). This command **shall** be sent on the *NWI HomeID* HomeID and **shall not** be sent on the currently assigned HomeID. Refer to 4.3.2.33.1.1 NWI HomeID (4 bytes) for details.

NWK:0175.1 The sending node **shall** listen and accept Assign IDs Command using the Auth HomeID. More details are provided in Figure 4.58 and 4.5.4.3 SmartStart Inclusion.

4.3.2.35.3 Effect on receipt

On receipt of this command, a controller node that intends to include the sending node **shall** return an Assign IDs Command using the SmartStart Auth HomeID.

NWK:0176.1 The Auth HomeID is used to confirm the sending node that the sending controller possesses its S2 DSK. This field **shall** be constructed from the S2 DSK of the node as follow:

- NWI HomeID 1..4 **shall** match byte 13..16 of the S2 DSK.
- Bits 7 and 6 of the Auth HomeID 1 **shall** be set to 1.
- Bit 0 of the Auth HomeID 4 byte **shall** be set to 0.

The Auth HomeID derivation is illustrated in Figure 4.58. Refer to 4.5.4.3 SmartStart Inclusion for more details.

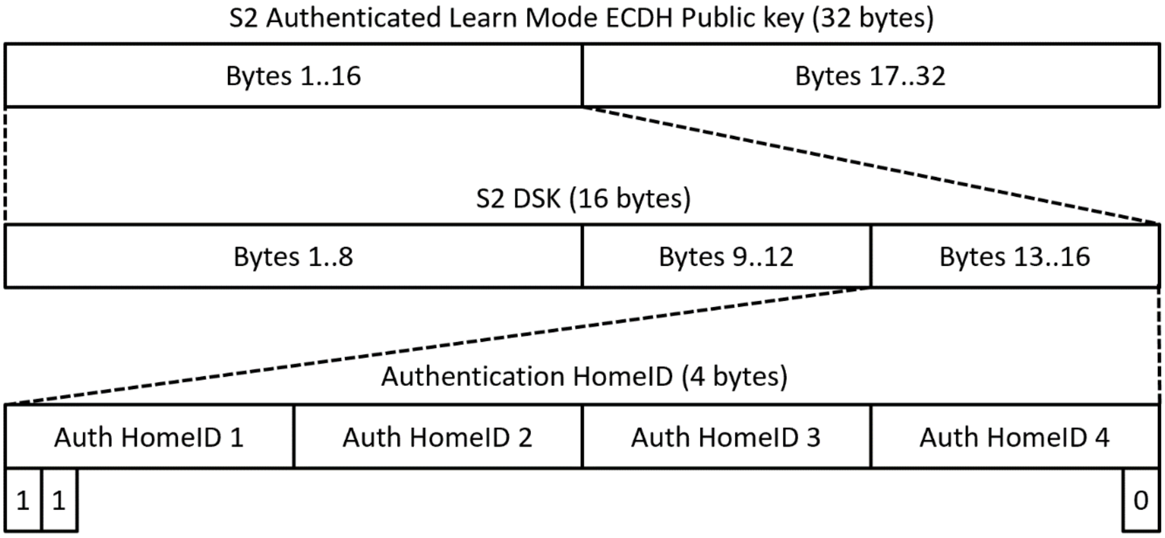


Figure 4.58 SmartStart Authentication HomeID construction

4.4 Constants

NWK:0177.1 The constants that define the Z-Wave NWK layer are presented in Table 4.27. Implementations **shall** comply with these values.

Table 4.27 NWK layer constants

Constant	Description	Value
nwkRecommendedSessionTxRandomInterval	Recommended value for the Session Tx Random Interval subfield in outgoing Explore Frames.	250
nwkMinTransmitAttemptsBeforeExploreFrame	Number of routing attempts that a node **shall** perform before issuing an Explore Frame to find a route to a destination.	3
nwkRecommendedNumberOfReturnRoutes	Recommended number of return routes that should be assigned to end nodes for a NodeID destination	4
nwkMinNetworkOperationBackOffTime	Minimum Backoff before trying an operation again, (such as Automatic Controller Update or Static Route Request)	1 minute
nwkTransferPresentationRepeatTime	Duration between Transfer Presentation Commands when a controller tries to include or exclude a node.	2 seconds
nwkSmartStartInclusionRequestDuration	Duration between SmartStart Prime Command and SmartStart Inclusion Request Command.	4 seconds (tolerance ± 1 second)
nwkSmartStartInclusionBackoffDuration	Duration between SmartStart inclusion requests	Table 4.29
nwkMinNWIModeSmartStartDuration	Minimum duration in which NWI mode **shall** stay enabled after a controller node has received a SmartStart DSK to include.	60 minutes
nwkNWIModeDefaultTimeout	Default duration for activating NWI Mode	5 minutes
nwkLearnModeNetworkWideMinDuration	Minimum duration in which node **shall** stay in Learn mode using Network Wide Inclusion/Exclusion.	4 seconds
nwkLearnModeMinDuration	Minimum duration in which a node **shall** stay in Learn Mode during using Classic Network Inclusion/Exclusion.	2 seconds

NWK:0178.1 The attributes defined by the Z-Wave NWK layer are presented in Table 4.28. Implementations **shall** comply with the indicated ranges.

Table 4.28 NWK layer attributes

Attribute	Description	Range
aNwkSmartStartMaxInclusionRequestInterval	Maximum Time interval between SmartStart Inclusion requests.	[4..99] * 128 seconds
aNwkRoutedAckTimeout	Timeout for considering that a particular Routed Frame has been lost and will not return any Routed Acknowledgement or Routed Error. Frame length, bitrate, number of repeaters and MAC Layer retransmissions (*aMacMaxFrameRetries*, refer to [G.9959]) **should** be taken into consideration for selecting the timeout.	18ms..1000ms
aNwkRandomHomeID	HomeID generated randomly after network exclusion or when a node is not part of a network	0xC0000000.. 0xFFFFFFFF

4.5 Functional description

NWK:0200.1 Z-Wave NWK commands in frame flows contained in this section **shall** use the No Operation Command Class and the Z-Wave Protocol Command Class. (refer to 4.3.1 and 4.3.2)

4.5.1 Routing

This section details how routing takes places in the network.

4.5.1.1 Assigning return routes

NWK:0179.1 A controller node **shall** assign return routes to end nodes, to provide them with reliable routes for reaching destinations in the network. This is done using the Assign Return Route Command (and Assign SUC Return Route Command for the SUC NodeID).

NWK:017A.1 The application layer **shall** be able to instruct the NWK layer to assign return routes for nodes.

NWK:017B.1 A controller **should** assign up to 4 return routes to a destination if possible.

NWK:017C.1 End nodes without return route assigned to a destination **should** try direct range transmission. If direct range transmission fails, they **should** issue their command using a Normal Explore Frame or **may** request a static route to the SUC. Refer to 4.5.9.5 End node route request. Controller nodes **should** ignore return routes and calculate their own route using the network topology information.

4.5.1.2 Priority routes

NWK:017D.1 If a Priority Route has been set for a NodeID destination (i.e. an Assign Return Route Priority Command or Assign SUC Return Route Priority Command has been received), an end node **shall** try to use this route for the first transmit attempt.

NWK:017E.1 Controller nodes **may** ignore priority routes and use the route of their choice for the first transmission attempt.

NWK:017F.1 An end node with a priority route set for a destination **may** try to use alternative routes only if at least one transmission using the priority route was unsuccessful.

4.5.1.3 General routing requirements

NWK:0180.1 A node sending or repeating a routing frame **should not** request a MPDU Acknowledgement (ACK Req subfield set to 0 in the MPDU Frame Control). The sending node **should** instead listen for the next repeater repeat frame and use this as a silent acknowledgement.

When routing to a FL node destination, the penultimate repeater will not be able to make use of the silent acknowledgement functionality when the last repeater must send a Wake Up beam prior to repeating the routed frame to the destination NodeID. Therefore, the last repeater **shall** return an ACK MPDU to the penultimate repeater even if Ack Req was set to 0 in the MPDU Frame Control. NWK:0181.1

NWK:0182.1 The repeater 0 node **shall** request an ACK MPDU to the destination NodeID when it repeats a Routed NPDU with the *Direction* field set to 1 (Routed Error or Routed Acknowledgement frame).

NWK:0183.1 A node receiving a Routed Frame **shall** return a Routed Acknowledgement using the same route as in the received Routed Frame.

NWK:0184.1 Nodes **shall** issue a minimum of *nwkMinTransmitAttemptsBeforeExploreFrame* Routed Frames or direct range frames to a destination before issuing a Normal Explore Frame to find a new route.
All Routed Frame transmissions attempts **may** use the same route. Controllers **should** try to calculate new routes based on their network topology information.

In general, the following route resolution is recommended:

- Priority route (if any) or last working route
- Calculated routes for controllers / Assigned return routes for end nodes
- Explorer Frames

NWK:0185.1 Nodes **may** omit the Wake Up Destination information for a FL node destination in the first Routed Frame transmission attempt to minimize beaming (in case the destination node is already awake).

4.5.1.3.1 AL nodes

4.5.1.3.1.1 Repeating frames

In normal operation:

AL nodes **shall** repeat Routed NPDUs sent on their HomeID, if their NodeID is in the repeater list. NWK:0186.1

NWK:0187.1 AL nodes **shall** repeat all new Normal Explore Frames sent on their HomeID.

NWK:0188.1 AL nodes **shall** repeat all Search Result Explore Frames sent on their HomeID, if their NodeID is in the repeater list.

NWI mode:

NWK:0189.1 If NWI mode is enabled, AL nodes **shall** repeat Routed Frames sent on a foreign HomeID, if their NodeID is in the repeater list.

NWK:018A.1 If NWI mode is enabled, AL nodes **shall** repeat Inclusion Request Explore Frames when:

- The Network HomeID set to 0x00000000
- The Network HomeID set to the repeating's node HomeID.

NWK:018B.1 AL nodes **shall not** repeat Inclusion Request Explore Frames if:

- NWI mode is disabled
- The Network HomeID is set to a foreign HomeID (and different than 0x00000000)

4.5.1.3.2 FL and NL nodes

NWK:018C.1 FL and NL nodes **shall not** repeat Routed and Explore NDPUs.

4.5.1.4 Successfully delivered Routed frame

This section specifies the procedure for routing a message across multiple hops to a destination.

4.5.1.4.1 Channel Configuration 1,2

Figure 4.59 depicts an example of a routed frame with 2 repeaters to a destination NodeID using Channel Configuration 1,2.

NWK:018D.1 The repeater list presented as Repeater = [2, 3] **shall** indicate that the *Repeater 0* field is set to 2 and the *Repeater 1* field is set to 3.

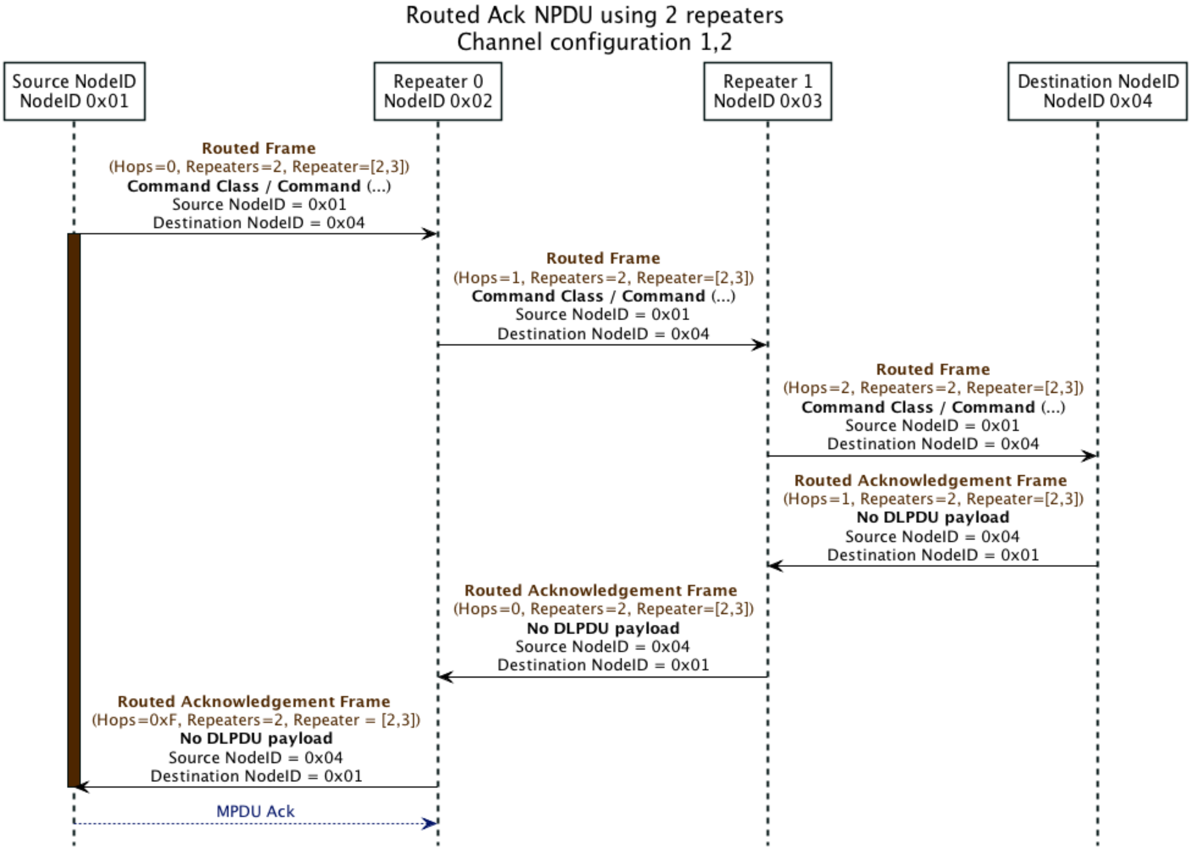

Routed Ack NPDU using 2 repeaters
Channel configuration 1,2
Source NodeID
NodeID 0x01
Repeater 0
NodeID 0x02
Repeater 1
NodeID 0x03
Destination NodeID
NodeID 0x04
Routed Frame
(Hops=0, Repeaters=2, Repeater=[2,3])
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Frame
(Hops=1, Repeaters=2, Repeater=[2,3])
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Frame
(Hops=2, Repeaters=2, Repeater=[2,3])
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Acknowledgement Frame
(Hops=1, Repeaters=2, Repeater=[2,3])
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
Routed Acknowledgement Frame
(Hops=0, Repeaters=2, Repeater=[2,3])
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
Routed Acknowledgement Frame
(Hops=0xF, Repeaters=2, Repeater = [2,3])
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
MPDU Ack

Figure 4.59 Routed frame to an AL node (Channel Configuration 1,2)

4.5.1.4.2 Channel Configuration 3

NWK:018E.1 A node using Channel Configuration 3 **shall** set the Destination Wake Up field to 0 when transmitting to an AL node. An example is depicted in Figure 4.60

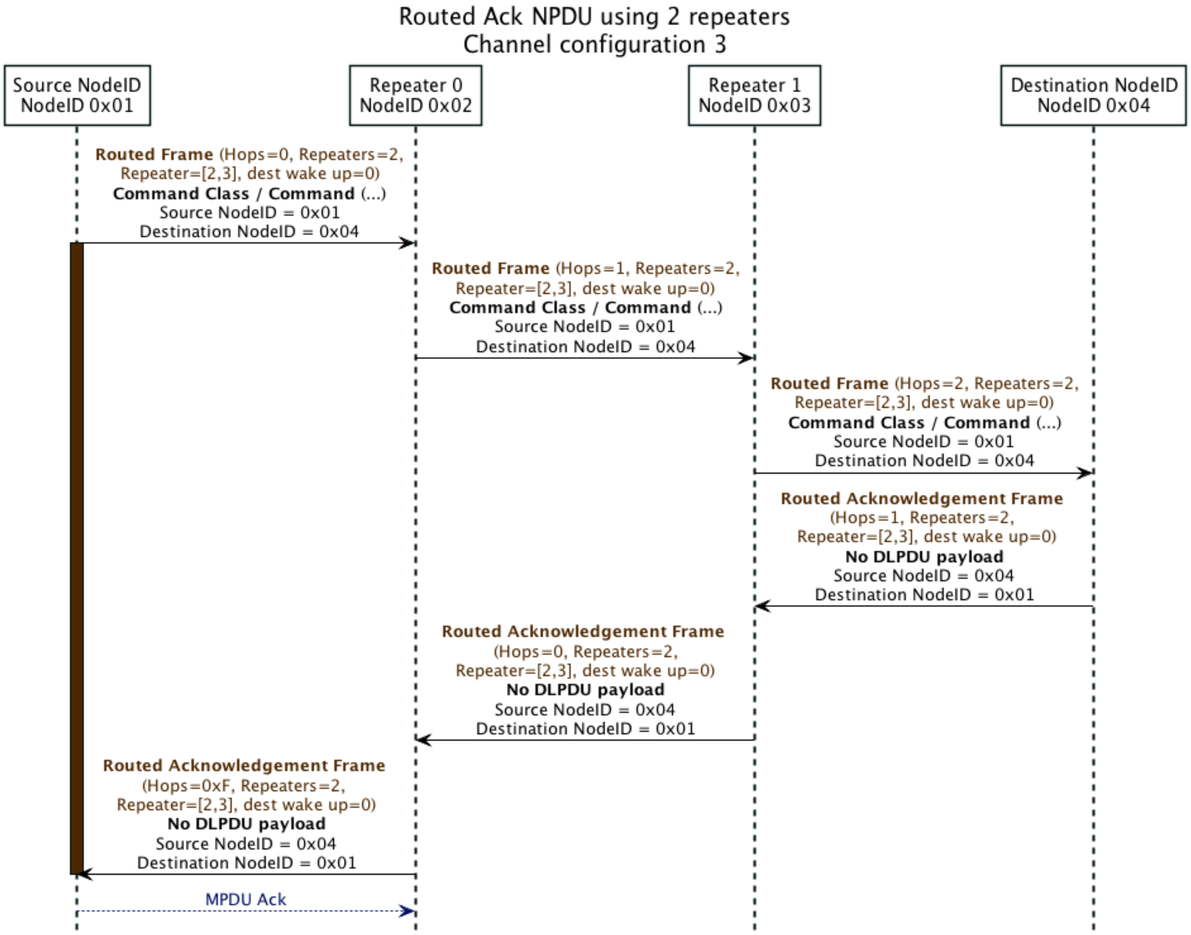

Routed Ack NPDU using 2 repeaters
Channel configuration 3
Source NodeID NodeID 0x01
Repeater 0 NodeID 0x02
Repeater 1 NodeID 0x03
Destination NodeID NodeID 0x04
Routed Frame (Hops=0, Repeaters=2, Repeater=[2,3], dest wake up=0)
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Frame (Hops=1, Repeaters=2, Repeater=[2,3], dest wake up=0)
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Frame (Hops=2, Repeaters=2, Repeater=[2,3], dest wake up=0)
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Acknowledgement Frame (Hops=1, Repeaters=2, Repeater=[2,3], dest wake up=0)
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
Routed Acknowledgement Frame (Hops=0, Repeaters=2, Repeater=[2,3], dest wake up=0)
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
Routed Acknowledgement Frame (Hops=0xF, Repeaters=2, Repeater=[2,3], dest wake up=0)
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
MPDU Ack

Figure 4.60 Routed frame to an AL node (Channel Configuration 3)

4.5.1.5 Routed singlecast to an FL node destination

NWK:018F.1 In the case of a FL node destination, the last repeater **shall** use the *Destination Wake Up* extension or field to wake up the destination with a Wake Up beam prior to repeating the routed message.

This process is illustrated in subsections 4.5.1.5.1 and 4.5.1.5.2.

4.5.1.5.1 Channel Configuration 1,2

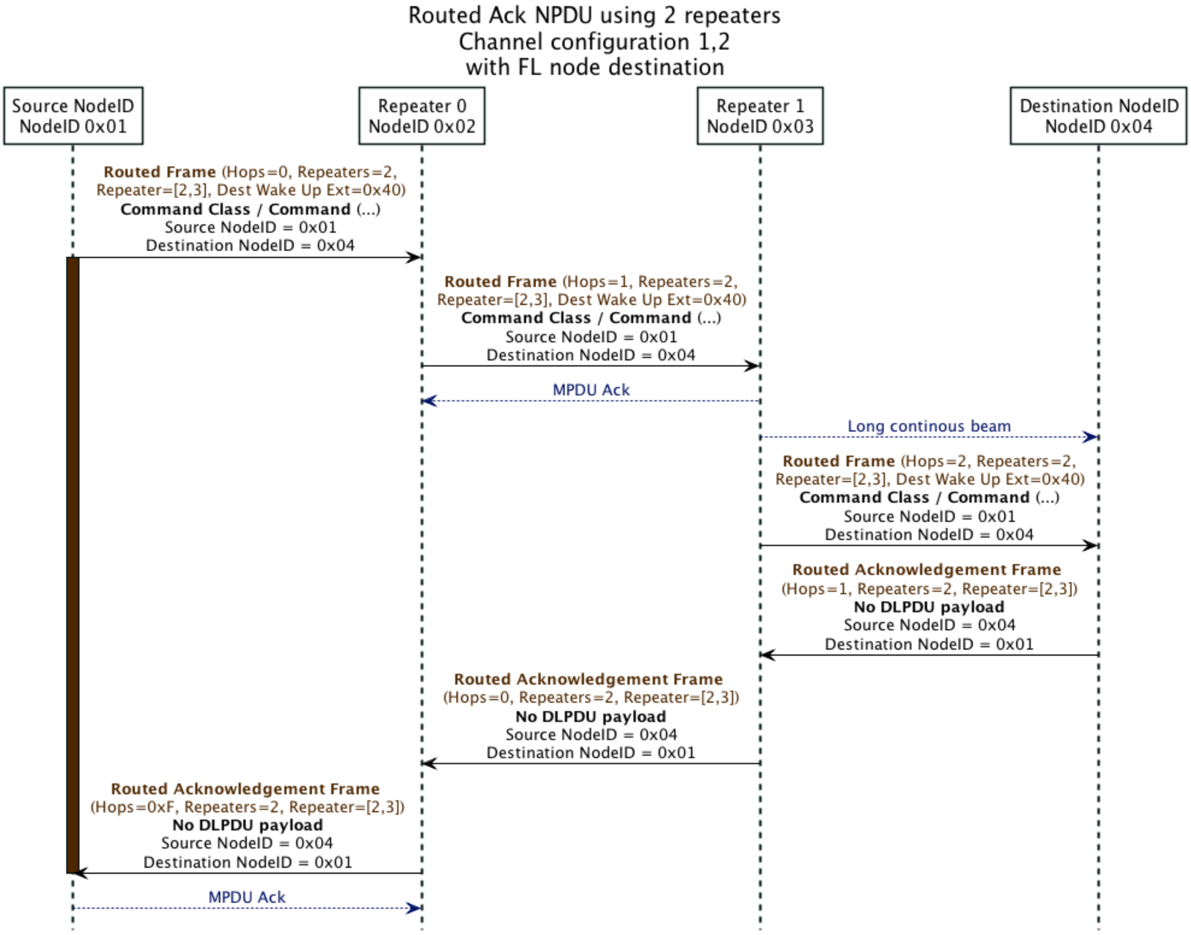

Routed Ack NPDU using 2 repeaters
Channel configuration 1,2
with FL node destination
Source NodeID NodeID 0x01
Repeater 0 NodeID 0x02
Repeater 1 NodeID 0x03
Destination NodeID NodeID 0x04
Routed Frame (Hops=0, Repeaters=2, Repeater=[2,3], Dest Wake Up Ext=0x40) Command Class / Command (...) Source NodeID = 0x01 Destination NodeID = 0x04
Routed Frame (Hops=1, Repeaters=2, Repeater=[2,3], Dest Wake Up Ext=0x40) Command Class / Command (...) Source NodeID = 0x01 Destination NodeID = 0x04
MPDU Ack
Long continous beam
Routed Frame (Hops=2, Repeaters=2, Repeater=[2,3], Dest Wake Up Ext=0x40) Command Class / Command (...) Source NodeID = 0x01 Destination NodeID = 0x04
Routed Acknowledgement Frame (Hops=1, Repeaters=2, Repeater=[2,3]) No DLPDU payload Source NodeID = 0x04 Destination NodeID = 0x01
Routed Acknowledgement Frame (Hops=0, Repeaters=2, Repeater=[2,3]) No DLPDU payload Source NodeID = 0x04 Destination NodeID = 0x01
Routed Acknowledgement Frame (Hops=0xF, Repeaters=2, Repeater=[2,3]) No DLPDU payload Source NodeID = 0x04 Destination NodeID = 0x01
MPDU Ack

Figure 4.61 Routed frame to an FL node (Channel Configuration 1,2)

4.5.1.5.2 Channel Configuration 3

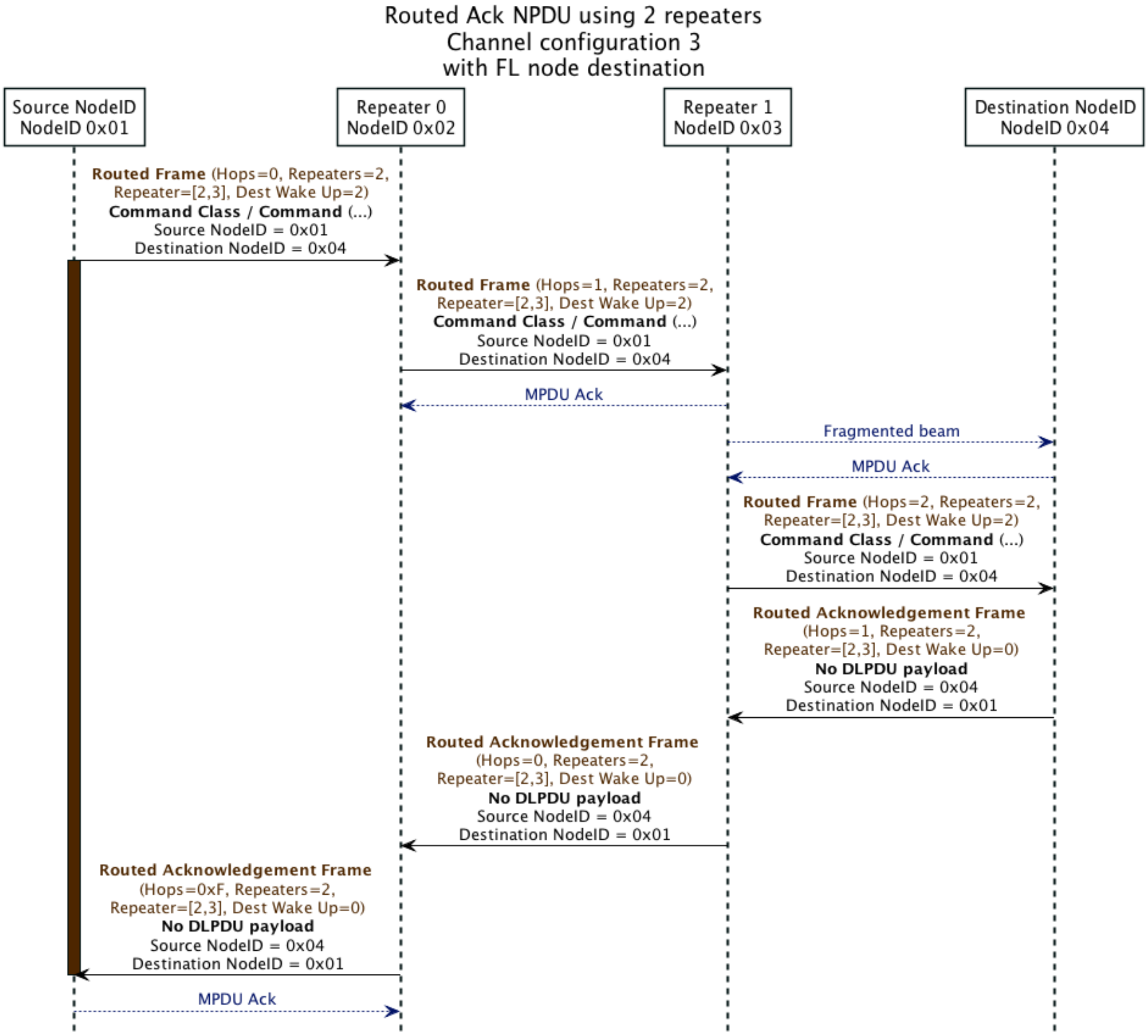

Routed Ack NPDU using 2 repeaters
Channel configuration 3
with FL node destination
Source NodeID NodeID 0x01
Repeater 0 NodeID 0x02
Repeater 1 NodeID 0x03
Destination NodeID NodeID 0x04
Routed Frame (Hops=0, Repeaters=2, Repeater=[2,3], Dest Wake Up=2)
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Frame (Hops=1, Repeaters=2, Repeater=[2,3], Dest Wake Up=2)
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
MPDU Ack
Fragmented beam
MPDU Ack
Routed Frame (Hops=2, Repeaters=2, Repeater=[2,3], Dest Wake Up=2)
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Acknowledgement Frame (Hops=1, Repeaters=2, Repeater=[2,3], Dest Wake Up=0)
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
Routed Acknowledgement Frame (Hops=0, Repeaters=2, Repeater=[2,3], Dest Wake Up=0)
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
Routed Acknowledgement Frame (Hops=0xF, Repeaters=2, Repeater=[2,3], Dest Wake Up=0)
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
MPDU Ack

Figure 4.62 Routed frame to an FL node (Channel Configuration 3)

4.5.1.6 Unsuccessful Routed frame with Routed Error frame

Figure 4.63 depicts the failed delivery of a routed frame. Retransmissions triggered by the MAC layer are not shown.

NWK:0190.1 A repeater node failing to transmit to the next hop **shall** return a Routed Error frame. The repeater node sending the Routed Error Frame **shall** set the Source NodeID of the frame as the value of the Destination NodeID of the Routed Frame which delivery failed.

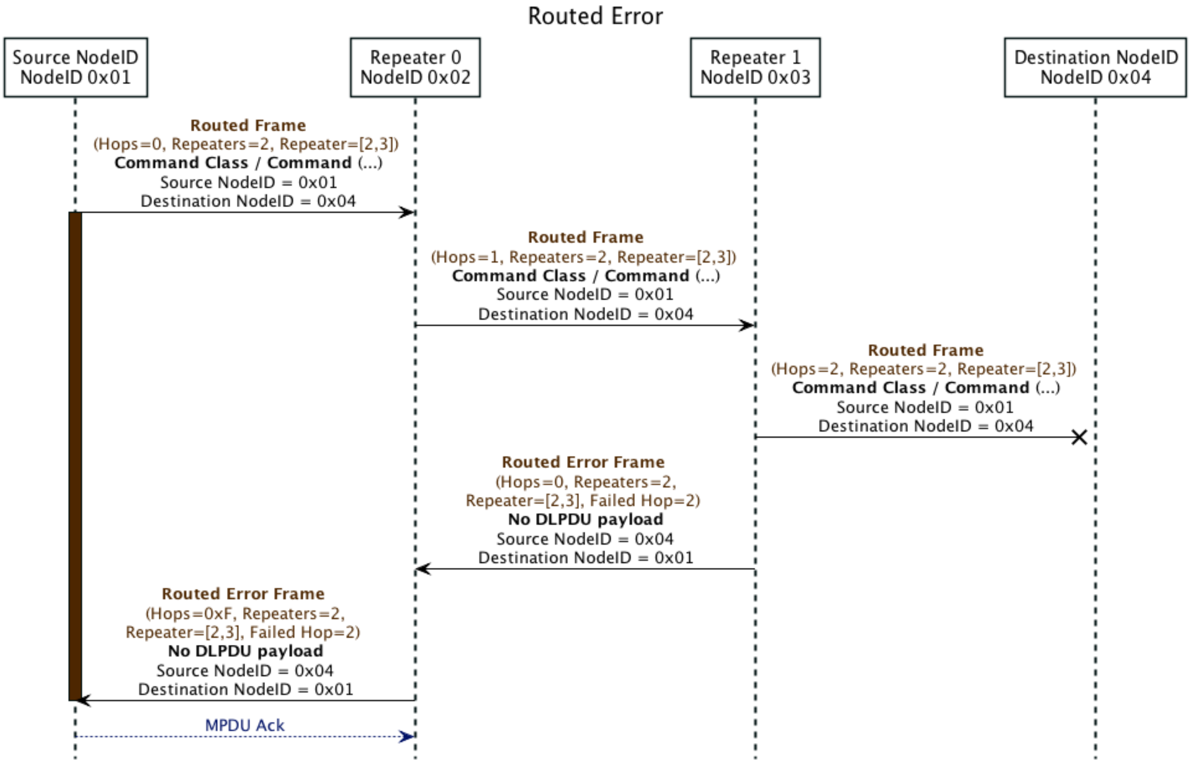

Routed Error
Source NodeID NodeID 0x01
Repeater 0 NodeID 0x02
Repeater 1 NodeID 0x03
Destination NodeID NodeID 0x04
Routed Frame
(Hops=0, Repeaters=2, Repeater=[2,3])
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Frame
(Hops=1, Repeaters=2, Repeater=[2,3])
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Frame
(Hops=2, Repeaters=2, Repeater=[2,3])
Command Class / Command (...)
Source NodeID = 0x01
Destination NodeID = 0x04
Routed Error Frame
(Hops=0, Repeaters=2,
Repeater=[2,3], Failed Hop=2)
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
Routed Error Frame
(Hops=0xF, Repeaters=2,
Repeater=[2,3], Failed Hop=2)
No DLPDU payload
Source NodeID = 0x04
Destination NodeID = 0x01
MPDU Ack

Figure 4.63 Unsuccessful Routed frame delivery

4.5.1.7 Unsuccessful Routed frame without Routed Error frame

NWK:0191.1 In the unlikely event that a Routed Acknowledgment frame or Routed Error frame is not returned to the sending node, the sending node **shall** timeout after a duration of their choice denoted *aNwkRoutedAckTimeout.*

Figure 4.64 depicts the failed delivery of a routed frame with a failure to transmit the Routed Error Frame. Retransmissions triggered by the MAC layer are not shown.

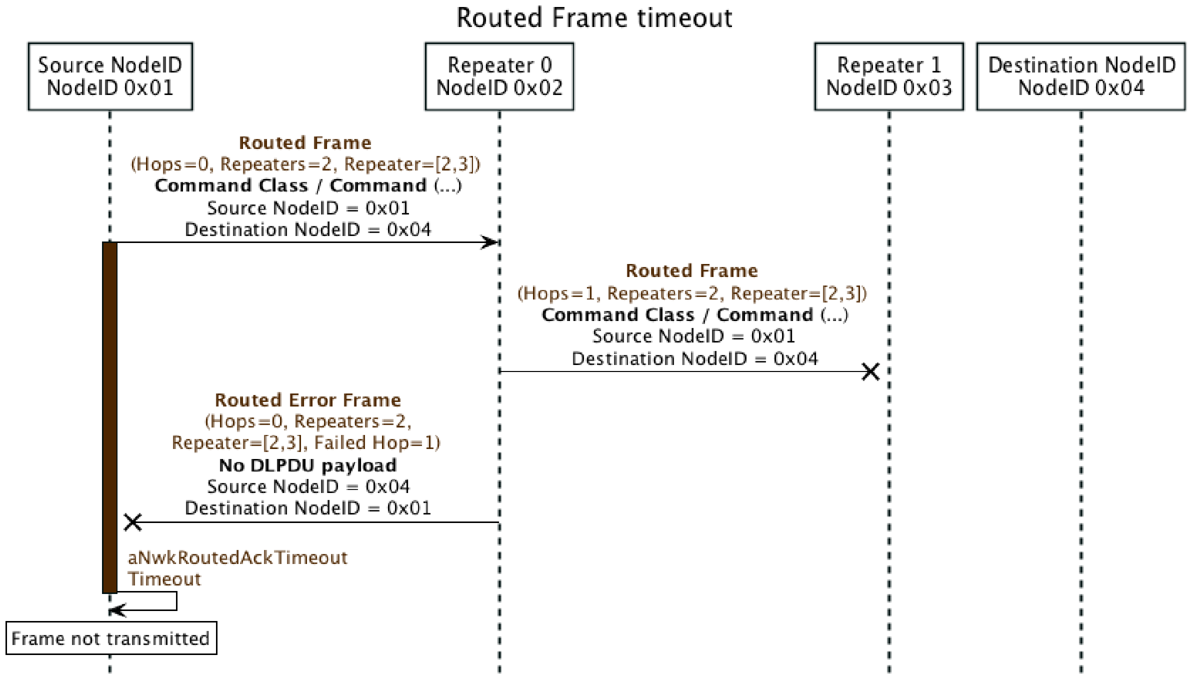


Figure 4.64 Routing - Timeout waiting for Routed Ack/Error frame

4.5.1.8 Normal and Search Result Explore frames

Figure 4.65 depicts how a Normal Explore frame reaches its destination. It illustrates how NodeID 1 is looking for a route to NodeID 9. Figure 4.66 depicts the route result.

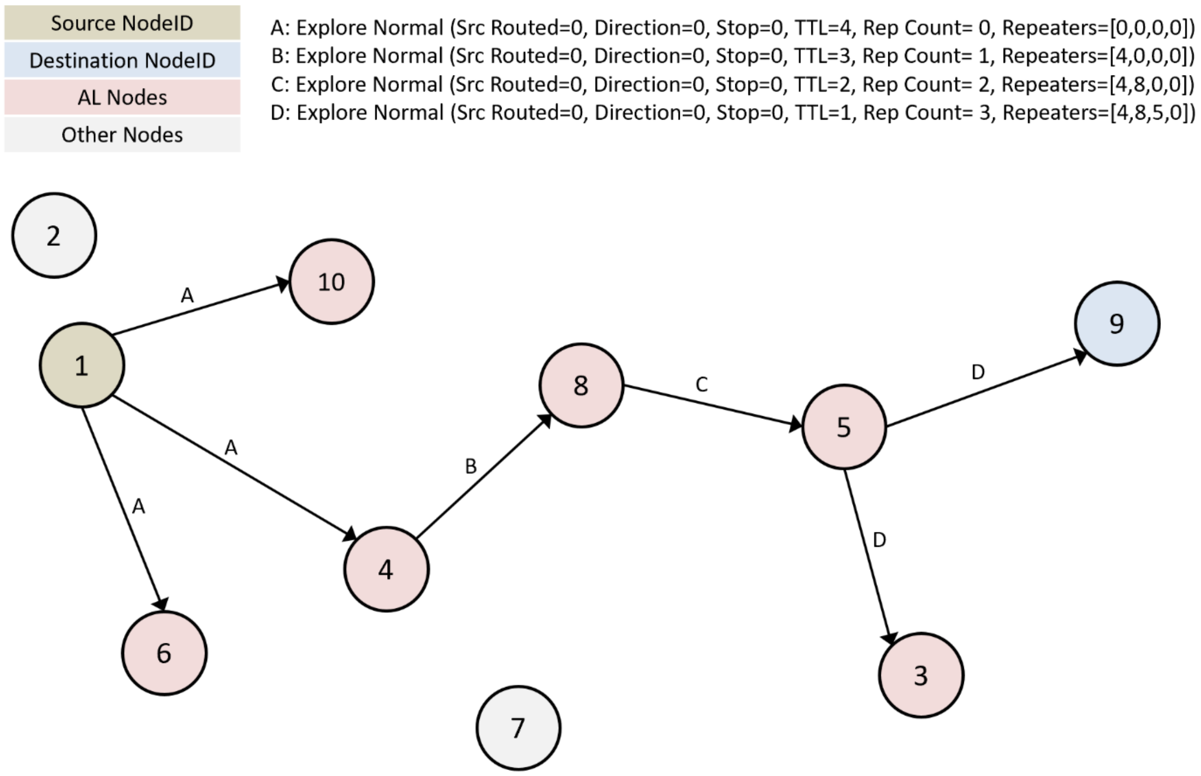

Source NodeID
Destination NodeID
AL Nodes
Other Nodes
A: Explore Normal (Src Routed=0, Direction=0, Stop=0, TTL=4, Rep Count= 0, Repeaters=[0,0,0,0])
B: Explore Normal (Src Routed=0, Direction=0, Stop=0, TTL=3, Rep Count= 1, Repeaters=[4,0,0,0])
C: Explore Normal (Src Routed=0, Direction=0, Stop=0, TTL=2, Rep Count= 2, Repeaters=[4,8,0,0])
D: Explore Normal (Src Routed=0, Direction=0, Stop=0, TTL=1, Rep Count= 3, Repeaters=[4,8,5,0])
2
10
1
A
A
A
6
4
B
8
C
5
D
9
D
3
7

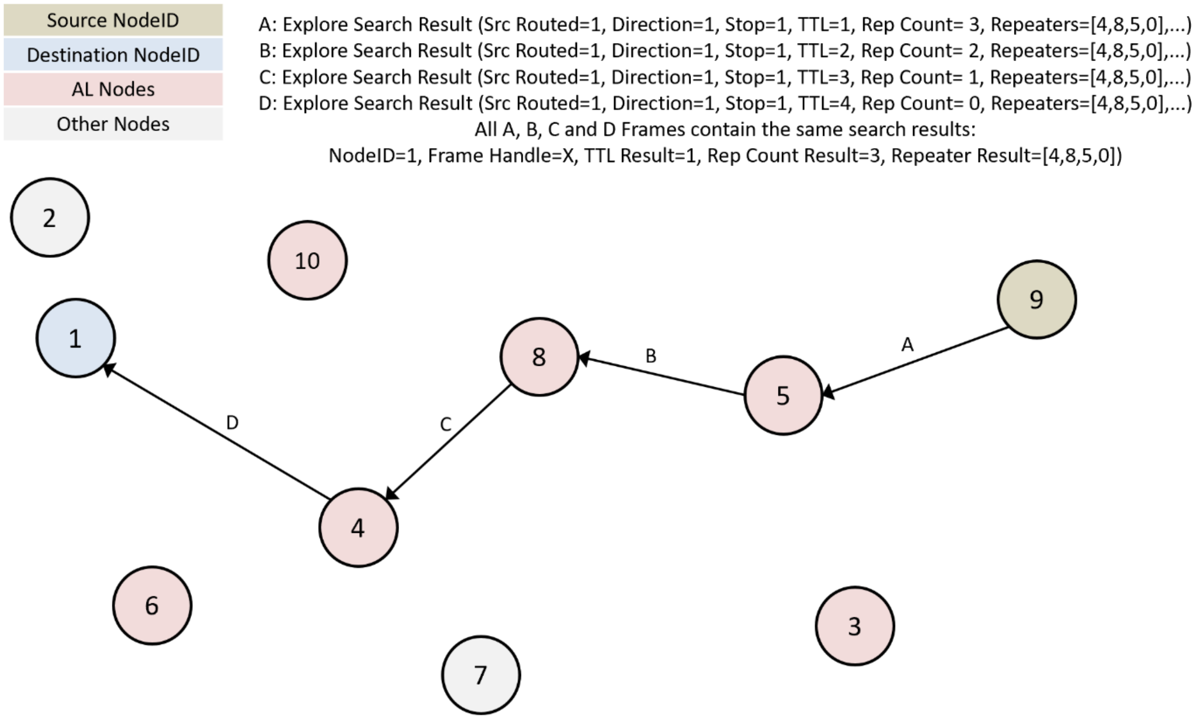
Figure 4.65 Normal Explore Frame example

NWK:0192.1 Every AL Node part of the same HomeID **shall** repeat Normal Explore frames the first time they receive it. Explore Frames are identified using the following fields from the MPDU Header:

- Source NodeID
- Sequence Number

Frame A is received by NodeID 6, 4 and 10. NodeID 6, 4 and 10 **shall** repeat frame A. NodeID 8 receives
NWK:0193.1 the repeated frame for the first time from NodeID 4 (Frame B), which also gets repeated.

When NodeID 9 receives the Normal Explore Frame, it **shall** return a Search Result Explore Frame, using
NWK:0194.1 the same route as the first Normal Explore Frame that it received.



A: Explore Search Result (Src Routed=1, Direction=1, Stop=1, TTL=1, Rep Count= 3, Repeaters=[4,8,5,0],...)
B: Explore Search Result (Src Routed=1, Direction=1, Stop=1, TTL=2, Rep Count= 2, Repeaters=[4,8,5,0],...)
C: Explore Search Result (Src Routed=1, Direction=1, Stop=1, TTL=3, Rep Count= 1, Repeaters=[4,8,5,0],...)
D: Explore Search Result (Src Routed=1, Direction=1, Stop=1, TTL=4, Rep Count= 0, Repeaters=[4,8,5,0],...)
All A, B, C and D Frames contain the same search results:
NodeID=1, Frame Handle=X, TTL Result=1, Rep Count Result=3, Repeater Result=[4,8,5,0])

Figure 4.66 Search Result Explore Frame example

4.5.2 Learn mode

NWK:0195.1 Z-Wave nodes **shall** provide functionalities that enable them to learn about the existing current network or a new network.

Learn Mode is used for several purposes:

- It is used for a node to accept changing network (joining or leaving).
- It is at times used for a controller node to accept receiving updated network topology information.

NWK:0196.1 Learn mode **should** only be enabled when necessary and disabled again as quickly as possible. The application layer **may** determine when learn mode is to be enabled/disabled.

NWK:0197.1 Nodes typically enter Learn Mode to join or leave a network. When a node enters to Learn Mode, it **may** have the following intents:

- Learn Mode Inclusion: the node is expecting a network inclusion.
- Learn Mode Exclusion: the node is expecting a network exclusion.
- SmartStart Learn Mode: the node operates with the SmartStart procedure for inclusion.

NWK:0198.1 When learn mode is enabled on a node, it **shall** accept Assign IDs Commands only if the payload matches their intent. More details are given in the individual scenarios described in 4.5.3 and 4.5.5.

A Secondary Controller **may** also enter to Learn Mode Inclusion to receive the network topology.

NWK:0199.1

NWK:019A.1 A node, that supports Classic Inclusion/Exclusion entering Learn mode, **shall** stay in Learn mode for a minimum duration of *nwkLearnModeMinDuration*.

NWK:019B.1 A node, that supports Network Wide Inclusion/Exclusion entering Learn mode, **shall** stay in Learn mode for a minimum duration of *nwkLearnModeNetworkWideMinDuration.*

4.5.3 Network Formation

NWK:019C.1 End nodes **shall not** start a new network and **shall** wait until they get included in a network by a controller node. When not included in a network, end nodes **shall** assign themselves a HomeID (*aNwkRandomHomeID)* using a random number generator.

NWK:019D.1 Controller nodes that do not belong to a network **shall** start a new network automatically by:

- Assigning themselves a HomeID and NodeID. The HomeID (*aNwkRandomHomeID)* **shall** be generated using a random number generator.
- Assuming the Primary Controller role
- Deciding if they take the SUC/SIS role.

4.5.4 Network Inclusion

Several procedures can be used for including a node into a network. They are:

- Classic network inclusion.
- Network wide Inclusion (NWI)
- SmartStart Inclusion (which is also network-wide)

4.5.4.1 Classic Network Inclusion

The Classic Network inclusion process will include new nodes to a network, using direct range communication only. No Routed NPDUs or Explore NPDUs are used in this case.

NWK:019E.1 The Classic Network Inclusion procedure **shall** be according to Figure 4.67.

NWK:019F.1 When starting a Learn Mode, a node **should** listen for Transfer Presentation Commands with the *Option* field indicating a controller trying to include a node. Upon reception, the joining node **shall** issue a Node Information Frame Command to the NodeID of the controller.

NWK:01A0.1 Alternatively, a node **may** ignore Transfer Presentation Commands and issue a Node Information Frame Command to the broadcast destination NodeID.

NWK:01A1.1 The including controller **shall** issue the Assign IDs Command using the NodeID and HomeID used by the joining node in its Node Information Frame Command, even if the NodeID is not a part of the valid NodeID range.

NWK:01A2.1 The Neighbour Discovery step in Figure 4.67 **shall** follow the description illustrated in Figure 4.91.

NWK:01A3.1 Joining nodes **may** use source NodeID 0xEF (instead of 0x00) while in learn mode to support old controller implementations.

NWK:01A4.1 Joining nodes **shall** consider network inclusion to be completed and learn mode to be deactivated automatically when they return a Range Info Command to the including controller.

NWK:01A5.1 Joining nodes **shall not** request MPDU Acknowledgement when issuing Node Information Frame Command, even if it is issued using Singlecast addressing.

NWK:01A6.1 Including controllers **shall** consider network inclusion to be completed and **should** deactivate add mode automatically when:

- they issued the SUC NodeID Command to an included controller node.
- they assigned the SUC Return Route to an included end node.

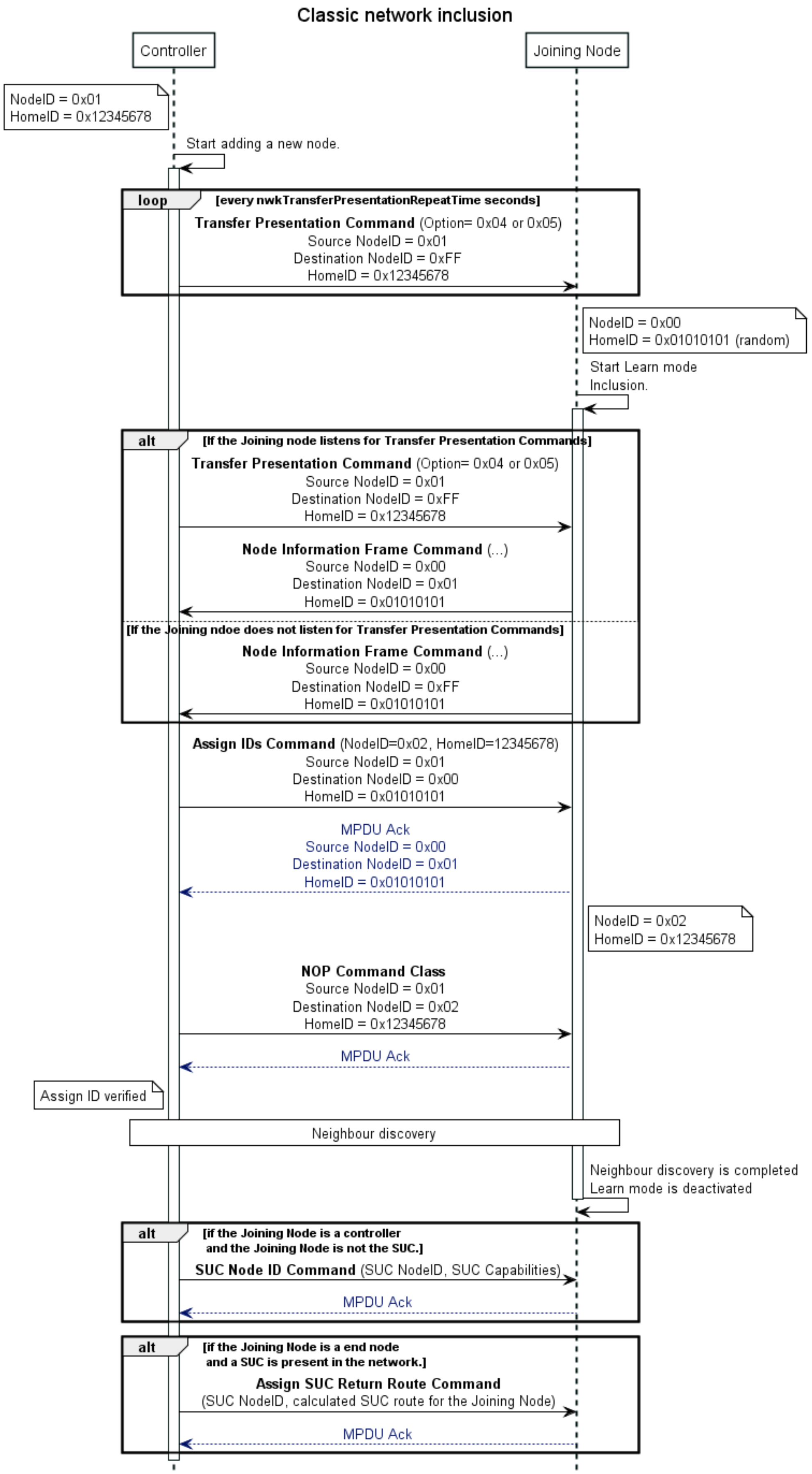

Classic network inclusion
Controller
Joining Node
NodeID = 0x01
HomeID = 0x12345678
Start adding a new node.
loop
[every nwkTransferPresentationRepeatTime seconds]
Transfer Presentation Command (Option= 0x04 or 0x05)
Source NodeID = 0x01
Destination NodeID = 0xFF
HomeID = 0x12345678
NodeID = 0x00
HomeID = 0x01010101 (random)
Start Learn mode
Inclusion.
alt
[If the Joining node listens for Transfer Presentation Commands]
Transfer Presentation Command (Option= 0x04 or 0x05)
Source NodeID = 0x01
Destination NodeID = 0xFF
HomeID = 0x12345678
Node Information Frame Command (...)
Source NodeID = 0x00
Destination NodeID = 0x01
HomeID = 0x01010101
[If the Joining ndoe does not listen for Transfer Presentation Commands]
Node Information Frame Command (...)
Source NodeID = 0x00
Destination NodeID = 0xFF
HomeID = 0x01010101
Assign IDs Command (NodeID=0x02, HomeID=12345678)
Source NodeID = 0x01
Destination NodeID = 0x00
HomeID = 0x01010101
MPDU Ack
Source NodeID = 0x00
Destination NodeID = 0x01
HomeID = 0x01010101
NodeID = 0x02
HomeID = 0x12345678
NOP Command Class
Source NodeID = 0x01
Destination NodeID = 0x02
HomeID = 0x12345678
MPDU Ack
Assign ID verified
Neighbour discovery
Neighbour discovery is completed
Learn mode is deactivated
alt
[if the Joining Node is a controller
and the Joining Node is not the SUC.]
SUC Node ID Command (SUC NodeID, SUC Capabilities)
MPDU Ack
alt
[if the Joining Node is a end node
and a SUC is present in the network.]
Assign SUC Return Route Command
(SUC NodeID, calculated SUC route for the Joining Node)
MPDU Ack

Figure 4.67 Network Inclusion - Classic inclusion procedure

4.5.4.2 Network Wide Inclusion (NWI)

Network-Wide Inclusion (NWI) allows a new node to be included across an existing Z-Wave network without direct range connectivity between the including controller and the joining node.

This procedure leverages Routed NPDUs and Explore NDPUs during the inclusion process.

NWK:01A7.1 NWI **should** be used as the default learn mode to ensure compatibility with all implementations of Z-Wave nodes.

NWK:01A8.1 When a controller node enters NWI add mode, it **shall** enable NWI Mode using the Set NWI Mode Command in a Normal Explore Frame.

NWK:01A9.1 The Network Wide Inclusion procedure **shall** be according to Figure 4.68.

NWK:01AA.1 When a node enters NWI Learn mode, it **shall** issue a Node Information Frame Command encapsulated in an Inclusion Request Explore Frame. This frame **shall** be sent with the MPDU Header Ack Request subfield set to 1.

NWK:01AB.1 Including controllers in NWI add mode **shall** send Transfer Presentation Commands with the *Option* field set to 0x05.

NWK:01AC.1 If routing is used for issuing an Assign IDs Command to the joining node, the Routed Acknowledgment frame returned by the joining node **shall** use the newly assigned HomeID. However, the Routed Acknowledgment frame **shall** use the same NodeIDs that were used in the Routed frame carrying the Assign IDs Command.
This is made to ensure that the repeaters will repeat the Routed Acknowledgment frame back to the including controller.

NWK:01AD.1 The Neighbour Discovery step in Figure 4.68 **shall** be according to Figure 4.91.

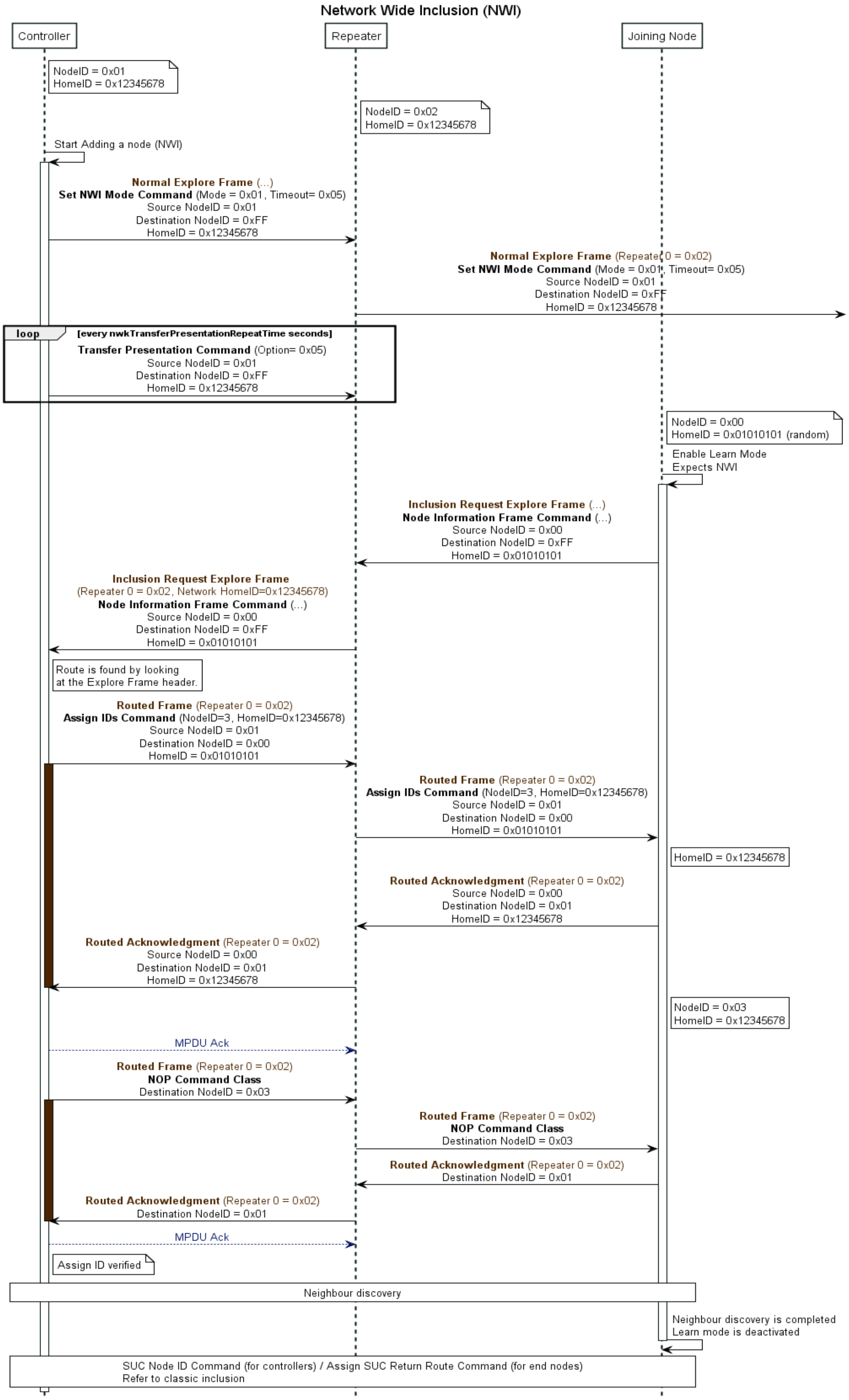

Network Wide Inclusion (NWI)
Controller
Repeater
Joining Node
NodeID = 0x01
HomeID = 0x12345678
NodeID = 0x02
HomeID = 0x12345678
Start Adding a node (NWI)
Normal Explore Frame (...)
Set NWI Mode Command (Mode = 0x01, Timeout= 0x05)
Source NodeID = 0x01
Destination NodeID = 0xFF
HomeID = 0x12345678
Normal Explore Frame (Repeater 0 = 0x02)
Set NWI Mode Command (Mode = 0x01, Timeout= 0x05)
Source NodeID = 0x01
Destination NodeID = 0xFF
HomeID = 0x12345678
loop
[every nwkTransferPresentationRepeatTime seconds]
Transfer Presentation Command (Option= 0x05)
Source NodeID = 0x01
Destination NodeID = 0xFF
HomeID = 0x12345678
NodeID = 0x00
HomeID = 0x01010101 (random)
Enable Learn Mode
Expects NWI
Inclusion Request Explore Frame (...)
Node Information Frame Command (...)
Source NodeID = 0x00
Destination NodeID = 0xFF
HomeID = 0x01010101
Inclusion Request Explore Frame
(Repeater 0 = 0x02, Network HomeID=0x12345678)
Node Information Frame Command (...)
Source NodeID = 0x00
Destination NodeID = 0xFF
HomeID = 0x01010101
Route is found by looking
at the Explore Frame header.
Routed Frame (Repeater 0 = 0x02)
Assign IDs Command (NodeID=3, HomeID=0x12345678)
Source NodeID = 0x01
Destination NodeID = 0x00
HomeID = 0x01010101
Routed Frame (Repeater 0 = 0x02)
Assign IDs Command (NodeID=3, HomeID=0x12345678)
Source NodeID = 0x01
Destination NodeID = 0x00
HomeID = 0x01010101
HomeID = 0x12345678
Routed Acknowledgment (Repeater 0 = 0x02)
Source NodeID = 0x00
Destination NodeID = 0x01
HomeID = 0x12345678
Routed Acknowledgment (Repeater 0 = 0x02)
Source NodeID = 0x00
Destination NodeID = 0x01
HomeID = 0x12345678
NodeID = 0x03
HomeID = 0x12345678
MPDU Ack
Routed Frame (Repeater 0 = 0x02)
NOP Command Class
Destination NodeID = 0x03
Routed Frame (Repeater 0 = 0x02)
NOP Command Class
Destination NodeID = 0x03
Routed Acknowledgment (Repeater 0 = 0x02)
Destination NodeID = 0x01
Routed Acknowledgment (Repeater 0 = 0x02)
Destination NodeID = 0x01
MPDU Ack
Assign ID verified
Neighbour discovery
Neighbour discovery is completed
Learn mode is deactivated
SUC Node ID Command (for controllers) / Assign SUC Return Route Command (for end nodes)
Refer to classic inclusion

Figure 4.68, Network Inclusion - NWI

4.5.4.3 SmartStart Inclusion

The Z-Wave SmartStart inclusion removes the user interactions and lets nodes join a network automatically if the including controller possesses the S2 DSK (refer to [TECC]) of the node to be included.

NWK:01AE.2 Nodes supporting to be included using SmartStart inclusion **shall** provide at least one of the following methods for entering SmartStart

1. Enter SmartStart Learn Mode automatically after powering on
2. **P**rovide a mechanism to enter SmartStart Learn Mode if already powered up.

4.5.4.3.1 SmartStart supporting nodes power-up

NWK:01AF.2 Nodes supporting SmartStart inclusion **should** initiate an inclusion procedure after powering up or alternatively it **should** initiate an inclusion procedure when triggered manually after power-up. This procedure depends on the inclusion state of a node and is described in the following subsections 4.5.4.3.1.1 and 4.5.4.3.1.2.

4.5.4.3.1.1 Not included nodes

NWK:01B0.1 Nodes that are not part of a network **shall** issue SmartStart Inclusion Requests at regular intervals. A SmartStart Inclusion Request **shall** consist of a SmartStart Prime Command and a SmartStart Inclusion Request Command.

NWK:01B1.1 It **shall** be according to Figure 4.69. The NWI HomeID construction is specified in 4.3.2.33.1.1 NWI HomeID (4 bytes).

NWK:01B2.1 Nodes entering Classic Inclusion or NWI Learn Mode **shall** stop issuing SmartStart inclusion requests until they return into SmartStart Learn Mode.

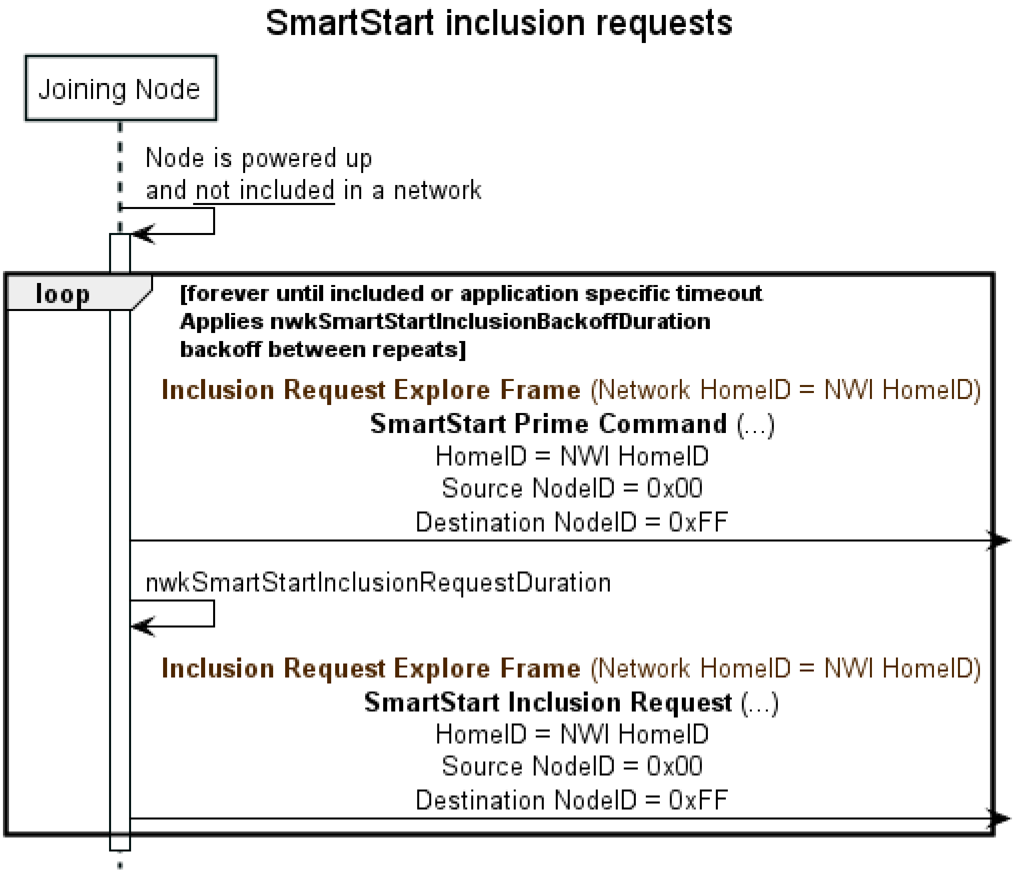

SmartStart inclusion requests
Joining Node
Node is powered up
and not included in a network
loop
[forever until included or application specific timeout
Applies nwkSmartStartInclusionBackoffDuration
backoff between repeats]
Inclusion Request Explore Frame (Network HomeID = NWI HomeID)
SmartStart Prime Command (...)
HomeID = NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFF
nwkSmartStartInclusionRequestDuration
Inclusion Request Explore Frame (Network HomeID = NWI HomeID)
SmartStart Inclusion Request (...)
HomeID = NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFF

Figure 4.69 Network Inclusion - SmartStart not included node power on

NWK:01B3.1 The timing of the inclusion requests **shall** be according to Table 4.29. When a range is indicated as duration, nodes **shall** use a new unique random value in that range every time.

Table 4.29 SmartStart backoff duration for inclusion requests

Request number	Backoff duration interval since last request (seconds)
1 (after power up)	N/A
2	[0..16]
3	[16..32]
4	[32..64]
5	[64..128]
6	[128..256]
7	[256..512]
8	512
>8	*aNwkSmartStartMaxInclusionRequestInterval* This is determined by the node Refer to Table 4.28 for more details.

4.5.4.3.1.2 Included nodes

NWK:01B4.1 A node already part of a network **shall** send a single SmartStart Included Node Information Command when entering SmartStart, i. e. after power up or when triggered manually. This is illustrated in Figure 4.70.

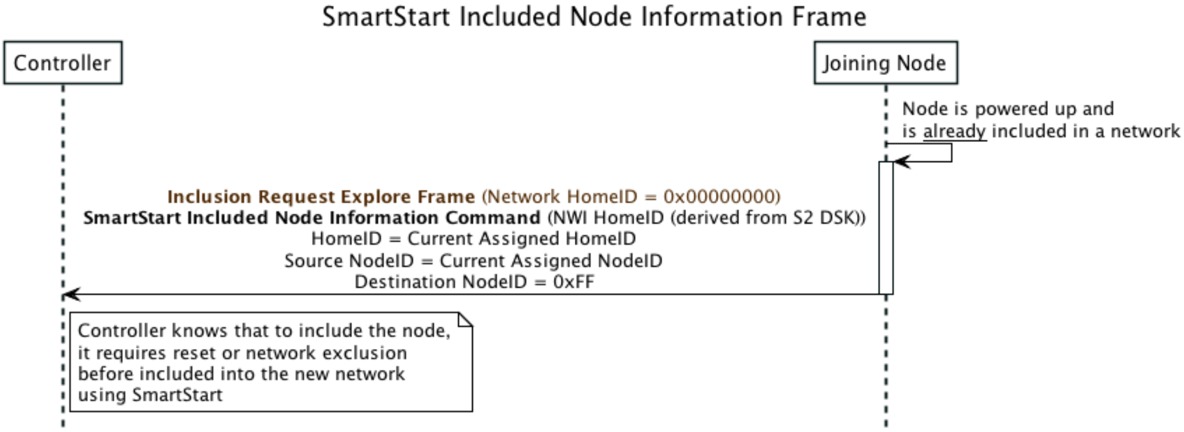

SmartStart Included Node Information Frame
Controller
Joining Node
Node is powered up and
is already included in a network
Inclusion Request Explore Frame (Network HomeID = 0x00000000)
SmartStart Included Node Information Command (NWI HomeID (derived from S2 DSK))
HomeID = Current Assigned HomeID
Source NodeID = Current Assigned NodeID
Destination NodeID = 0xFF
Controller knows that to include the node,
it requires reset or network exclusion
before included into the new network
using SmartStart

Figure 4.70 Network Inclusion - SmartStart included node power on

4.5.4.3.2 SmartStart including controllers

NWK:01B5.1 A controller **shall** have the Primary Controller role in the network to perform a SmartStart inclusion. Secondary Controllers (and Inclusion Controllers) **shall not** perform SmartStart inclusions.

NWK:01B6.1 A controller **shall** be given the S2 DSK of a node to perform a SmartStart inclusion. Refer to [TECC].

NWK:01B7.1 When a SmartStart including controller has received the DSK of a node for inclusion, it **shall** keep NWI Mode enabled for at least *nwkMinNWIModeSmartStartDuration* minutes if the node does not get included. This is illustrated in Figure 4.71.

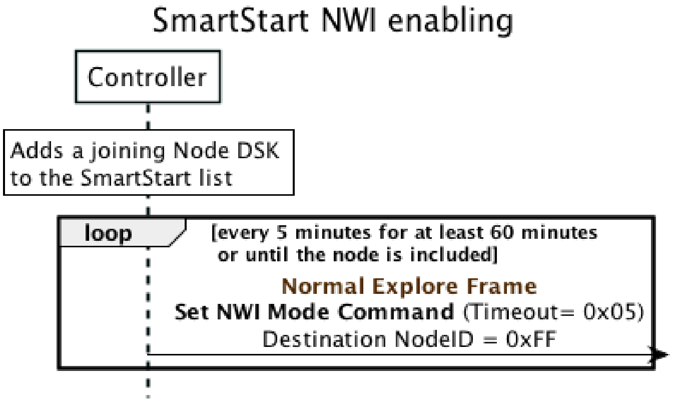

SmartStart NWI enabling
Controller
Adds a joining Node DSK to the SmartStart list
loop
[every 5 minutes for at least 60 minutes or until the node is included]
Normal Explore Frame
Set NWI Mode Command (Timeout= 0x05)
Destination NodeID = 0xFF

Figure 4.71 Network Inclusion - SmartStart NWI mode enabling

4.5.4.3.3 Successful SmartStart inclusion

NWK:01B8.1 A controller performing a SmartStart network inclusion **shall** perform S2 bootstrapping. (even if the joining node does not show the S2 Command Class in its supported Command Class list). Refer to [TECC] for the detailed S2 bootstrapping procedure.

NWK:01B9.1 A SmartStart inclusion **shall** be according to Figure 4.72. Figure 4.73 depicts a SmartStart inclusion using one repeater.

NWK:01BA.1 As for NWI, if routing is used for issuing an Assign IDs Command to the joining node, the Routed Acknowledgment frame returned by the joining node **shall** use the newly assigned HomeID. However, the Routed Acknowledgment frame **shall** use the same NodeIDs that were used in the Routed frame carrying the Assign IDs Command.
This is made to ensure that the repeaters will repeat the Routed Acknowledgment frame back to the including controller.

NWK:01BB.1 The Neighbour Discovery step in Figure 4.72 and Figure 4.73 **shall** be according to Figure 4.91.

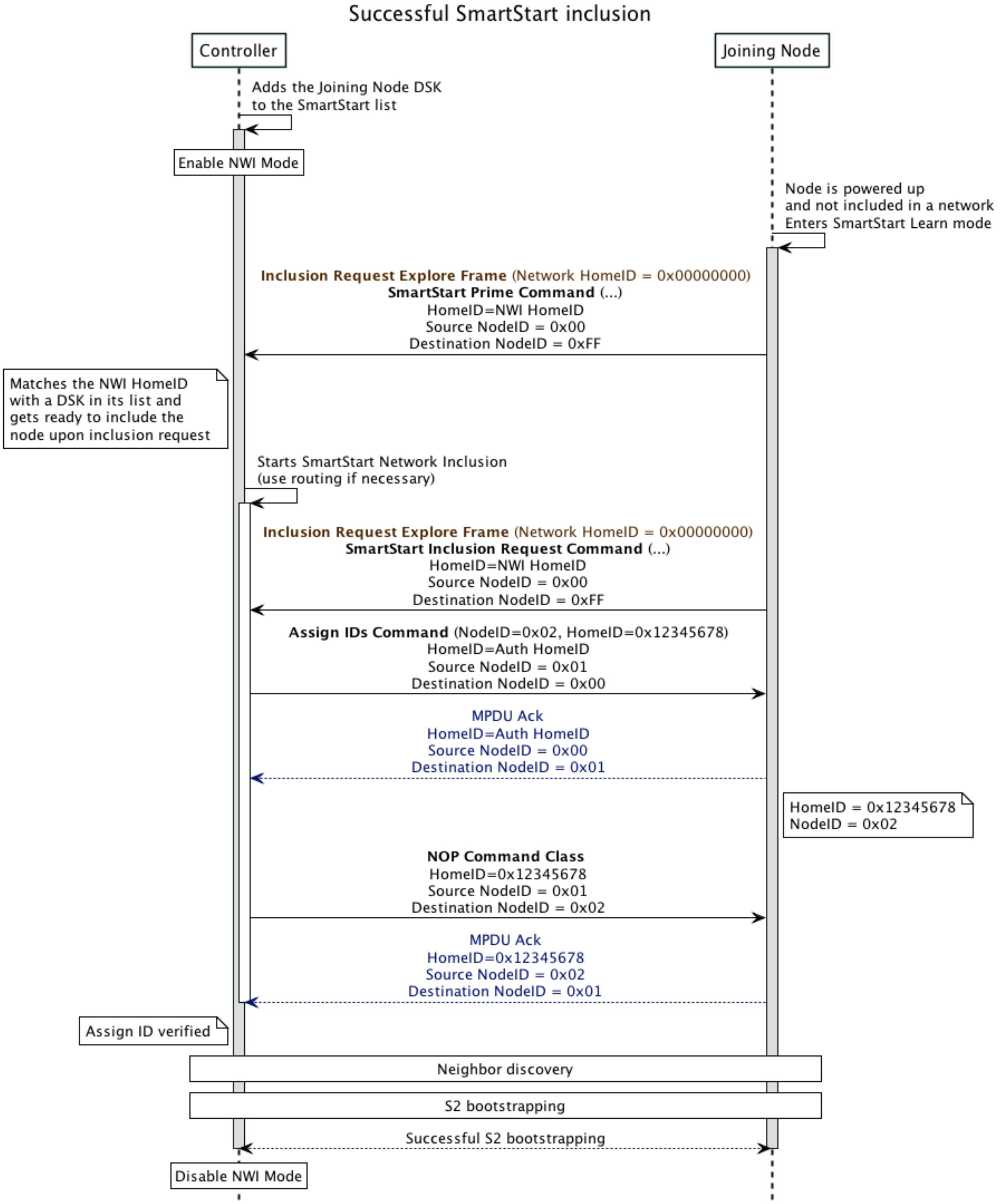

Successful SmartStart inclusion
Controller
Joining Node
Adds the Joining Node DSK
to the SmartStart list
Enable NWI Mode
Node is powered up
and not included in a network
Enters SmartStart Learn mode
Inclusion Request Explore Frame (Network HomeID = 0x00000000)
SmartStart Prime Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFF
Matches the NWI HomeID
with a DSK in its list and
gets ready to include the
node upon inclusion request
Starts SmartStart Network Inclusion
(use routing if necessary)
Inclusion Request Explore Frame (Network HomeID = 0x00000000)
SmartStart Inclusion Request Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFF
Assign IDs Command (NodeID=0x02, HomeID=0x12345678)
HomeID=Auth HomeID
Source NodeID = 0x01
Destination NodeID = 0x00
MPDU Ack
HomeID=Auth HomeID
Source NodeID = 0x00
Destination NodeID = 0x01
HomeID = 0x12345678
NodeID = 0x02
NOP Command Class
HomeID=0x12345678
Source NodeID = 0x01
Destination NodeID = 0x02
MPDU Ack
HomeID=0x12345678
Source NodeID = 0x02
Destination NodeID = 0x01
Assign ID verified
Neighbor discovery
S2 bootstrapping
Successful S2 bootstrapping
Disable NWI Mode

Figure 4.72 Successful SmartStart inclusion (direct range)

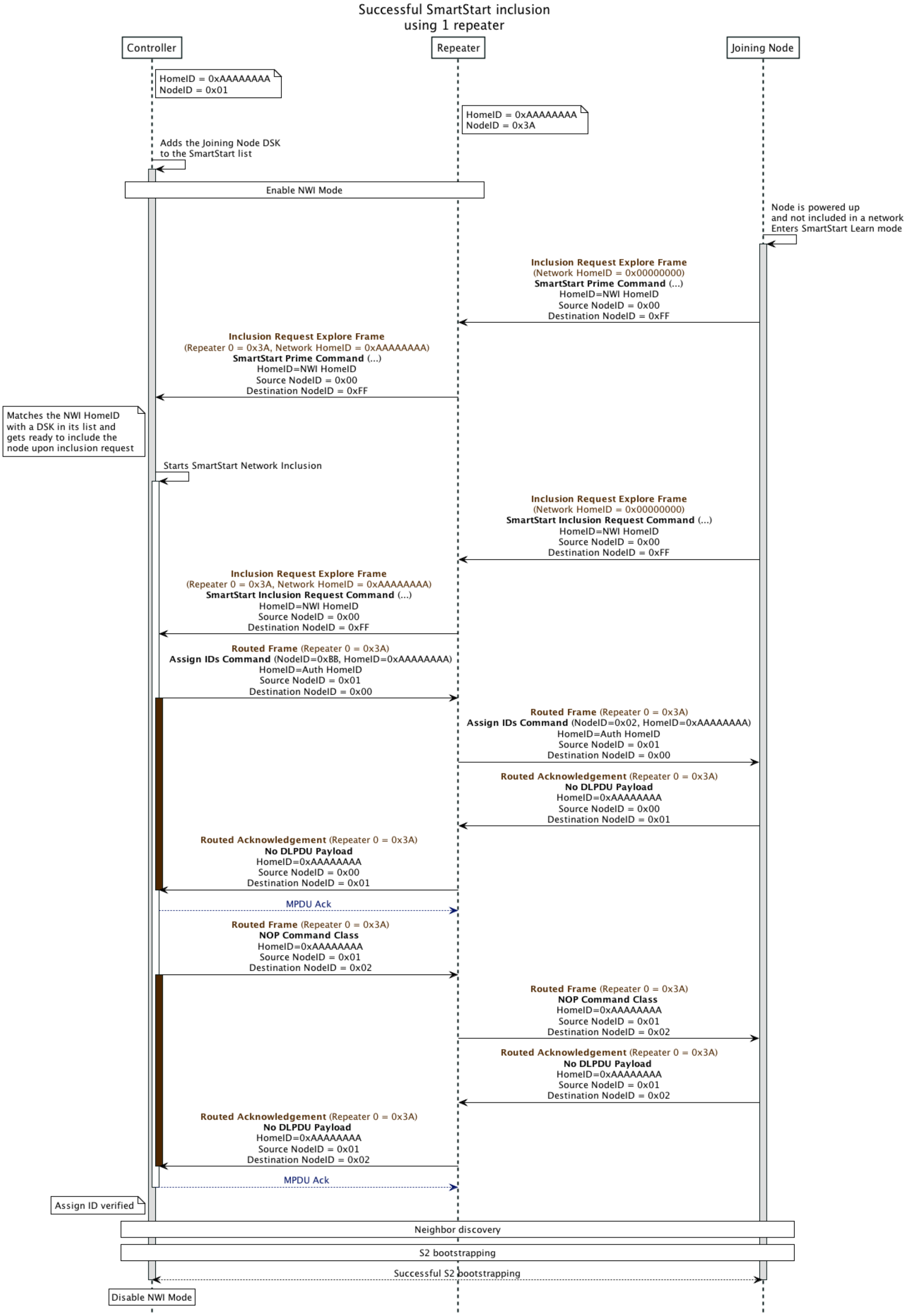

Successful SmartStart inclusion
using 1 repeater
Controller
Repeater
Joining Node
HomeID = 0xAAAAAAAA
NodeID = 0x01
HomeID = 0xAAAAAAAA
NodeID = 0x3A
Adds the Joining Node DSK
to the SmartStart list
Enable NWI Mode
Node is powered up
and not included in a network
Enters SmartStart Learn mode
Inclusion Request Explore Frame
(Network HomeID = 0x00000000)
SmartStart Prime Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFF
Inclusion Request Explore Frame
(Repeater 0 = 0x3A, Network HomeID = 0xAAAAAAAA)
SmartStart Prime Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFF
Matches the NWI HomeID
with a DSK in its list and
gets ready to include the
node upon inclusion request
Starts SmartStart Network Inclusion
Inclusion Request Explore Frame
(Network HomeID = 0x00000000)
SmartStart Inclusion Request Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFF
Inclusion Request Explore Frame
(Repeater 0 = 0x3A, Network HomeID = 0xAAAAAAAA)
SmartStart Inclusion Request Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFF
Routed Frame (Repeater 0 = 0x3A)
Assign IDs Command (NodeID=0xBB, HomeID=0xAAAAAAAA)
HomeID=Auth HomeID
Source NodeID = 0x01
Destination NodeID = 0x00
Routed Frame (Repeater 0 = 0x3A)
Assign IDs Command (NodeID=0x02, HomeID=0xAAAAAAAA)
HomeID=Auth HomeID
Source NodeID = 0x01
Destination NodeID = 0x00
Routed Acknowledgement (Repeater 0 = 0x3A)
No DLPDU Payload
HomeID=0xAAAAAAAA
Source NodeID = 0x00
Destination NodeID = 0x01
Routed Acknowledgement (Repeater 0 = 0x3A)
No DLPDU Payload
HomeID=0xAAAAAAAA
Source NodeID = 0x00
Destination NodeID = 0x01
MPDU Ack
Routed Frame (Repeater 0 = 0x3A)
NOP Command Class
HomeID=0xAAAAAAAA
Source NodeID = 0x01
Destination NodeID = 0x02
Routed Frame (Repeater 0 = 0x3A)
NOP Command Class
HomeID=0xAAAAAAAA
Source NodeID = 0x01
Destination NodeID = 0x02
Routed Acknowledgement (Repeater 0 = 0x3A)
No DLPDU Payload
HomeID=0xAAAAAAAA
Source NodeID = 0x01
Destination NodeID = 0x02
Routed Acknowledgement (Repeater 0 = 0x3A)
No DLPDU Payload
HomeID=0xAAAAAAAA
Source NodeID = 0x01
Destination NodeID = 0x02
MPDU Ack
Assign ID verified
Neighbor discovery
S2 bootstrapping
Successful S2 bootstrapping
Disable NWI Mode

Figure 4.73 Successful SmartStart inclusion (using one repeater)

4.5.4.3.4 Unsuccessful SmartStart inclusion

SmartStart inclusion attempts may be unsuccessful, as the S2 Bootstrapping procedure is added to the inclusion. Any SmartStart inclusion attempt that does not complete with a successful S2 Bootstrapping **shall** be considered as unsuccessful. NWK:01BC.1

Refer to [TECC] for S2 Bootstrapping.

NWK:0201.1 If an error occurred during S2 bootstrapping (S2 bootstrapping started and aborted), the inclusion attempt **shall not** be considered successful.

NWK:0202.1 If S2 bootstrapping did not start, the inclusion attempt **shall not** be considered successful.

NWK:0203.1 If the including controller granted fewer keys than what the joining node requested, the inclusion attempt **shall** be considered successful.

In case of an unsuccessful SmartStart inclusion:

- NWK:01BD.2 The joining node **shall** leave the network automatically and consider itself not included in any network. The joining node **shall** return to SmartStart learn mode.
- The joining node **may** continue with SmartStart learn mode until successful SmartStart inclusion or re-try at least up to 2 times and give up.
- NWK:01BE.1 The including controller **should** consider the joining node removed from the network. It **may** verify if the joining node has left the network properly using the NOP Command Class.

An example is given in Figure 4.74.

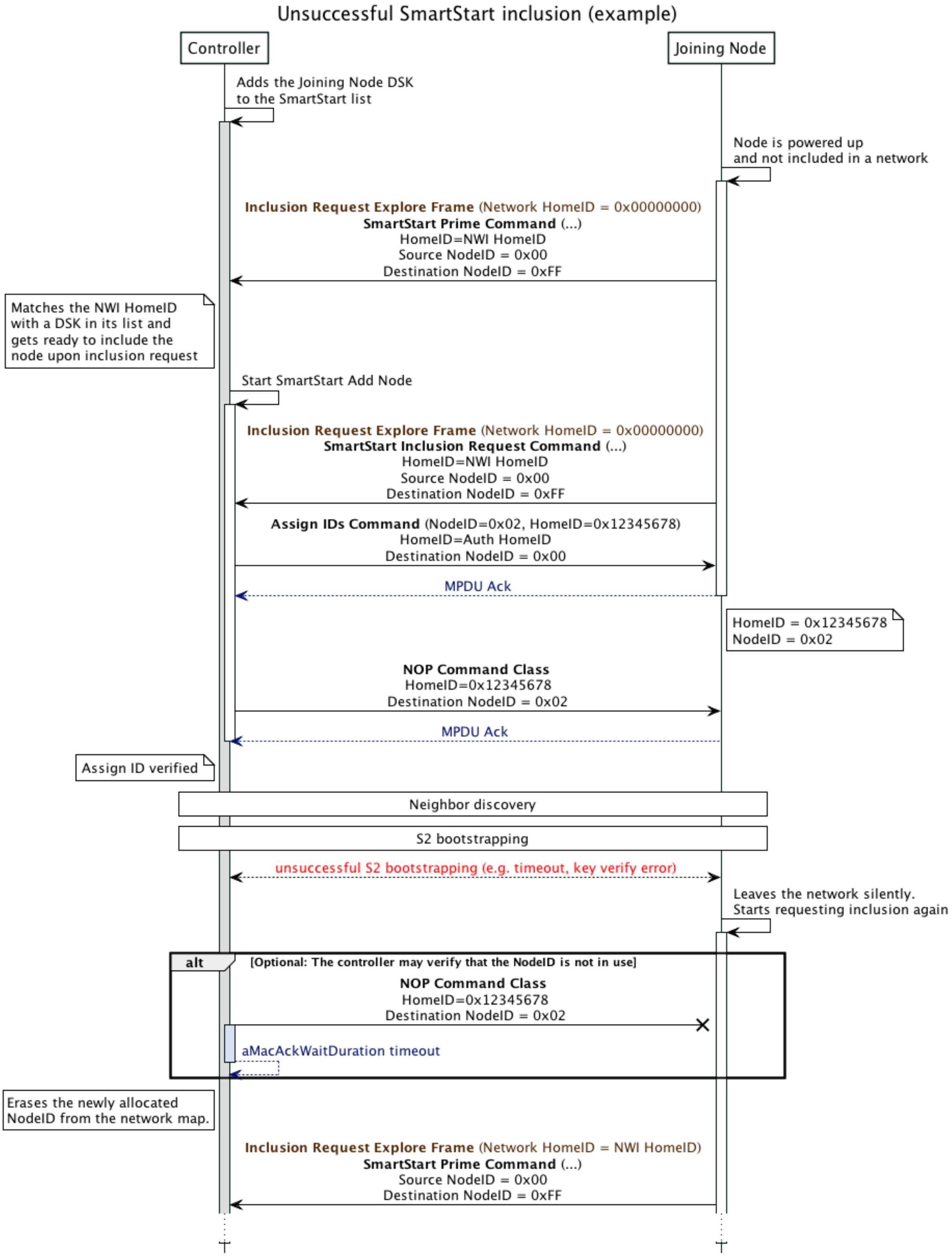

Unsuccessful SmartStart inclusion (example)
Controller
Joining Node
Adds the Joining Node DSK to the SmartStart list
Node is powered up and not included in a network
Inclusion Request Explore Frame (Network HomeID = 0x00000000)
SmartStart Prime Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFF
Matches the NWI HomeID with a DSK in its list and gets ready to include the node upon inclusion request
Start SmartStart Add Node
Inclusion Request Explore Frame (Network HomeID = 0x00000000)
SmartStart Inclusion Request Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFF
Assign IDs Command (NodeID=0x02, HomeID=0x12345678)
HomeID=Auth HomeID
Destination NodeID = 0x00
MPDU Ack
HomeID = 0x12345678
NodeID = 0x02
NOP Command Class
HomeID=0x12345678
Destination NodeID = 0x02
MPDU Ack
Assign ID verified
Neighbor discovery
S2 bootstrapping
unsuccessful S2 bootstrapping (e.g. timeout, key verify error)
Leaves the network silently. Starts requesting inclusion again
alt
[Optional: The controller may verify that the NodeID is not in use]
NOP Command Class
HomeID=0x12345678
Destination NodeID = 0x02
aMacAckWaitDuration timeout
Erases the newly allocated NodeID from the network map.
Inclusion Request Explore Frame (Network HomeID = NWI HomeID)
SmartStart Prime Command (...)
Source NodeID = 0x00
Destination NodeID = 0xFF

Figure 4.74 Unsuccessful SmartStart inclusion example

4.5.5 Network Exclusion

Several procedures can be used for excluding a node from a network. They are:

- Classic Network Exclusion
- Network Wide Exclusion

4.5.5.1 Classic Network Exclusion

NWK:01BF.1 The Classic Network inclusion process will exclude nodes from a network, using direct range communication only. No Routed NPDU or Explore NPDUs **shall** be used in this case.

NWK:01C0.1 The Classic Network Exclusion procedure **shall** be according to Figure 4.75.

NWK:01C1.1 When starting Learn Mode, a node **should** listen for Transfer Presentation Commands with the *Option* field indicating a controller trying to exclude a node. Upon reception, the leaving node **shall** issue a Node Information Frame Command to the NodeID of the controller.

NWK:01C2.1 Alternatively, a node **may** ignore Transfer Presentation Commands and issue a Node Information Frame Command to the broadcast destination NodeID.

NWK:01C3.1 Leaving nodes **shall not** request acknowledgement when issuing the Node Information Frame Command, even if it is issued using singlecast addressing.

NWK:01C4.1 Excluding controllers **should** issue a Transfer End Command, using HomeID 0x00000000 and NodeID 0x00 if they exclude a controller to increase compatibility with old implementations.

NWK:01C5.1 If the excluded node is a controller, it **shall** acknowledge Transfer End Commands issued to HomeID 0x00000000 and NodeID 0x00 shortly after network exclusion.

NWK:01C6.1 End nodes excluded from a network **shall** assume the NodeID 0x00 after exclusion and **shall** assume a new random HomeID (*aNwkRandomHomeID*). The new HomeID **shall** be generated using a random number generator.
Controller nodes excluded from a network **shall** start a new network (refer to 4.5.3 Network Formation).

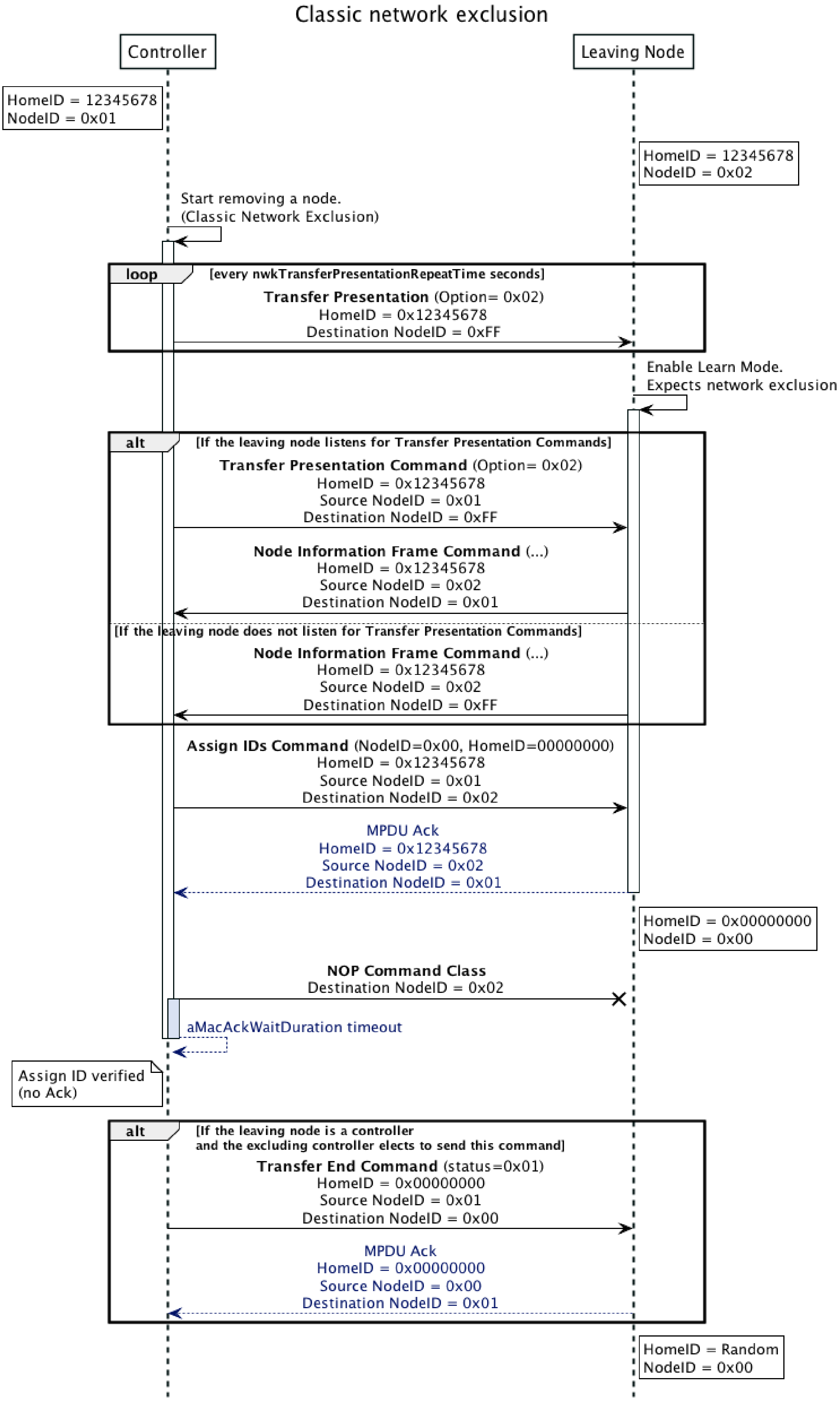

Classic network exclusion
Controller
Leaving Node
HomeID = 12345678
NodeID = 0x01
HomeID = 12345678
NodeID = 0x02
Start removing a node.
(Classic Network Exclusion)
loop
[every nwkTransferPresentationRepeatTime seconds]
Transfer Presentation (Option= 0x02)
HomeID = 0x12345678
Destination NodeID = 0xFF
Enable Learn Mode.
Expects network exclusion
alt
[If the leaving node listens for Transfer Presentation Commands]
Transfer Presentation Command (Option= 0x02)
HomeID = 0x12345678
Source NodeID = 0x01
Destination NodeID = 0xFF
Node Information Frame Command (...)
HomeID = 0x12345678
Source NodeID = 0x02
Destination NodeID = 0x01
[If the leaving node does not listen for Transfer Presentation Commands]
Node Information Frame Command (...)
HomeID = 0x12345678
Source NodeID = 0x02
Destination NodeID = 0xFF
Assign IDs Command (NodeID=0x00, HomeID=00000000)
HomeID = 0x12345678
Source NodeID = 0x01
Destination NodeID = 0x02
MPDU Ack
HomeID = 0x12345678
Source NodeID = 0x02
Destination NodeID = 0x01
HomeID = 0x00000000
NodeID = 0x00
NOP Command Class
Destination NodeID = 0x02
aMacAckWaitDuration timeout
Assign ID verified
(no Ack)
alt
[If the leaving node is a controller
and the excluding controller elects to send this command]
Transfer End Command (status=0x01)
HomeID = 0x00000000
Source NodeID = 0x01
Destination NodeID = 0x00
MPDU Ack
HomeID = 0x00000000
Source NodeID = 0x00
Destination NodeID = 0x01
HomeID = Random
NodeID = 0x00

Figure 4.75 Network Exclusion: Classic Network Exclusion

4.5.5.2 Network Exclusion from a foreign network

NWK:01C7.1 Controllers instructed to remove a node **shall** also remove nodes from foreign networks.

NWK:01C8.1 A controller that has started to remove a node using Classic Exclusion or NWE **shall** return an Assign IDs Command if a Node Information Frame Command has been issued in another HomeID. The Assign IDs Command issued by the excluding controller **shall** be on its own HomeID.

NWK:01C9.1 A node in Learn Mode (exclusion) **shall** accept Assign IDs Command issued in another HomeID.

Exclusions from foreign networks will only work in direct range. Figure 4.76 illustrates a network exclusion in a foreign network.

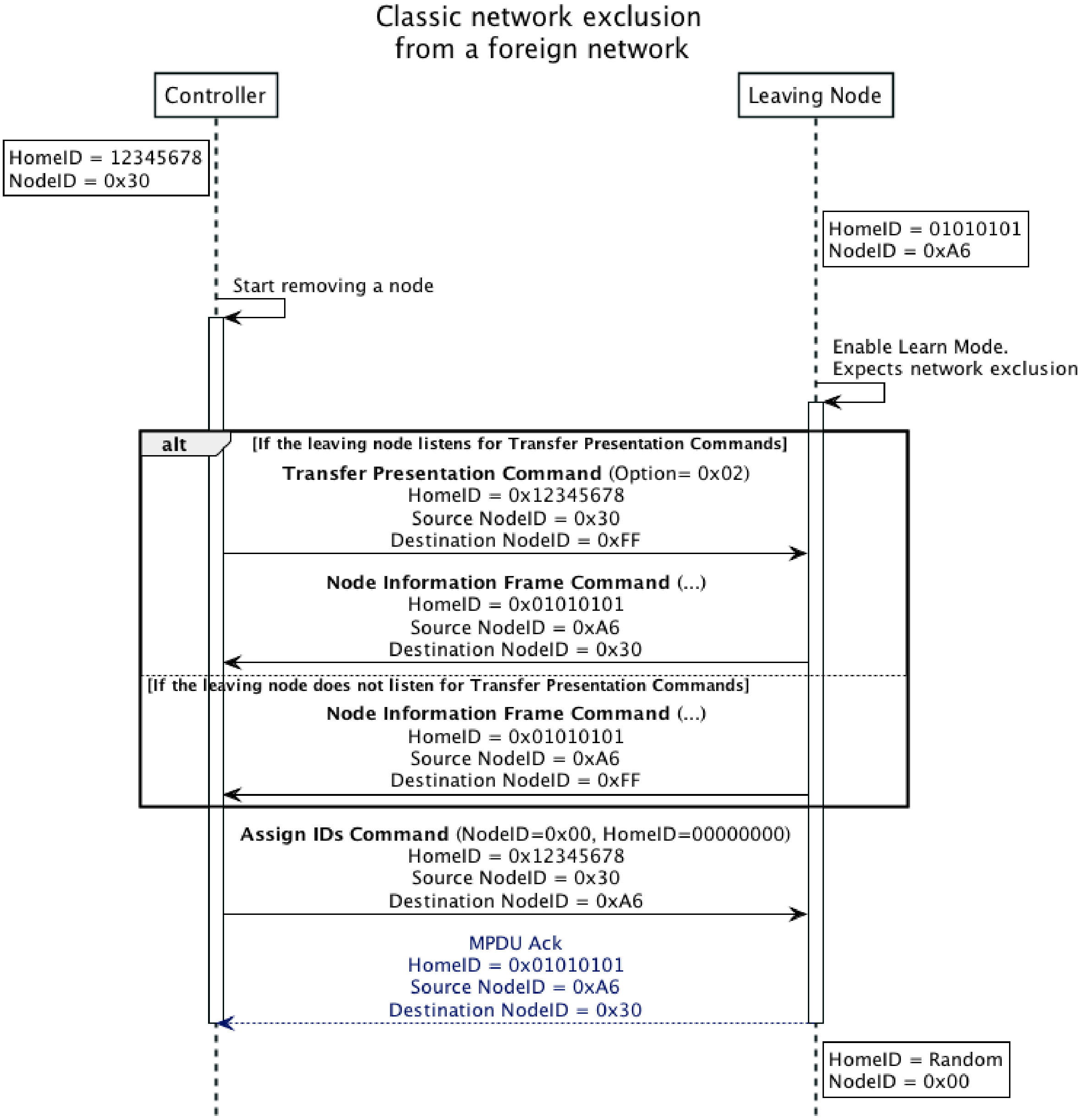

Classic network exclusion
from a foreign network
Controller
Leaving Node
HomeID = 12345678
NodeID = 0x30
HomeID = 01010101
NodeID = 0xA6
Start removing a node
Enable Learn Mode.
Expects network exclusion
alt
[If the leaving node listens for Transfer Presentation Commands]
Transfer Presentation Command (Option= 0x02)
HomeID = 0x12345678
Source NodeID = 0x30
Destination NodeID = 0xFF
Node Information Frame Command (...)
HomeID = 0x01010101
Source NodeID = 0xA6
Destination NodeID = 0x30
[If the leaving node does not listen for Transfer Presentation Commands]
Node Information Frame Command (...)
HomeID = 0x01010101
Source NodeID = 0xA6
Destination NodeID = 0xFF
Assign IDs Command (NodeID=0x00, HomeID=00000000)
HomeID = 0x12345678
Source NodeID = 0x30
Destination NodeID = 0xA6
MPDU Ack
HomeID = 0x01010101
Source NodeID = 0xA6
Destination NodeID = 0x30
HomeID = Random
NodeID = 0x00

Figure 4.76 Network Exclusion from a foreign HomeID

4.5.5.3 Network Wide Exclusion (NWE)

Network-Wide Exclusion (NWE) allows a node to be removed across an existing Z-Wave network without direct range connectivity between the excluding controller and the leaving node.

This procedure leverages Routed NPDUs and Explore NDPUs during the exclusion process.

NWK:01CA.1 The Network Wide Exclusion procedure **shall** be according to Figure 4.77. All frames in Figure 4.77 **shall** use the same HomeID.

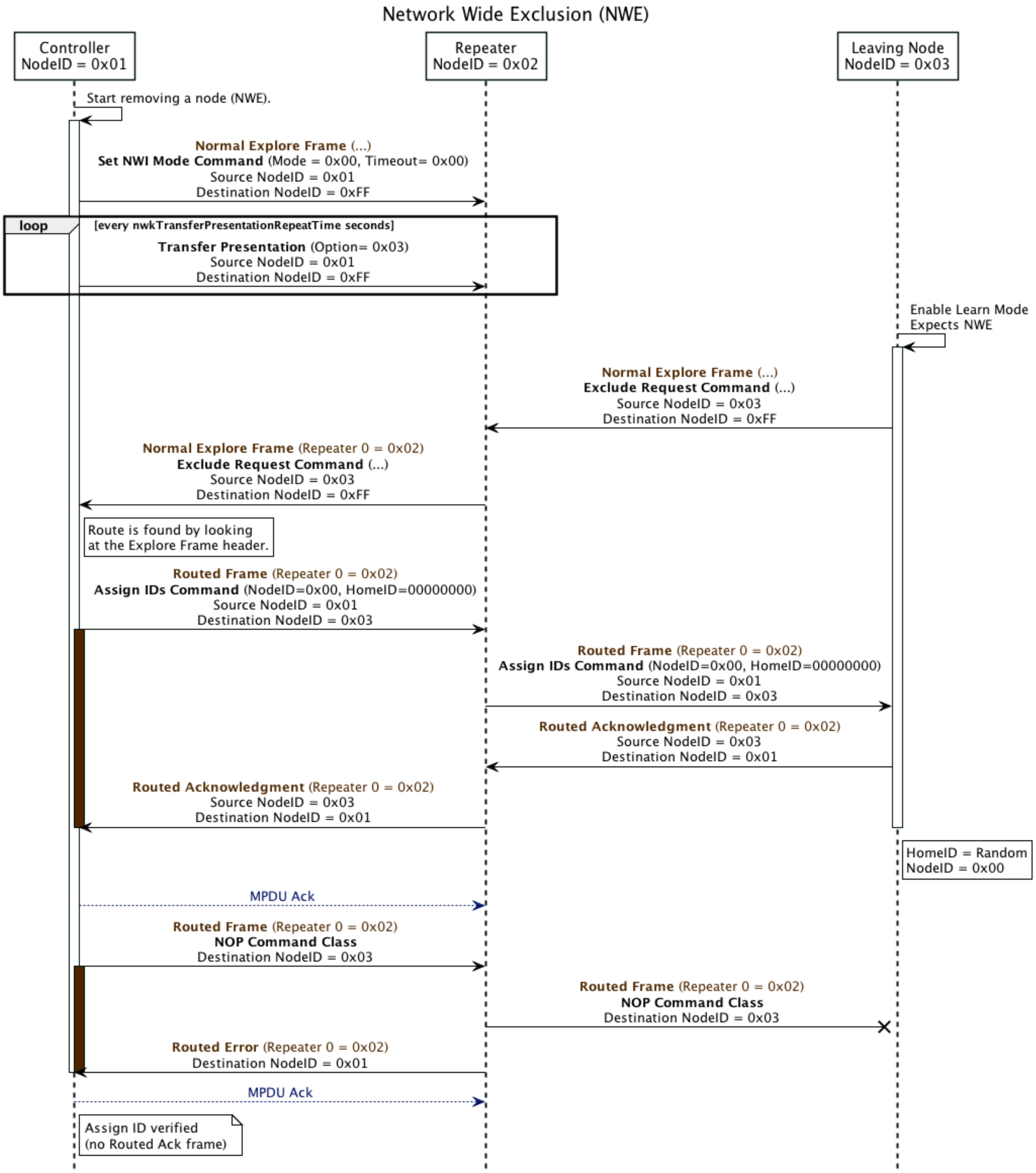

Network Wide Exclusion (NWE)
Controller NodeID = 0x01
Repeater NodeID = 0x02
Leaving Node NodeID = 0x03
Start removing a node (NWE).
Normal Explore Frame (...)
Set NWI Mode Command (Mode = 0x00, Timeout= 0x00)
Source NodeID = 0x01
Destination NodeID = 0xFF
loop
[every nwkTransferPresentationRepeatTime seconds]
Transfer Presentation (Option= 0x03)
Source NodeID = 0x01
Destination NodeID = 0xFF
Enable Learn Mode
Expects NWE
Normal Explore Frame (...)
Exclude Request Command (...)
Source NodeID = 0x03
Destination NodeID = 0xFF
Normal Explore Frame (Repeater 0 = 0x02)
Exclude Request Command (...)
Source NodeID = 0x03
Destination NodeID = 0xFF
Route is found by looking
at the Explore Frame header.
Routed Frame (Repeater 0 = 0x02)
Assign IDs Command (NodeID=0x00, HomeID=00000000)
Source NodeID = 0x01
Destination NodeID = 0x03
Routed Frame (Repeater 0 = 0x02)
Assign IDs Command (NodeID=0x00, HomeID=00000000)
Source NodeID = 0x01
Destination NodeID = 0x03
Routed Acknowledgment (Repeater 0 = 0x02)
Source NodeID = 0x03
Destination NodeID = 0x01
Routed Acknowledgment (Repeater 0 = 0x02)
Source NodeID = 0x03
Destination NodeID = 0x01
HomeID = Random
NodeID = 0x00
MPDU Ack
Routed Frame (Repeater 0 = 0x02)
NOP Command Class
Destination NodeID = 0x03
Routed Frame (Repeater 0 = 0x02)
NOP Command Class
Destination NodeID = 0x03
Routed Error (Repeater 0 = 0x02)
Destination NodeID = 0x01
MPDU Ack
Assign ID verified
(no Routed Ack frame)

Figure 4.77 Network Exclusion: NWE

4.5.6 Failing nodes

NWK:01CB.1 In a Z-Wave network, a node **may** be considered to be failing or non-responsive when a controller cannot reach the node, using routing and explorer frames.

4.5.6.1 Remove a Failing node

4.5.6.1.1 AL and FL nodes

NWK:01CD.1 A Remove Failed Node procedure **may** be used to remove non-responsive nodes from a network.

NWK:01CE.1 Before removing a non-responsive NodeID from a network, a controller **shall** issue NOP commands to the non-responsive NodeID. If the node is not responding, the controller **shall** proceed with removing the NodeID and updating the network.

NWK:01CF.1 A responding node **shall not** be removed from the network by a controller without using Classic Exclusion or Network Wide Exclusion.

NWK:01D0.1 Removing a failing AL or FL node **shall** be according to Figure 4.78

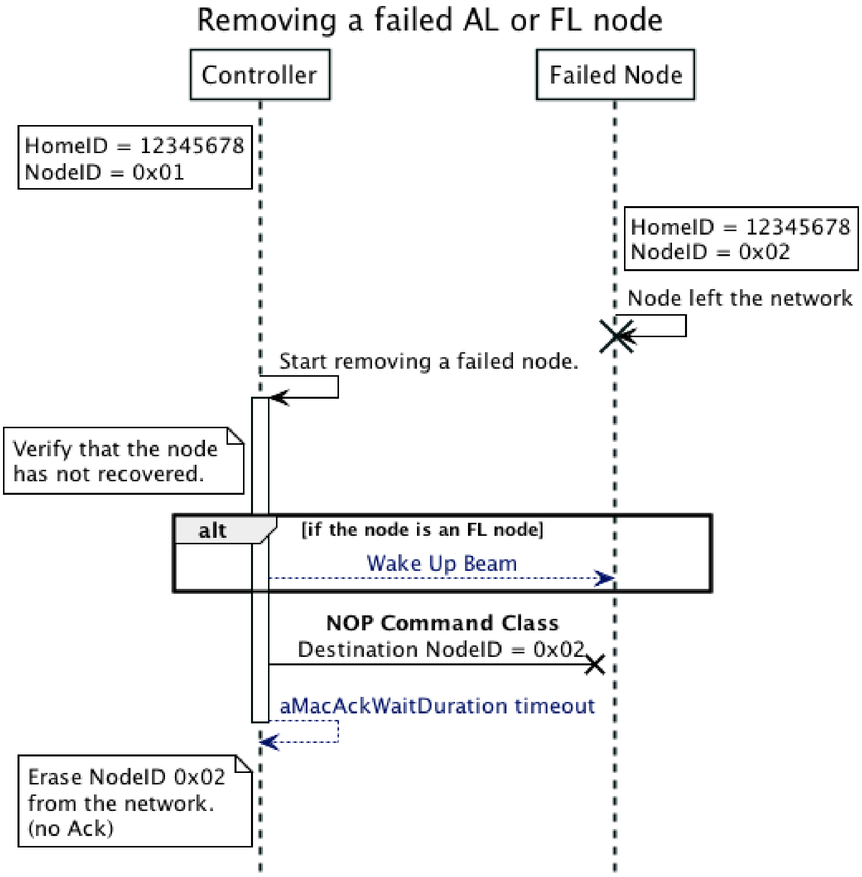

Removing a failed AL or FL node
Controller
Failed Node
HomeID = 12345678
NodeID = 0x01
HomeID = 12345678
NodeID = 0x02
Node left the network
Start removing a failed node.
Verify that the node has not recovered.
alt
[if the node is an FL node]
Wake Up Beam
NOP Command Class
Destination NodeID = 0x02
aMacAckWaitDuration timeout
Erase NodeID 0x02 from the network. (no Ack)

Figure 4.78 Removing a failed AL or FL node from a network

4.5.6.1.2 NL nodes

NWK:01D1.1 NL nodes **should** be considered as failing after missing more than two consecutive Wake Up Periods (no commands were received or transmitted to the node). Refer to the Wake Up Command Class for more details [MCC].

NWK:01D2.1 A controller **may** remove an NL node after any arbitrary duration without receiving any frame from the node.

This is illustrated in Figure 4.79.

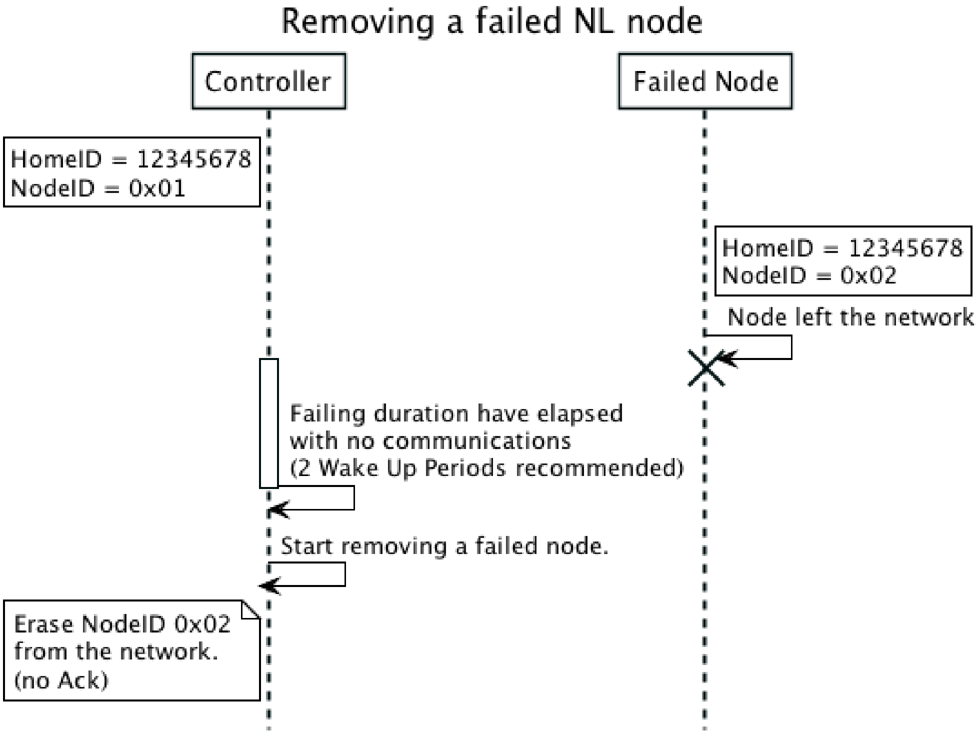


Figure 4.79 Removing a failed NL node from a network

4.5.7 Controller roles

Controllers can change roles in a network using several procedures. The following subsections describe possible transitions and associated procedures.

4.5.7.1 Role transitions

Controllers starting a network are the Primary Controllers for their network. (refer to 4.5.3 Network Formation)

NWK:01D3.1 A controller being included in a network **shall** assume the Secondary Controller role by default.

NWK:01D4.1 A Secondary Controller receiving a Transfer New Primary Controller Complete Command while it is in Learn Mode **shall** assume the Primary Controller role. (refer to 4.5.7.2 Primary Controller shift)

NWK:01D5.1 A Secondary Controller receiving a SUC Node ID Command with the *SUC Capabilities* field bit 0 set to 1 **shall** assume the Inclusion Controller role.

NWK:01D6.1 A controller node receiving a Set SUC Command from the Primary Controller (and accepting it) **shall** assume the SUC/SIS role.

4.5.7.2 Primary Controller shift

NWK:01D7.1 Primary controllers **may** give the Primary Controller role to a Secondary Controller. This procedure **may** be done as part of an inclusion, or with a node that is already part of the network. The procedure **shall** be according to Figure 4.80.

NWK:01D8.1 All controller nodes **shall** accept to become Primary Controller.

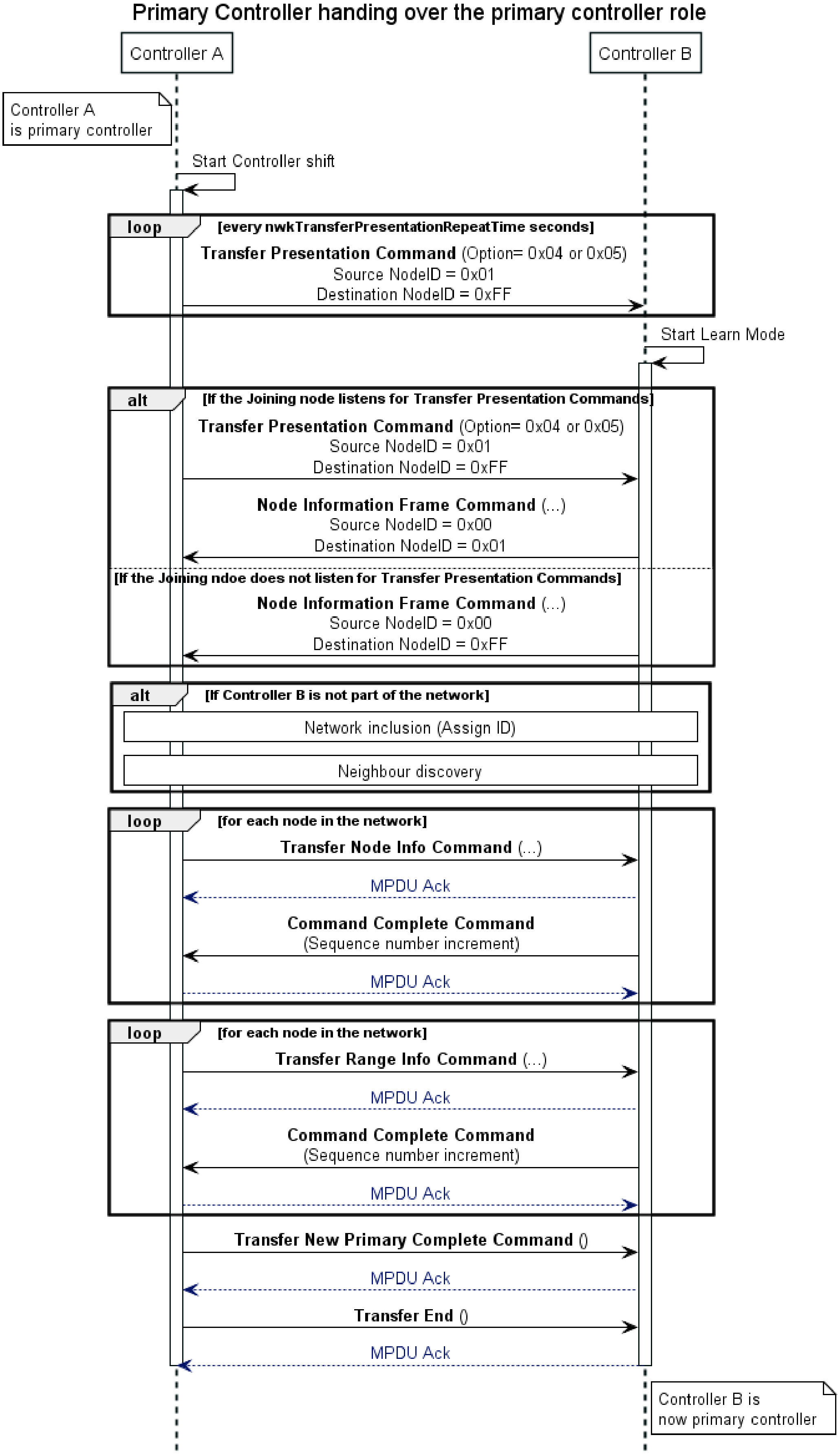

Primary Controller handing over the primary controller role
Controller A
Controller B
Controller A is primary controller
Start Controller shift
loop
[every nwkTransferPresentationRepeatTime seconds]
Transfer Presentation Command (Option= 0x04 or 0x05) Source NodeID = 0x01 Destination NodeID = 0xFF
Start Learn Mode
alt
[If the Joining node listens for Transfer Presentation Commands]
Transfer Presentation Command (Option= 0x04 or 0x05) Source NodeID = 0x01 Destination NodeID = 0xFF
Node Information Frame Command (...) Source NodeID = 0x00 Destination NodeID = 0x01
[If the Joining ndoe does not listen for Transfer Presentation Commands]
Node Information Frame Command (...) Source NodeID = 0x00 Destination NodeID = 0xFF
alt
[If Controller B is not part of the network]
Network inclusion (Assign ID)
Neighbour discovery
loop
[for each node in the network]
Transfer Node Info Command (...)
MPDU Ack
Command Complete Command (Sequence number increment)
MPDU Ack
loop
[for each node in the network]
Transfer Range Info Command (...)
MPDU Ack
Command Complete Command (Sequence number increment)
MPDU Ack
Transfer New Primary Complete Command ()
MPDU Ack
Transfer End ()
MPDU Ack
Controller B is now primary controller

Figure 4.80 Primary Controller – Primary Controller shift

4.5.7.3 Give the SUC/SIS role

NWK:01D9.1 Primary Controllers **may** give the SUC role to another controller by issuing a Set SUC Command. The NWK layer **shall not** perform this procedure automatically. The application layer **shall** be able to instruct the NWK layer to give the SUC/SIS role to another controller.

NWK:01DA.1 Secondary Controllers and Inclusion Controller **shall not** give the SUC role to another controller.

NWK:01DB.1 Primary controllers **shall not** assign the SUC/SIS role if another node already has this role in the network.

NWK:01DC.1 The procedure **shall** be according to Figure 4.81.

NWK:01DD.1 A Primary Controller **shall** take the Inclusion Controller role if the other controller accepted the SUC/SIS role by returning a Set SUC ACK Command with the *State* and *SUC Capabilities* fields set to 0x01.

NWK:01DE.1 Controllers with SUC/SIS capabilities that start a network (first node in the network) **should** assign themselves the SUC/SIS role (silently, without sending commands).
Controllers with SUC/SIS capabilities included in a network that are handed over the Primary Controller role and assigning themselves the SUC/SIS role **should** issue a SUC Node ID Command to all controllers present in the network.

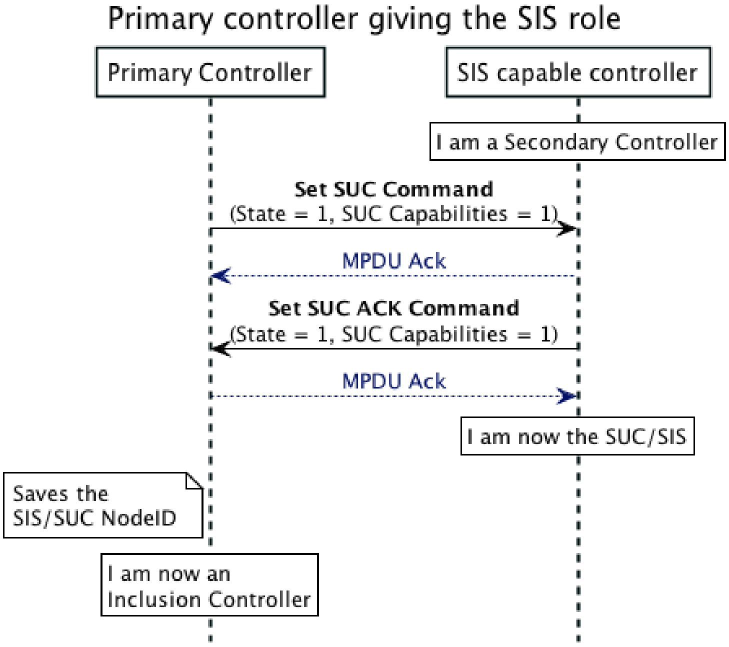

Primary controller giving the SIS role
Primary Controller
SIS capable controller
I am a Secondary Controller
Set SUC Command
(State = 1, SUC Capabilities = 1)
MPDU Ack
Set SUC ACK Command
(State = 1, SUC Capabilities = 1)
MPDU Ack
I am now the SUC/SIS
Saves the SIS/SUC NodeID
I am now an Inclusion Controller

Figure 4.81 Primary controllers - give the SUC/SIS role to another controller

NWK:01DF.1 Controller with no SUC capabilities **shall** decline the SUC/SIS role in the network. This is illustrated in Figure 4.82.

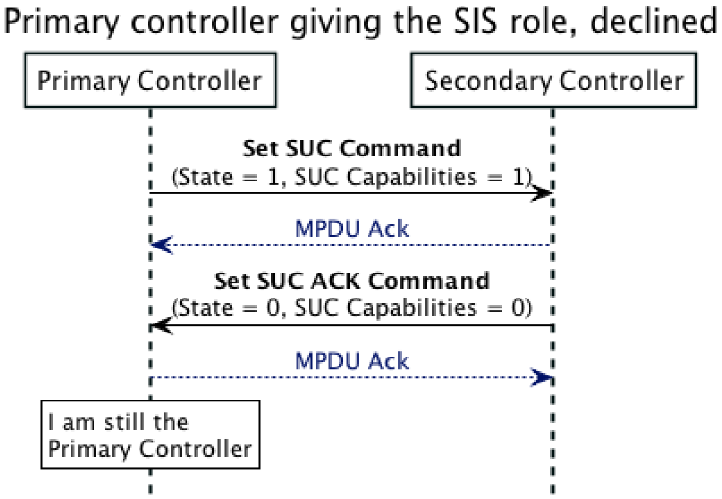

Primary controller giving the SIS role, declined
Primary Controller
Secondary Controller
Set SUC Command
(State = 1, SUC Capabilities = 1)
MPDU Ack
Set SUC ACK Command
(State = 0, SUC Capabilities = 0)
MPDU Ack
I am still the Primary Controller

Figure 4.82 Primary Controllers - Give the SUC/SIS Role to anther controller declined

NWK:01E0.1 Controllers with no SIS capabilities **should** decline the SUC role. However, in the event where a controller only supports the SUC capability, the primary controller **shall** keep the primary controller role. This is illustrated in Figure 4.83.

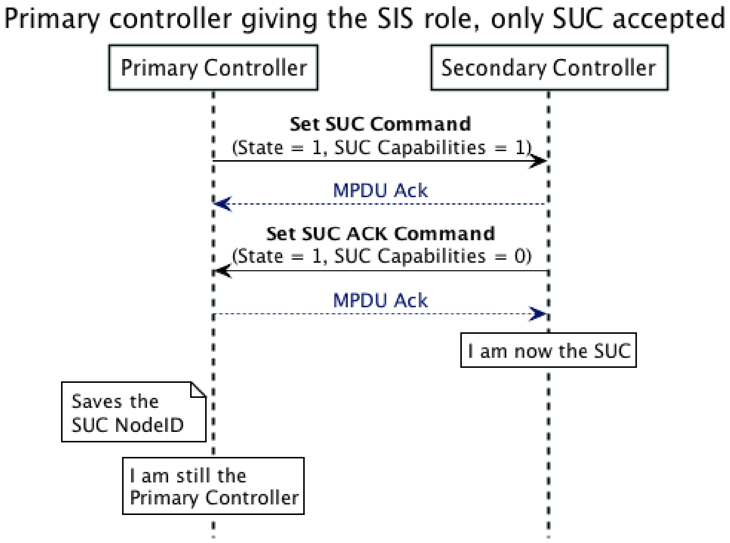

Primary controller giving the SIS role, only SUC accepted
Primary Controller
Secondary Controller
Set SUC Command
(State = 1, SUC Capabilities = 1)
MPDU Ack
Set SUC ACK Command
(State = 1, SUC Capabilities = 0)
MPDU Ack
I am now the SUC
Saves the SUC NodeID
I am still the Primary Controller

Figure 4.83 Primary Controllers - Give the SUC/SIS Role to anther controller partially accepted

4.5.8 Inclusion Controllers functionalities

When a SIS is present in a network, it enables Inclusion Controllers to perform network management operations.

A SUC with SIS capability will enable Inclusion Controller to perform network inclusions and exclusions on its behalf.

Inclusion Controllers network management operations are described in the sections below.

4.5.8.1 Add new nodes on behalf of the SIS

NWK:01E1.1 Inclusion controller **may** add new nodes to a network on behalf of the SIS.

NWK:01E2.1 An Inclusion Controller **shall** make an Automatic Controller Update request to the SIS prior to attempting a network inclusion.

NWK:01E3.1 An Inclusion Controller **shall** request a reserved NodeID, to use for the network inclusion, to the SIS controller prior to attempting a network inclusion.

NWK:01E4.1 An Inclusion Controller **shall** issue a New Node Registered Command and a New Range Registered Command to the SIS after a successful network inclusion.

NWK:01E5.1 A network inclusion by an Inclusion Controller **shall** be according to Figure 4.84

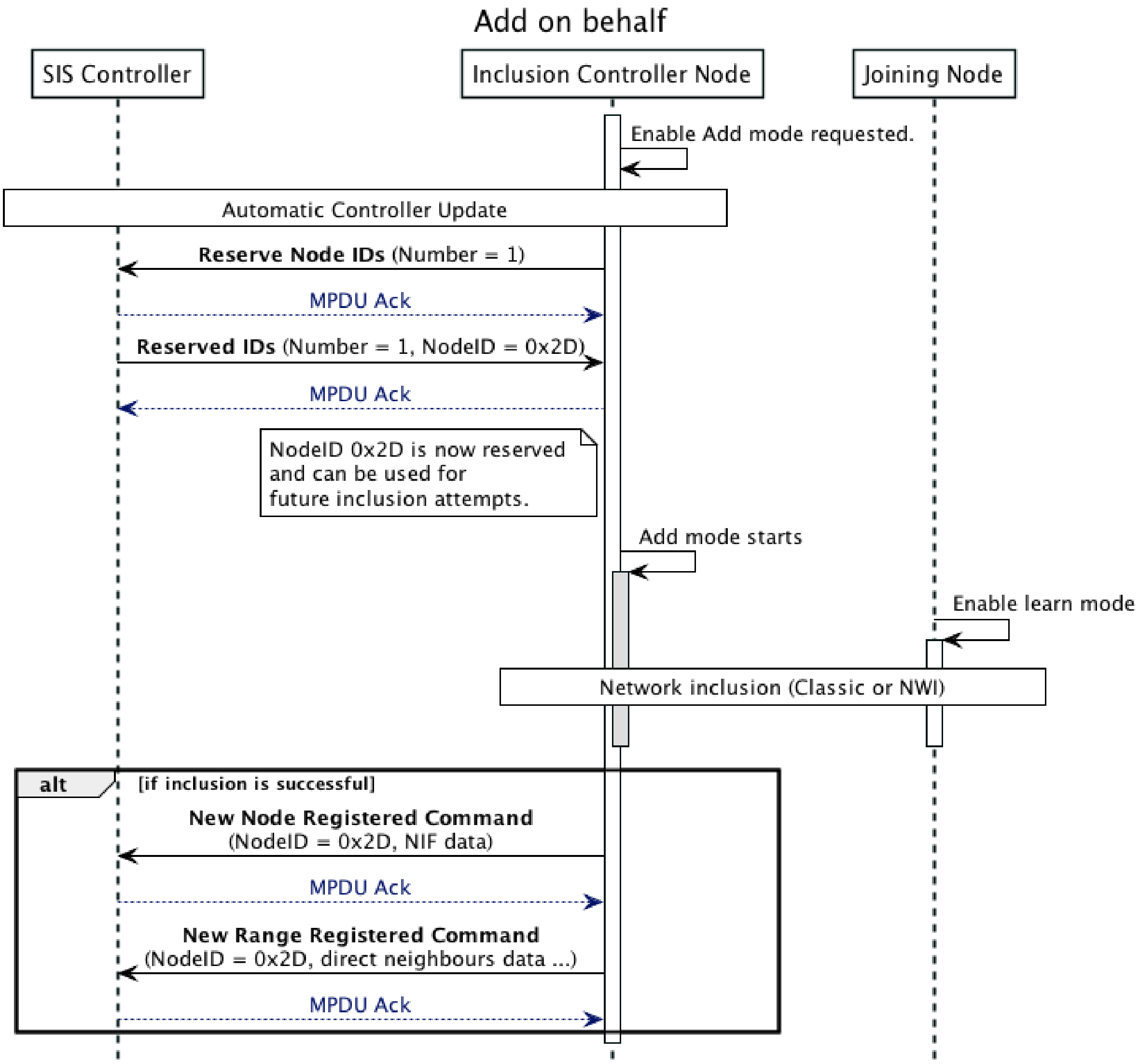

Add on behalf
SIS Controller
Inclusion Controller Node
Joining Node
Enable Add mode requested.
Automatic Controller Update
Reserve Node IDs (Number = 1)
MPDU Ack
Reserved IDs (Number = 1, NodeID = 0x2D)
MPDU Ack
NodeID 0x2D is now reserved and can be used for future inclusion attempts.
Add mode starts
Enable learn mode
Network inclusion (Classic or NWI)
alt
[if inclusion is successful]
New Node Registered Command (NodeID = 0x2D, NIF data)
MPDU Ack
New Range Registered Command (NodeID = 0x2D, direct neighbours data ...)
MPDU Ack

Figure 4.84 Inclusion Controllers - Add on behalf of the SIS

4.5.8.2 Remove nodes on behalf of the SIS

NWK:01E6.1 Inclusion controller **may** remove nodes from a network on behalf of the SIS.

NWK:01E7.1 An inclusion controller **shall** issue a New Node Registered Command to the SIS after a successful network exclusion of a node that belongs to the same network.

NWK:01E8.1 An inclusion controller **shall not** issue a New Node Registered Command if excluding a node from a foreign network.

NWK:01E9.1 A network exclusion by an Inclusion Controller **shall** be according to Figure 4.85.

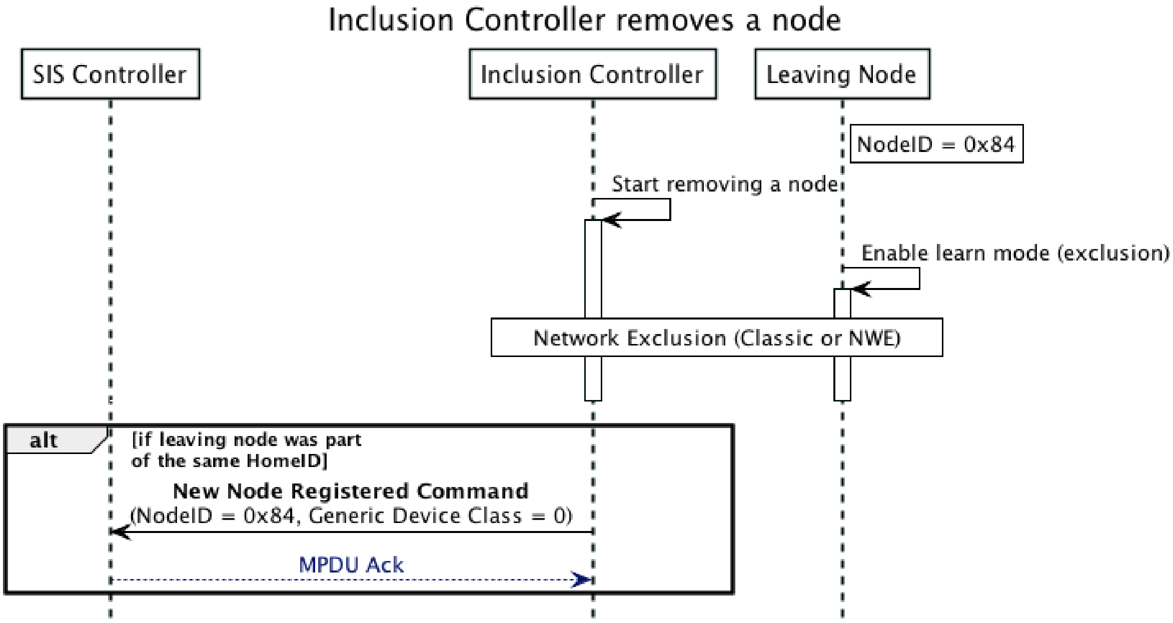

Inclusion Controller removes a node
SIS Controller
Inclusion Controller
Leaving Node
NodeID = 0x84
Start removing a node
Enable learn mode (exclusion)
Network Exclusion (Classic or NWE)
alt
[if leaving node was part of the same HomeID]
New Node Registered Command (NodeID = 0x84, Generic Device Class = 0)
MPDU Ack

Figure 4.85 Network Exclusion by an Inclusion Controller

4.5.9 Network maintenance

4.5.9.1 Automatic Controller Update

NWK:01EA.1 In a Z-Wave network, the SUC **shall** store the network topology and it **shall** keep it updated when new nodes are added/removed.

NWK:01CC.1 The SUC **shall** deliver the network topology to any nodes that request a network topology update.

NWK:01EB.1 A non-SUC controller's network topology is dated from last time a node was included or it requested a network update from the SUC. To get updated, the Inclusion Controller **may** request updated network topology using the Automatic Controller Update process.

NWK:01EC.1 The SUC **shall** provide the network topology information to controllers when requested. The SUC controller may provide this information in two ways:

1. It **may** transfer all available network topology information.
2. It **may** transfer the list of changes since the last time the specific controller requested the network topology.

NWK:01ED.1 During Automatic Controller Updates, the requesting controller **shall not** erase network topology, but instead it **shall** update the topology using the data provided by the SUC.

NWK:01EE.1 An Inclusion Controller **shall** send an Automatic Controller Update Start Command to the SUC to start an Automatic Controller Update procedure. The SUC **shall** send the topology information to the controller using New Node Registered Commands, New Range Registered Commands. A Nodes Exist Command **shall** be sent if performing a full update.

NWK:01EF.1 The Automatic Controller Update procedure **shall** be according to Figure 4.86 or Figure 4.87.

NWK:01F0.1 The application layer **shall** be able to instruct the NWK layer to trigger an Automatic Controller Update.

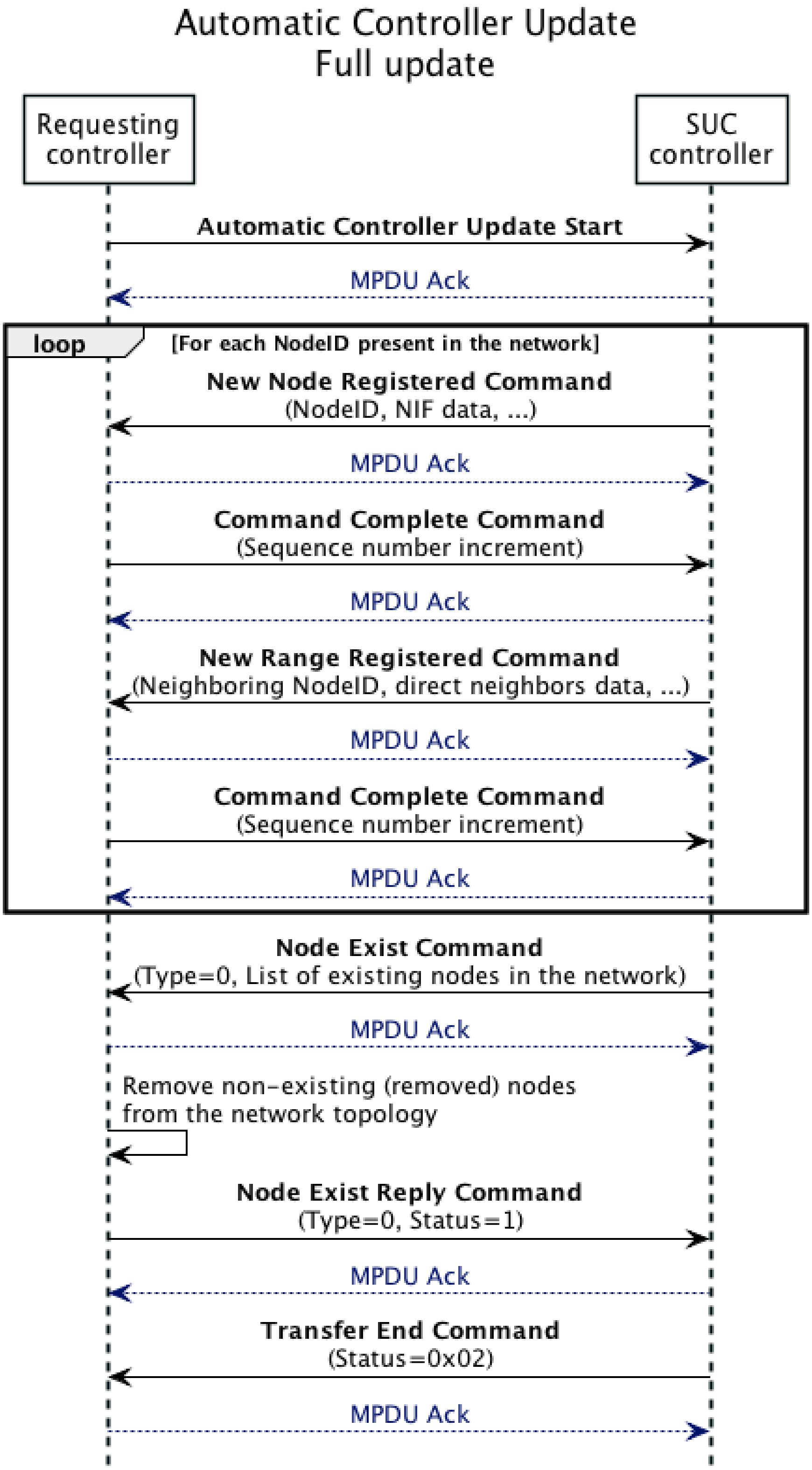

Automatic Controller Update
Full update
Requesting controller
SUC controller
Automatic Controller Update Start
MPDU Ack
loop
[For each NodeID present in the network]
New Node Registered Command
(NodeID, NIF data, ...)
MPDU Ack
Command Complete Command
(Sequence number increment)
MPDU Ack
New Range Registered Command
(Neighboring NodeID, direct neighbors data, ...)
MPDU Ack
Command Complete Command
(Sequence number increment)
MPDU Ack
Node Exist Command
(Type=0, List of existing nodes in the network)
MPDU Ack
Remove non-existing (removed) nodes
from the network topology
Node Exist Reply Command
(Type=0, Status=1)
MPDU Ack
Transfer End Command
(Status=0x02)
MPDU Ack

Figure 4.86 Automatic Controller Update (full update)

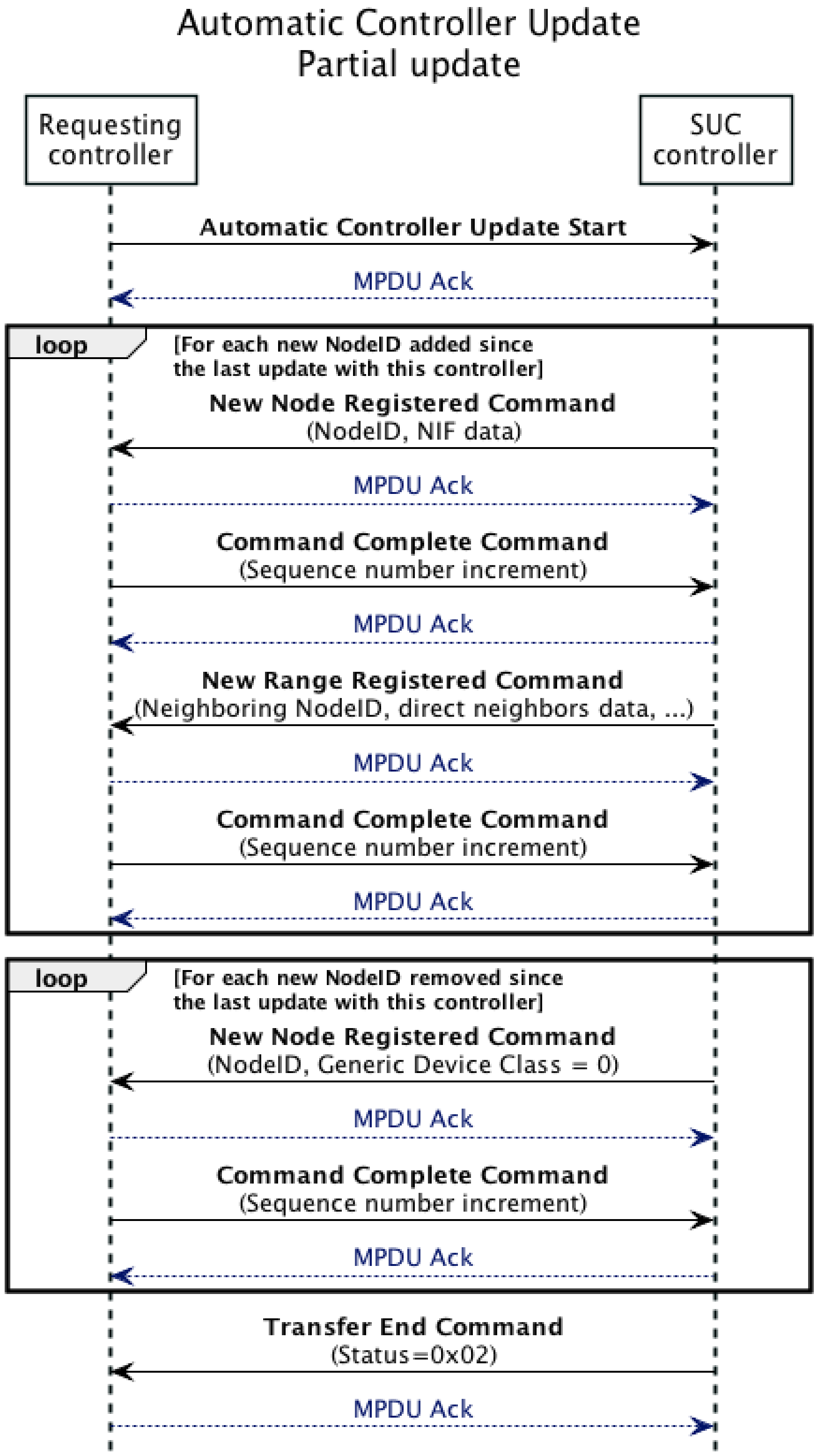

Automatic Controller Update
Partial update
Requesting controller
SUC controller
Automatic Controller Update Start
MPDU Ack
loop
[For each new NodeID added since the last update with this controller]
New Node Registered Command (NodeID, NIF data)
MPDU Ack
Command Complete Command (Sequence number increment)
MPDU Ack
New Range Registered Command (Neighboring NodeID, direct neighbors data, ...)
MPDU Ack
Command Complete Command (Sequence number increment)
MPDU Ack
loop
[For each new NodeID removed since the last update with this controller]
New Node Registered Command (NodeID, Generic Device Class = 0)
MPDU Ack
Command Complete Command (Sequence number increment)
MPDU Ack
Transfer End Command (Status=0x02)
MPDU Ack

Figure 4.87 Automatic Controller Update (Partial update)

4.5.9.2 SUC updates by the Primary Controller

NWK:01F1.1 In a network where the SUC is not the Primary Controller, Primary Controllers **shall** update the SUC with the updated network topology after including or excluding a node.

NWK:01F2.1 A Primary Controller **shall** issue a New Node Registered Command and a New Range Registered Command to the SUC after a successful network inclusion.

The network inclusion update **shall** be according to Figure 4.88.

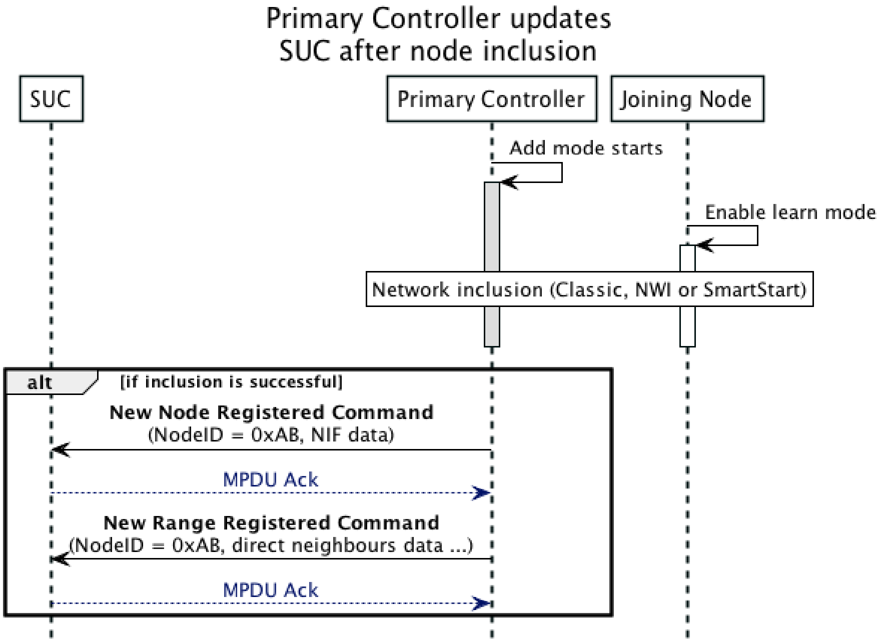

Primary Controller updates
SUC after node inclusion
SUC
Primary Controller
Joining Node
Add mode starts
Enable learn mode
Network inclusion (Classic, NWI or SmartStart)
alt
[if inclusion is successful]
New Node Registered Command
(NodeID = 0xAB, NIF data)
MPDU Ack
New Range Registered Command
(NodeID = 0xAB, direct neighbours data ...)
MPDU Ack

Figure 4.88 Primary Controller includes a node and notifies the SUC

NWK:01F3.1 A Primary Controller **shall** issue a New Node Registered Command to the SUC after a successful network exclusion of a node that belongs to the same network.

NWK:01F4.1 A Primary Controller **shall not** issue a New Node Registered Command if excluding a node from a foreign network.

NWK:01F5.1 The network exclusion update **shall** be according to Figure 4.89.

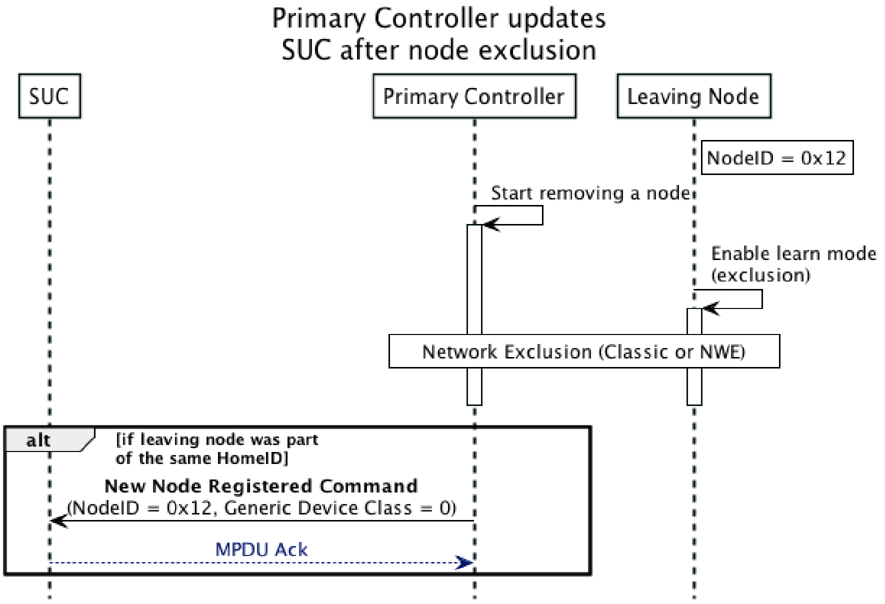

Primary Controller updates
SUC after node exclusion
SUC
Primary Controller
Leaving Node
NodeID = 0x12
Start removing a node
Enable learn mode
(exclusion)
Network Exclusion (Classic or NWE)
alt
[if leaving node was part
of the same HomeID]
New Node Registered Command
(NodeID = 0x12, Generic Device Class = 0)
MPDU Ack

Figure 4.89 Primary Controller excludes a node and notifies the SUC

4.5.9.3 Controller Replication

A Controller Replication is a procedure that can be initiated by a Primary Controller and intents to:

- Optionally include a new controller into the network
- Transfer the network topology to another controller.

NWK:01F6.1 The controller receiving the Controller Replication **shall** be in Learn Mode to accept initiating the procedure.

NWK:01F7.1 The Controller Replication procedure **shall** be as illustrated in Figure 4.90.

Controller replication

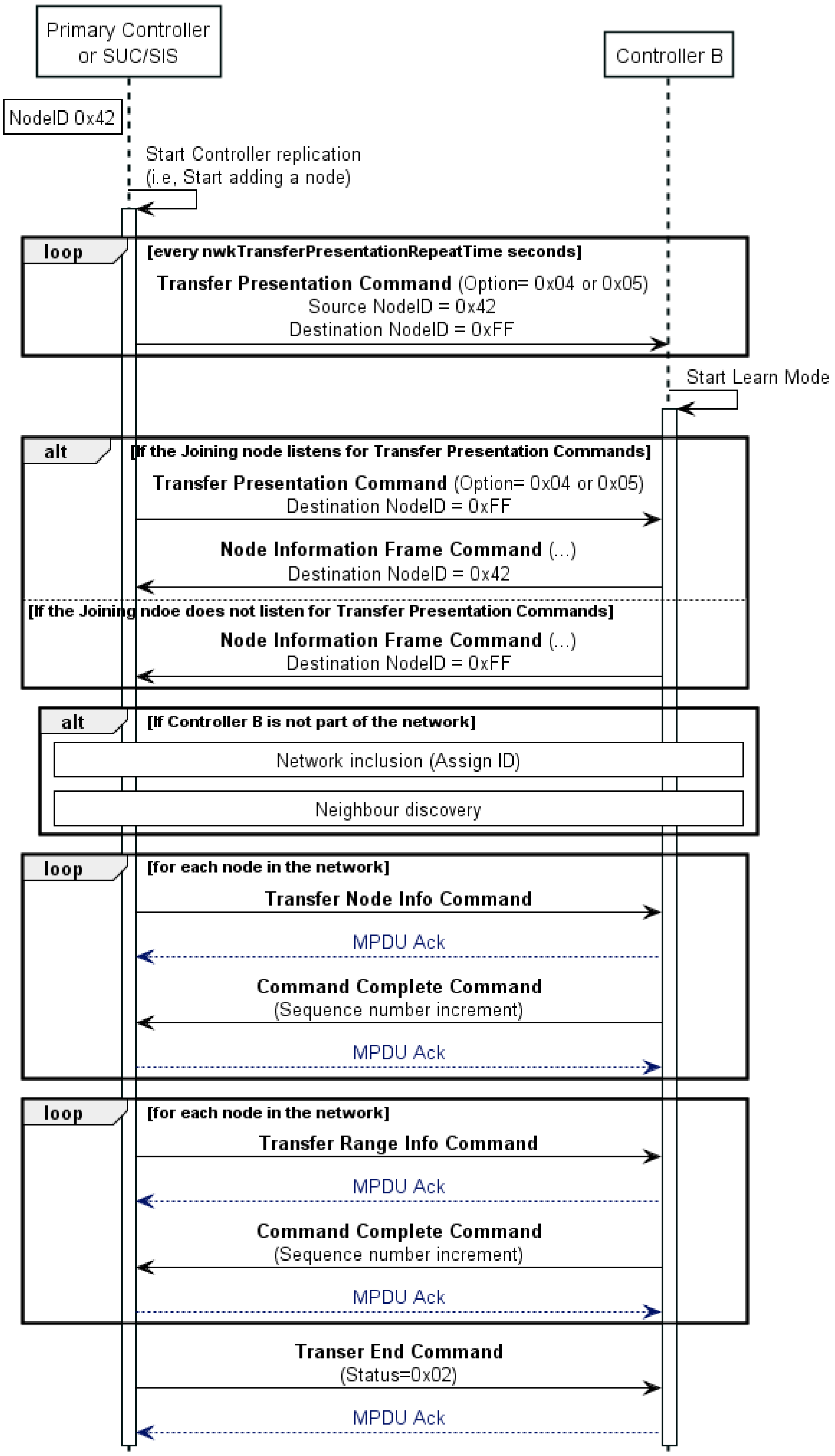

Primary Controller or SUC/SIS
Controller B
NodeID 0x42
Start Controller replication (i.e, Start adding a node)
loop
[every nwkTransferPresentationRepeatTime seconds]
Transfer Presentation Command (Option= 0x04 or 0x05)
Source NodeID = 0x42
Destination NodeID = 0xFF
Start Learn Mode
alt
[If the Joining node listens for Transfer Presentation Commands]
Transfer Presentation Command (Option= 0x04 or 0x05)
Destination NodeID = 0xFF
Node Information Frame Command (...)
Destination NodeID = 0x42
[If the Joining ndoe does not listen for Transfer Presentation Commands]
Node Information Frame Command (...)
Destination NodeID = 0xFF
alt
[If Controller B is not part of the network]
Network inclusion (Assign ID)
Neighbour discovery
loop
[for each node in the network]
Transfer Node Info Command
MPDU Ack
Command Complete Command
(Sequence number increment)
MPDU Ack
loop
[for each node in the network]
Transfer Range Info Command
MPDU Ack
Command Complete Command
(Sequence number increment)
MPDU Ack
Transer End Command
(Status=0x02)
MPDU Ack

Figure 4.90 Network Maintenance - Controller Replication

4.5.9.4 Neighbour Discovery / Range test

This procedure is used to request a node to perform a discovery of its direct range neighbours.

NWK:01F8.1 A SIS/SUC or Inclusion Controller **shall** perform a Neighbour Discovery as part of the network inclusion (refer to 4.5.3).

NWK:01F9.1 A Primary or SUC/SIS controller **may** perform a Neighbour Discovery periodically to keep the network topology accurate.

NWK:01FA.1 The Neighbour Discovery procedure **shall** be as illustrated in Figure 4.91 or Figure 4.92.

NWK:01FB.1 250ms FL nodes and 1000ms FL nodes **may** be included in the same Find Nodes In Range Command if the *Wake Up Time* field is set to 0x01 (1000ms). (250ms FL nodes will wake up with a 1000ms beam). This is shown in Figure 4.92

NWK:01FC.1 FL nodes **may** be asked to find other FL nodes in the neighbor discovery process, but it is not required as FL nodes are not acting as repeaters.

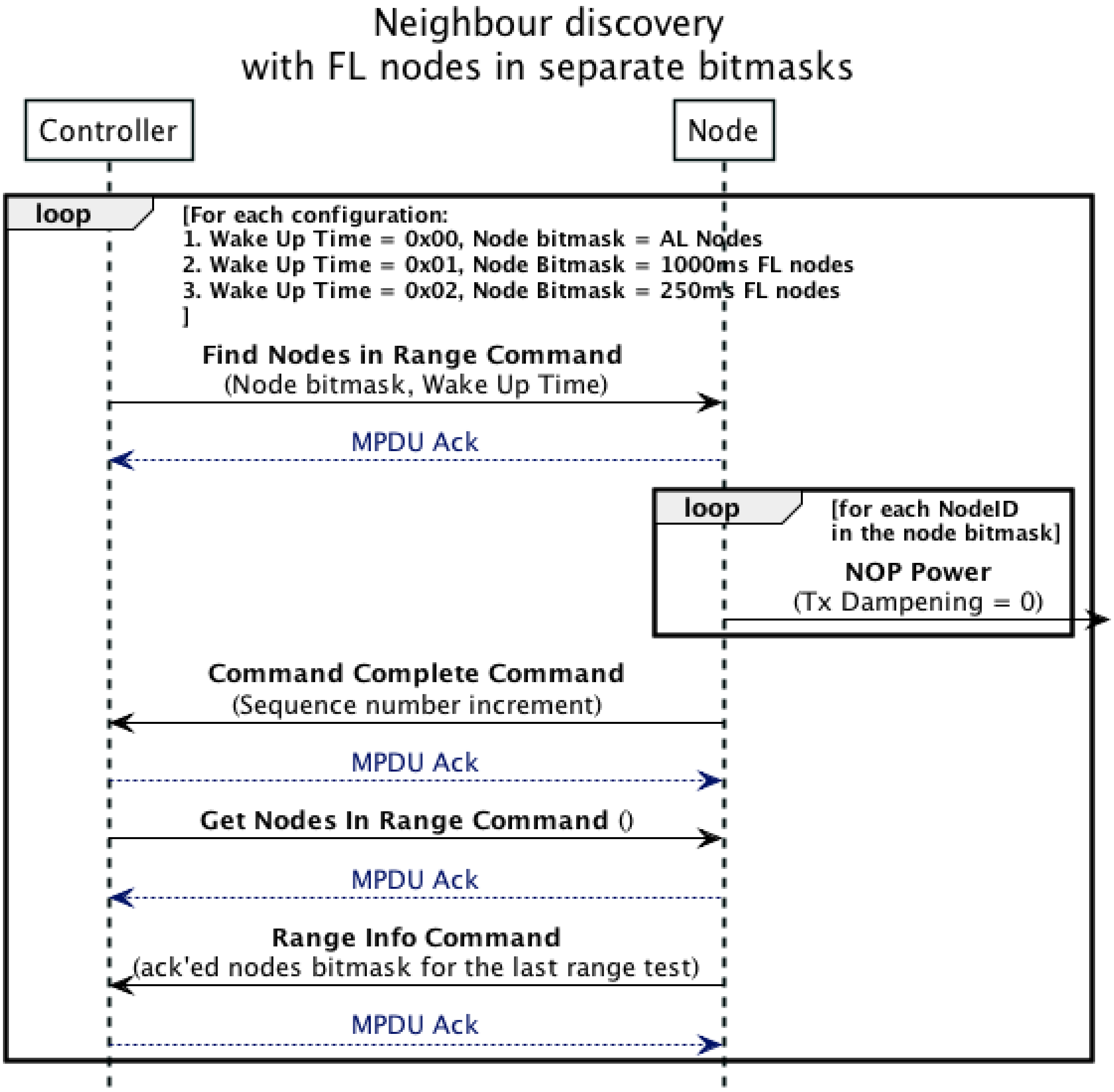

Neighbour discovery
with FL nodes in separate bitmasks
Controller
Node
loop
[For each configuration:
1. Wake Up Time = 0x00, Node bitmask = AL Nodes
2. Wake Up Time = 0x01, Node Bitmask = 1000ms FL nodes
3. Wake Up Time = 0x02, Node Bitmask = 250ms FL nodes
]
Find Nodes in Range Command
(Node bitmask, Wake Up Time)
MPDU Ack
loop
[for each NodeID
in the node bitmask]
NOP Power
(Tx Dampening = 0)
Command Complete Command
(Sequence number increment)
MPDU Ack
Get Nodes In Range Command ()
MPDU Ack
Range Info Command
(ack'ed nodes bitmask for the last range test)
MPDU Ack

Figure 4.91 Network Maintenance - Neighbour discovery (FL nodes in separate bitmasks)

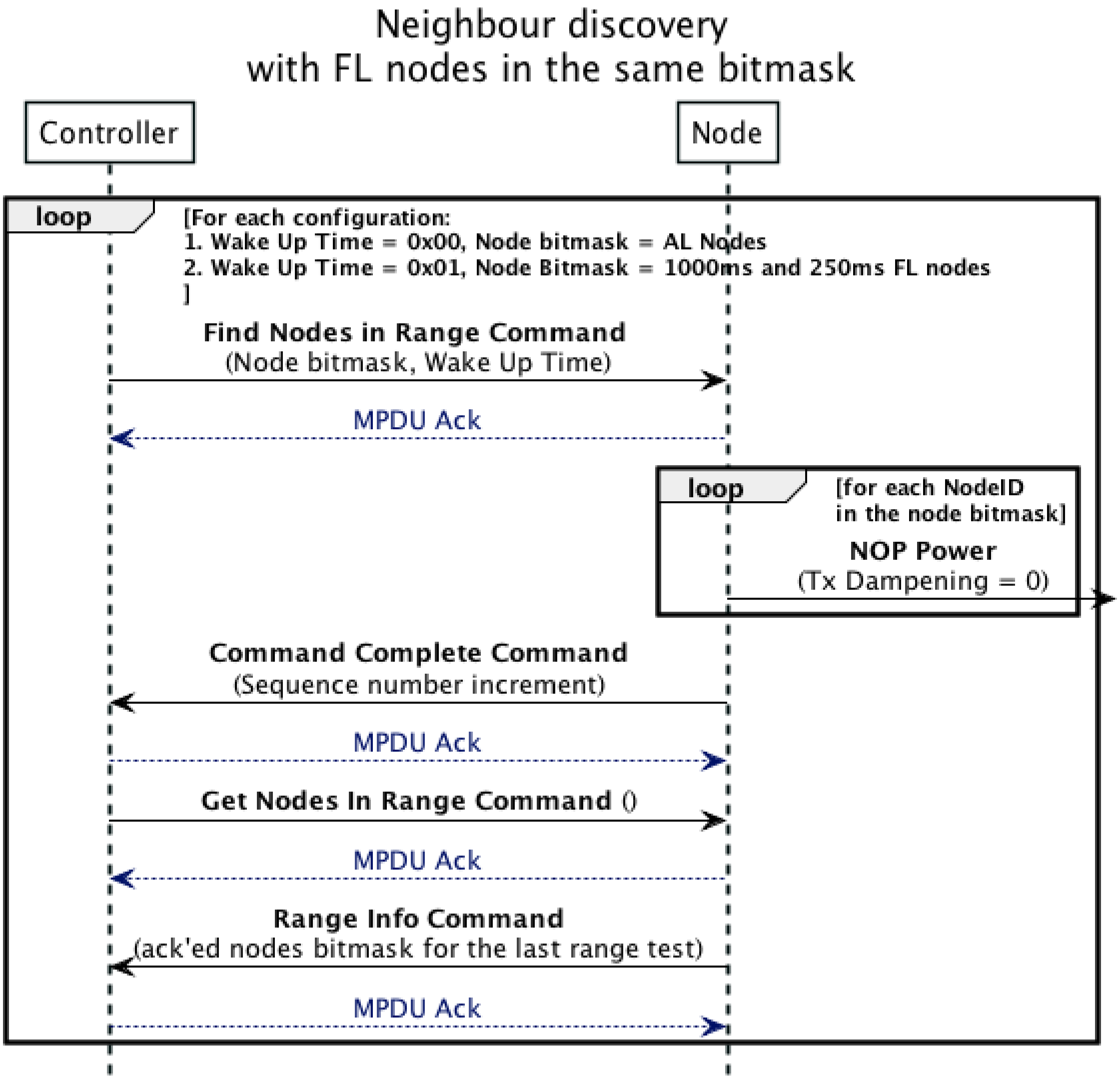

Neighbour discovery
with FL nodes in the same bitmask
Controller
Node
loop
[For each configuration:
1. Wake Up Time = 0x00, Node bitmask = AL Nodes
2. Wake Up Time = 0x01, Node Bitmask = 1000ms and 250ms FL nodes
]
Find Nodes in Range Command
(Node bitmask, Wake Up Time)
MPDU Ack
loop
[for each NodeID
in the node bitmask]
NOP Power
(Tx Dampening = 0)
Command Complete Command
(Sequence number increment)
MPDU Ack
Get Nodes In Range Command ()
MPDU Ack
Range Info Command
(ack'ed nodes bitmask for the last range test)
MPDU Ack

Figure 4.92 Network Maintenance - Neighbour discovery (FL nodes in the same bitmask)

4.5.9.5 End node route request

NWK:01FC.1 End nodes **should not** try to calculate routes automatically, as they do not have a full network topology available.

NWK:01FD.1 End nodes **may** request network update information to the SUC using the Static Route Request Command.

End nodes learn about the identity of the SUC node when they receive an Assign SUC Return Route Command or Assign SUC Return Route Priority Command.

NWK:01FE.1 The process of end node route request **shall** follow the procedure illustrated in Figure 4.93. The SUC Controller **should** assign *nwkRecommendedNumberOfReturnRoutes* return routes for each NodeID destination.

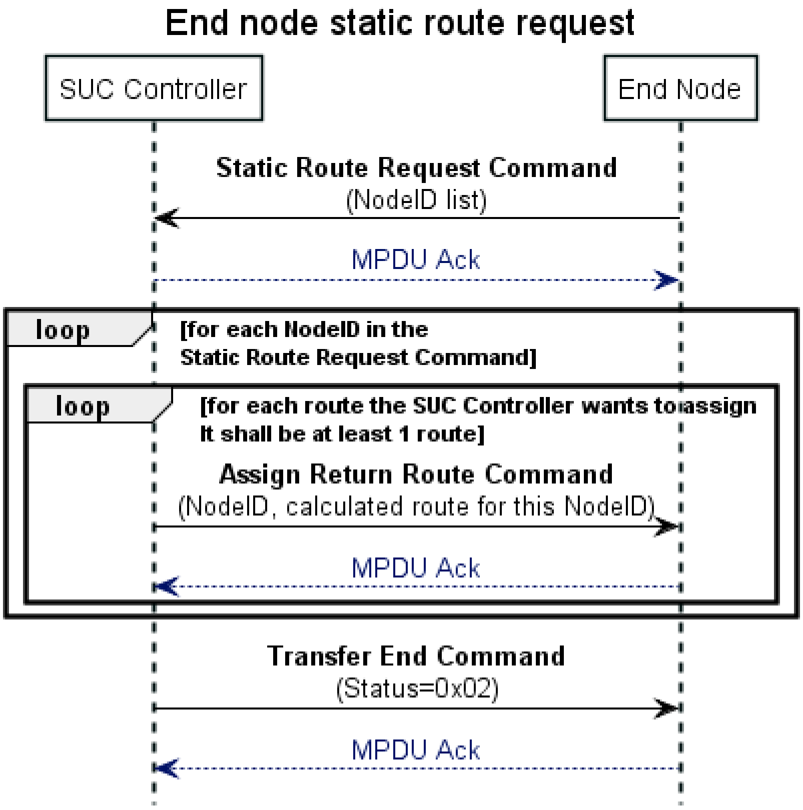

End node static route request
SUC Controller
End Node
Static Route Request Command
(NodeID list)
MPDU Ack
loop
[for each NodeID in the
Static Route Request Command]
loop
[for each route the SUC Controller wants to assign
It shall be at least 1 route]
Assign Return Route Command
(NodeID, calculated route for this NodeID)
MPDU Ack
Transfer End Command
(Status=0x02)
MPDU Ack

Figure 4.93 Network Maintenance - End node Route Request process

If the SUC is busy carrying another network operation, it **may** return a Transfer End Command with the *Status* field set 0x04. This is illustrated in Figure 4.94

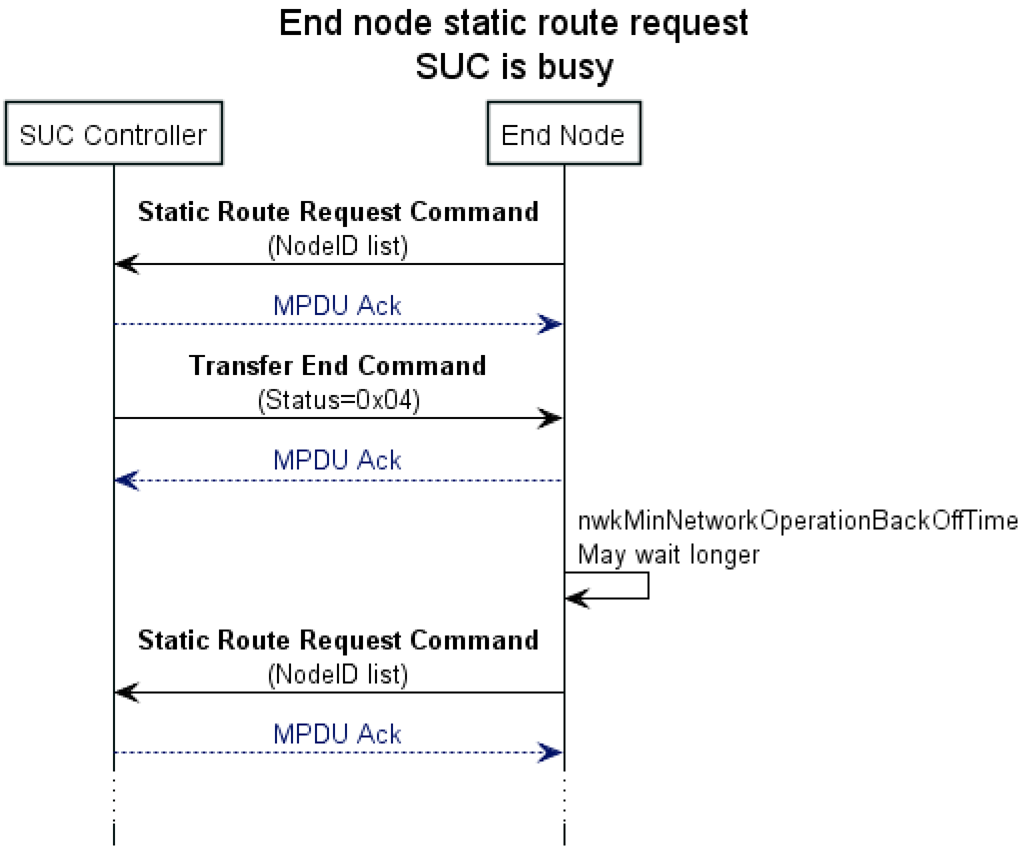


Figure 4.94 - Network Maintenance - Static Route request with SUC busy

5 Z-WAVE LONG RANGE PROTOCOL OVERVIEW

5.1 The Z-Wave Long Range protocol stack architecture

The Z-Wave Long Range protocol stack is similar to the Z-Wave protocol stack. This is illustrated in Figure 5.1.

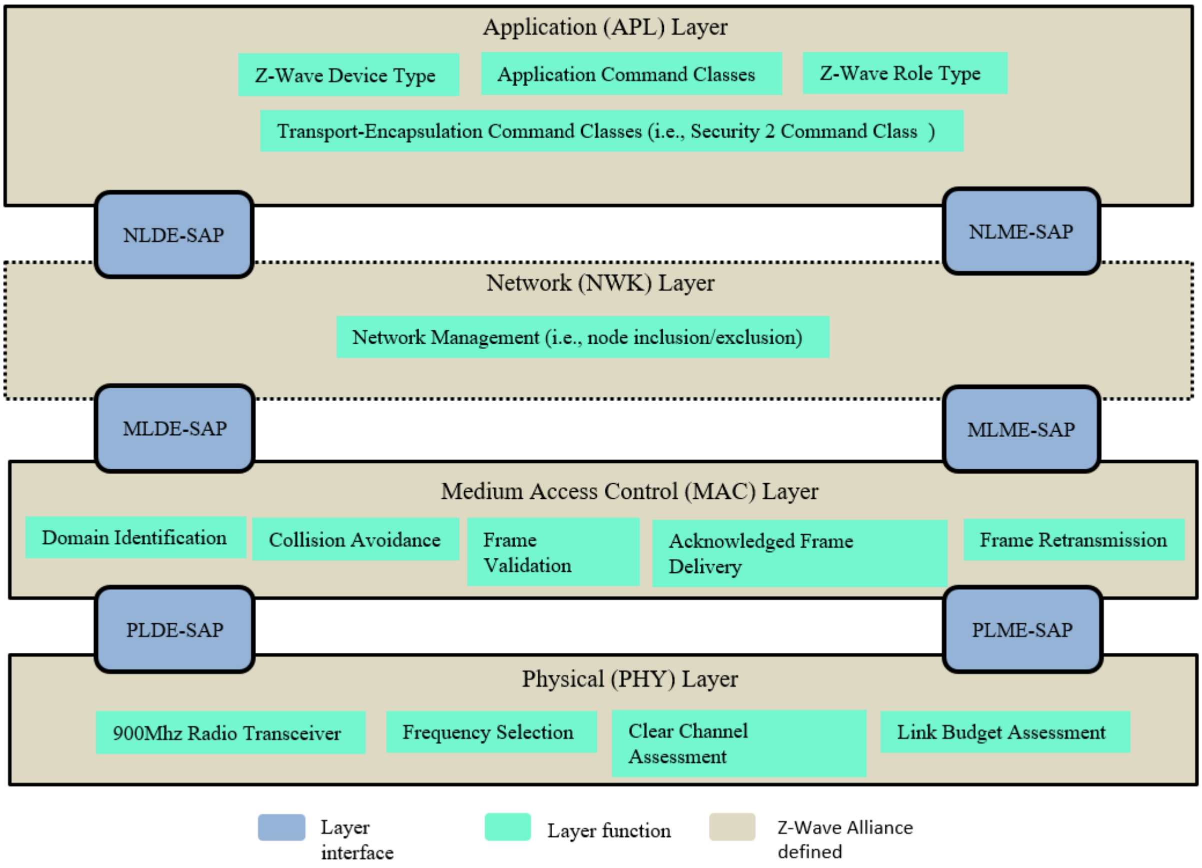


Figure 5.1, Z Wave Long Range protocol stack architecture

Each layer has two main interfaces to facilitate the communication with upper layers through an SAP. The interfaces are described as a data entity and management entity that provide a data transmission service and all other services, respectively.

[LR_PHY] defines the physical layer and [LR_MAC] defines the medium access control layer.

On the foundation of those two lower layers, the Z-Wave alliance defines the Network layer (NWK) and application layers.

The Z-Wave Long Range NWK layer is responsible for network formation (i.e., inclusion/exclusion of nodes to/from a network). The Z-Wave Long Range NWK layer manages the network establishment using command frames known as the Z-Wave Long Range Command Class (defined in section 6.3). These NWK commands are designed for network formation specific purposes.

The Z-Wave application layer is responsible for building applications using dedicated Command Classes, (defined in [ACC], [MCC], [TECC], [NPCC]).In order to be certifiable, applications **shall** comply with Z-Wave device types defined in [DT] and [DTV2]. Finally, the applications layer is also responsible for providing some network management functionalities using the NWK interface (for details, refer to [RT]).

5.2 Z-Wave Long Range Network Layer reference model

The Z-Wave Long Range NWK layer provides an interface between the application layer and the MAC layer. The Z-Wave Long Range NWK layer relies on services provided by the MAC layer and offers services to higher layers though the Network Layer Data Entity (NLDE) and Network Layer Management Entity (NLME) service point interfaces. Figure 3.2 illustrates the components and interface of the Z-Wave Long Range NWK layer.

LR-NWK:0001.1 The Z-Wave Long Range NWK layer **shall** provide two services to the Application layer that are accessed through two SAPs:

- The data service, accessed through NLDE-SAP, and
- The network management service accessed through the NLME-SAP.

The detailed description of the Z-Wave Long Range NWK functional model is presented in chapter 6.

5.3 Z-Wave Long Range definitions

5.3.1 Z-Wave Long Range network principles

The following is a summary of the network principles established by [LR_PHY], [LR_MAC]:

1. Groups of nodes are divided into domains:
   - The division of physical nodes into domains is logical. Domains may fully or partially overlap each other's radio frequency ranges.
   - The Z-Wave Network Layer supports up to 2^{32} domains.
   - Each domain is identified by a unique **HomeID**.

2. The domain is a set of nodes connected to the same medium:
   - Each domain may contain up to *4000* nodes.
   - Each node in the domain is identified by a **NodeID** that is unique within the actual domain.
   - Nodes of the same domain can only communicate with the controller using direct range transmissions.

5.3.2 Controller and end nodes

Refer to 3.3.4 Z-Wave controller roles

5.3.3 Network topology

Refer to 3.3.3 Network topology

Nodes added to a network using Z-Wave Long Range will only have one known neighbour, which is the Primary Controller.

5.3.4 Z-Wave controller roles

Refer to 3.3.4 Z-Wave controller roles

LR-NWK:0002.1 A controller starting a Z-Wave Long Range network **shall** assume the Primary controller role.

The SUC/SIS functionalities will not be used in a Z-Wave Long Range network and included controllers will be Secondary Controllers.

5.3.5 Node operation modes

Refer to 3.3.5 Node operation modes

5.3.6 Network addressing

Z-Wave Long Range supports the following type of addressing:

- Singlecast
- Broadcast

The type of addressing and its frame format are defined in the MPDU Header (refer to [LR_MAC]).

LR-NWK:0003.1 In this specification, some commands **shall not** be sent using broadcast addressing (0xFFF) and **shall** be ignored if received via broadcast addressing.

6 Z-WAVE LONG RANGE NETWORK LAYER SPECIFICATION

6.1 General Description

The Network Layer provides transmission services for higher layers using services provided by the MAC layer. The Z-Wave Long Range NWK layer services only include network formation.

6.1.1 Z-Wave Long Range NWK Layer overview

The network layer is required to provide functionality to ensure correct operation of the MAC sub-layer and provide a suitable service interface to the application layer. To interface with the application layer, the network layer conceptually includes two service entities that provide the necessary functionalities. These service entities are the data service and the management service. The Z-Wave Long Range NWK layer data entity (NLDE) provides the data transmission service via its associated SAP (NLDE-SAP), and the Z-Wave Long Range NWK layer management entity (NLME) provides the management service via its associated SAP (NLME-SAP). The NLME utilizes the NLDE to achieve some of network management tasks. Also, the NLME maintains a database known as the Network Information Base (NIB) that contains information regarding the network topology.

6.1.2 Network Layer Data Entity (NLDE)

LR-NWK:0004.1 The NLDE **shall** provide a data service for the application layer to transport Data Link Protocol Data Unit (DLPDU) to a destination located in the same network. The DLPDU format is shown in Figure 4.1

LR-NWK:0005.1 The NLDE **shall** provide the following services:

LR-NWK:0006.1

- **Generation of NPDUs (Network Protocol Data Unit)**: The NLDE **shall** be capable to generate appropriate NPDUs from application data.

6.1.2.1 Network Layer Management Entity (NLME)

LR-NWK:0007.1 The NLME **shall** provide a management service to leverage the network's routing capabilities. The NLME
LR-NWK:0008.1 **shall** provide the following services:

- **Network Inclusion:** This is the ability to join or create a network.
- **Network Exclusion:** This is the ability to leave a network.

6.2 Frame format

The Z-Wave Long Range Network layer does not provide any functionality other than network management.

6.2.1 NPDU formats

LR-NWK:0009.1 When using a Z-Wave Long Range PHY/MAC, the Z-Wave Long Range NWK layer **shall** add a DLPDU payload, either containing a Z-Wave Long Range NWK command or a command received from the application layer.

This is illustrated in Figure 6.1

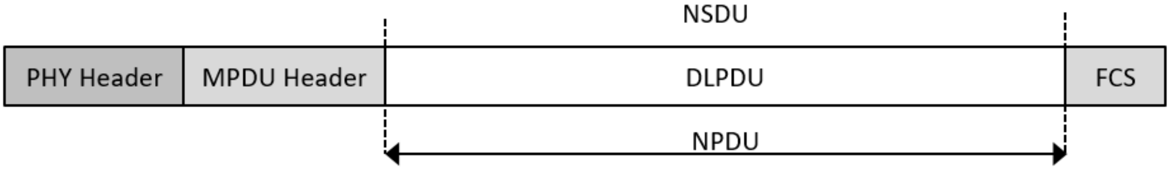


Figure 6.1 General Z-Wave Long Range NPDU format

6.3 Command frames

The commands defined by the Z-Wave Long Range NWK layer are categorized in Command Classes. These Command Classes are listed in Table 4.12.

Table 6.1 Z-Wave Long Range NWK Layer Command Classes

PHY/MAC	Command Class Identifier	Command Class Name	Reference
Z-Wave Long Range	0x04	Z-Wave Long Range Command Class	6.3.1

The following sections illustrates how the Network Layer Management (NLME) shall build the individual commands for transmission.

LR-NWK:000A.1 During the transmission of each of these commands, the NLME **shall** construct the network layer protocol data (NPDU) part of the frame as illustrated in Figure 6.2.

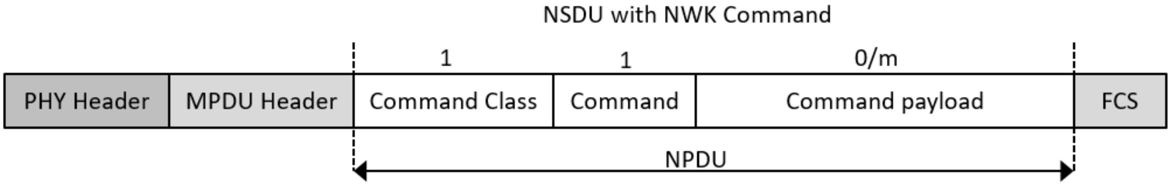


Figure 6.2 The Z-Wave Long Range Network Layer Command Frame Format

LR-NWK:000B.1 LR-NWK:000C.1 Z-Wave Long Range NWK Commands **shall not** use any segmentation or encryption. No application payload **shall** be added from upper layers when using NWK Commands. This is illustrated in Figure 4.18.

Command Classes in the range 0x00..0x1F shall be considered as NWK Commands.

6.3.1 Z-Wave Long Range Command Class

This Command Class is used for including and excluding nodes in a Z-Wave Long Range network.

LR-NWK:000D.1 The Z-Wave Long Range Command Class **shall** be supported by all nodes operating with Z-Wave Long Range.

The Command frames defined by this Command Class are listed in Table 6.2

Table 6.2 Z-Wave Long Range Network Formation Command Class Commands

Command Frame Identifier	Command Name	Reference
0x00	No Operation Command	6.3.1.1
0x01	Node Information Frame Command	6.3.1.2
0x02	Request Node Information Frame Command	6.3.1.3
0x03	Assign IDs Command	6.3.1.4
0x23	Exclude Request Command	6.3.1.5
0x26	SmartStart Included Node Information Command	6.3.1.6
0x27	SmartStart Prime Command	6.3.1.7
0x28	SmartStart Inclusion Request Command	6.3.1.8
0x29	Exclude Request Confirmation Command	6.3.1.9
0x2A	Non Secure Inclusion Step Complete Command	6.3.1.10

LR-NWK:000E.1 All commands in this command class **shall** be supported by a node operating on a Z-Wave Long Range network.

6.3.1.1 No Operation Command

The No Operation Command is used to test the availability of a node in a network.

LR-NWK:000F.1 This Command **may** be used to verifying if an excluded node is still part of the network. This Command may also be used on application level e.g. checking if a node is still operational.

6.3.1.1.1 Frame format

The No Operation Command **shall** be formatted as illustrated in Figure 6.3.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE_LONG_RANGE (0x04)							
Command = ZWAVE_CMD_NO_OPERATION (0x00)							

Figure 6.3 Z-Wave Long Range No Operation Command

6.3.1.1.2 When generated

LR-NWK:0010.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using broadcast addressing.

LR-NWK:0011.1 The application layer **shall** be able to instruct the NWK layer to issue a No Operation Command to a node.

6.3.1.1.3 Effect on receipt

LR-NWK:0012.1 On receipt of this command, a receiving node **shall not** do anything. The sending node will be notified that the receiving node is operational by receiving a MAC layer acknowledgment.

6.3.1.2 Node Information Frame Command

The Node Information Frame command is used to advertise the capabilities of the sending node.

6.3.1.2.1 Frame format

LR-NWK:0013.1 The Node Information Frame Command **shall** be formatted as illustrated in Figure 6.4.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_ZWAVE_LONG_RANGE (0x04)</td></tr>
<tr><td colspan="8">Command = ZWAVE_CMD_NODE_INFO_FRAME (0x01)</td></tr>
<tr><td>Listening</td><td colspan="7">Reserved</td></tr>
<tr><td>Reserved</td><td>Sensor 1000ms</td><td colspan="4">Reserved</td><td>Controller</td><td>Reserved</td></tr>
<tr><td colspan="5">Reserved</td><td colspan="3">Supported Speed</td></tr>
<tr><td colspan="8">Generic Device Class</td></tr>
<tr><td colspan="8">Specific Device Class</td></tr>
<tr><td colspan="8">Command Class List Length</td></tr>
<tr><td colspan="8">Command Class 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Command Class N</td></tr>
</table>

Figure 6.4 Node Information Frame Command format

LR-NWK:0014.1 All fields not described below **shall** be identical to the Node Information Frame Command described in section 4.3.2.1.

6.3.1.2.1.1 Supported Speed (3 bits)

LR-NWK 0088.1 The *Supported Speed* field indicates the transmission data rate supported by the sending node. This field **shall** be treated as a bitmask and **shall** have at least one speed bits set as depicted in Table 6.3

Table 6.3 Maximum Speed supported by the node

Bit	Description
0	*Reserved*
1	100kbps
2	*Reserved*

6.3.1.2.1.2 Command Class List Length (8 bits)

This field is used to advertise the length in bytes of the *Command Class* field.

6.3.1.2.1.3 Command class (N bytes)

This field is used to advertise the list of Command Classes (Refer to [ACC], [MCC], [TECC] and [NPCC]) supported by the sending node using non-secure communication.

LR-NWK:0015.1 The length of this field in bytes **shall** be according to the *Command Class List Length* field value.

LR-NWK:0016.1 LR-NWK:0017.1 LR-NWK:0018.1 The field **shall** advertise the list of Command Classes that the node supports. Command Classes advertised in this field **shall** be in the range 0x21..0xFFFF. Command Classes in the range 0x00..0x20 s**hall not** be advertised in this field.

6.3.1.2.2 When generated

No requirement

6.3.1.2.3 Effect on receipt

On receipt of this command, a receiving node is notified of the sender's Z-Wave Long Range capabilities and non-secure supported Command Classes.

LR-NWK:0019.1 If the network layer does not expect to receive a Node Information Frame Command (i.e. did not issue a Request Node Information Frame Command), the command (or its data) **shall** be forwarded to the upper protocol layer (application).

6.3.1.3 Request Node Information Frame Command

This command is used to request a node to return a Node Information Frame Command.

6.3.1.3.1 Frame format

LR-NWK:001A.1 The Request Node Information Frame Command **shall** be formatted as illustrated in Figure 6.5.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_ZWAVE_LONG_RANGE (0x04)</td></tr>
<tr><td colspan="8">Command = ZWAVE_CMD_REQUEST_NODE_INFO (0x02)</td></tr>
</table>

Figure 6.5 Request Node Information Frame Command format

6.3.1.3.2 When generated

LR-NWK:001B.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using broadcast addressing.

LR-NWK:001C.1 The application layer **shall** be able to instruct the NWK layer to issue a Request Node Information Frame Command to a node.

6.3.1.3.3 Effect on receipt

LR-NWK:001D.1 On receipt of this command, a receiving node **shall** return a Node Information Frame Command in
LR-NWK:001E.1 response. A receiving node **shall not** return a response if this command is received via broadcast
addressing.

6.3.1.4 Assign IDs Command

This command is used to assign a new NodeID and HomeID to the receiving node.

6.3.1.4.1 Frame format

LR-NWK:001F.1 The Assign IDs Command **shall** be formatted as illustrated in Figure 6.6.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE_LR_NETWORK (0x04)							
Command = ZWAVE_CMD_ASSIGN_IDS (0x03)							
Reserved				NodeID (11-8)			
NodeID (7-0)							
HomeID 1							
HomeID 2							
HomeID 3							
HomeID 4							

Figure 6.6 Z-Wave Long Range Assign IDs Command format

6.3.1.4.1.1 Reserved (4 bits)

LR-NWK:0020.1 The *Reserved* field **shall** be set to 0 by a sending node and **shall** be ignored by a receiving node.

6.3.1.4.1.2 NodeID (12 bits)

The *NodeID* field is used to assign a NodeID to a node.

LR-NWK:0021.1 Values in the range 0x100..0xFA0 **shall** indicate the new NodeID assigned to the receiving node.

6.3.1.4.1.3 HomeID (4 bytes)

The *HomeID* field is used to assign an HomeID to a node.

LR-NWK:0022.1 This field **shall** indicate the new HomeID assigned to the receiving node.

6.3.1.4.2 When generated

LR-NWK:0023.1 This command **shall** be sent using singlecast addressing and **shall not** be sent using broadcast addressing.

6.3.1.4.3 Effect on receipt

LR-NWK:0024.1 On receipt of this command, a receiving node **shall** update its HomeID and NodeID if and only if it is currently in SmartStart Learn Mode after issuing a SmartStart Inclusion Request Command and it is received on the Auth HomeID.

Refer to 6.5.4.1 SmartStart Inclusion for details.

LR-NWK:0025.1 A receiving node **shall** ignore the command if it is received via broadcast addressing.

LR-NWK:0026.1 A receiving node **shall** ignore this command if the NodeID field is set to 0x00.

6.3.1.5 Exclude Request Command

The Exclude Request Command is used by a node looking to be excluded from its current network.

6.3.1.5.1 Frame Format

LR-NWK:0027.1 The Exclude Request Command **shall** be formatted as illustrated in Figure 6.7.

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_ZWAVE_LR_NETWORK (0x04)</td></tr>
<tr><td colspan="8">Command = ZWAVE_CMD_EXCLUDE_REQUEST (0x23)</td></tr>
</table>

Figure 6.7 Exclude Request Command format

6.3.1.5.2 When generated

LR-NWK:0028.1 This command **shall** be sent to the broadcast destination (NodeID 0xFFF).

LR-NWK:0029.1 The fields values set in the Exclude Request Command **shall** be identical to the fields set by the sending node in its Node Information Frame Command.

LR-NWK:002A.1 The Exclude Request Command **shall** be sent only if send sending node is in Learn Mode Exclusion. Refer to 6.5.5 Z-Wave Long Range Network Exclusion for details.

6.3.1.5.3 Effect on receipt

LR-NWK:002B.1 On receipt of this command, a controller node that has been instructed to remove a node **shall** return an Exclude Request Confirmation Command to the sending node.

LR-NWK:002C.1 Controllers nodes not trying to exclude a node **shall** ignore this command.

Refer to 6.5.5 Z-Wave Long Range Network Exclusion for details.

6.3.1.6 SmartStart Included Node Information Command

The SmartStart Included Node Info Frame command is used by nodes to notify a controller that it was just powered up and is already part of a network.

6.3.1.6.1 Frame format

LR-NWK:002D.1 The SmartStart Included Node Information Frame Command **shall** be formatted as illustrated in Figure 6.8.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE_LR_NETWORK (0x04)							
Command = ZWAVE_CMD_INCLUDED_NODE_INFO (0x26)							
NWI HomeID 1							
NWI HomeID 2							
NWI HomeID 3							
NWI HomeID 4							

Figure 6.8 SmartStart Included Node Information Frame Command format

6.3.1.6.1.1 NWI HomeID (4 bytes)

Refer to 4.3.2.33.1.1 NWI HomeID (4 bytes)

6.3.1.6.2 When generated

LR-NWK:002E.1 This command **shall** be sent to the broadcast destination (NodeID 0xFFF).

6.3.1.6.3 Effect on receipt

LR-NWK:002F.1 On receipt of this command, a controller node trying to perform a SmartStart inclusion of a node whose S2 DSK matches the *NWI HomeID* field of this command **should** indicate to the application layer that the node to be included is currently included in another network and needs to be removed from the foreign network before it can be included.

6.3.1.7 SmartStart Prime Command

The Smart Prime command is used to notify SmartStart including controllers that a node is about to make an inclusion request.

6.3.1.7.1 Frame format

LR-NWK:0030.1 The SmartStart Included Node Information Frame Command **shall** be formatted as illustrated in Figure 6.9

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE_LR_NETWORK (0x04)							
Command = ZWAVE_CMD_SMARTSTART_PRIME (0x27)							
Listening	Reserved						
Reserved	Sensor 1000ms	Sensor 250ms	Beam capability	Reserved		Controller	Reserved
Generic Device Class							
Specific Device Class							
Command Class list length							
Command Class 1							
…							
Command Class N							

Figure 6.9 SmartStart Prime Command format

LR-NWK:0031.1 All fields configuration **shall** be identical to the Node Information Frame Command described in section 4.3.2.1.

6.3.1.7.2 When generated

LR-NWK:0032.1 This command **shall** be sent to the broadcast destination (NodeID 0xFFF).

LR-NWK:0033.1 This command **shall** be sent on the *NWI HomeID* HomeID and **shall not** be sent on the currently assigned HomeID. Refer to 4.3.2.33.1.1 NWI HomeID (4 bytes) for details.

LR-NWK:0034.1 The sending node **shall** subsequently send a SmartStart Inclusion Request Command after *nwkSmartStartInclusionRequestDuration* seconds.

LR-NWK:0035.1 Nodes not operating in AL mode **may** return to sleep between issuing the SmartStart Prime Command and the SmartStart Inclusion Request Command.

6.3.1.7.3 Effect on receipt

LR-NWK:0036.1 On receipt of this command, a controller node that intends to include any node using SmartStart **shall** verify if the Network HomeID of the frame header matches the NWI HomeID of any of the DSKs present in its SmartStart list.

LR-NWK:0037.1 If it finds a match, the controller node **shall** enter SmartStart Inclusion and attempt to include the node when it issues a SmartStart Inclusion Request Command.

LR-NWK:0038.1 In the unlikely event of several DSK matches for the received NWI HomeID, the controller node **shall** enter SmartStart Inclusion alternating between the possible DSK candidates.

6.3.1.8 SmartStart Inclusion Request Command

The SmartStart Inclusion Request Command is used to request to initiate a SmartStart inclusion.

6.3.1.8.1 Frame format

LR-NWK:0039.1 The SmartStart Inclusion Request Command **shall** be formatted as illustrated in Figure 6.10

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_ZWAVE_LR_NETWORK (0x04)</td></tr>
<tr><td colspan="8">Command = ZWAVE_CMD_SMARTSTART_INCLUDE (0x28)</td></tr>
<tr><td>Listening</td><td colspan="7">Reserved</td></tr>
<tr><td>Reserved</td><td>Sensor 1000ms</td><td>Sensor 250ms</td><td>Beam capability</td><td colspan="2">Reserved</td><td>Controller</td><td>Reserved</td></tr>
<tr><td colspan="8">Generic Device Class</td></tr>
<tr><td colspan="8">Specific Device Class</td></tr>
<tr><td colspan="8">Command Class list length</td></tr>
<tr><td colspan="8">Command Class 1</td></tr>
<tr><td colspan="8">...</td></tr>
<tr><td colspan="8">Command Class N</td></tr>
</table>

Figure 6.10 SmartStart Inclusion Request Command format

LR-NWK:003A.1 All fields configuration **shall** be identical to the Node Information Frame Command described in section 4.3.2.1.

6.3.1.8.2 When generated

LR-NWK:003B.1 LR-NWK:003C.1 This command **shall** be sent to the broadcast destination (NodeID 0xFFF). This command **shall** be sent on the NWI HomeID and **shall not** be sent on the currently assigned HomeID. Refer to 4.3.2.33.1.1 NWI HomeID (4 bytes) for details.

LR-NWK:003D.1 The sending node **shall** listen to and accept Assign IDs Commands using the Auth HomeID. More details are provided in Figure 4.58 and 4.5.4.3 SmartStart Inclusion.

6.3.1.8.3 Effect on receipt

LR-NWK:003E.1 On receipt of this command, a controller node that intends to include the sending node **shall** return an Assign IDs Command using the SmartStart Auth HomeID.

For SmartStart Auth HomeID definition, refer to 4.3.2.35.3 Effect on receipt

6.3.1.9 Exclude Request Confirmation Command

The Exclude Request Confirmation Command is used by a controller node to confirm to a node that it can leave the current network.

6.3.1.9.1 Frame Format

LR-NWK:003F.1 The Exclude Request Confirmation Command **shall** be formatted as illustrated in Figure 6.11.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE_LR_NETWORK (0x04)							
Command = ZWAVE_CMD_EXCLUDE_REQUEST_CONFIRMATION (0x29)							
Reserved				Requesting NodeID (11-8)			
Requesting NodeID (7-0)							
Requesting HomeID 1							
Requesting HomeID 2							
Requesting HomeID 3							
Requesting HomeID 4							

Figure 6.11 Exclude Request Confirmation Command format

Requesting NodeID (12 bits)

This field is used to indicate the NodeID that has issued an Exclude Request Command for which this command is returned.

Requesting HomeID (4 bytes)

This field is used to indicate the HomeID on which the Exclude Request Command was issued.

6.3.1.9.2 When generated

LR-NWK:0040.1 This command **shall** be sent to a node that has issued an Exclude Request Command.

LR-NWK:0041.1 The Exclude Request Confirmation Command **shall** be sent by a controller that has been instructed to exclude a node from a Z-Wave Long Range network.

Refer to 6.5.5 Z-Wave Long Range Network Exclusion for details.

6.3.1.9.3 Effect on receipt

LR-NWK:0042.1 On receipt of this command, a node currently trying to be excluded from its network **shall** leave its current network.

LR-NWK:0043.1 Nodes **shall** ignore this command if they are not in Learn Mode Exclusion.

Refer to 6.5.5 Z-Wave Long Range Network Exclusion for details.

6.3.1.10 Non Secure Inclusion Step Complete Command

The Non Secure Inclusion Step Complete Command is used by a controller node to indicate to a joining node that the non-secure part of the network inclusion is completed.

6.3.1.10.1 Frame Format

LR-NWK:0044.1 The Non-Secure Inclusion Step Complete Command **shall** be formatted as illustrated in Figure 6.12.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZWAVE_LR_NETWORK (0x04)							
Command = ZWAVE_CMD_NON_SECURE_INCLUSION_STEP_COMPLETE (0x2A)							

Figure 6.12 Non Secure Inclusion Step Complete Command format

6.3.1.10.2 When generated

LR-NWK:0045.1 A sending node **shall** initiate S2 Security Bootstrapping after issuing this command.

Refer to 6.5.4 Z-Wave Long Range Network Inclusion for details.

6.3.1.10.3 Effect on receipt

LR-NWK:0046.1 On receipt of this command, a joining node is instructed that the non-secure inclusion is completed, and the S2 Security Bootstrapping **shall** now take place.

LR-NWK:0047.1 A receiving node **shall** start its S2 TB1 timer upon reception of this command. Refer to [TECC] for details.

Refer to 6.5.4 Z-Wave Long Range Network Inclusion for details.

6.4 Constants

LR-NWK:0048.1 The constants that define the Z-Wave Long Range NWK layer are presented in Table 6.4. Implementations **shall** comply with these values.

Table 6.4 Z-Wave Long Range NWK layer constants

Constant	Description	Value
nwkSmartStartInclusionRequestDuration	Duration between SmartStart Prime Command and SmartStart Inclusion Request Command.	4 seconds (tolerance ± 1 second)
nwkSmartStartInclusionBackoffDuration	Duration between SmartStart inclusion requests	Refer to Table 4.29
nwkLearnModeMinDuration	Minimum duration in which a node **shall** stay in Learn Mode during Network Exclusion.	3 seconds
nwkLRExcludeRequestForeignNetBackOff	Minimum delay for a controller after receiving an Exclude Request command to issue an Exclusion Request Confirmation command on a foreign Network.	2 seconds
nwkAssignIDConfirmationRetries	Maximum number of retransmissions of the No Operation Command to verify that Assign IDs Command has been accepted	10
nwkNonSecureInclusionCommandTimeout	Timeout between each command of the non-secure part of a network inclusion for joining nodes	10 seconds

LR-NWK:0049.1 The attributes defined by the Z-Wave Long Range NWK layer are presented in Table 6.5. Implementations **shall** comply with the indicated ranges.

Table 6.5 Z-Wave Long Range NWK layer attributes

Attribute	Description	Range
aNwkSmartStartMaxInclusionRequestInterval	Maximum Time interval between SmartStart Inclusion requests.	[4..99] * 128 seconds

6.5 Functional description

NWK commands in frame flows contained in this section shall use the Z-Wave Long Range Command Class. (refer to 6.3.1)

6.5.1 Communication between Z-Wave Long Range nodes.

LR-NWK:004A.1 All communications **shall** use direct range when operating on a Z-Wave Long Range PHY/MAC.

6.5.2 Learn mode

LR-NWK:004C.1 Z-Wave Long Range nodes **shall** provide functionalities that enable them to learn about the existing current network or a new network.

On a Z-Wave Long Range network, Learn Mode is used for a node to accept changing network (joining or leaving).

LR-NWK:004D.1 Nodes typically enter Learn Mode to join or leave a network. When a node enters to Learn Mode, it **may** have the following intents:

- Learn Mode Exclusion: the node is expecting a network exclusion.
- SmartStart Learn Mode: the node operates with the SmartStart procedure for inclusion.

LR-NWK:004E.1 LR-NWK:004F.1 Learn Mode Exclusion **should** only be enabled when necessary and disabled again as quickly as possible. A node entering Learn Mode Exclusion **shall** stay in Learn Mode Exclusion for a minimum duration of *nwkLearnModeMinDuration*.

LR-NWK:0050.1 The application layer **may** determine when Learn Mode Exclusion is to be enabled/disabled.

More details are given in the individual scenarios described in 6.5.4 and 6.5.5

6.5.3 Network Formation

LR-NWK:0051.1 LR-NWK:0052.1 End nodes **shall not** start a new network and **shall** wait until they get included in a network by a controller node. When not included in a network, end nodes **shall** assign themselves their NWI HomeID (refer to 4.3.2.33.1.1 NWI HomeID (4 bytes)).

LR-NWK:0053.1 Controller nodes that do not belong to a network **shall** start a new Z-Wave Long Range network automatically by:

- LR-NWK:0054.1 Assigning themselves a HomeID and the NodeID 0x01. The HomeID (*aNwkRandomHomeID)* **shall** be generated using a random number generator.
- Assuming the Primary Controller role.

6.5.4 Z-Wave Long Range Network Inclusion

Nodes join a Z-Wave Long Range network using SmartStart Inclusion

6.5.4.1 SmartStart Inclusion

The Z-Wave SmartStart inclusion removes the user interactions and lets nodes join a network automatically if the including controller possesses the S2 DSK (refer to [TECC]) of the node to be included.

LR-NWK:0055.2 Nodes supporting to be included using SmartStart inclusion **shall** provide at least one of the following methods for entering SmartStart

1 Enter SmartStart Learn Mode automatically after powering on
2 Provide a mechanism to enter SmartStart Learn Mode if already powered up.

6.5.4.1.1 SmartStart supporting nodes power-up

LR-NWK:0056.2 Nodes supporting SmartStart inclusion **should** initiate an inclusion procedure after powering up or alternatively it **should** initiate an inclusion procedure when triggered manually after power-up. This procedure depends on the inclusion state of a node and is described in subsections 6.5.4.1.1.1 and 6.5.4.1.1.2.

LR-NWK:0057.1 Nodes capable of operating both on Z-Wave and Z-Wave Long Range **shall** issue SmartStart Prime and SmartStart Inclusion Requests both on the Z-Wave and Z-Wave Long Range channels.

6.5.4.1.1.1 Not included nodes

LR-NWK:0058.1 Nodes that are not part of a network **shall** issue SmartStart Inclusion Requests at regular intervals. A
LR-NWK:0059.1 SmartStart Inclusion Request **shall** consist of a SmartStart Prime Command and a SmartStart Inclusion Request Command.

LR-NWK:005A.1 It **shall** be according to Figure 6.13. The NWI HomeID construction is specified in 4.3.2.33.1.1 NWI HomeID (4 bytes).

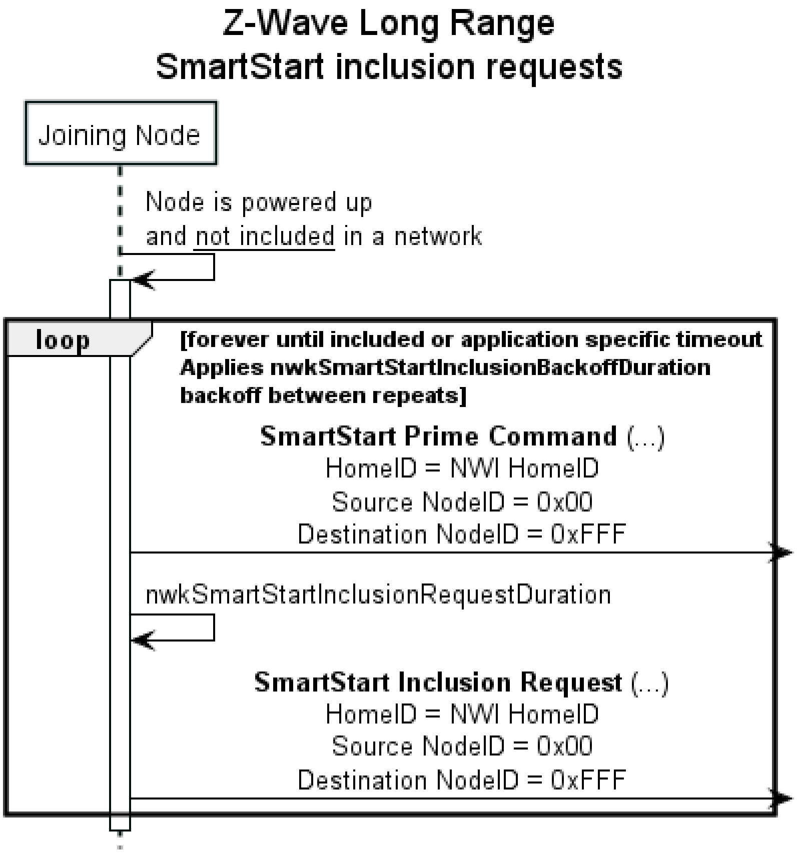

Z-Wave Long Range
SmartStart inclusion requests
Joining Node
Node is powered up
and not included in a network
loop
[forever until included or application specific timeout
Applies nwkSmartStartInclusionBackoffDuration
backoff between repeats]
SmartStart Prime Command (...)
HomeID = NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFFF
nwkSmartStartInclusionRequestDuration
SmartStart Inclusion Request (...)
HomeID = NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFFF

Figure 6.13 Z-Wave Long Range Network Inclusion - SmartStart not included node power on

LR-NWK:005B.1 LR-NWK:005C.1 The timing of the inclusion requests **shall** be according to Table 4.29. When a range is indicated as duration, nodes **shall** use a new unique random value in that range every time.

6.5.4.1.1.2 Included nodes

LR-NWK:005D.1 A node already part of a network **shall** send a single SmartStart Included Node Information Command when entering SmartStart, i. e. after power up or when triggered manually. This is illustrated in Figure 6.14.

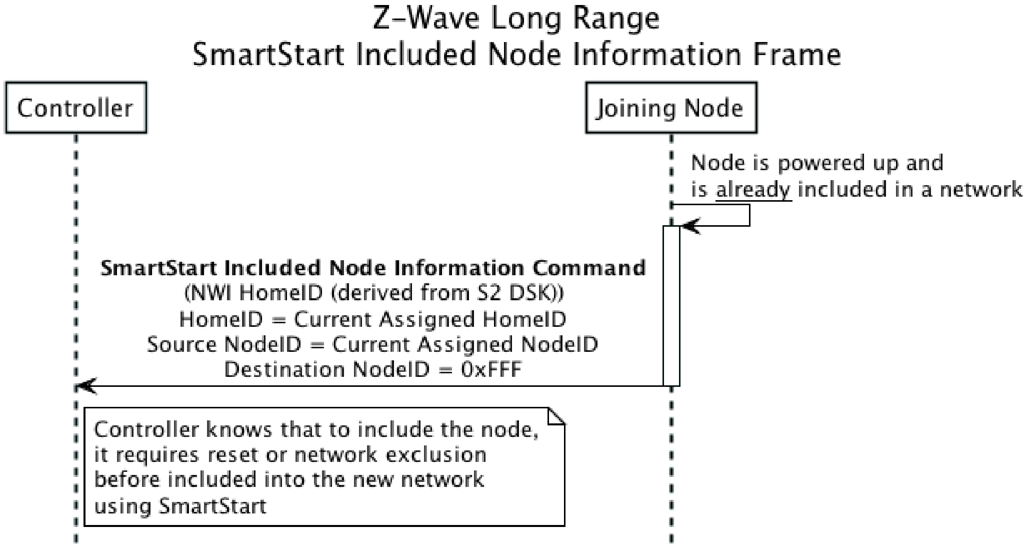

Z-Wave Long Range
SmartStart Included Node Information Frame
Controller
Joining Node
Node is powered up and
is already included in a network
SmartStart Included Node Information Command
(NWI HomeID (derived from S2 DSK))
HomeID = Current Assigned HomeID
Source NodeID = Current Assigned NodeID
Destination NodeID = 0xFFF
Controller knows that to include the node,
it requires reset or network exclusion
before included into the new network
using SmartStart

Figure 6.14 Z-Wave Long Range Network Inclusion - SmartStart included node power on

6.5.4.1.2 SmartStart including controllers

LR-NWK:005E.1 A controller **shall** have the Primary Controller role in the network to perform a SmartStart inclusion. LR-NWK:005F.1 Secondary Controllers **shall not** perform SmartStart inclusions.

A controller **shall** be given the S2 DSK of a node to perform a SmartStart inclusion. Refer to [TECC].

6.5.4.1.3 Successful SmartStart inclusion

LR-NWK:0060.1 A controller performing a SmartStart network inclusion **shall** perform S2 bootstrapping (even if the joining node does not show the S2 Command Class in its supported Command Class list). Refer to [TECC] for the detailed S2 bootstrapping procedure.

LR-NWK:0061.1 A SmartStart inclusion **shall** be according to Figure 6.15.

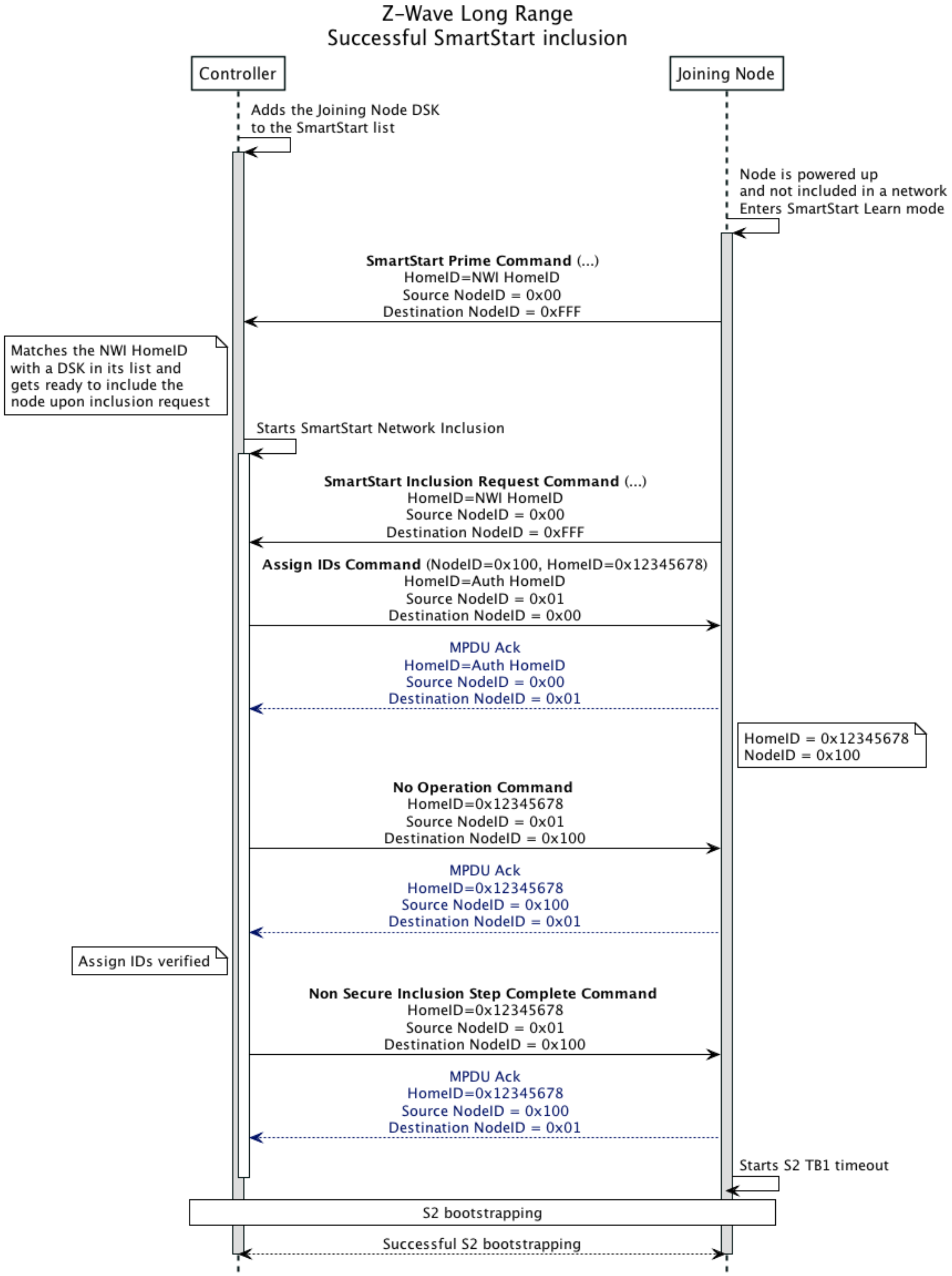

Z-Wave Long Range
Successful SmartStart inclusion
Controller
Joining Node
Adds the Joining Node DSK to the SmartStart list
Node is powered up and not included in a network Enters SmartStart Learn mode
SmartStart Prime Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFFF
Matches the NWI HomeID with a DSK in its list and gets ready to include the node upon inclusion request
Starts SmartStart Network Inclusion
SmartStart Inclusion Request Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFFF
Assign IDs Command (NodeID=0x100, HomeID=0x12345678)
HomeID=Auth HomeID
Source NodeID = 0x01
Destination NodeID = 0x00
MPDU Ack
HomeID=Auth HomeID
Source NodeID = 0x00
Destination NodeID = 0x01
HomeID = 0x12345678
NodeID = 0x100
No Operation Command
HomeID=0x12345678
Source NodeID = 0x01
Destination NodeID = 0x100
MPDU Ack
HomeID=0x12345678
Source NodeID = 0x100
Destination NodeID = 0x01
Assign IDs verified
Non Secure Inclusion Step Complete Command
HomeID=0x12345678
Source NodeID = 0x01
Destination NodeID = 0x100
MPDU Ack
HomeID=0x12345678
Source NodeID = 0x100
Destination NodeID = 0x01
Starts S2 TB1 timeout
S2 bootstrapping
Successful S2 bootstrapping

Figure 6.15 Successful Z-Wave Long Range SmartStart inclusion

6.5.4.1.4 Unsuccessful SmartStart inclusion

LR-NWK:0062.1 SmartStart Network inclusion attempts may be unsuccessful. Any SmartStart Network Inclusion attempt that does not complete with a successful S2 Bootstrapping **shall** be considered as unsuccessful.

LR-NWK:0063.1 If an error occurred during S2 bootstrapping (S2 bootstrapping started and aborted), the inclusion attempt **shall not** be considered successful.

LR-NWK:0064.1 If S2 bootstrapping did not start, the inclusion attempt **shall not** be considered successful.

LR-NWK:0065.1 If the including controller did not grant any key requiring authentication (Public Key Report bytes obfuscated by zeros), the inclusion attempt **shall not** be considered successful.

LR-NWK:0066.1 If the including controller granted fewer keys than what the joining node requested, the inclusion attempt **shall** be considered successful if at least one authenticated key was granted.

LR-NWK:0067.1 If more than *nwkNonSecureInclusionCommandTimeout* seconds elapsed between receiving commands part of the SmartStart inclusion prior to S2 bootstrapping, a joining node **shall** abort the inclusion attempt and **shall not** consider the inclusion successful.

LR-NWK:0068.1 If more than *nwkNonSecureInclusionCommandTimeout* seconds elapsed between receiving acknowledgements for commands part of the SmartStart inclusion prior to S2 bootstrapping, a controlling node **shall** abort the inclusion attempt and **shall not** consider the inclusion successful.

Refer to [TECC] for S2 Bootstrapping.

In case of an unsuccessful SmartStart Network Inclusion:

- LR-NWK:0069.1 LR-NWK:006A.1 The joining node **shall** leave the network automatically and consider itself not included in any network. The joining node **shall** return to SmartStart learn mode.
- The joining node **may** continue with SmartStart learn mode until successful SmartStart inclusion or re-try at least up to 2 times and give up.
- LR-NWK:006B.1 LR-NWK:006C.1 The including controller **should** consider the joining node removed from the network. It **may** verify if the joining node has left the network properly using the NOP Command.

An example is given in Figure 6.16. MAC layer Ack frames are omitted from Figure 6.16.

LR-NWK:006D.1 If receiving no MPDU acknowledgement, an including controller **may** issue up to *nwkAssignIDConfirmationRetries* No Operation Command frames to verify that the node was included. LR-NWK:006E.1 It **should not** send the Non Secure Inclusion Step Complete Command and **should not** initiate the S2 bootstrapping procedure if the node is non-responsive and in this case it **shall** consider the network inclusion unsuccessful. LR-NWK:006F.1

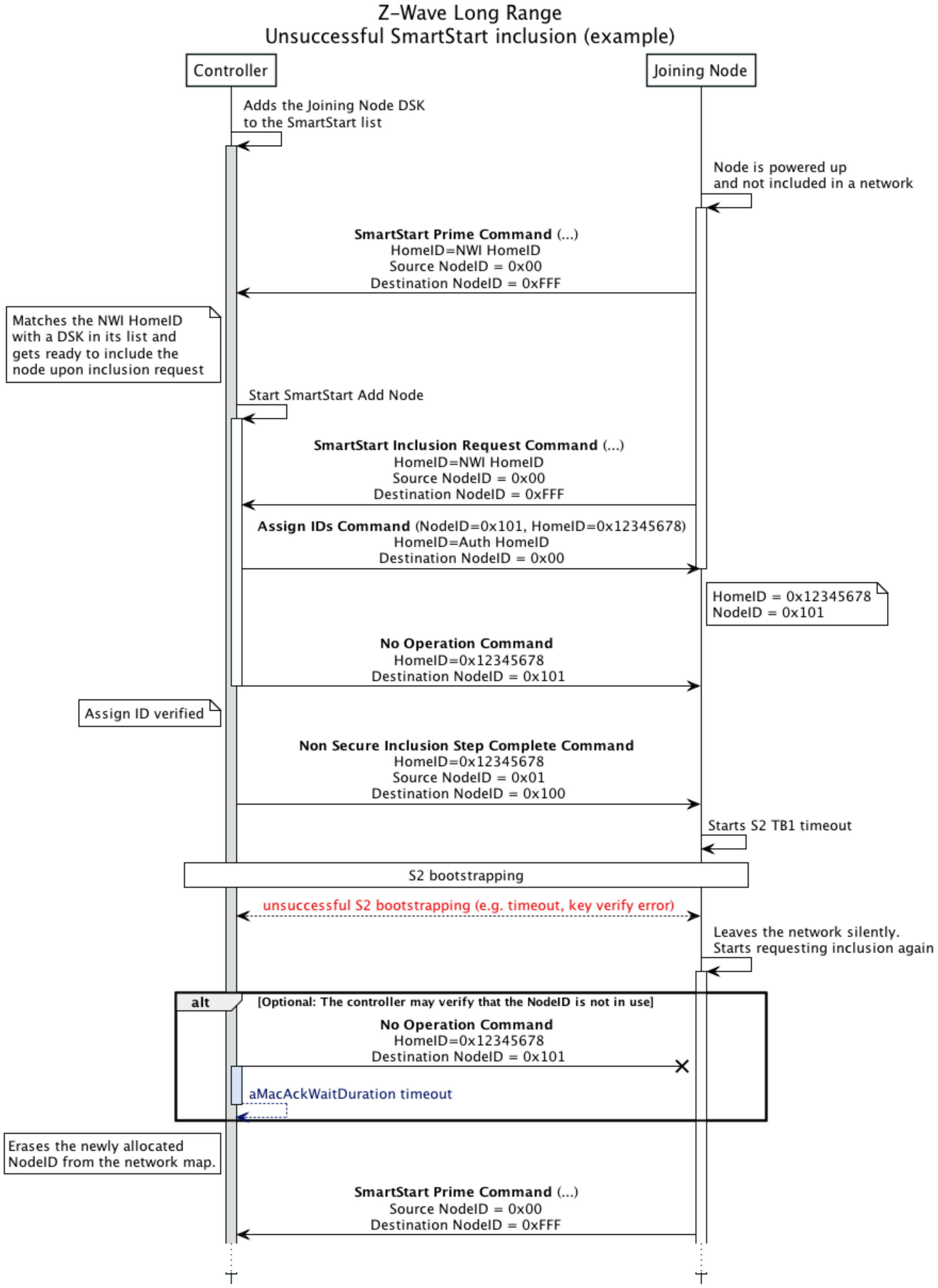

Z-Wave Long Range
Unsuccessful SmartStart inclusion (example)
Controller
Joining Node
Adds the Joining Node DSK
to the SmartStart list
Node is powered up
and not included in a network
SmartStart Prime Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFFF
Matches the NWI HomeID
with a DSK in its list and
gets ready to include the
node upon inclusion request
Start SmartStart Add Node
SmartStart Inclusion Request Command (...)
HomeID=NWI HomeID
Source NodeID = 0x00
Destination NodeID = 0xFFF
Assign IDs Command (NodeID=0x101, HomeID=0x12345678)
HomeID=Auth HomeID
Destination NodeID = 0x00
HomeID = 0x12345678
NodeID = 0x101
No Operation Command
HomeID=0x12345678
Destination NodeID = 0x101
Assign ID verified
Non Secure Inclusion Step Complete Command
HomeID=0x12345678
Source NodeID = 0x01
Destination NodeID = 0x100
Starts S2 TB1 timeout
S2 bootstrapping
unsuccessful S2 bootstrapping (e.g. timeout, key verify error)
Leaves the network silently.
Starts requesting inclusion again
alt
[Optional: The controller may verify that the NodeID is not in use]
No Operation Command
HomeID=0x12345678
Destination NodeID = 0x101
aMacAckWaitDuration timeout
Erases the newly allocated
NodeID from the network map.
SmartStart Prime Command (...)
Source NodeID = 0x00
Destination NodeID = 0xFFF

Figure 6.16 Unsuccessful Z-Wave Long Range SmartStart inclusion example

6.5.5 Z-Wave Long Range Network Exclusion

Nodes exit a Z-Wave Long Range network using direct range Network Exclusion.

6.5.5.1 Network Exclusion

The Network Exclusion process will exclude nodes from a network.

LR-NWK:0070.1 The Network Exclusion procedure **shall** be according to Figure 6.17.

LR-NWK:0071.1 When starting Learn Mode Exclusion, a node **shall** issue an Exclude Request Command to the broadcast destination NodeID.

LR-NWK:0072.1 End nodes excluded from a network **shall** assume the NodeID 0x00 after exclusion and **shall** assume their NWI HomeID as new HomeID.

LR-NWK:0073.1 Controller nodes excluded from a network **shall** start a new network (refer to 6.5.3 Network Formation).

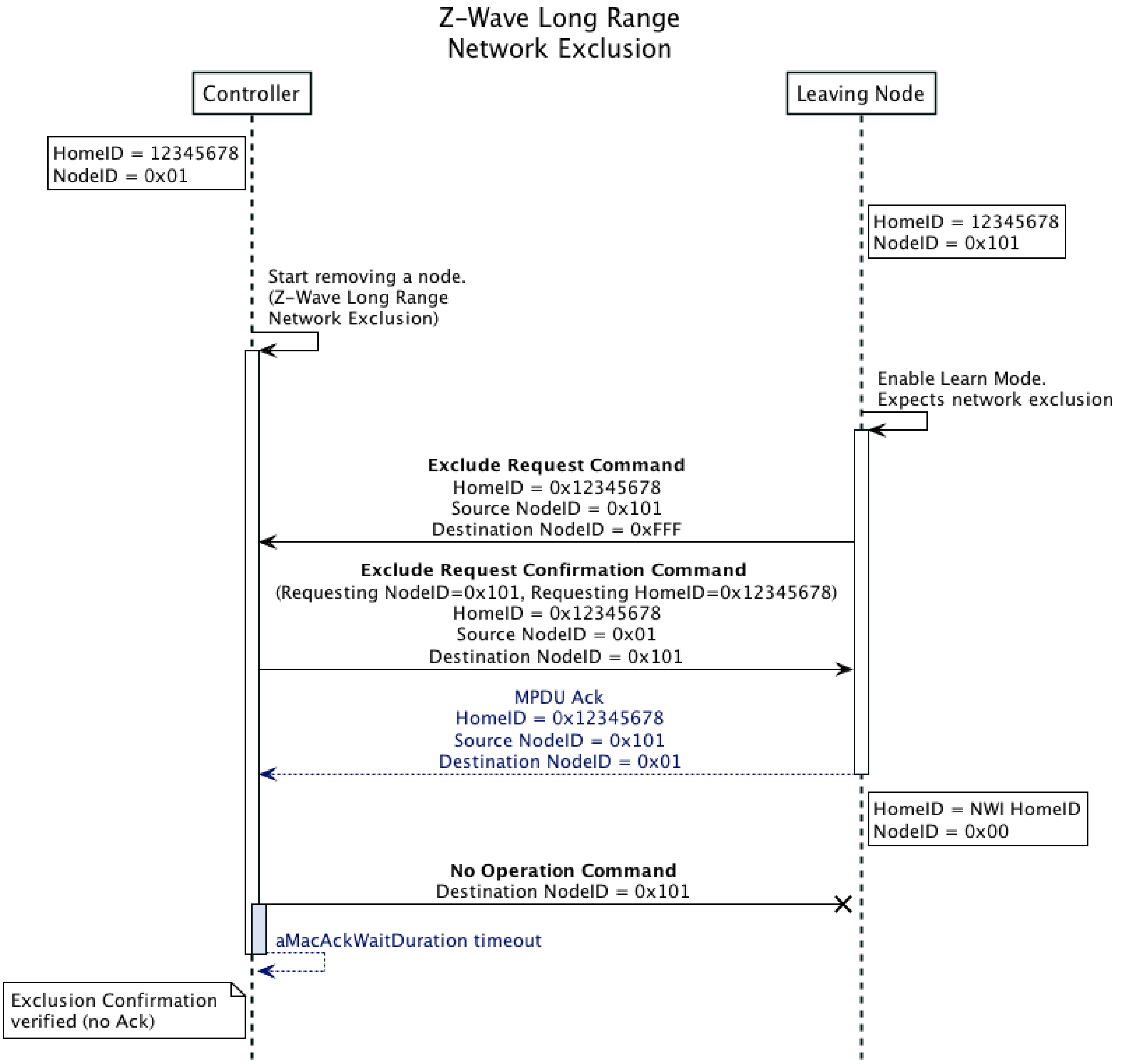


Figure 6.17 Z-Wave Long Range Network Exclusion

6.5.5.2 Network Exclusion from a foreign network

LR-NWK:0074.1 Controllers instructed to remove a node **shall** also remove nodes from foreign networks.

LR-NWK:0075.1 A controller that has started a node removal **shall** return an Exclude Request Confirmation Command if an Exclude Request Command has been issued in another HomeID. The Exclude Request Confirmation LR-NWK:0076.1 Command if issued by the excluding controller **shall** be on its own HomeID.

A controller shall issue the Exclude Request Confirmation Command on its own HomeID. When LR-NWK:0077.1 returning an Exclude Request Confirmation Command to a node in a foreign network, an MDPU Ack **shall not** be requested.

LR-NWK:0078.1 A randomized delay in the range 0..1 second **should** be added by controller nodes to the *nwkLRExcludeRequestForeignNetBackOff* time .

LR-NWK:0079.1 A node in Learn Mode (exclusion) **shall** accept Exclude Request Confirmation Command if issued in another HomeID.

Figure 6.18 illustrates a network exclusion in a foreign network.

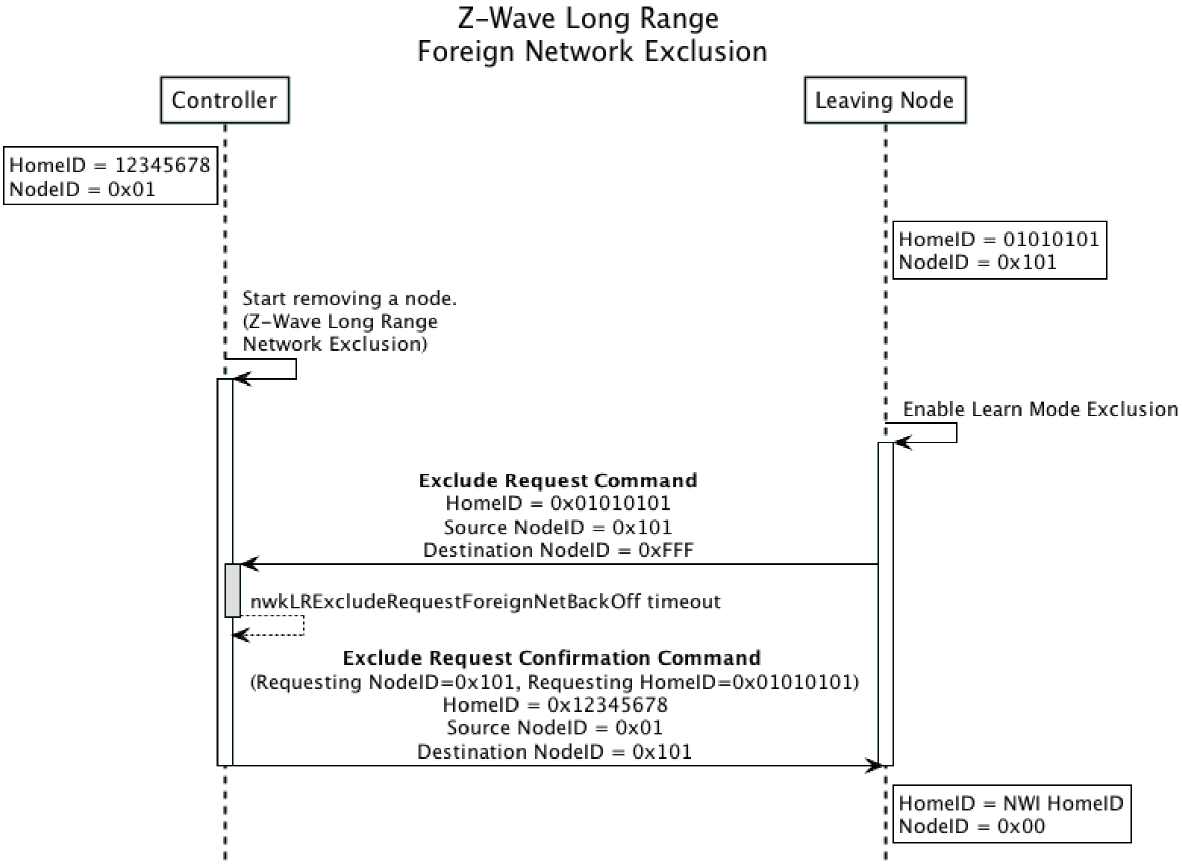


Figure 6.18 Z-Wave Long Range Network Exclusion from a foreign HomeID

6.5.6 Failing nodes

LR-NWK:007A.1 In a Z-Wave network, a node **may** be considered failing or non-responsive when a controller cannot reach the node.

6.5.6.1 Remove a Failing node

6.5.6.1.1 AL and FL nodes

LR-NWK:007B.1 A Remove Failed Node procedure **may** be used to remove non-responsive nodes from a network.

LR-NWK:007C.1 LR-NWK:007D.1 Before removing a non-responsive NodeID from a network, a controller **shall** issue No Operation Commands to the non-responsive NodeID. If the node is not responding, the controller **shall** proceed with removing the NodeID and updating the network.

LR-NWK:007E.1 A responding node **shall not** be removed from the network by a controller without using Network Exclusion.

LR-NWK:007F.1 Removing a failing AL or FL node **shall** be according to Figure 6.19

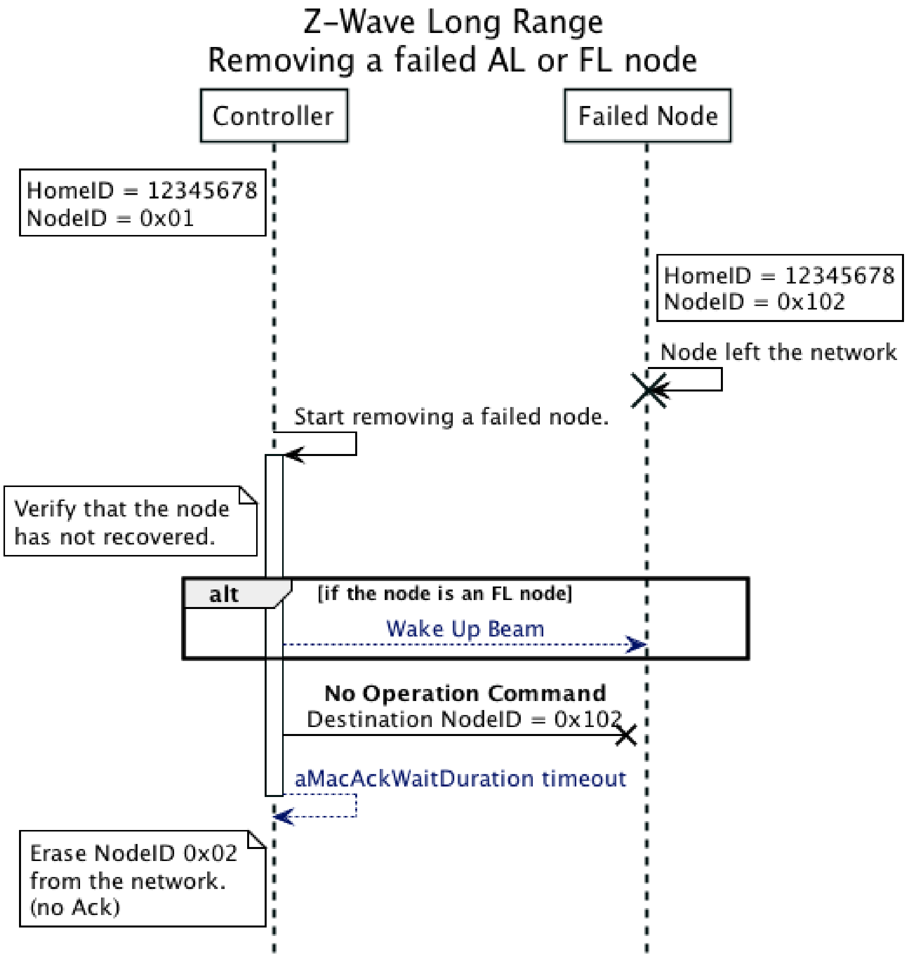

Z-Wave Long Range
Removing a failed AL or FL node
Controller
Failed Node
HomeID = 12345678
NodeID = 0x01
HomeID = 12345678
NodeID = 0x102
Node left the network
Start removing a failed node.
Verify that the node has not recovered.
alt
[if the node is an FL node]
Wake Up Beam
No Operation Command
Destination NodeID = 0x102
aMacAckWaitDuration timeout
Erase NodeID 0x02 from the network. (no Ack)

Figure 6.19 Z-Wave Long Range Removing a failed AL or FL node from a network

6.5.6.1.2 NL nodes

LR-NWK:0080.1 The process **shall** be identical to 4.5.6.1.2 NL nodes.

6.5.7 Controller roles

LR-NWK:0081.1 Controllers in a Z-Wave Long Range network cannot change role. The controller that has created the network **shall** be Primary Controller and all other subsequently included controllers **shall** be Secondary Controllers.

6.5.8 Dual Z-Wave and Z-Wave Long Range networks

LR-NWK:0082.1 Controllers **may** create both a Z-Wave and Z-Wave Long Range network that they operate on
LR-NWK:0083.1 simultaneously. In this case, they **shall** use the same HomeID.

LR-NWK:0084.1 Controllers using S2 **shall** use different key for identical S2 Security Classes in the Z-Wave and the Z-Wave Long Range network. i.e. the S2 Authenticated key in the Z-Wave network cannot decrypt S2 Authenticated messages from the Z-Wave Long Range network.

LR-NWK:0085.1 All other nodes in that network **shall** operate either on the Z-Wave or Z-Wave Long Range PHY/MAC, but not both simultaneously.

LR-NWK:0086.1 Nodes supporting both Z-Wave and Z-Wave Long Range entering SmartStart learn mode **shall** issue SmartStart Prime and SmartStart Inclusion Requests on both PHY/MAC.

LR-NWK:0087.1 Nodes included in a network **shall** only use the PHY/MAC that they got included with. i.e. a node included with Z-Wave Long Range **shall not** operate on Z-Wave and vice-versa.

REFERENCES

[G.9959]	ITU-T G.9959, Short range narrowband digital radiocommunication transceivers – PHY & MAC layer specifications
[LR_PHY]	ZWA_Z-Wave Long Range PHY and MAC Layer
[LR_MAC]	ZWA_Z-Wave Long Range PHY and MAC Layer Specification
[ACC]	ZWA_Z-Wave Application Command Class Specification ACC
[MCC]	ZWA_Z-Wave Management Command Class SpecificationMCC
[TECC]	ZWS_Z-Wave Transport-Encapsulation Command Class SpecificationTECC
[NPCC]	ZWA_Z-Wave Network-Procotol Command Class Specification NPCC
[RT]	ZWA_Z-Wave Plus Role Type Specification RT
[DT]	ZWA_Z-Wave Plus Device Type SpecificationDT
[DTV2]	ZWA_Z-Wave Plus v2 Device Type SpecificationDTV2
[DC]	ZWA_Z-Wave Device Class Specification
[NPITR]	ZWS_Node Provisioning Information Type Registry