

AUTOSAR4.2.2 CAN网络管理栈培训

普华汽车电子事业部
iSOFT AE BU

2022年1月28日

普华基础软件股份有限公司

iSOFT INFRASTRUCTURE SOFTWARE CO.,
LTD



1

网络管理简介



2

CanNm



3

Nm



4

ComM



5

CanSm



Part 1

网络管理简介

1.1 网络管理功能概述



1.2 Autosar网络管理概要



1.3 申请网络



1.4 释放网络



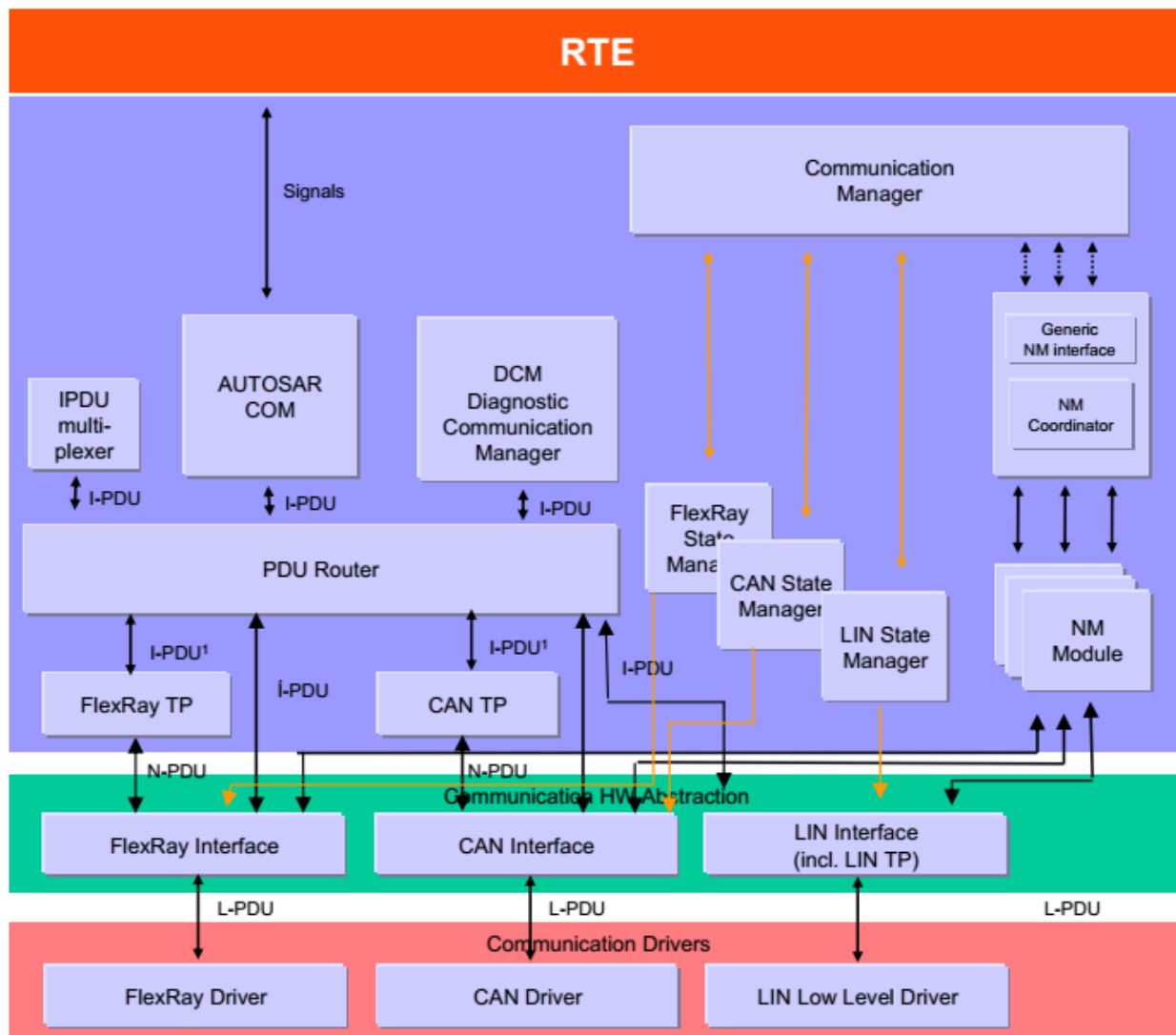
1.1 网络管理功能概述

- 不管是传统的燃油车还是新能源汽车，车上都有各种各样的ECU，所有这些ECU都是需要用电的。车上的供电单元一般是蓄电池，而蓄电池的电量是有限的。为了尽可能的省电，所以需要ECU在不工作时进入低功耗模式(睡眠)，为实现这个目的就提出了网络管理。也就是说网络管理一个最重要的作用就是为了节能。
- 网络管理是通过在各个ECU的网络(Can、Lin、Ethernet等)上，发送一些命令，制定一套规则，来实现各个ECU的协同睡眠和唤醒。

1.2 Autosar协议栈概要

- 目前，AUTOSAR4.2.2 网络管理规范支持Can、Lin、FlexRay以及Ethernet总线。以Can总线为例，涉及到的模块自顶向下包括ComM、CanSm、Nm、CanNm、CanIf、CanTDrv。
- ① ComM 简化了用户对网络管理的使用，同时协调一个ECU上多个独立的软件对总线通信模型的分时复用；
 - ② CanSm设置Control、Tranceiver的状态，同时负责Bus-Off的恢复；
 - ③ Nm为ComM和CanNm之间的桥梁，为上层屏蔽了具体的网络管理细节；同时，协调多个网络之间的同步睡眠(Coordinated shutdown)。
 - ④ CanNm为网络管理策略；
 - ⑤ CanIf和CanTrcv为报文的收发提供了底层支持。

1.2.1 AUTOSAR架构中的网络管理

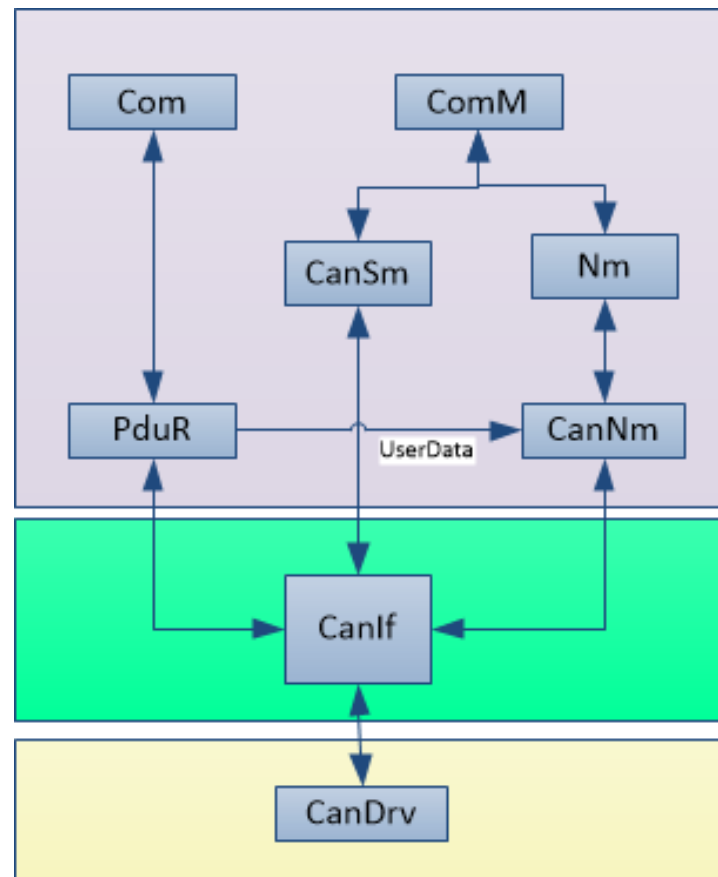


其中：
黄色线条表示控制流；
黑色线条表示数据流。

1.3 各模块之间的关系

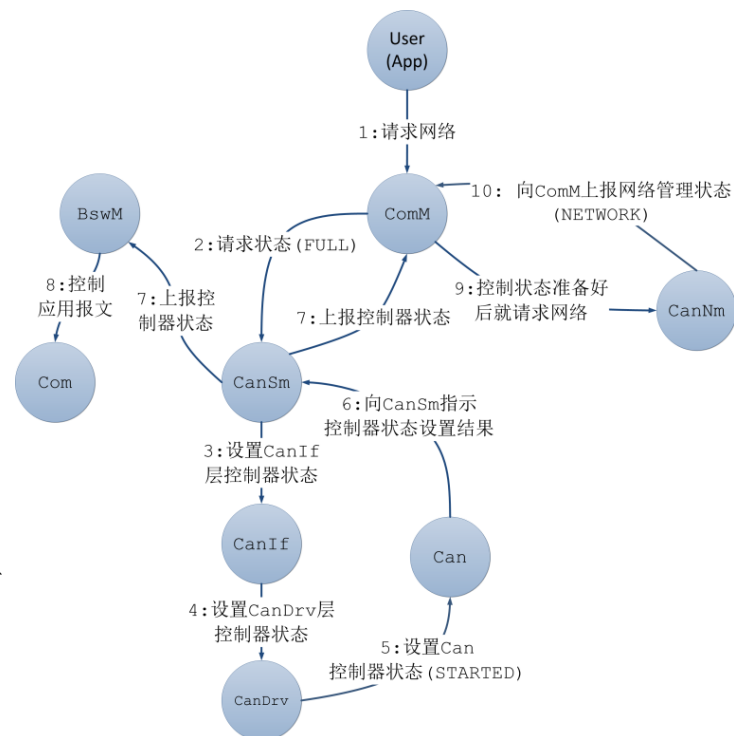
➤ 以Can总线为例, 如右图所示, 网络管理涉及到ComM模块、CanSm模块、Nm模块、CanNm模块、CanIf模块以及CanDrv模块。这么多模块在网络管理是以什么顺序运行, 他们之间有什么关系。下面以请求网络和释放网络为例, 讲解其流程。

➤ 备注: 右图中的Com、PduR、CanIf、CanDrv属于Can通信栈, 只是网络管理与之有少许的交互, 故也画了出来。



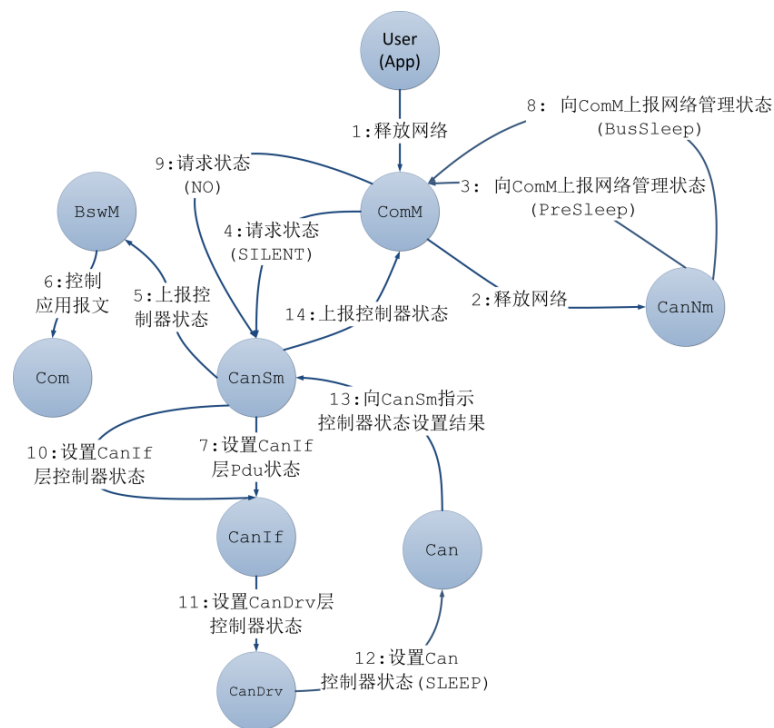
1.3.1 申请网络

1. 上层通过user向ComM请求网络(1)；
2. ComM向CanSm请求状态，以将CAN控制器切换到正常收发模式(STARTED) (2)；
3. CAN控制器状态切换成功后(3, 4, 5, 6)：
 - ◆ 向BswM上报，以使能应用报文的收发(7, 8)；
 - ◆ 向ComM答复，请求状态设置成功(7)；
4. ComM通过Nm向CanNm请求网络，并发出网络管理报文(9)；
5. CanNm通过Nm向ComM上报，状态切换成功(10)；



1.3.2 释放网络

1. 上层通过user向ComM请求释放网络(1)；
2. ComM通过Nm向CanNm请求释放网路(2)；
3. CanNm从Network Mode进入Prepare Bus-Sleep Mode后，向ComM上报(3)；
4. ComM向CanSm请求SILENT后，CanSm将设置Pdu模式为TX_OFFLINE；此时可以通过BswM去控制应用报文(发送能力关闭，接收能力开启)(4, 5, 6, 7)；
5. CanNm从Prepare Bus-Sleep Mode进入Bus Sleep Mode后，向ComM上报(8)；
6. ComM向CanSm请求NO后，CanSm将设置控制状态到SLEEP(9, 10, 11, 12, 13)；



Part 2

CanNm

2.1 CanNm简介



2.2 网络管理



2.3 Bus Load Reduced



2.4 User Data



2.5 Remote Sleep



2.6 Partial Networking



2.7 Coordinator Synchronization



2.1 CanNm简介

➤ CanNm是分布式的直接网络管理策略。这种策略可总结为如下两点：

- ① 每个节点如果想保持总线通信，就会一直周期性的发送NM消息；如果它不想再需要保持总线通信，它就不再发送NM消息。
- ② 如果总线通信已经被释放，并且在一段时间内没有发送或者接收到NM消息，则切换到Bus-Sleep Mode。

➤ CanNm主要功能有：

1. 网络管理
2. Bus Load Reduced
3. User Data
4. Remote Sleep
5. Partial Networking
6. Coordinator Synchronization

2.2.1 网络管理报文格式

➤ CanNm的帧结构如下图所示。

- ◆ Source Node Identifier(NID)、Control bit Vector(CBV)表示节点ID和控制域，可分别通过CanNmPduNidPosition和CanNmPduCbvPosition来配置，取值为byte0、byte1和0ff。
- ◆ User Data用于存放用户定义的数据。420

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 7	User data 5							
Byte 6	User data 4							
Byte 5	User data 3							
Byte 4	User data 2							
Byte 3	User data 1							
Byte 2	User data 0							
Byte 1	Control Bit Vector							
Byte 0	Source Node Identifier							

2.2.1 网络管理报文格式

其中的控制域如下图所示：

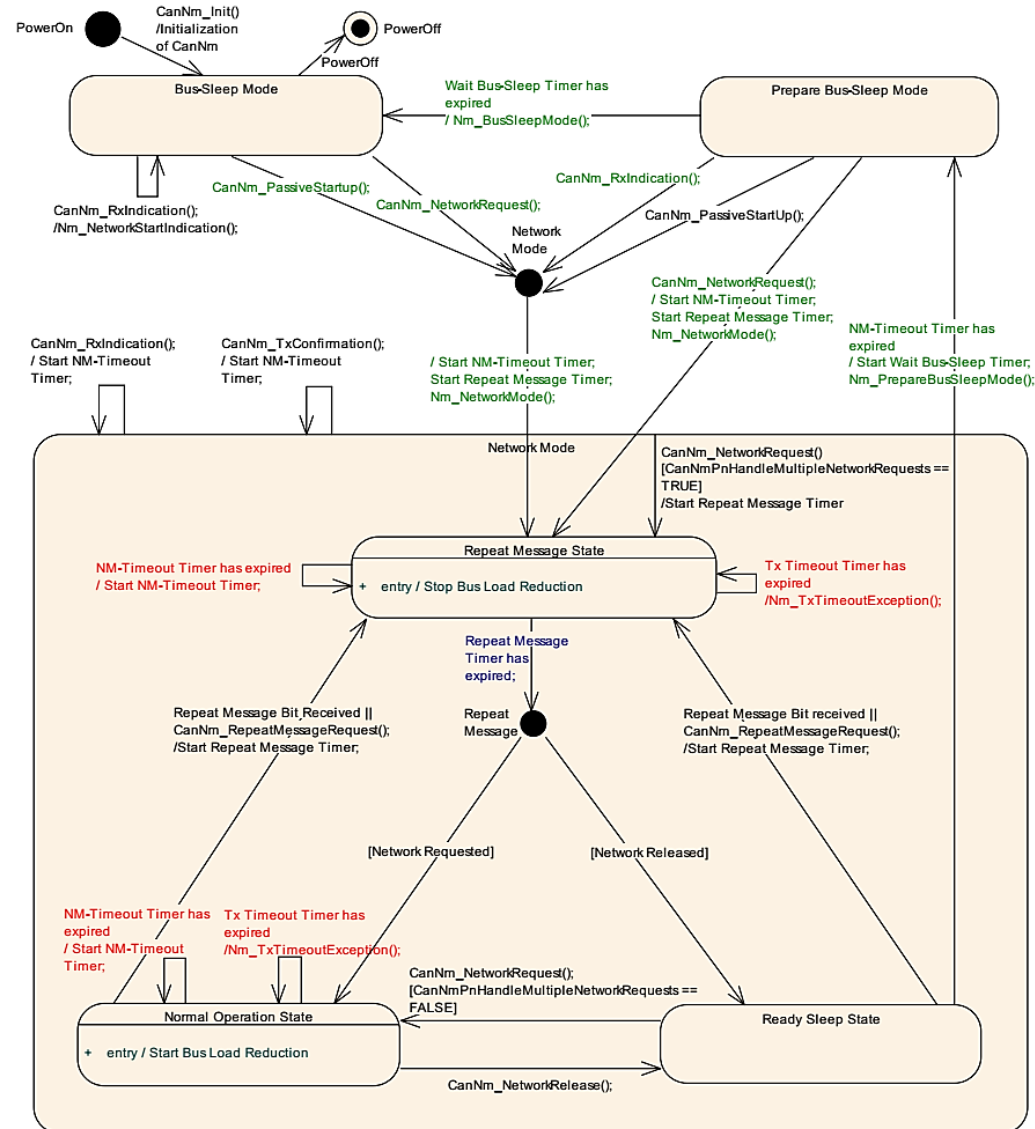
Repeat Message Request	为1表示外部节点请求节点进入Repeat Message State； 通过CanNmNodeDetectionEnabled开启； 调用CanNm_RepeatMessageRequest时设置为1，退出Repeat Message State时清0；
NM Coordinator Id	Coordinator Cluster Id
Nm Coordinator Sleep Ready bit	为1表示主网关请求开始睡眠； 调用CanNm_SetSleepReadyBit来设置；
Active Wakeup Bit	为1表示节点是主动唤醒的，否则表示被动唤醒； 通过CanNmActiveWakeupBitEnabled 开启； 进入Network Mode置1，退出Network Mode清0；
Partial Network Information bit	为1表示该报文包含PN请求信息；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CBV	Reserved	Partial Network Information Bit	Reserved	Active Wakeup Bit	NM Coordinator Sleep Ready Bit	Reserved R3.2 NM Coordinator or ID (High Bit)	Reserved R3.2 NM Coordinator ID (Low Bit)	Repeat Message Request

2.2 网络管理

- 网络管理报文通过广播发送，网络中的所有节点都可以接收到。
 - ✓ 某个节点发出网络管理报文, 表示该节点需要维持在Network Mode。
 - ✓ 如果某节点准备进入Bus Sleep Mode，它就停止发送网络管理报文。
- 但是，只要它还能够接收到其他节点发来的网络管理报文，它就推迟转换到Bus Sleep Mode。最终，在一定的时限内，由于不再接收到网络管理报文，每个节点都将同步转换到Bus-Sleep Mode。

2.2.1 State Machine



2.2.2 网络管理的状态

➤ CanNm为每个网络都维护一个状态机。主要是为了协调网络在正常操作模式和睡眠模式之间的切换，这也是它的核心功能。状态机中有三个状态，分别是：

1. Bus Sleep Mode
2. Prepare Bus-Sleep Mode
3. Network Mode。其中Network Mode又分为3个子状态：
 - ① Repeat Message State
 - ② Normal Operation State
 - ③ Ready Sleep State

➤ 下面讲一下每个状态它的作用。

2.2.2 网络管理的状态

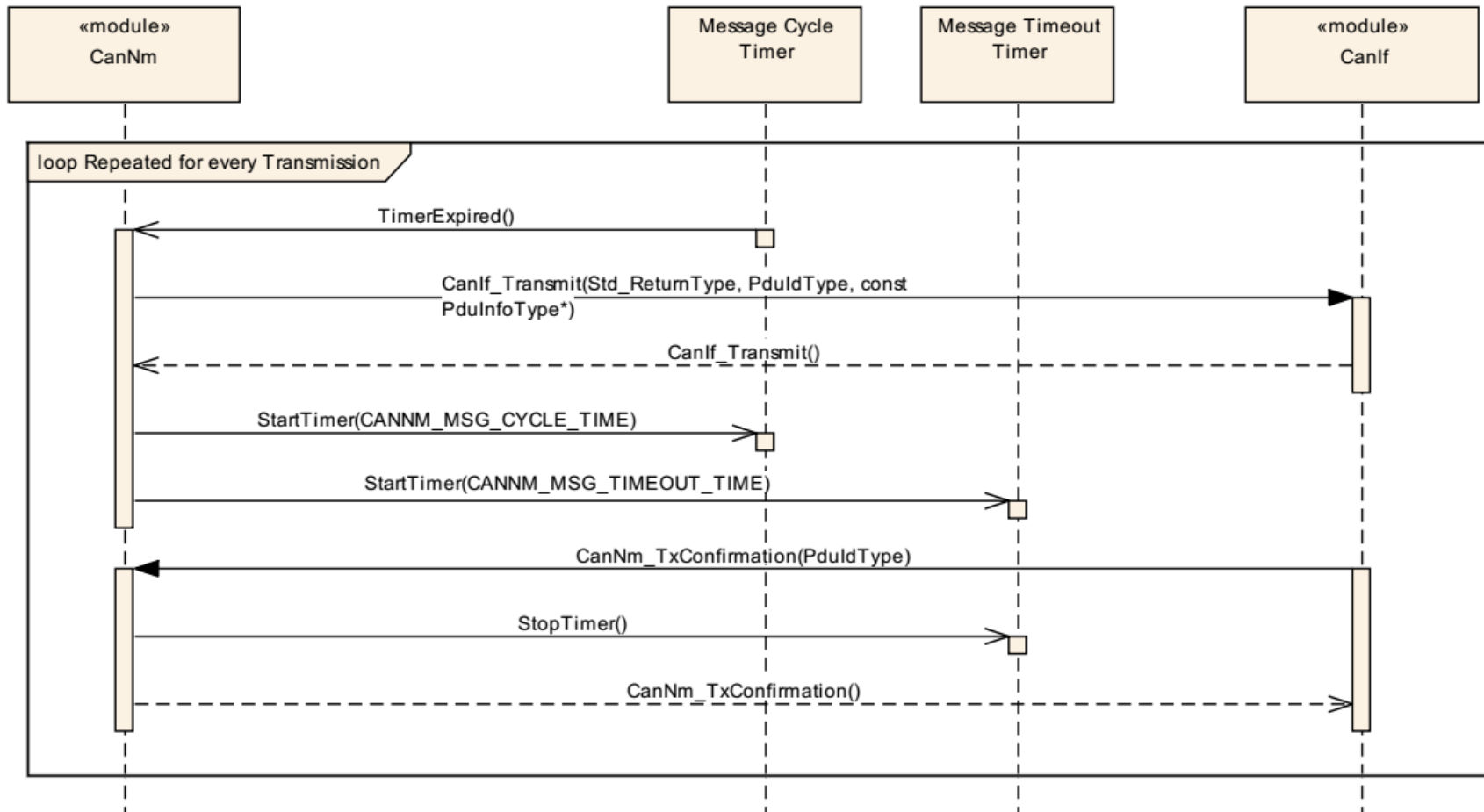
Bus Sleep Mode		这个状态就是我们所说的睡眠状态，也是CanNm初始化后的状态。该状态下不收发网络管理报文也不收发应用报文
Prepare Bus-Sleep Mode		<p>这个状态不发送网络管理报文，也不发送应用报文，只是等待其他ECU一起睡眠。</p> <p>任何到睡眠的过程，都需要经过这个状态，也就是说睡眠前有些准备工作是必须要完成的。比如数据存储等情况，因此这个状态就是用来做一些休眠前的准备工作的。</p>
Network Mode	Repeat Message State	<p>该状态有2个作用：</p> <ol style="list-style-type: none"> 1. 确保节点维持最小的Network Mode； 2. 用于检测网络中当前已经存在的节点； <p>该状态下以CanNmImmediateNmCycleTime为周期，发出CanNmImmediateNmTransmissions个报文，并维持CanNmRepeatMessageTime</p>
	Normal Operation State	该状态是节点正常工作的状态。该状态下节点以CanNmMsgCycleTime(所有节点必须相同)为周期发出网络管理报文。
	Ready Sleep State	这个状态下不发出网络管理报文，但可以接收网络管理报文。这个状态的目的是为了实现在网络内的节点的同步睡眠。

2.2.2 网络管理的状态

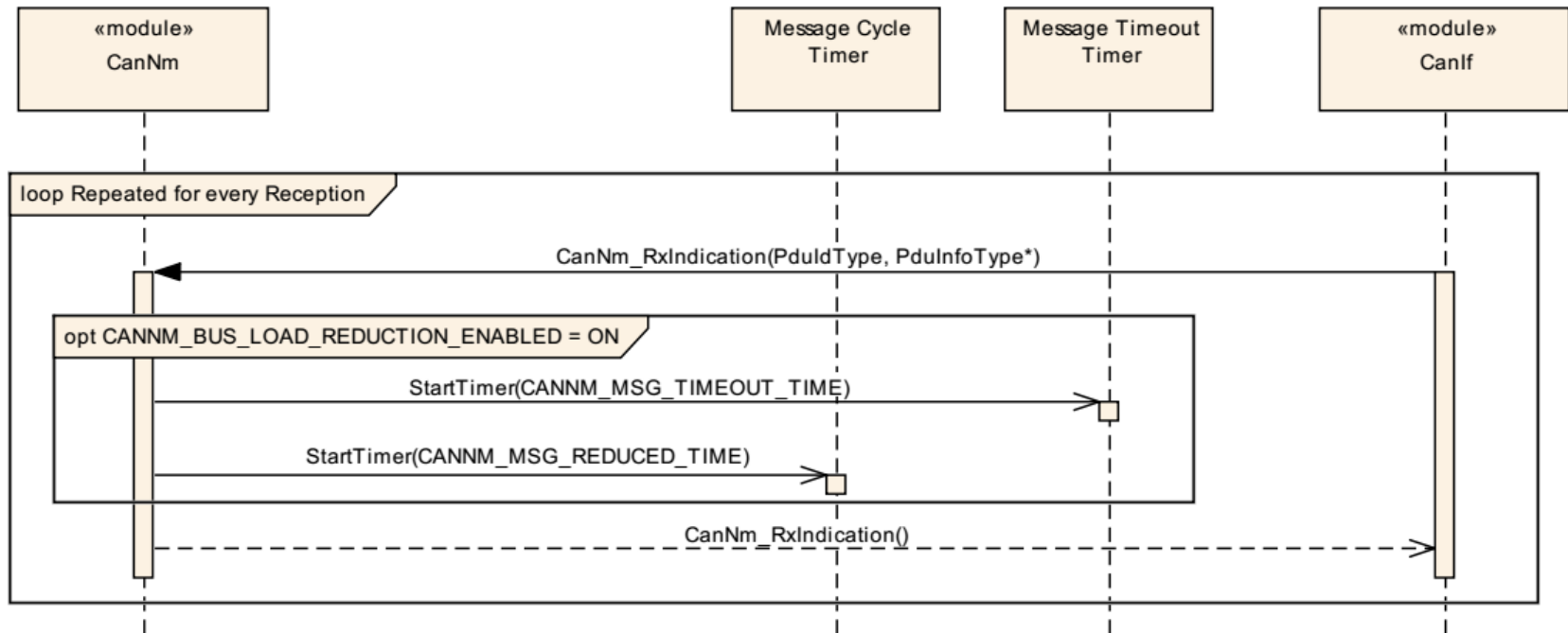
➤ 各模式下，网络管理报文和应用报文的收发功能如下。

	Repeat message state	Normal operation state	Ready sleep state	Prepare Bus-sleep Mode	Bus sleep Mode
Nm Pdu	TX:ON RX :ON	TX:ON RX :ON	TX:OFF RX :ON	TX:OFF RX :OFF	TX:OFF RX :OFF
Com Pdu	TX:ON RX :ON	TX:ON RX :ON	TX:ON RX :ON	TX:OFF RX :ON	TX:OFF RX :OFF

2.2.3 网络管理报文的发送



2.2.4 网络管理报文的接收



2.2.5 网络管理是如何实现同步睡眠的

- 在上面提到，Ready Sleep State主要用于同步睡眠，那CanNm是如何保证同步睡眠的呢？
- 在CanNm的规范当中，有一个定时器NM-Timeout Timer。该定时器进入Network Mode启动，退出Network Mode后关闭。该定时器有如下特性：
 1. 成功发送/接收一帧网络管理报文后，该定时器被重启
 2. Repeat Message State/Normal Operation State下，该定时器超时后被重启（正常运行不会出现。）
 3. Ready Sleep State下超时后，切换到Prepare Bus-Sleep Mode。
- 通过上面的2个特性，即可保证网络内所有节点“同步睡眠”。
- 备注：网络内所有节点的该参数必须相同，且大于`CanNmMsgCycleTime`

2.2.5 网络管理是如何实现同步睡眠的

➤ 相关配置

CanNmTimeoutTime

表示Ready Sleep State的维持时间

2.3 Bus Load Reduced

➤ 背景

通过上面的介绍可以知道，若存在多个节点，他们同时进入Network Mode 将导致一个问题：总线负载在某些时间点会显著上升。

➤ 为解决这个问题有两种解决方式：

- ① 配置CanNmMsgCycleOffset。该方法主要是给每个节点配置一个发送偏移时间，可以缓解该问题，但这不能从根本上解决该问题。
- ② 开启Bus Load Reduced功能。该方法只在“Normal Operation State”下有效。

➤ 具体策略

- ① 当节点接收一帧网络管理报文，则切换自己的发送周期为CanNmMsgReducedTime。每个节点该值不同，并且满足

$$\frac{\text{CanNmMsgCycleTime}}{2} < \text{CanNmMsgReducedTime} < \text{CanNmMsgCycleTime}。$$

- ② 当节点发送一帧网络管理报文，则切换自己的发送周期为CanNmMsgCycleTime

。

2.3 Bus Load Reduced

➤ 限制

与PN功能互斥。

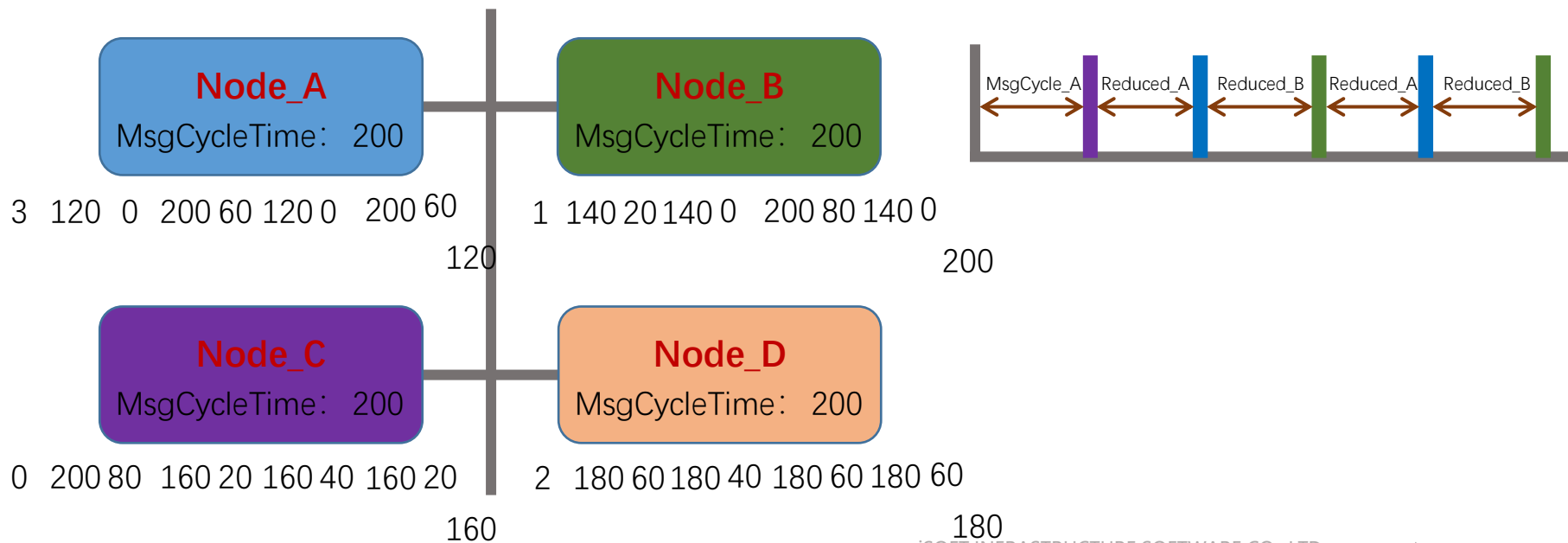
➤ 表现

这样保证CanNmMsgReducedTime最小的两个节点在交替发送NM消息。一个CanNmMsgCycleTime内最多会有2个NM消息被发送出去。

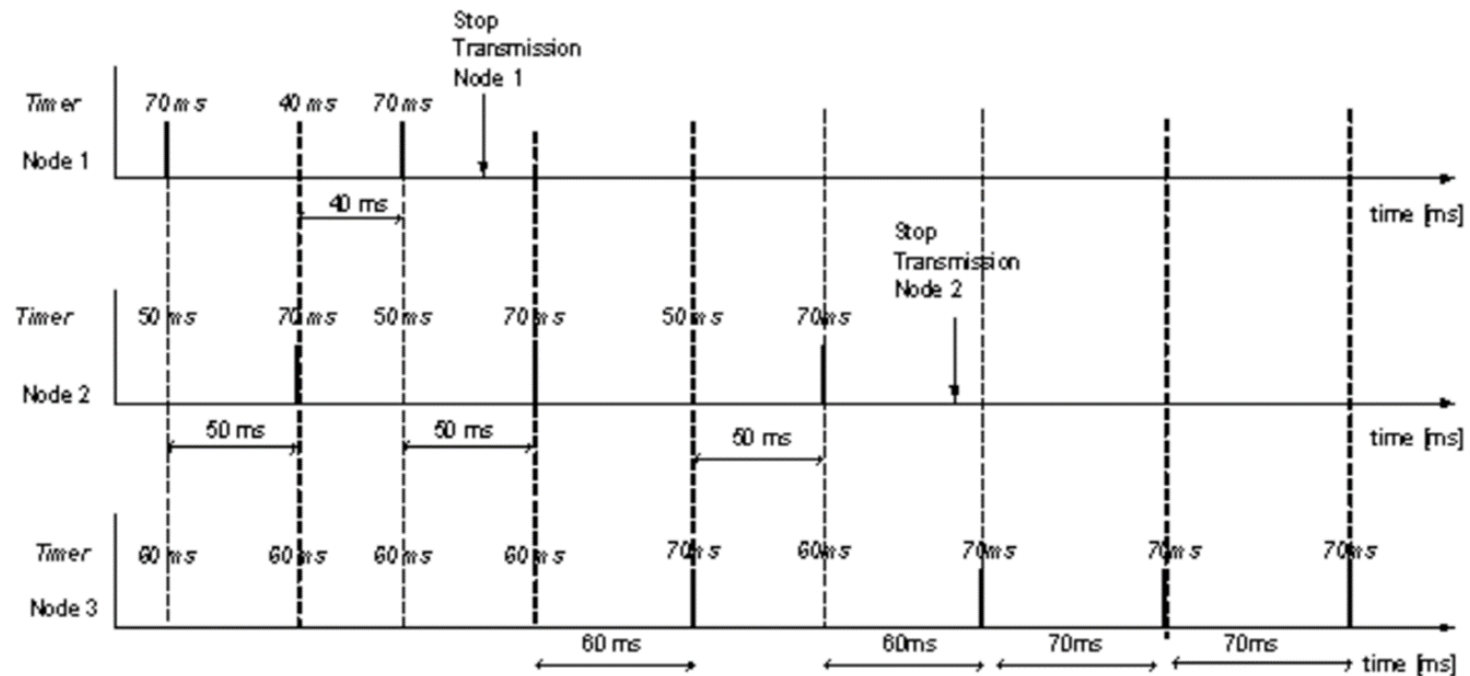
2.3 Bus Load Reduced

假设有4个节点A, B, C, D。进入Network Mode的顺序分别为C → B → D → A。

	CanNmMsgCycleTime	CanNmMsgReducedTime
Node_A	200ms	120ms
Node_B	200ms	140ms
Node_C	200ms	160ms
Node_D	200ms	180ms



2.3 Bus Load Reduced



:Transmission of a NM Message

CANNM_MSG_CYCLE_TIME 70 ms

CANNM_MSG_REDUCED_TIME Node 1: 40 ms

CANNM_MSG_REDUCED_TIME Node 2: 50 ms

CANNM_MSG_REDUCED_TIME Node 3: 60 ms

2.4 User Data

➤ 简介

User Data用于存放用户的自定义数据。

- ① 发送网络管理报文时，若CanNmComUserDataSupport设置为TRUE，CanNm通过PduR来获取user data数据；否则通过专用的接口 (CanNm_SetUserData) 设置数据；
- ② 接收网络管理报文时，通过专用的接口 (CanNm_GetUserData) 获取user data数据

➤ 相关配置

CanNmUserDataEnabled		开启/关闭user data功能。 等于NmUserDataEnabled
CanNmComUserDataSupport		是否通过Com层获取user data数据。 为TRUE表示通过 PduR_CanNmTriggerTransmit获取user data。 为FALSE表示通过CanNm_SetUserData获取 user data。
CanNmUser DataTxPdu	CanNmTxUserDataPduRef	User data关联的Pdu
	CanNmTxUserDataPduId	User data关联的Pdu的Id

2.5 Remote Sleep

➤ 简介

该功能是为了实现Nm层的Coordinated Shutdown功能。只会在actively channel收到。

➤ 方法

- ① 处于*Normal Operation State*的节点一段时间内未收到任何报文，会调用Nm_RemoteSleepIndication向Nm层上报Remote Sleep Indication，表示该网络内的其他节点都准备睡眠了；
- ② 若已经上报了该信息，处于*Normal Operation State*或*Ready Sleep State*的节点**接收到网络管理报文**后，将会调用Nm_RemoteSleepCancellation以取消该信息，表示网络内有节点不想睡眠；
- ③ 若已经上报了该信息，处于*Normal Operation State*或*Ready Sleep State*的节点**切换到Repeat Message State**，将会调用Nm_RemoteSleepCancellation以取消该信息，表示网络内有节点不想睡眠；

2.5 Remote Sleep

➤ 限制

只针对Gateway节点。

➤ 相关配置

CanNmRemoteSleepIndEnabled	开启/关闭Remote Sleep indication功能
CanNmRemoteSleepIndTime	该时间内未收到任何报文就向Nm上报。 $\text{CanNmRemoteSleepIndTime} \geq \text{CanNmMsgCycleTime}$

2.6 Coordinator Synchronization

➤ 简介

该功能是为了实现Nm层的Coordinated Shutdown功能。

➤ 方法

- ① 若进入了*Network Mode*，或先前调用过Nm_CoordReadyToSleepCancellation，则第一次收到NmCoordinatorSleepReady Bit为1的网络管理报文时，CanNm调用Nm_CoordReadyToSleepIndication通知NM层，主网关请求开始睡眠；
- ② 若*在Network Mode下的节点*调用过Nm_CoordReadyToSleepIndication，则第一次收到NmCoordinatorSleepReady Bit为0的网络管理报文时，CanNm调用Nm_CoordReadyToSleepCancellation通知NM层，主网关取消睡眠；

➤ 限制

只针对存在Nested Gateway节点场景。

2.6 Coordinator Synchronization

➤ 相关配置

CanNmCoordinatorSyncSupport	开启/关闭Coordinator Synchronization功能
-----------------------------	------------------------------------

2.7 Partial Networking

➤ 简介

该功能是为ComM层的Partial Networking (PN) 提供支持，主要是将收到的PN请求通过Com模块上传到ComM, 并为PN请求提供复位机制。PN请求根据应用场景的不同，分为2类。

- ① EIRA: 用于需要切换I-Pdu-Group的节点;
- ② ERA: 用于Gateway节点, 用于转发PN请求;

➤ 方法

- ① 对于EIRA, 若发出/接收的报文中, 存在与本节点相关(与配置的mask按位与)的PN请求。CanNm应该存储该Pn信息, 并更新定时器, 并将本地PN请求上传到ComM。
- ② 对于ERA, 若接收的报文中, 存在与本节点相关(与配置的mask按位与)的PN请求。CanNm应该存储该Pn信息, 并更新定时器, 并将本地PN请求上传到ComM。
- ③ CanNm为每个PN请求都设置了一个定时器, 当定时器超时, 应该清除本地PN请求, 并将本地PN请求上传到ComM。

2.7 Partial Networking

➤ 限制

该功能必须开启user data功能，PN Info Range是作为user data中的一个signal上传到上层的。

➤ 相关配置

CanNmPnResetTime	CanNmPnResetTime > CanNmMsgCycleTime CanNmPnResetTime < CanNmTimeoutTime
CanNmPnEiraRxNSduRef	EIRA关联到Com层的Pdu
CanNmPnEraCalcEnabled	使能EIRA相关功能

CanNmAllNmMessageKeepAwake	<ol style="list-style-type: none"> 若配置为TRUE： <ol style="list-style-type: none"> ① 接收到报文中PNI为0，则不丢弃该报文。 ② 接收的报文中PNI为1，但无与还节点相关的PN请求，则不丢弃该报文。 配置为FALSE： <ol style="list-style-type: none"> ① 接收到报文中PNI为0，则丢弃该报文。 ② 接收的报文中PNI为1，但无与还节点相关的PN请求，则丢弃该报文。 主要针对Gateway，确保它收到任何报文都能保持网络。
CanNmPnEraCalcEnabled	使能ERA功能
CanNmPnHandleMultipleNetworkRequests	若该参数为TRUE，则Normal Operation或Ready Sleep state下重新请求网络，将进入Repeat Message State状态。
CanNmPnEraNSduRef	ERA关联到Com层的Pdu
CanNmPnInfoOffset	PN Info Range网络管理报文中的偏移
CanNmPnInfoLength	PN Info Range的长度(字节)
CanNmPnFilterMaskByteIndex	Index of filter mask byte
CanNmPnFilterMaskByteValue	Filter mask byte

Part 3

Nm

3.1 Nm简介



3.2 基本的网络管理



3.3 State Change Notification



3.4 Car Wake Up



3.5 Coordinator Shutdown



3.1 Nm简介

➤ 简介

Nm模块为ComM和CanNm之间的桥梁，为上层屏蔽了具体的网络管理细节；同时，协调多个网络之间的同步睡眠(Coordinated shutdown)。

➤ Nm主要功能有：

- ① 基本网络管理功能
- ② State Change Notification
- ③ Car Wake Up
- ④ Coordinated Shutdown

3.2 基本的网络管理

➤ 支持总线类型

Can, J1939, Flexray, Lin以及Udp。

➤ 主要接口：

<BusNm>_PassiveStartup	<BusNm>_GetPduData
<BusNm>_NetworkRequest	<BusNm>_RepeatMessageState
<BusNm>_NetworkRelease	<BusNm>_GetNodeIdentifier
<BusNm>_DisableCommunication	<BusNm>_GetLocalNodeIdentifier
<BusNm>_EnableCommunication	<BusNm>_CheckRemoteSleepIndication
<BusNm>_SetUserData	<BusNm>_GetState
<BusNm>_GetUserData	<BusNm>_GetVersionInfo
<BusNm>_RequestBusSynchronization	<BusNm>_SetSleepReadyBit

3.3 State Change Notification

➤ 简介

NM提供了一种机制，通过网络管理报文，将NM的内部状态指示出来。

➤ 方法

通过关联Com内部的一个信号，CanNm状态修改时，就修改该signal的值，并发送出去。该signal一般处于CanNm的user data区域，每次发出网络管理报文时，一同发送出去。信号的值与状态的对应关系如下。

Bit	Value	Name	Description
0	1	NM_RM_BSM	NM in state RepeatMessage (transition from BusSleepMode)
1	2	NM_RM_PBSM	NM in state RepeatMessage (transition from PrepareBusSleepMode)
2	4	NM_NO_RM	NM in state NormalOperation (transition from RepeatMessage)
3	8	NM_NO_RS	NM in state NormalOperation (transition from ReadySleep)
4	16	NM_RM_RS	NM in state RepeatMessage (transition from ReadySleep)
5	32	NM_RM_NO	NM in state RepeatMessage (transition from NormalOperation)

3.3 State Change Notification

➤ 相关配置

NmStateReportEnabled	是否支持state change Notification
NmStateReportSignalRef	Reference to the signal for setting the NMS by calling Com_SendSignal for the respective channel.

3.4 Car Wake Up

➤ 简介

该功能与CanNm Car WakeUp功能协同工作。

➤ 处理流程

- ① 若CanNm配置CanNmCarWakeUpFilterEnabled为FALSE，接收的网络管理报文中CWB为1，CanNm调用Nm_CarWakeUpIndication。
- ② 若CanNm配置CanNmCarWakeUpFilterEnabled为TRUE，接收的网络管理报文中CWB为1且NID等于CanNmCarWakeUpFilterNodeId，CanNm调用Nm_CarWakeUpIndication。

➤ 相关配置

NmCarWakeUpRxEnabled	使能/禁止Car Wake Up 功能
NmCarWakeUpCallout	Nm_CarWakeUpIndication被调用时，若配置了该回调函数，Nm就调用该函数。否则调用BswM_CarWakeUpIndication

3.5 Coordinated Shutdown

➤ 背景

通过上面的介绍，CanNm能够保证一个网络中所有节点同步睡眠。但存在Gateway时，该节点上必然存在多种类型的网络，那要实现不同网络上节点的同步睡眠，又该如何做呢。

➤ 方法

这就需要用到Nm模块的Coordinated Shutdown功能。用于同步关闭不同的网络。

➤ 依赖

该功能除了需要CanNm的网络管理功能之外，还需要Remote Sleep(作为Nm处理Cluster中 Active Channel 关闭的条件)以及Coordinator Synchronization功能的支持。

3.5 Coordinated Shutdown

➤ 简介

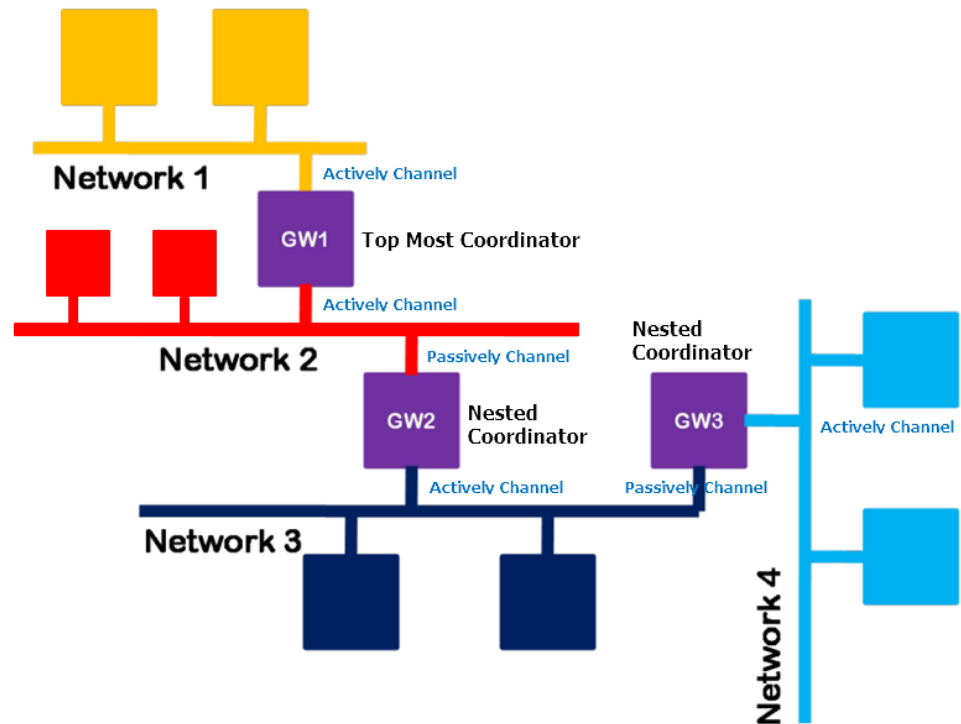
首先介绍几个概念。

- ① **Actively Channel**: NMActiveCoordinator为TRUE的通道; 该通道只会发出NmCoordinatorSleepReady Bit为1/0的报文
- ② **Passively Channel**: NMActiveCoordinator为FALSE的通道; 该通道只会接收NmCoordinatorSleepReady Bit为1/0的报文
- ③ **Top Most Coordinator**: 该Gateway是主Gateway, 用于控制睡眠操作的启停。所有通道都是Actively Channel。
- ④ **Nested Coordinator**: 该Gateway是除Top Most Coordinator的其他Gateway。只存在一个Passively Channle和多个Actively Channel。

3.5 Coordinated Shutdown

➤ 睡眠流程

如图所示，是一个包含了3个网关的网络。其中的GW1作为Top Most Coordinator，与之相连的channel全部是Actively channel。GW2和GW3是Nested Coordinator。



3.5 Coordinated Shutdown

➤ 睡眠流程

- ① Net4的所有节点释放网络，进入Ready Sleep State。CanNmRemoteSleepIndTime后，GW3收到Actively channel的Nm_RemoteSleepIndication;

注意，这个时候，由于GW3的Actively Channel还未释放，任然处于Normal Operation State，会一直发出报文，故Net4的节点会一致处于Ready Sleep State。

- ② GW3 Nm模块判断出与之相连的所有Actively Channel都收到了Nm_RemoteSleepIndication，且GW准备睡眠，则释放GW3的Passively Channel;
- ③ Net3的所有节点释放网络，进入Ready Sleep State。由于步骤②中GW3已经释放了Passively Channel，不发出报文。因此，CanNmRemoteSleepIndTime后，GW2收到Actively channel的Nm_RemoteSleepIndication;
- ④ GW2 Nm模块判断出与之相连的所有Actively Channel都收到了Nm_RemoteSleepIndication，且GW准备睡眠，则释放GW2的Passively Channel;

3.5 Coordinated Shutdown

➤ 睡眠流程

- ⑤ Net2的所有节点释放网络，进入Ready Sleep State。由于步骤④中GW2已经释放了Passively Channel，不发出报文。因此，CanNmRemoteSleepIndTime后，GW1收到Net2 Actively channel的Nm_RemoteSleepIndication;
- ⑥ Net1的所有节点释放网络，进入Ready Sleep State。GW1收到Net1 Actively channel的Nm_RemoteSleepIndication;
- ⑦ 到此，除GW1、GW2和GW3的Actively Channel外，所有普通节点都释放了网络。若GW1也准备睡眠，GW1将发出一帧报文，其中的NmCoordinatorSleepReady Bit为1，并通过Actively Channel发出，Nested Coord从其Passively Channel收到后，从其Actively Channel发出，并传遍所有的GW。

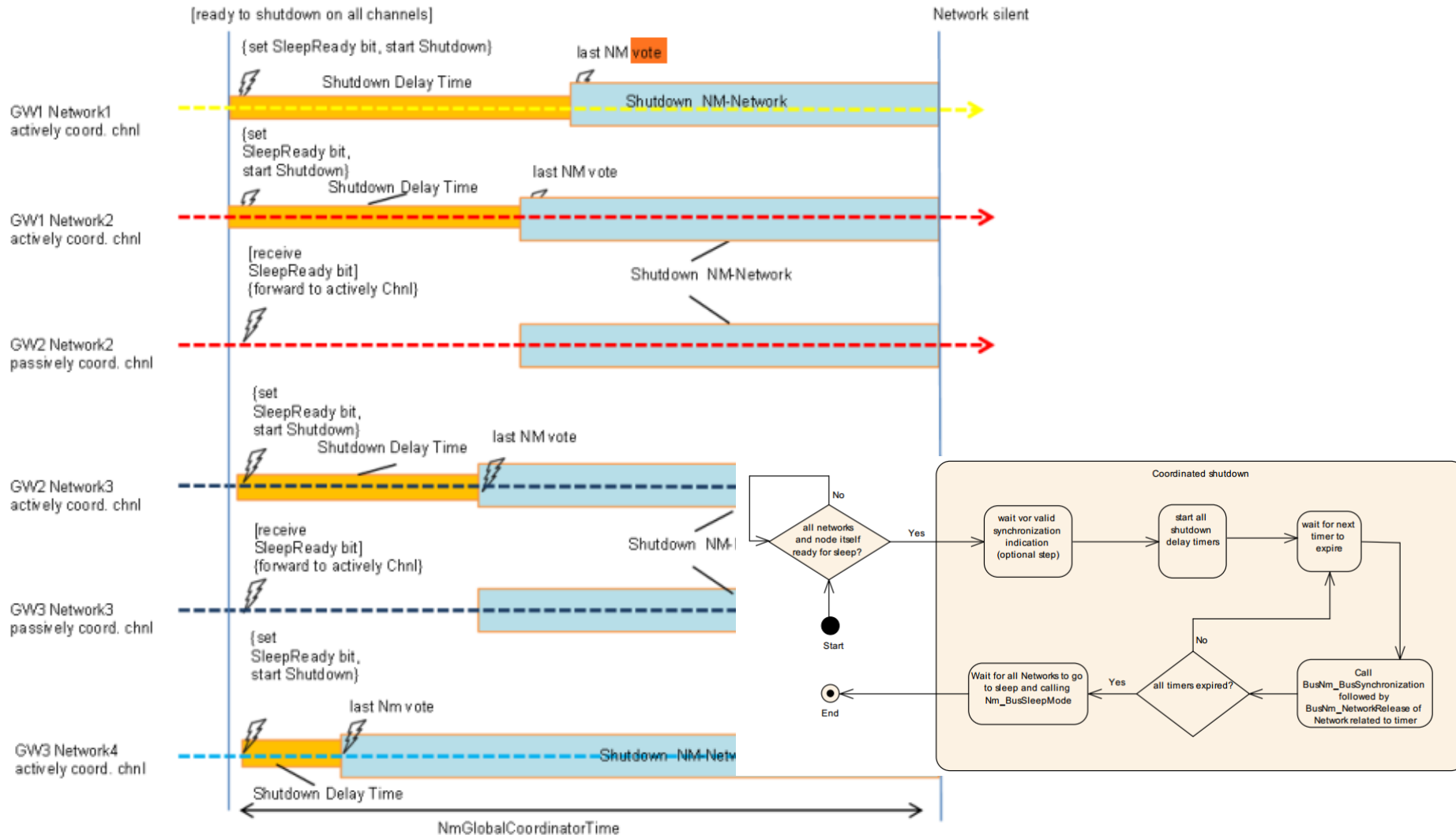
3.5 Coordinated Shutdown

➤ 睡眠流程

- ⑧ 步骤⑦发出这帧报文后，GW1会启动每个通道shutdown delay Timer，GW2和GW3收到CRB为1的报文后，也会启动actively channel的shutdown delay Timer。该定时器超时后，GW在每个通道调用BusNm_BusSynchronization、BusNmNetworkRelease。
- ⑨ 对于Can，BusNm_BusSynchronization会发出一帧网络管理报文。它可以同步网络内各节点的NM-Timeout，使网络内节点同步进入Ready Sleep State。
- ⑩ 经过以上步骤，经过CanNmTimeoutTime和CanNmWaitBusSleepTime后同步进入Bus Sleep Mode。

注意：一个网络中，CanNmTimeoutTimer和CanNmWaitBusSleepTime必须一致，但不同channel之前却可以不一致。为此，Nm提供了一个shutdown delay timer，用于实现多个channel之前的同步睡眠。

3.5 Coordinated Shutdown



3.5 Coordinated Shutdown

睡眠流程

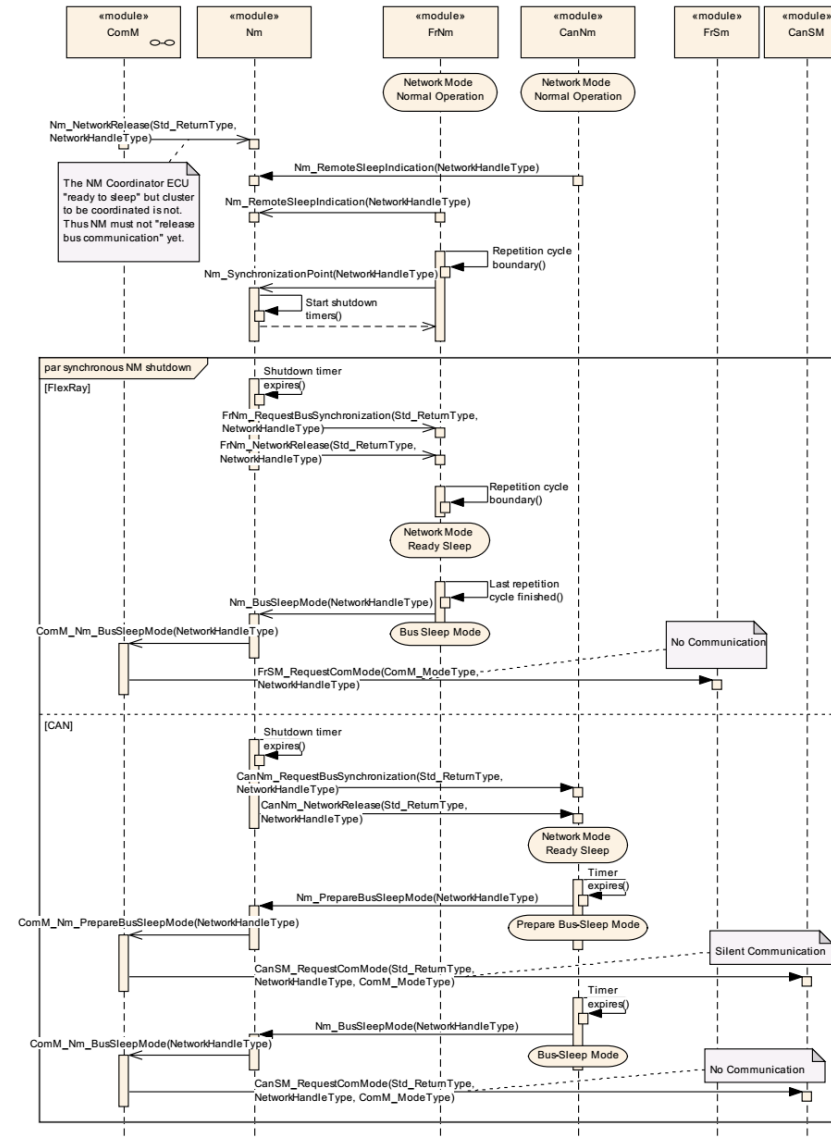


Figure 9.1: Nm Coordination

3.5 Coordinated Shutdown

➤ 中止睡眠

GW通过以下几种途径来中止睡眠：

- ① 处于Network Mode的一个节点重新请求网络，GW中的NM会收到Nm_RemoteSleepCancellation()
 - ② 处于Prepare Bus-Sleep Mode的节点重新请求网络，GW中的NM会收到Nm_NetworkMode()
 - ③ GW上层的ComM模块在NM Coordination Cluster上请求网络
 - ④ GW从Passively Channel收到CRB=0的报文，NM会调用Nm_CoordReadyToSleepCancellation
- (SWS_Nm_00183)若Coordinated Shutdown被中止，NM应该为处于Bus Sleep Mode的节点调用ComM_Nm_RestartIndication;
- (SWS_Nm_00182)若Coordinated Shutdown被中止，NM应该为未处于Bus Sleep Mode的节点重新请求网络。

Part 4

ComM

4.1 ComM简介



4.2 Channel State Machine



4.3 PN State Machine



4.4 ComM Communication inhibition



4.1 ComM简介

➤ 简介

ComM模块简化用户对通信栈和网络管理的使用，同时协调一个ECU上多个独立的软件对总线通信模型的分时复用。

➤ ComM主要功能有：

- ① Channel State Machine
- ② PN State Machine

4.2 Channel State Machine

➤ 简介

ComM为每一个通道维护了一个状态机以达到对每个通道的单独管理功能。ComM状态机有三个主状态：COMM_NO_COMMUNICATION(默认状态)、COMM_SILENT_COMMUNICATION和COMM_FULL_COMMUNICATION。

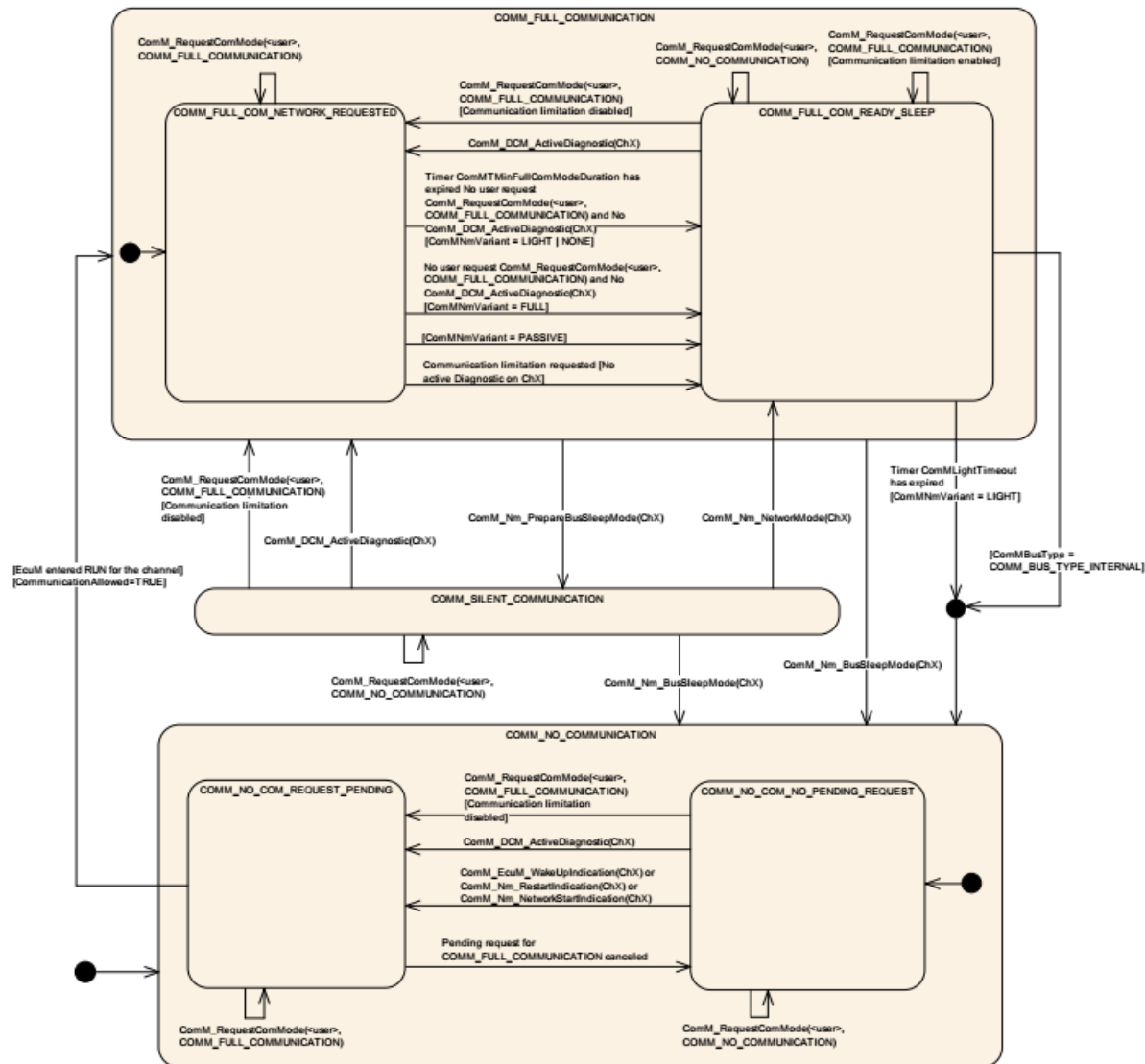
① COMM_NO_COMMUNICATION包含两个子状态

- ✓ COMM_NO_COM_REQUESTED_PENDING
- ✓ COMM_NO_COM_NO_PENDING_REQUEST

② COMM_FULL_COMMUNICATION包含两个子状态

- ✓ COMM_FULL_COM_NETWORK_REQUESTED
- ✓ COMM_FULL_COM_READY_SLEEP

4.2 Channel State Machine



4.2.1 ComM Network management dependencies

➤ 简介

配置参数ComMNmVariant取值为None、LIGHT、PASSIVE和FULL。各参数的意思如下。

NM variant	Keep bus awake capability	Shutdown synchronization
NONE		No shutdown synchronization by ComM. Shutdown by switching off the power of the ECU.
LIGHT		Shutdown synchronization by ComM with means of a timeout (configured with ComMNmLightTimeout, ECUC ComM 00606)
PASSIVE	ECU is not allowed to keep the bus awake	Shutdown synchronization by ComM with means of AUTOSAR NM.
FULL	ECU is allowed to keep the bus awake.	Shutdown synchronization by ComM with means of AUTOSAR NM.

4.2.2 开始发送/接收

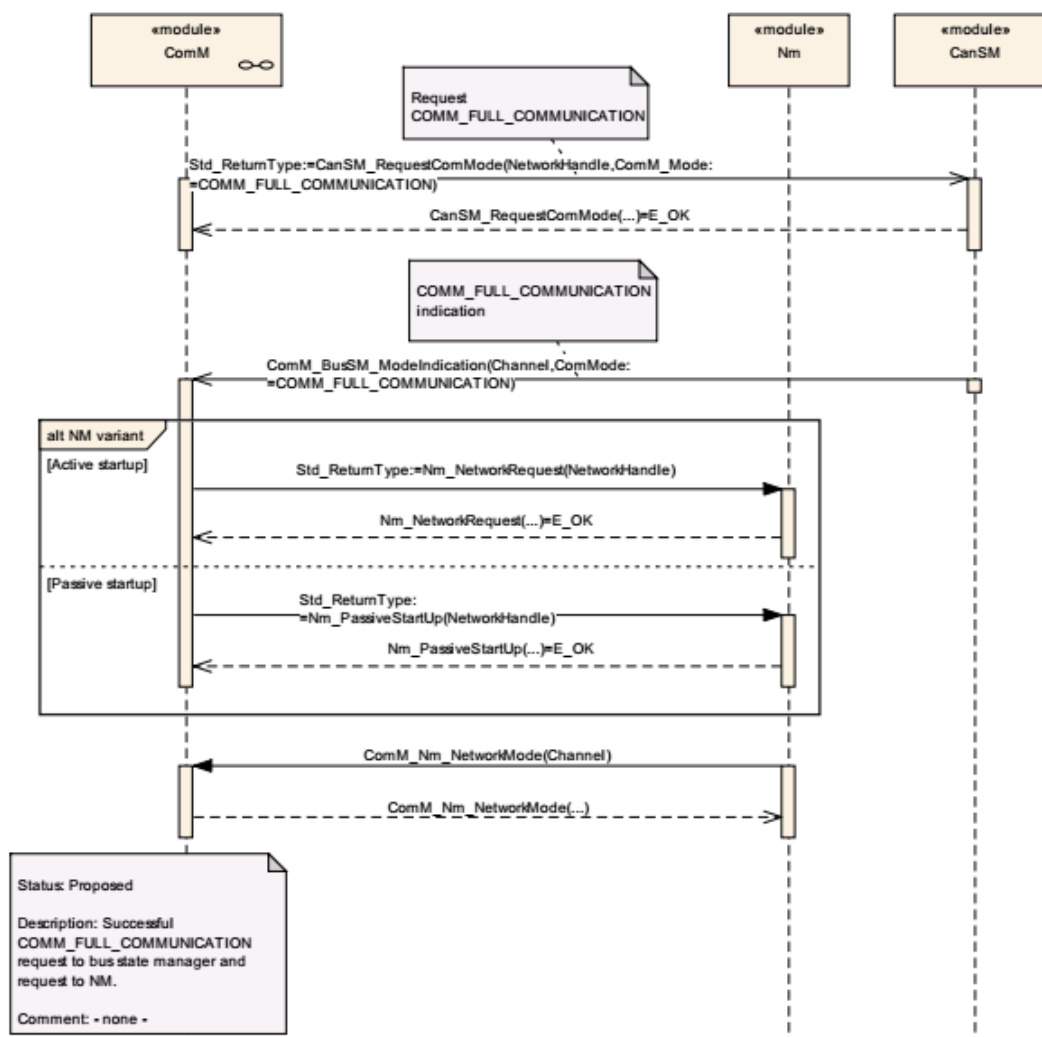
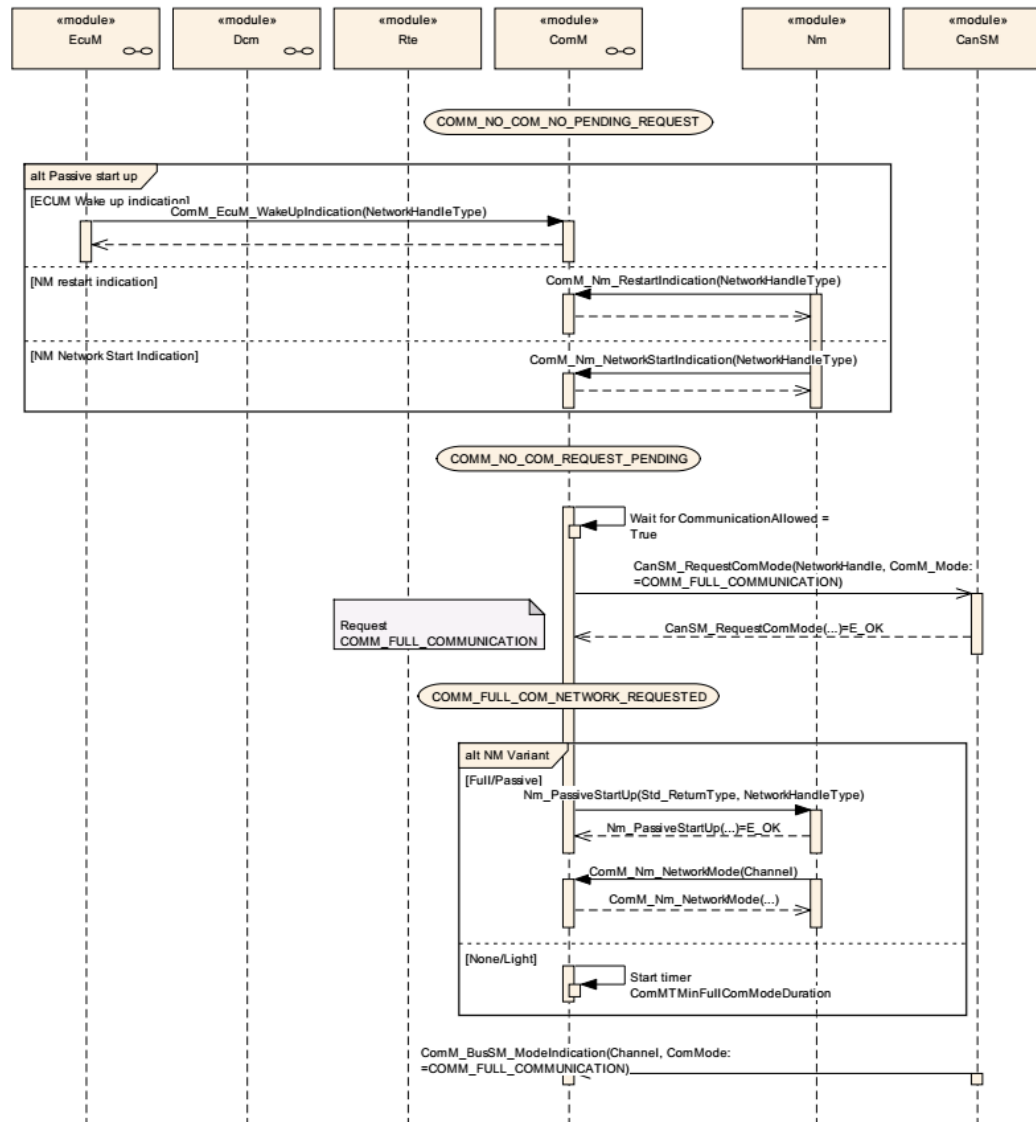
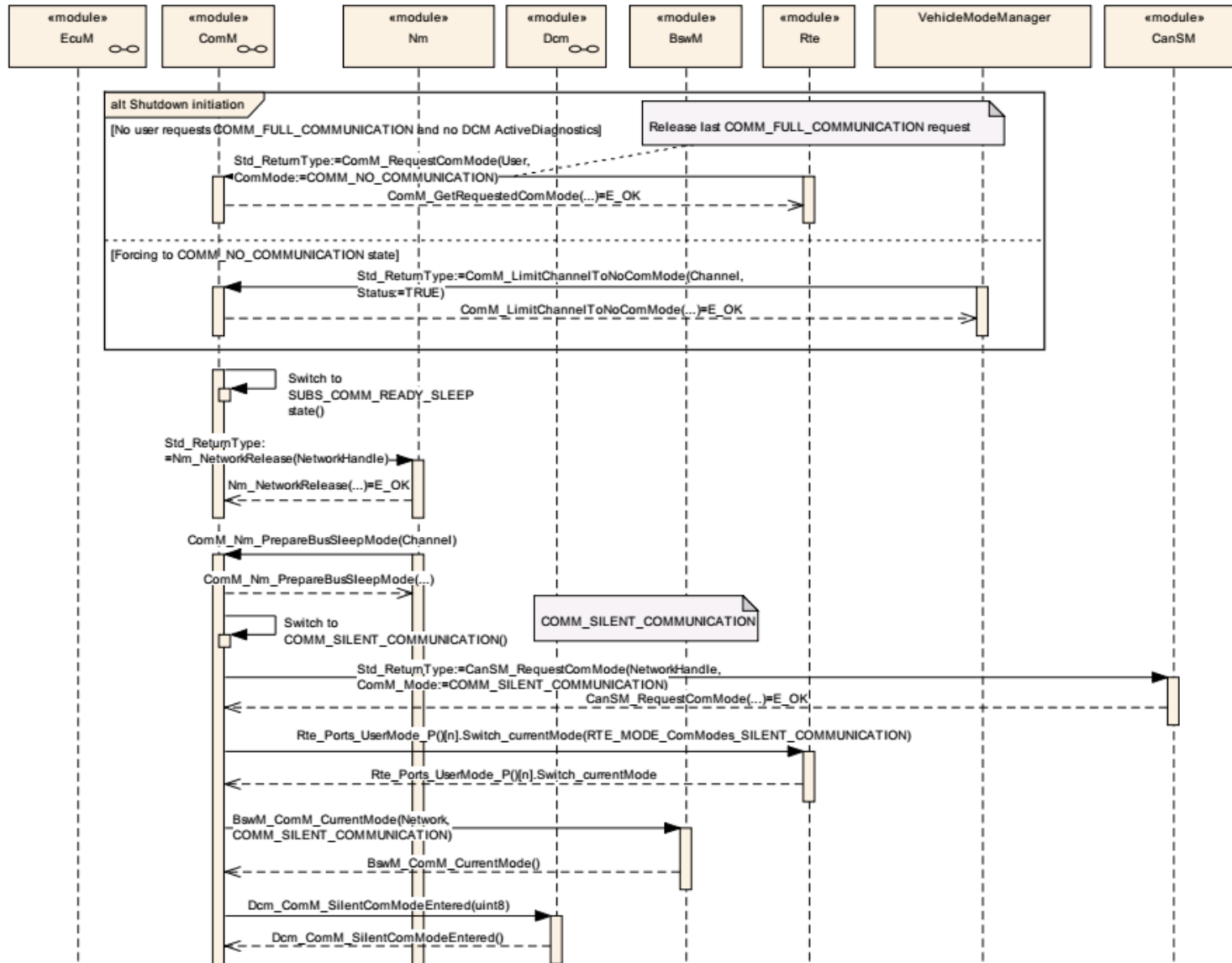


Figure 9: Starting transmission and reception on CAN

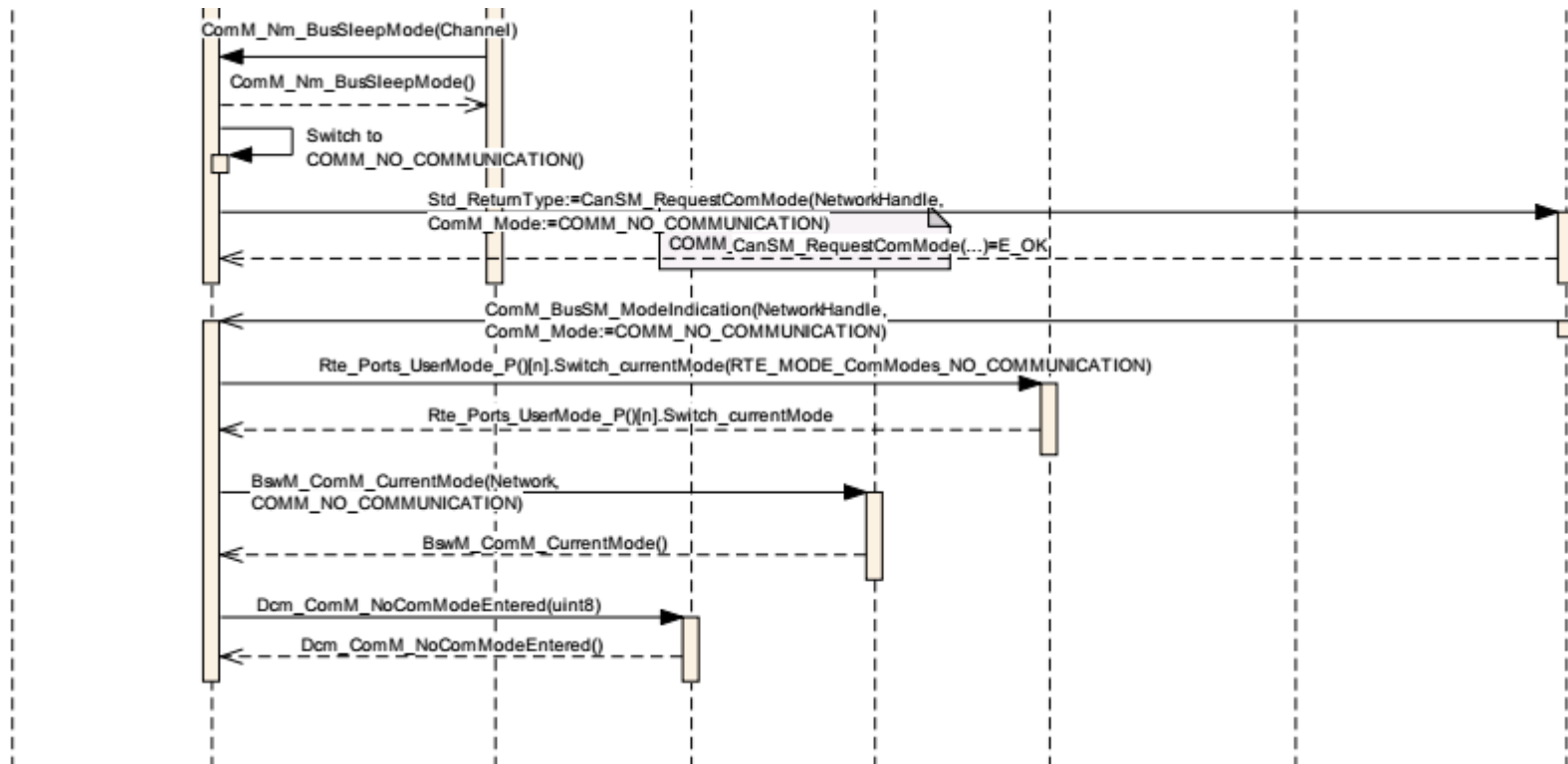
4.2.3 Passive Wake-Up



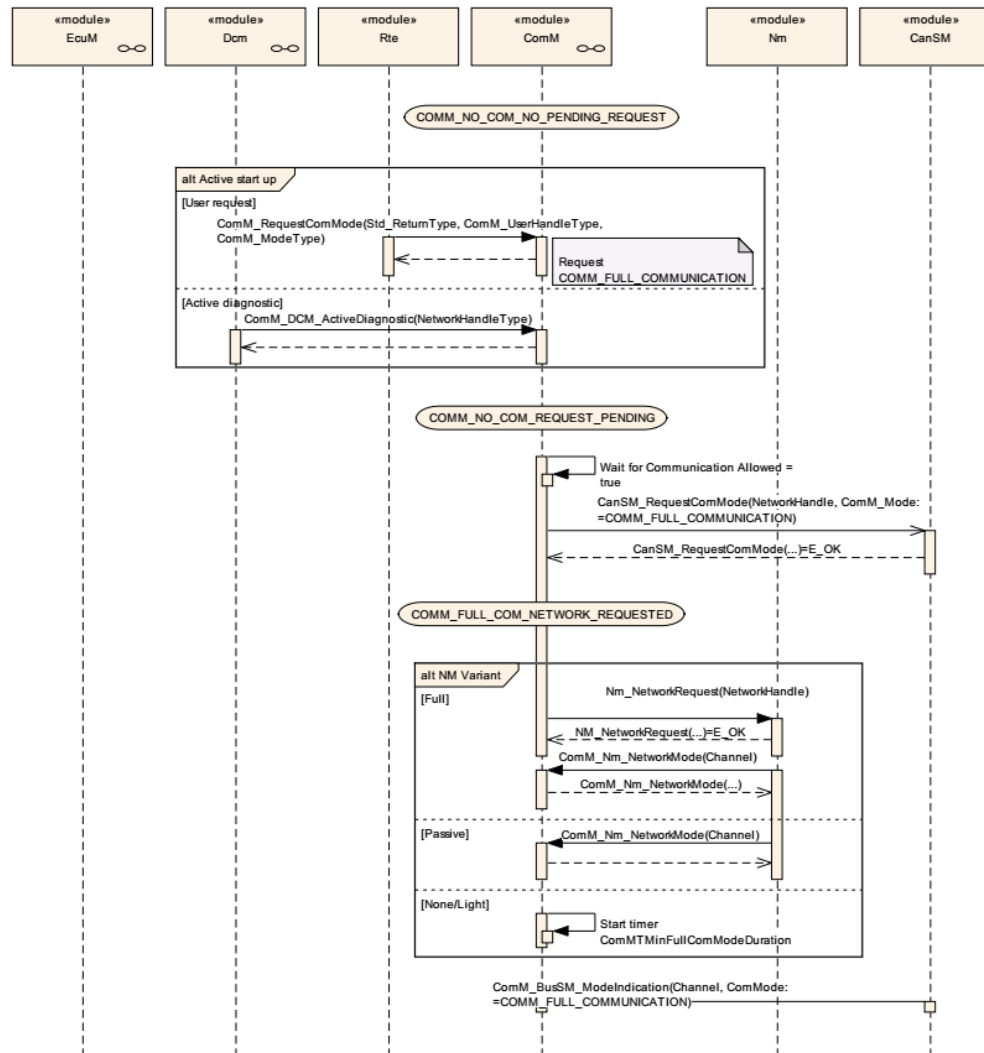
4.2.4 Network Shutdown



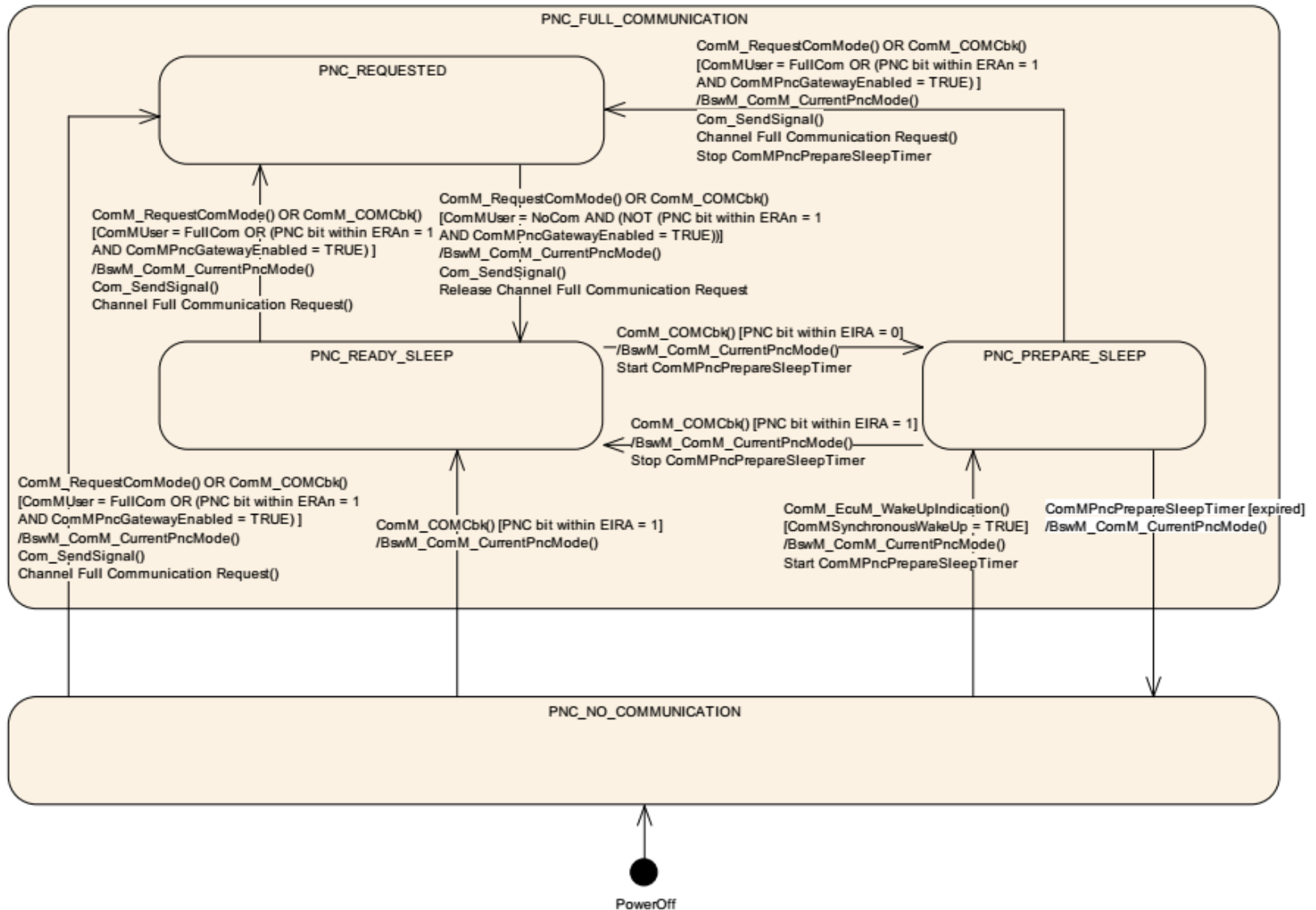
4.2.4 Network Shutdown



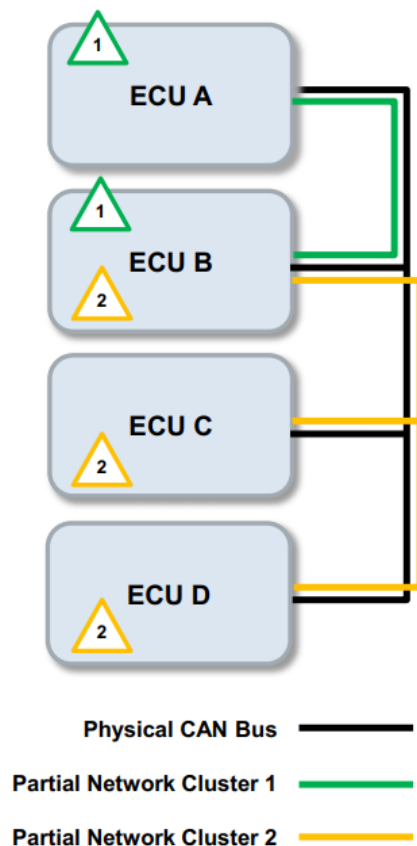
4.2.5 Communication Request



4.3 PN State Machine



4.3.1 PNC的应用场景



Initial situation:

- ECUs “A” and “B” are members of Partial Network Cluster (PNC) 1. ECUs “B”, “C” and “D” are members of PNC 2.
- All functions of the ECUs are organized either in PNC 1 or PNC 2.
- Both PNCs are active.
- PNC 2 is only requested by ECU “C”.
- The function requiring PNC 2 on ECU “C” is terminated, therefore ECU “C” can release PNC 2.

This is what happens:

- ECU “C” stops requesting PNC 2 to be active.
- ECUs “C” and “D” are no longer participating in any PNC and can be shutdown.
- ECU “B” ceases transmission and reception of all signals associated with PNC 2.
- ECU “B” still participates in PNC 1. That means it remains awake and continues to transmit and receive all signals associated with PNC 1.
- ECU “A” is not affected at all.

4.4 ComM Communication inhibition

➤ Bus wake up inhibition

- ① 避免被错误的唤醒。表现为非 COMM_FULL_COMMUNICATION 不处理User对网络的Full请求。根据需要，抑制ECU从No Communication到Full Communication的转化。即，忽略Full Communication/Diagnostic active的请求。
- ② 配置参数：ComMNoWakeup
- ③ 运行调用接口：ComM_PreventWakeUp

➤ Limit to COMM_NO_COMMUNICATION mode

- ① channel强制进入Ready sleep子状态，忽略User对在COMM_FULL_COM_NETWORK_REQUESTED 状态下对Channe的网络请求。
- ② 配置参数：ComMNoCom
- ③ 运行调用接口：ComM_LimitECUToNoComMode、ComM_LimitChannelToNoComMode

Part 5

CanSm

5.1 CanSm简介



5.2 Network State Machine



5.3 Bus Off Recovery



5.4 与PN相关功能



5.1 CanSm简介

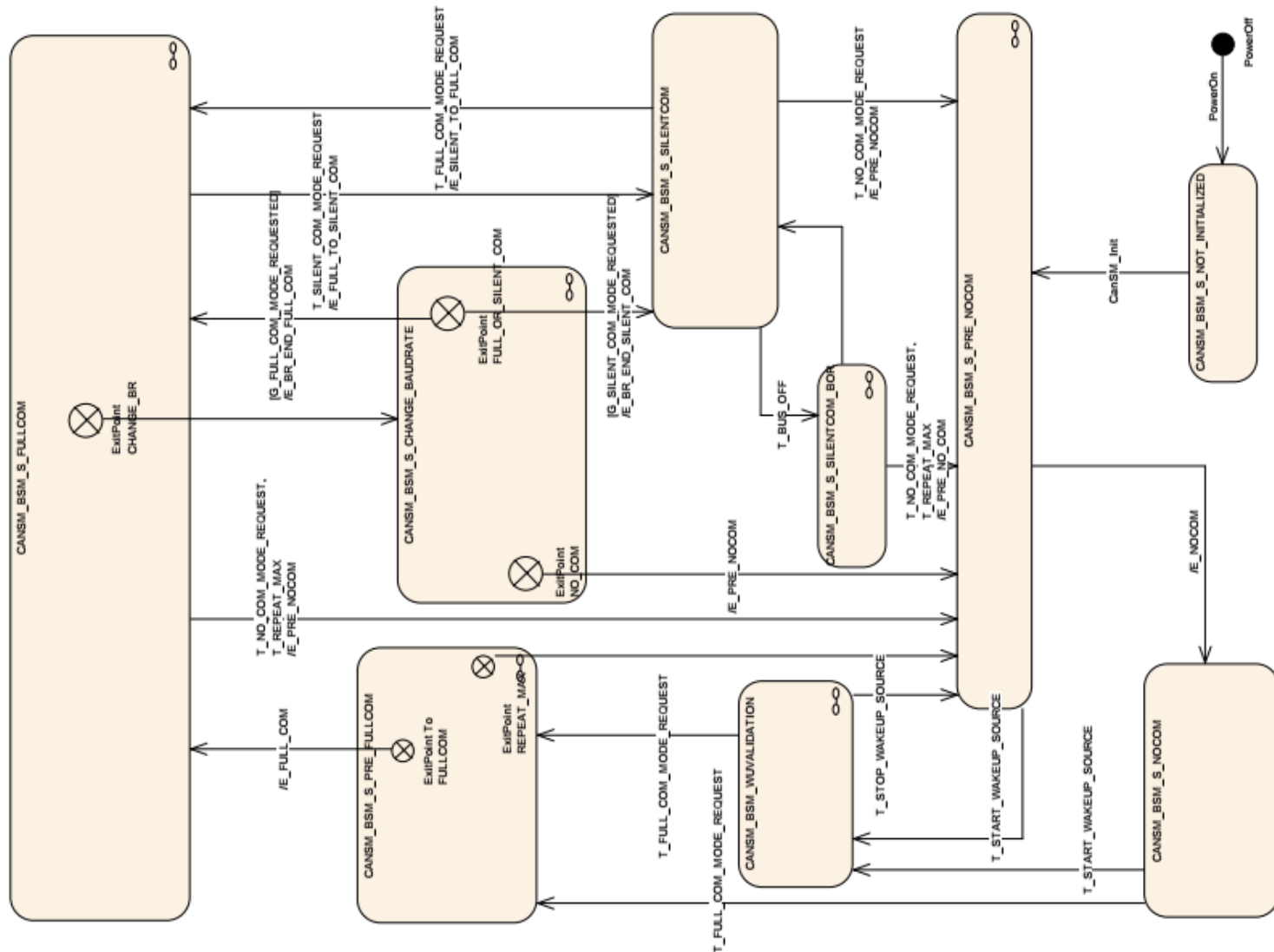
➤ 简介

CanSM为每个CAN网络提供了一个网络通信模式状态机，状态机根据ComM模式设置Control、Transceiver状态。

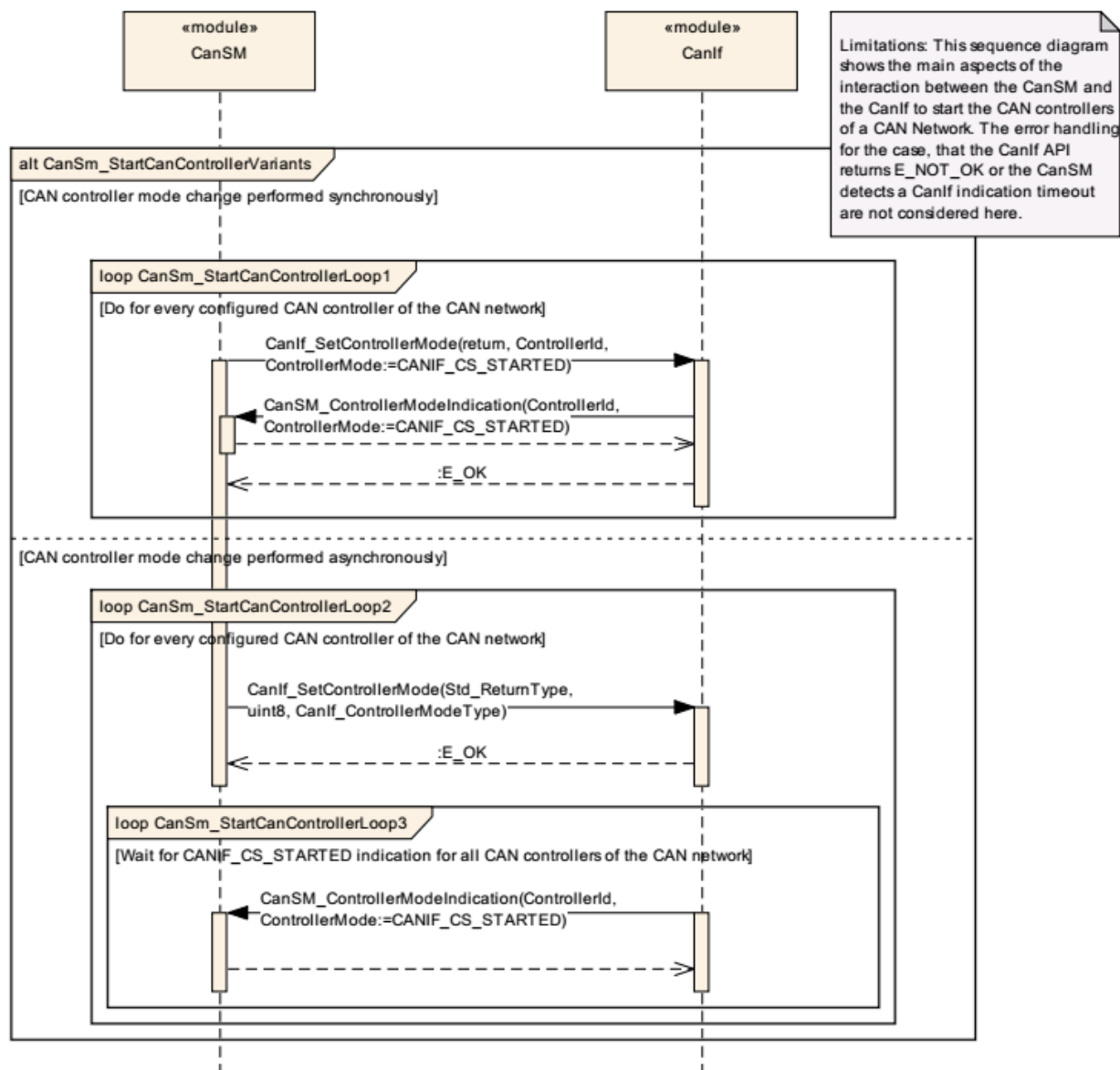
➤ ComM主要功能有：

- ① Network State Machine
- ② Bus Off Recovery
- ③ 与Pn相关的功能

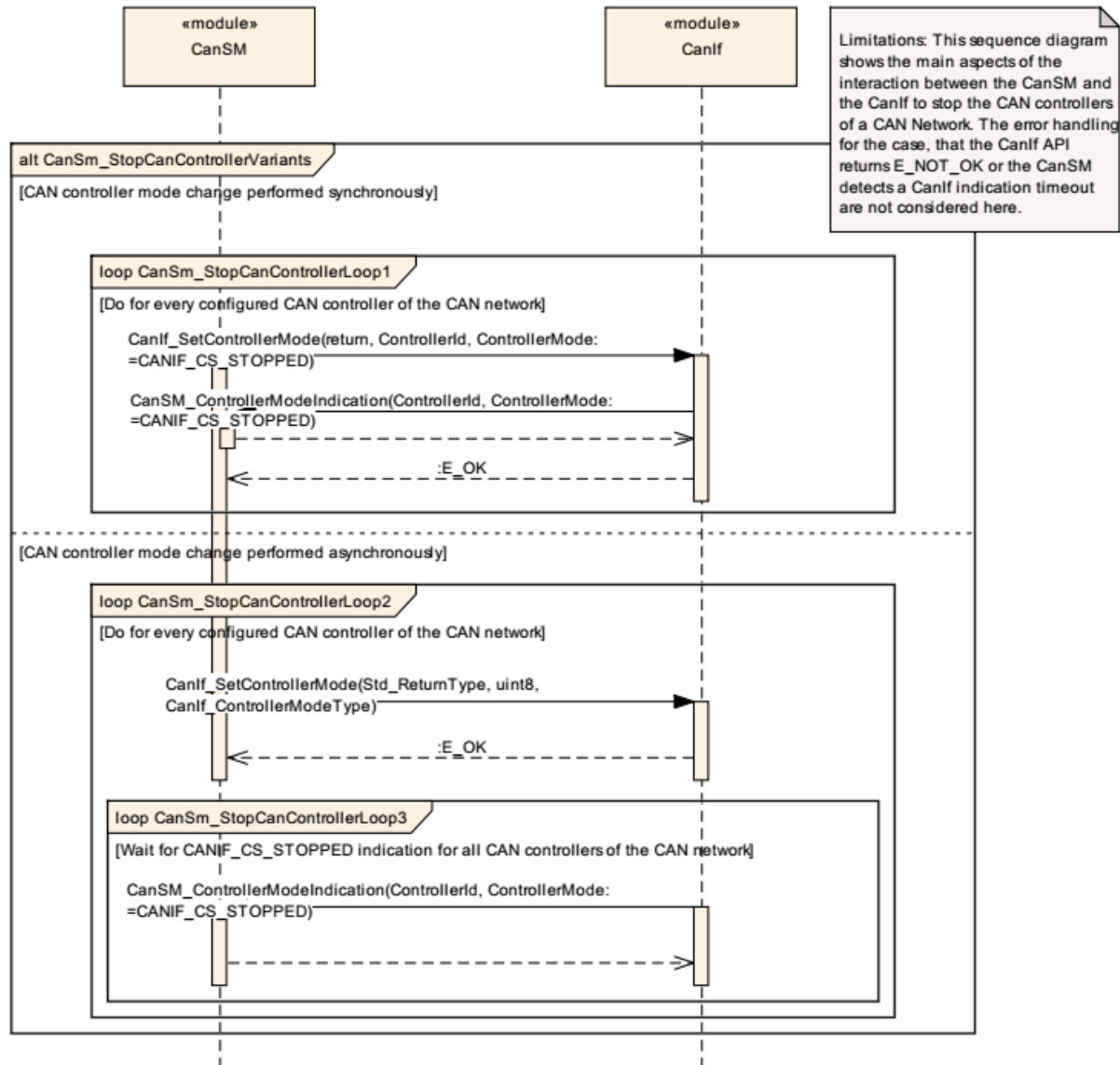
5.2 Network State Machine



5.2.1 启动控制器



5.2.2 停止控制器



5.3 Bus-Off Recovery

➤ Bus-Off恢复流程（以FULL模式下为例）。

① 发生Bus-Off中断。

✓ CanDrv将控制器状态设置成STOP；

✓ CanDrv调用CanIf_ControllerBusOff接口，通知CanIf模块。

② CanIf将控制器状态设置成STOP，将PDU状态设置成TX_OFFLINE；

CanIf调用CanSM_ControllerBusOff接口，通知CanSm模块。

③ CanSm将内部状态机切换到S_RESTART_CC；将内部控制器状态切换到STOP。

④ CanSm将控制器状态切换到START，若切换成功，将状态机切换到S_TX_OFF。

⑤ 在S_TX_OFF模式下维持一定时间(CanSMBorTimeL1 / CanSMBorTimeL2)。

⑥ 若时间到了，将CanIf中PDU状态设置成ONLINE；CanSm将内部状态机切换到S_BUS_OFF_CHECK。

⑦ 确认恢复成功后，CanSm将内部状态机切换到S_NO_BUS_OFF。

5.3 Bus-Off Recovery

➤ 相关配置

CanSmModeRequestRepetitionMax	若CanSm对Can控制器的状态连续请求N次都失败，则返回CANSMBSM_S_PRE_NOCOM状态(T_REPEAT_MAX)
CanSmModeRequestRepetitionTime	连续请求Can控制器的时间间隔
CanSMBorTimeL1	快恢复的时间
CanSMBorTimeL2	慢恢复的时间
CanSmBorCounterL1ToL2	快恢复的次数
CanSMBorTxConfirmationPolling	确保Bus-Off恢复成功
CanSMBorTimeTxEnsured	若持续该参数时间再发生Bus-Off，则确认恢复成功
CanSMBorTimeTxEnsured	<p>依赖：CanSMBorTxConfirmationPolling为FALSE通过调用CanIf_GetTxConfirmationState确保Bus-Off恢复成功。</p> <p>Bus-Off恢复后，需要成功发送一帧报文后才确认恢复成功</p>

5.3 Bus-Off Recovery

- FULL模式下发生Bus-Off。

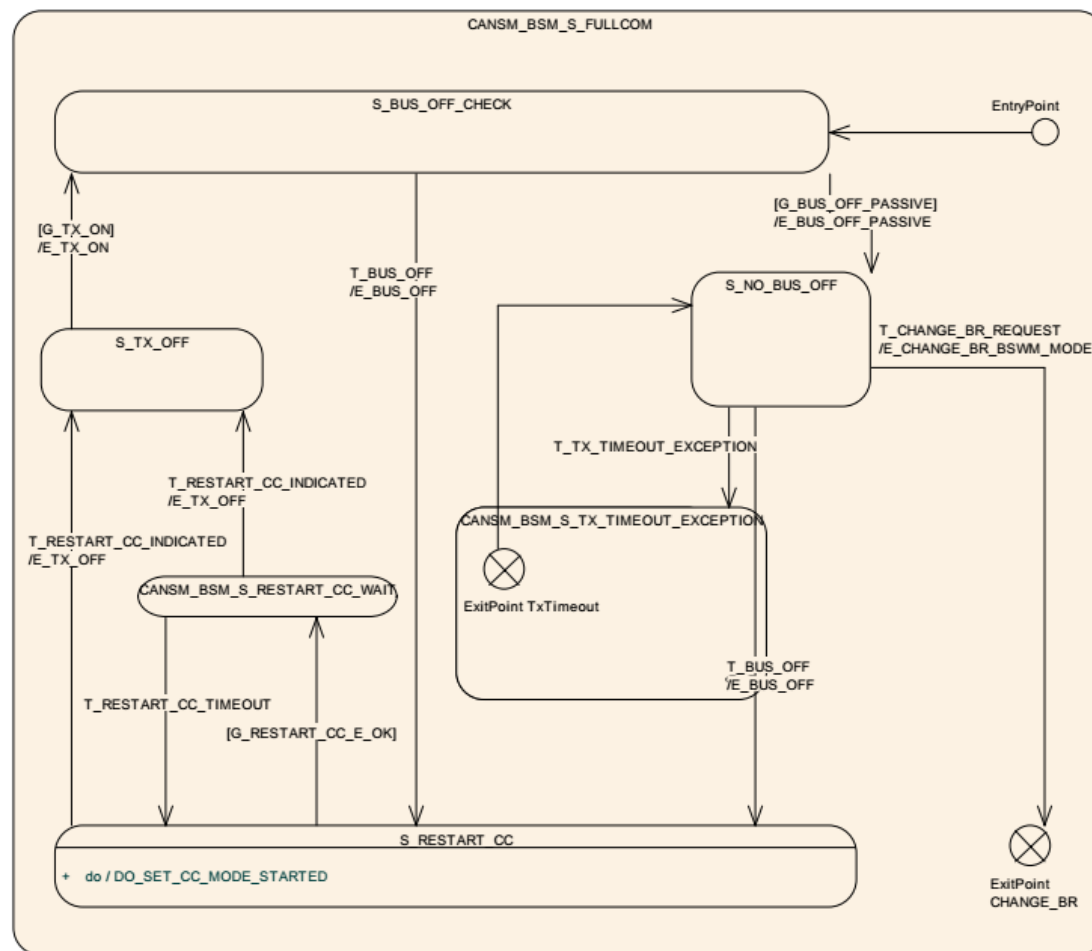
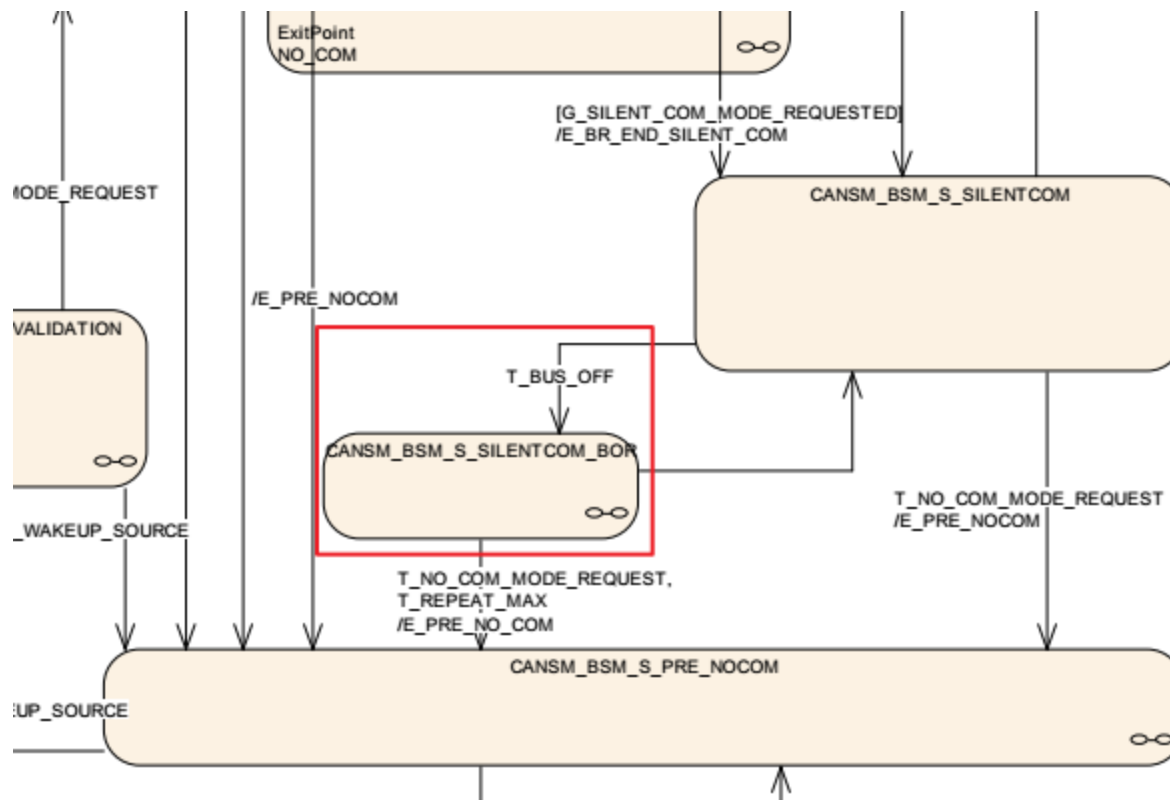


Figure 7-8: CANSM_BSM_S_FULLCOM, sub state machine of CANSM_BSM

5.3 Bus-Off Recovery

➤ SILENT模式下发生Bus-Off



5.3 Bus-Off Recovery

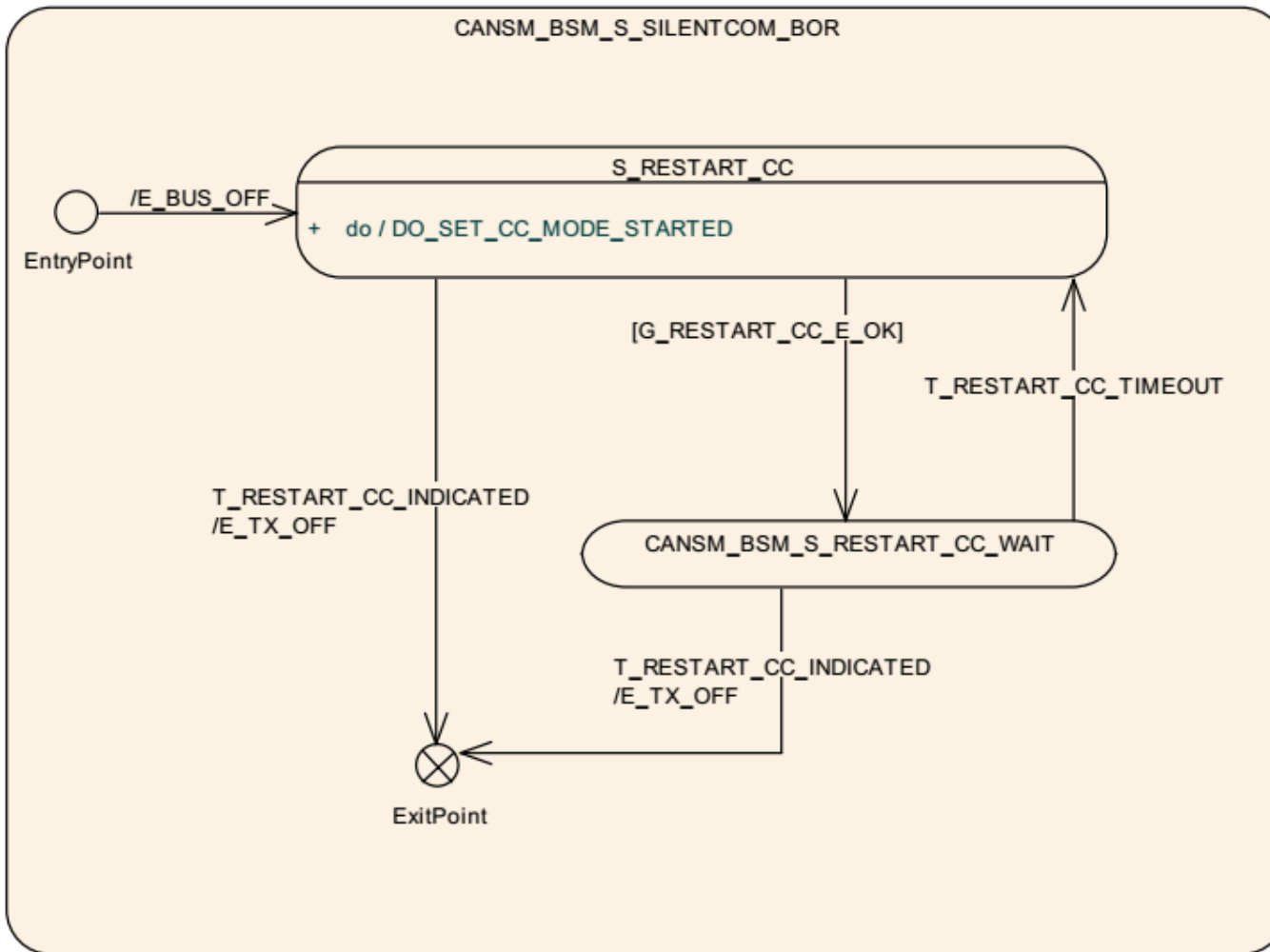
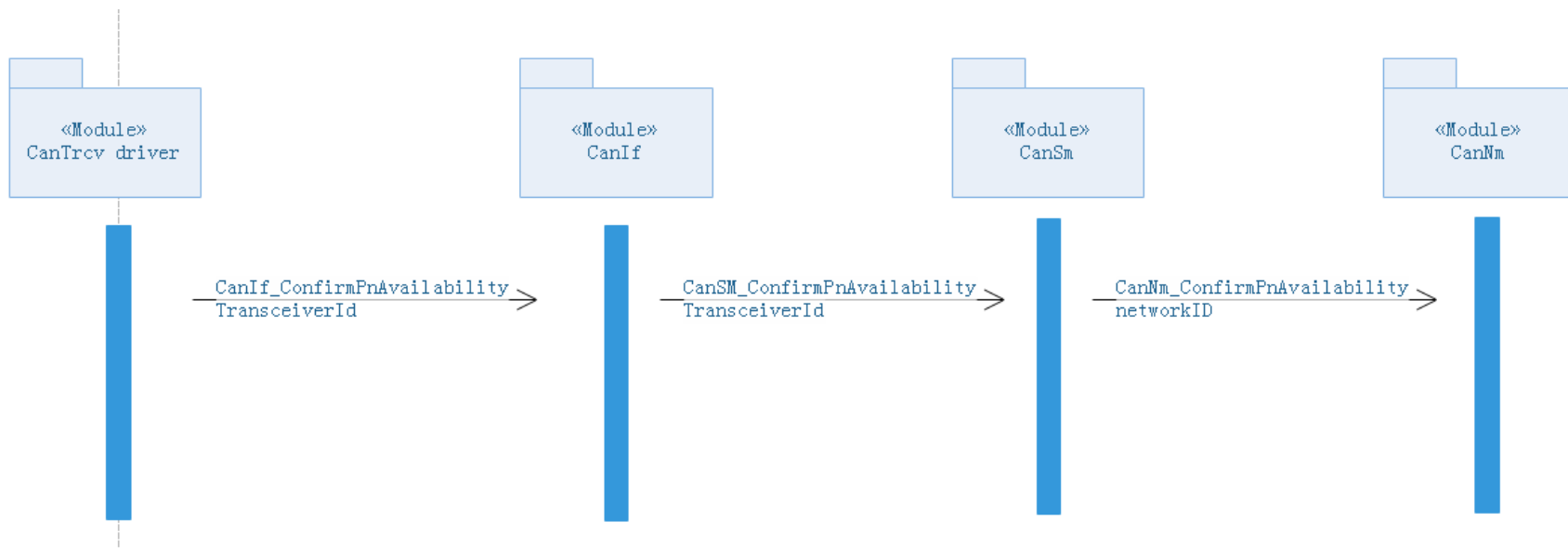


Figure 7-6: CANSM_BSM_S_SILENTCOM_BOR, sub state machine of CANSM_BSM

5.4 与PN相关功能

- 当Transceiver 当前在PN模式, 则通知到CanNm 中用于对接受的PN 类型的NM Pdu进行过滤。



感谢关注!

责任 创新 卓越 共享



网址: www.i-soft.com.cn
信箱: Marketing@i-soft.com.cn
热线: 400-650-9325