

# OPEN Sleep/Wake-up Specification

---

Sleep/Wake-up Specification for Automotive Ethernet



Author & Company	Philip Axer (NXP), Charles Hong (Realtek), Antony Liu (Realtek)
Title	OPEN Sleep/Wake-up Specification
Version	2.0
Date	Feb 21, 2017
Status	Final
Restriction Level	Public

---

**Version Control of Document**

Version	Author	Description	Date
1.0	Antony Liu,	First Draft	05/10/2015
1.1	Philip Axer	Second Draft	17/11/2015
1.3	Philip Axer	Third Draft	11/01/2016
1.4	Charles Hong	Fourth Draft	20/04/2016
1.5	Philip Axer	Updated electrical characteristics of the local wakeup and sleep forwarding interface. Beautification	22/04/2016
1.6	Philip Axer	Integrated review feedback	19/04/2016
1.7	Charles Hong	Added Appendix: System Timing Diagram	28/06/2016
1.8	Philip Axer	Refined Wake pin timing	14/07/2016
1.9	Philip Axer	Added service primitives Variables renamed	05/12/2016
2.0	Philip Axer	Final	21/02/2017

**Restriction level history of Document**

Version	Restriction Level	Description	Date
1.0-1.9	OPEN internal only	Internal Draft	05/12/2016
2.0	Public	Final	21/02/2017

## Contents

1	Disclaimer.....	5
2	Introduction .....	6
2.1	Objectives.....	6
2.2	Relationship to IEEE802.3bw specification .....	6
2.3	Power Sequencing.....	6
3	Wakeup/Sleep Interface Behavior .....	7
4	Timing Behavior .....	8
4.1	ACK_timer .....	8
4.2	REQ_timer.....	8
4.3	TWU_Link_passive .....	8
4.4	TWU_Link_active .....	8
4.5	TWU_Forward_passive .....	8
4.6	TWU_Forward_active .....	8
5	Service Primitives and Interfaces.....	9
5.1	Sleep.request .....	9
5.2	Sleep.indication.....	9
5.3	SleepFail.indication .....	9
5.4	SleepAbort.request.....	9
5.5	Wakeup.request.....	9
5.6	Wakeup.indication .....	9
5.7	Inhibit.indication.....	9
6	Command Definitions .....	9
6.1.1	Low Power Sleep (LPS) .....	10
6.1.2	Wake-Up Request (WUR).....	10
6.1.3	Wake-Up Pulse (WUP) .....	10
7	Annex IEEE802.3bw.....	11
7.1	Service Primitives and Interfaces.....	12
7.2	States.....	12
7.3	Command Definitions .....	13
7.3.1	Low Power Sleep (LPS).....	13
7.3.2	Wake-Up Request (WUR).....	13

7.3.3	Wake-Up Pulse (WUP) .....	13
7.4	Generation of scrambling bits $Sd_n[2:0]$ .....	14
7.5	PCS PHY Control State Diagram .....	14
7.5.1	Variables.....	16
7.5.2	Timers.....	16
Appendix: System Timing Diagram .....		17
A.1	System timing for WUP/WUR forwarding and transmission.....	17

## 1 Disclaimer

**NOTICE TO USERS WHO ARE OPEN ALLIANCE SIG MEMBERS:** Members of OPEN Alliance have the right to use and implement this Specification, subject to the Member's continued compliance with the OPEN Alliance SIG's governance documents, Intellectual Property Rights Policy, and the applicable OPEN Alliance Promoter or Adopters Agreement. OPEN Specification documents may only be reproduced in electronic or paper form or utilized in order to achieve the Scope, as defined in the OPEN Alliance Intellectual Property Rights Policy. Reproduction or utilization for any other purposes as well as any modification of the Specification document, in any form or by any means, electronic or mechanical, including photocopying and microfilm, is explicitly excluded.

**NOTICE TO NON-MEMBERS OF OPEN ALLIANCE SIG:** If you are not a Member of OPEN Alliance and you have obtained a copy of this document, you only have a right to review this document for informational purposes. You do not have the right to reproduce, distribute, make derivative works of, publicly perform or publicly display this document in any way.

All OPEN Specifications are provided on an "as is" basis and all warranties, either explicit or implied, are excluded unless mandatory under law. Accordingly, the OPEN Alliance Members who have contributed to the OPEN Specifications make no representations or warranties with regard to the OPEN Specifications or the information (including any software) contained therein, including any warranties of merchantability, fitness for purpose, or absence of third party rights and make no representations as to the accuracy or completeness of the OPEN Specifications or any information contained therein.

The OPEN Alliance Members who have contributed to the OPEN Specifications will not be liable for any losses, costs, expenses or damages arising in any way out of use or reliance upon any OPEN Specification or any information therein. Nothing in this document operates to limit or exclude any liability for fraud or any other liability which is not permitted to be excluded or limited by operation of law.

The material contained in OPEN Specifications is protected by copyright and may be subject to other types of Intellectual Property Rights.

The distribution of OPEN Specifications shall not operate as an assignment or license to any recipient of any OPEN Specification of any patents, registered designs, unregistered designs, trademarks, trade names or other rights as may subsist in or be contained in or reproduced in any OPEN Specification. The commercial exploitation of the material in this document may require such a license, and any and all liability arising out of use without such a license is excluded.

Without prejudice to the foregoing, the OPEN Alliance Specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

OPEN Alliance reserves the right to withdraw, modify, or replace any OPEN Specification at any time, without notice.

## 2 Introduction

This specification defines new service primitives provided by the ISO/OSI layer 1 (PHY) and supporting a controlled link shutdown and a fast global wake-up within an Ethernet network. Higher layers like the network management can access those service primitives to realize partial networking, where selected parts of a network are inactive. The coordination of switching off selected nodes of a network is handled by the network management and is not part of this specification. This partial networking concept relying on selective link shutdown and fast global wake-up is especially suited for automotive Ethernet networks.

### 2.1 Objectives

The following are the objectives of the Sleep/Wake-up specification:

- a) Comply with the CSMA/CD MAC
- b) Comply with the specifications for the xMII (MII, RMII, RGMII etc.)
- c) Support global network wake-up (incl. link start-up time) within less 250ms
- d) Support wake-up process completely covered in ISO/OSI layer 1
- e) Support controlled link shutdown to deactivate selective parts of network
- f) Comply with Autosar network management
- g) No unwanted wakeup in presence of interference noise
- h) Applicable for 100Base-T1

### 2.2 Relationship to IEEE802.3bw specification

The IEEE802.3bw specification does not define mechanisms for a controlled link shut-down and wake-up. Therefore the new service primitives defined in this specification can be regarded as a supplement to the IEEE802.3bw specification.

The new services primitives make use of LPS, WUR and WUP commands. When not using the new service primitives and commands, implementation of these extensions will not impact the interoperability to a “basic” IEEE802.3bw PHY.

### 2.3 Power Sequencing

This shows a high-level overview of the power sequencing. The actual implementation is physical layer dependent. The high level INH variable indicates the state of an optional inhibit pin. The PHY variable indicates the high level PHY state, where ENABLED means that the PHY is powered up and DISABLED means that the PHY is in power-down mode.



## 4 Timing Behavior

The sleep and wake up process in a PHY shall fulfill the following requirements<sup>1</sup>:

- ACK\_timer
- REQ\_timer
- TWU\_Link\_passive
- TWU\_Link\_active
- TWU\_Forward\_passive
- TWU\_Forward\_active

### 4.1 ACK\_timer

The time duration in SLEEP\_ACK state shall expire in 8ms.

ACK\_timer = 8ms.

### 4.2 REQ\_timer

The time duration in SLEEP\_REQUEST state and SLEEP\_SILENT state shall expire in 16ms.

REQ\_timer = 16ms.

### 4.3 TWU\_Link\_passive

The wake-up transmission time over a passive link (WUP) shall be less than 2 ms.

TWU\_Link\_passive < 2ms.

### 4.4 TWU\_Link\_active

Wake-up transmission time over an active link (WUR) shall be less than 1 ms.

TWU\_Link\_active < 1ms (for MTU = 1500 byte).<sup>2</sup>

### 4.5 TWU\_Forward\_passive

Wake-up forwarding time for a passive switch shall be less than 15ms, including 1ms until power supply is stable (This includes boot and config from flash). A passive switch is a switch with all ports in passive state (power down or deep sleep).

TWU\_Forward\_passive < 15 ms.

### 4.6 TWU\_Forward\_active

Wake-up forwarding time for an active switch shall be less than 2ms. An active switch is a switch with at least one port active.

---

<sup>1</sup> For the mentioned timer values a 1 % tolerance is expected.

<sup>2</sup> For the case of JUMBO packets (16kB), TWU\_Link\_active < 3 ms is acceptable.



TWU\_Forward\_active < 2ms.

## 5 Service Primitives and Interfaces

Beside the service primitives and interfaces, specified in IEEE802.3, new service primitives are offered by the ISO/OSI layer data link layer to the upper management layer. These services are needed to realize the sleep and wake-up behavior.

In typical implementations these services primitives are implemented over a system management interface and indications are signaled over an IRQ mechanism. This document does not specify an SMI address layout.

### 5.1 Sleep.request

The purpose of the *Sleep.request* service primitive is to shut down a link in a controlled manner, without generating unwanted link failure interrupts.

### 5.2 Sleep.indication

The purpose of the *Sleep.indication* service primitive is to indicate a received sleep request.

### 5.3 SleepFail.indication

The purpose of the optional *SleepFail.indication* service primitive is to indicate an aborted or unsuccessful sleep handshake.

### 5.4 SleepAbort.request

The purpose of the *SleepAbort.request* service primitive is abort an initiated wakeup.

### 5.5 Wakeup.request

The purpose of the *Wakeup.request* service primitive is to generate a WUP or WUR command leading to a global wake-up within the Ethernet network.

### 5.6 Wakeup.indication

The purpose of the *Wakeup.indication* service primitive is to indicate a detected wake-up event. This includes a wakeup over a passive link, a wakeup over an active link as well as over a local wakeup pin.

### 5.7 Inhibit.indication

Signals the state of an optional power supply inhibit interface.

## 6 Command Definitions

This specification defines three commands which are used to request a power down and signal a wakeup over an active as well as a passive link.

### **6.1.1 Low Power Sleep (LPS)**

The Low Power Sleep (LPS) is a command to indicate a sleep request to the link partner. The LPS command is sent by a node requesting a transition to SLEEP, while the link is up.

### **6.1.2 Wake-Up Request (WUR)**

The Wake-Up Request (WUR) is a command to indicate a wake-up request to the link partner. It can be sent by a node PHY or switch PHY to distribute the wake-up request over a link, which is already active.

### **6.1.3 Wake-Up Pulse (WUP)**

The Wake-up pulses (WUP) is a command to indicate a wake-up request to the link partner.

## 7 Annex IEEE802.3bw

This section describes the modification of the Physical Coding Sublayer of IEEE802.3bw. The following state diagram shows the power state machine which implements the two-way handshake protocol.

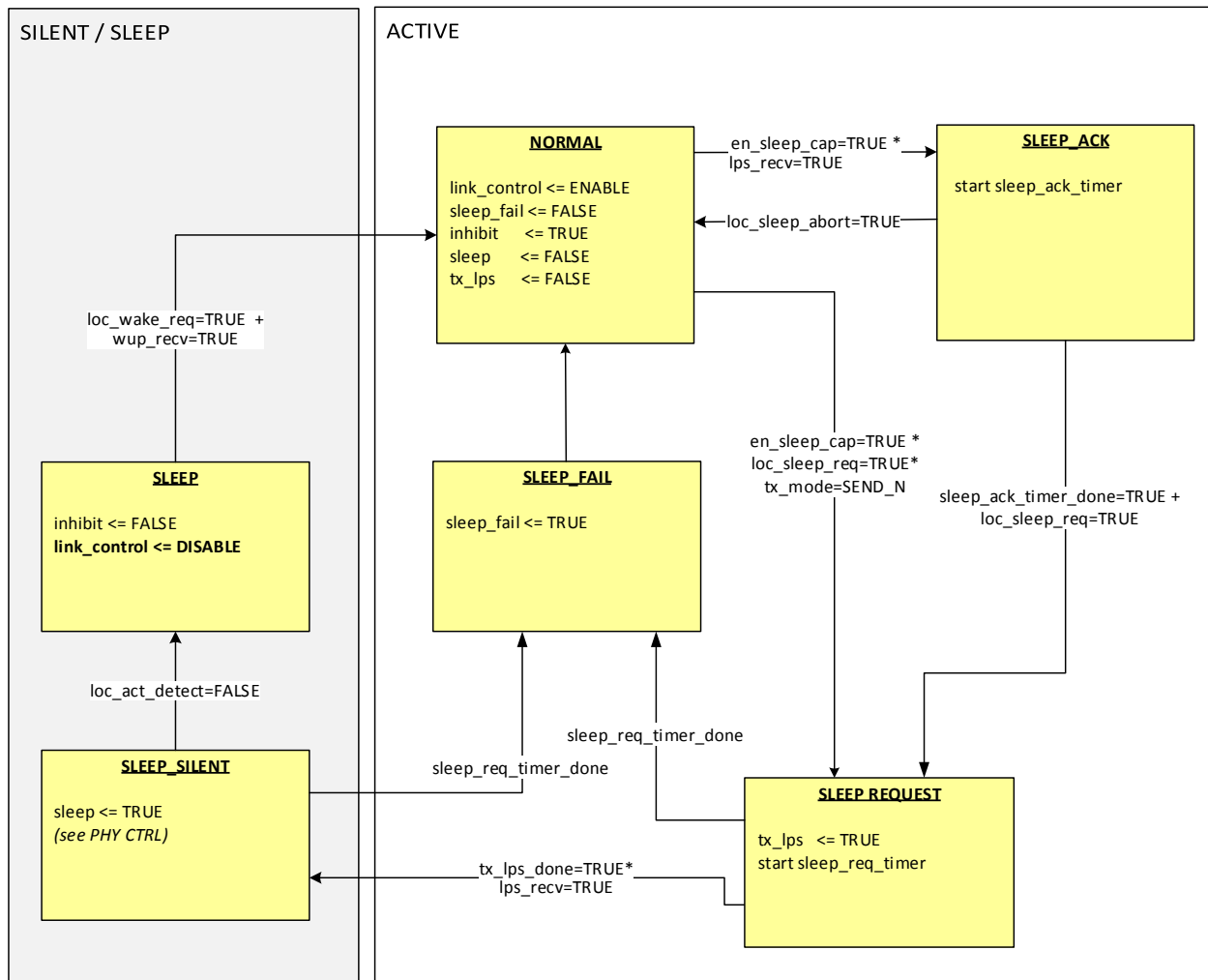


Figure 2: PHY power modes

In case the link is up (`tx_mode = SEND_N`) and *LinkSleep.request* is asserted the PHY will enter the Sleep Request state and will send LPS commands. The link partner receiving those LPS commands enters SLEEP\_ACK state and starts `sleep_ack_timer`. If `loc_sleep_abort` is asserted, the sleep is aborted because of incoming data message. If sleep reject is not done, the link partner will enter SLEEP\_REQUEST state and send LPS commands. If the PHY detects that it has sent and received LPS commands it transits to SLEEP\_SILENT state and eventually to SLEEP. On the other hand, if the handshaking is not done before `sleep_req_timer` timeout, the PHY enters SLEEP\_FAIL and back to SEND\_IDLE\_OR\_DATA state.

The signaling of a *Wakeup.request* depends on the state of the link. If the link is up (`tx_mode = SEND_N`) the PHY will transmit a WUR command over the active link during IDLE times. If the link is down (`tx_mode`

= SEND\_Z) the PHY will transmit a WUP pulse. If the link is not yet established (!loc\_rcvr\_status) for instance because the link is still in training (tx\_mode = SEND\_I) the link is first established, then a WUR command is sent.

Multi-PHY devices (e.g. switches) shall implement a selective wakeup forwarding mechanism. If a multi-PHY device detects a *Wakeup.request* (either WUR or WUP) on one a port, it must be possible to forward the *request* to other PHYs of the device. It shall be possible forward a wakeup over a passive link (WUP) as well as a wakeup over an active link (WUR) immediately to another port (or PHY).

The *Wakeup.indicate* should be generated upon wakeup events. In case the link is down this service primitive is generated upon the reception of WUP pulses (wup\_rcv) or if a local wakeup request (loc\_wake\_req) or if a WUR is received (wur\_rcv) was signaled. The implementation of the energy detection process is left to the PHY vendor. The energy detection process must not take longer than 2ms. It must be ensured that any transmitted IDLE pattern on the link will trigger the energy detection (wup\_rcv=TRUE).

### 7.1 Service Primitives and Interfaces

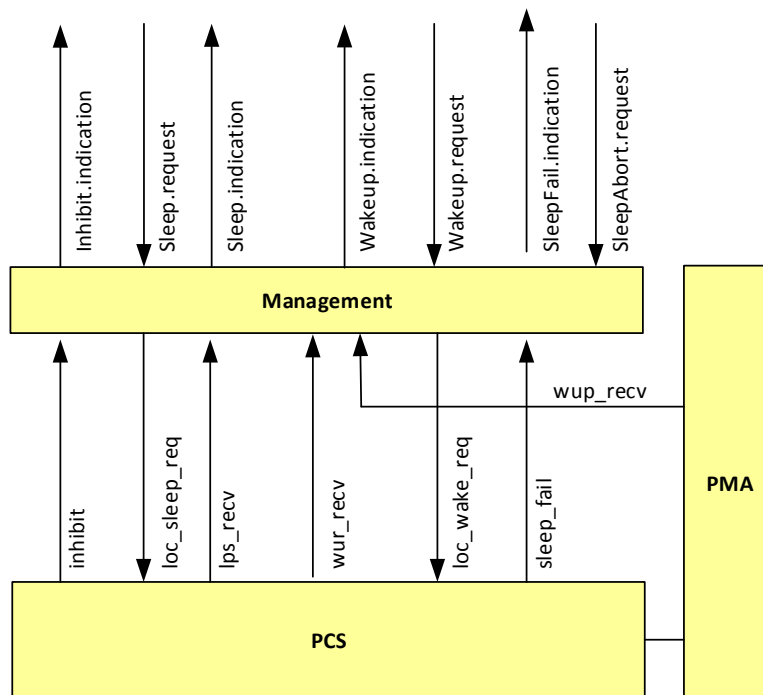


Figure 3: IEEE802.3bw Service Primitives

### 7.2 States

**NORMAL** state is the sub-state of SEND\_IDLE\_OR\_DATA with normal data transmission. On entering NORMAL state, rx\_rcv and loc\_sleep\_abort are deasserted.

**SLEEP\_ACK** state is also a sub-state of SEND\_IDLE\_OR\_DATA which allows an higher-layer to implicitly acknowledgement during the time ACK\_timer. During SLEEP\_ACK the sleep handshake can be aborted (loc\_sleep\_abort=TRUE) and the PHY transitions to NORMAL. The link partner will sense this (sleep\_req\_timer\_done) and transitions eventually to SLEEP\_FAIL.

The **SLEEP\_REQ** state is entered if the PHY is requested to enter SLEEP state over the LinkSleep.request primitive or after an implicit acknowledgement (ACK\_timer done). In this state it is expected that the Peer PHY also sends an LPS to acknowledge the flow. If the PHY has send its own LPS command and also received LPS, then SLEEP\_SILENT is entered. Otherwise the PHY enters SLEEP\_FAIL after sleep\_req\_timer expires.

In **SLEEP\_SILENT** the PHYs transmitter remains silent (tx\_mode=SEND\_Z) but the energy detection circuitry remains disabled to prevent spurious wup\_recv glitches. This acts as a safeguard to prevent a mutual wakeup through LPS commands. SLEEP state is entered if both PHYs are silent (loc\_act\_detect=FALSE).

**SLEEP\_FAIL** state is entered if either one or both PHYs cannot finish handshaking properly before sleep\_req\_timer timeout. The sleep flow is terminated and the PHYs go back to SEND\_IDLE\_OR\_DATA state.

In **SLEEP** the transmitter is powered down and waits for a wakeup pulse or software wakeup.

## 7.3 Command Definitions

This specification defines three commands which are used to request a power down and signal a wakeup over an active as well as a passive link.

### 7.3.1 Low Power Sleep (LPS)

LPS is encoded in the scrambler stream as defined in Section 7.3. The LPS command must be sent for a minimum of 64 bits. The detection of a LPS command is left to the implementer. Aborting a LPS command may lead to sending less than 64 bits.

### 7.3.2 Wake-Up Request (WUR)

The WUR is encoded in the scrambler stream as defined in Section 7.3. The WUR command must be sent for a minimum of 64 bits. The detection of a WUR command is left to the implementer. Aborting a WUR command may lead to sending less than 64 bits. In addition, the maximum number of hops of a wake-up network is 4.

### 7.3.3 Wake-Up Pulse (WUP)

WUP are link training codes transmitted on the network by a node in tx\_mode=SEND\_I or switch PHY to distribute the wake-up request over a link, which is down. The activity on the twisted-pair lines will be detected by the partner PHY as a remote wakeup. The wake-up pulse has a duration of 1 ms (+/- 0.3ms) to allow reliable detection. The energy detection of a WUP command is left to the implementer.

## 7.4 Generation of scrambling bits $Sd_n[2:0]$

Commands are transmitted in the IDLE state of the PCS state machine. Therefore the side stream scrambler is modified. The generation of  $Sd_n[1:0]$  of IEEE802.3bw is modified. The bit  $Sd_n[2]$  remains identical to IEEE802.3bw. The bit  $Sd_n[1]$  is used to scramble the data bit  $tx\_data_n[1]$  during data mode and to encode LPS otherwise. It is defined as follows:

$$Sd_n[1] = \begin{cases} Sc_n[1] \wedge tx\_data_n[1] & \text{if } (tx\_enable_{n-3} = 1) \\ Sc_n[1] \wedge 1 & \text{else if } ((tx\_lps = TRUE) \& (loc\_wake\_req = FALSE) \& (tx\_mode = SEND\_N)) \\ Sc_n[1] & \text{else} \end{cases}$$

The bit  $Sd_n[0]$  is used to scramble the data bit  $tx\_data_n[0]$  during data mode and to encode WUR otherwise. It is defined as follows:

$$Sd_n[0] = \begin{cases} Sc_n[0] \wedge tx\_data_n[0] & \text{if } (tx\_enable_{n-3} = 1) \\ Sc_n[0] \wedge 1 & \text{else if } ((tx\_lps = FALSE) \& (loc\_wake\_req = TRUE) \& (tx\_mode = SEND\_N)) \\ Sc_n[0] & \text{else} \end{cases}$$

## 7.5 PCS PHY Control State Diagram

The following Figure shows the PCS PHY Control State machine which implements parts of the power sequencing state machine.

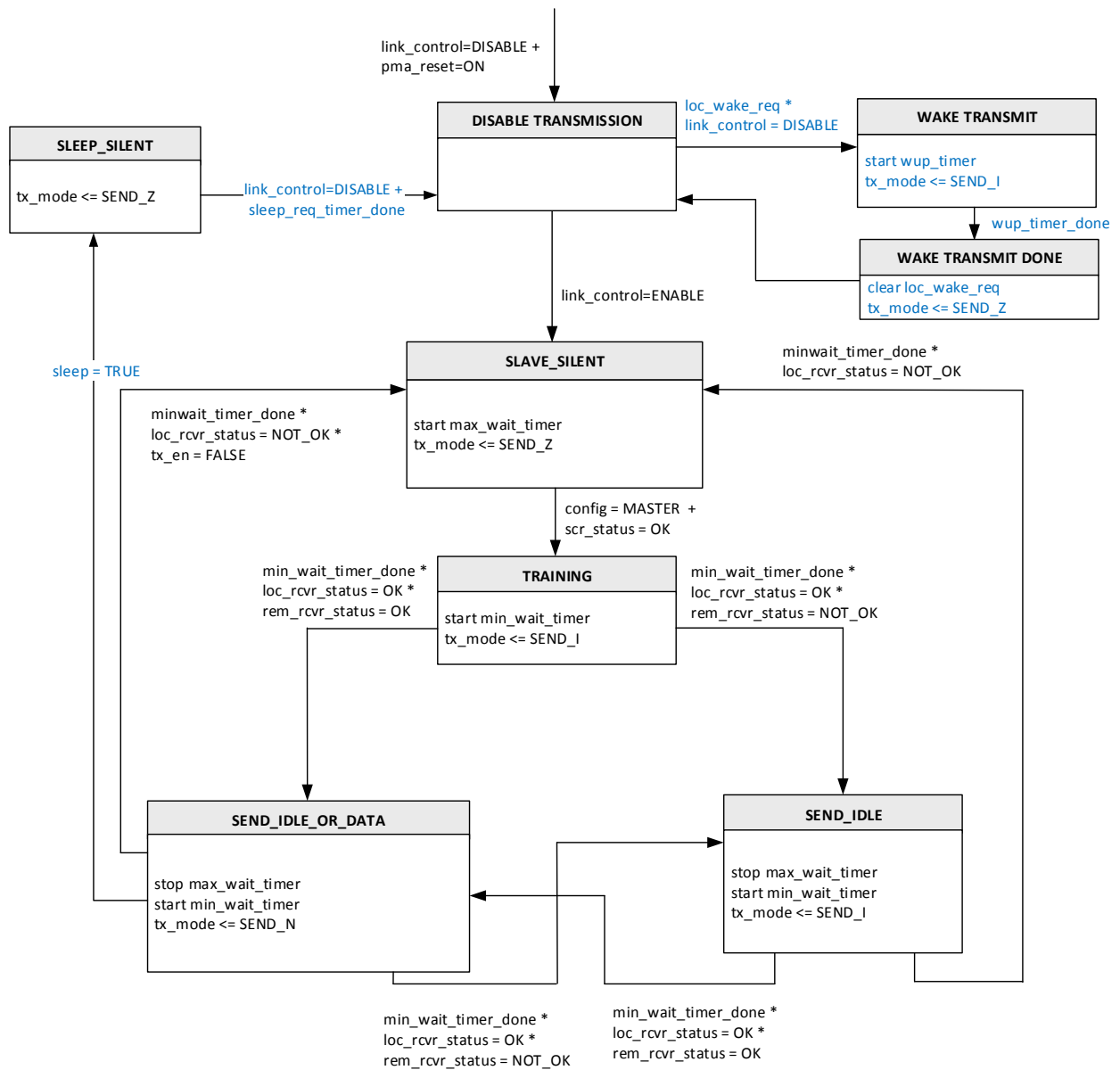


Figure 4 PHY Control State Diagram to replace the BroadR Reach PHY Control State Diagram

### 7.5.1 Variables

<i>loc_act_detect:</i>	<i>Local activity detection signal. The variable is set to FALSE if consecutive symbols of zeros were received; otherwise set to TRUE. The value of loc_act_detect shall be set to TRUE (FALSE) within 1 us.</i>
<i>lps_rcv:</i>	<i>Set if a LPS command was entirely received.</i>
<i>wur_rcv:</i>	<i>Set if a WUR command was entirely received.</i>
<i>wup_rcv:</i>	<i>Set if WUP pulses were sensed.</i>
<i>tx_lps:</i>	<i>If set, LPS bits are transmitted by the scrambler.</i>
<i>tx_lps_done:</i>	<i>Set after the LPS command was entirely sent (at least 64 bits).</i>
<i>loc_sleep_req:</i>	<i>Set if a sleep is requested by the local PHY.</i>
<i>loc_wake_req:</i>	<i>Set if a wakeup is requested by the local PHY.</i>
<i>loc_sleep_abort:</i>	<i>Set if a remote sleep request is to be rejected while still in SLEEP_ACK phase.</i>
<i>sleep_fail:</i>	<i>Set if a sleep handshake was aborted by the link partner.</i>
<i>inhibit:</i>	<i>Set if the (external) power supply shutdown is inhibited.</i>
<i>sleep:</i>	<i>Set by the power statemachine to notify PHY CTRL to disable transmission.</i>
<i>en_sleep_cap:</i>	<i>Indicated whether sleep capability is enabled.</i>

### 7.5.2 Timers

<i>wup_timer:</i>	<i>A timer used to wait for reliable detection of WUP pulse. The timer shall expire 1ms +/- 0.5ms after being started.</i>
<i>sleep_ack_timer:</i>	<i>A timer used in SLEEP_ACK state to check whether NM decides to reject sleep flow on an incoming data message or not. The timer shall expires 8ms after being started.</i>
<i>sleep_req_timer:</i>	<i>A timer set up in SLEEP_REQ to check if the handshaking is properly done by both PHYs. If the PHY doesn't enter SLEEP state before timeout, it enters SLEEP_FAIL state and back to NORMAL. The timer shall expires 16ms after being started.</i>



## Appendix: System Timing Diagram

### A.1 System timing for WUP/WUR forwarding and transmission

The timing sequence for WUP/WUR forwarding and transmission is shown in Figure 4. LP1, LP2, LP3 can be either PHYs or switches. Their ports P1 and P2 can be either MDI or wakeup I/O pins. The forwarding starts when the device detects WUP/WUR or a local wakeup, and ends when its port start transmitting WUP/WUR or outputs a wake signal from the wakeup I/O. The transmission starts when LP starts transmitting WUP/WUR and stops when the receiving LP detects WUP/WUR.

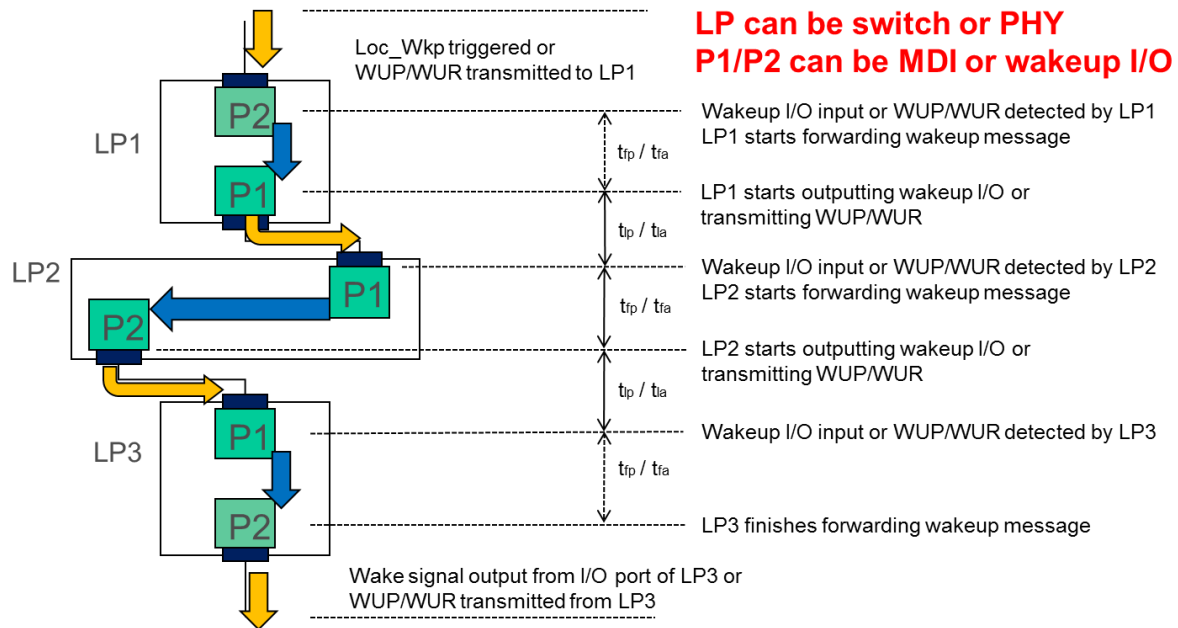


Fig 4 Detail timeline for WUP/WUR forwarding and transmission