Signaling Transport Working Group J. Craig
Internet Draft Tekelec
Expires: November 8, 2006 May 8, 2006

M2PA Implementer's guide
<draft-craig-sigtran-m2pa-ig-00.txt>

Status of this Memo

Copyright Notice

Abstract

This document contains a compilation of all defects found up until the publication date for M2PA RFC4165 [RFC4165]. These defects may be of an editorial or technical nature. This document may be thought of as a companion document to be used in the implementation of M2PA to remove ambiguities and correct errors in the original M2PA document. This document updates RFC4165 [RFC4165] and text within this document supersedes the text found in RFC4165.

Table of Contents

1. Introduction

1.1. Overview

This document contains a compilation of all defects found up until the publication date for the Signaling System 7 (SS7) Message Transfer Part 2 (MTP2) User Peer-to-Peer Adaptation Layer (M2PA) [RFC4165]. These defects may be of an editorial or technical nature. This document may be thought of as a companion document to be used in the implementation of M2PA to remove ambiguities abd correct errors in the original M2PA document. This document updates RFC4165 and text within this document, where noted, supersedes the text found in RFC4165. Each error and clarification will be detailed within this document in the form of:

- The problem description,
- A description of the solution,
- The text quoted from RFC4165, if applicable,
- The new or replacement text.

1.2. Terminology

This document uses the terms described in [RFC4165], in addition to the following:

Empty User Data Message - a M2PA message having a message type value of 'User Data' and a Message Data field having a length of zero. This message is used to acknowledge reception of non-empty User Data messages when there are no non-empty User Data messages to be sent. This kind of message does not contain user data.

Non-Empty User Data Message - a M2PA message having a message type value of 'User Data' and a Message Data field having a length greater than zero. This kind of message contains user data.

1.3. Abbreviations

This document uses the abbreviations described in [RFC4165], in addition to the following:

ACK - Acknowledgement
AERM - Alignment Error Rate Monitor
L2 - M2PA or MTP2
L3 - MTP3
LS - Link Status
PO - Processor Outage
PR - Processor Recovered
RB - Receive Buffer
RTB - Retransmit Buffer
SUERM - Signaling Unit Error Rate Monitor
TB - Transmit Buffer

2. Changes to RFC4165

2.1. Add Error Rate Monitors

2.1.1. Problem Description

RFC4165 section 4.1.1 states "Since SCTP uses a checksum to detect transmission errors, there is no need for an M2PA checksum, as is needed in MTP2. This also eliminates the need for the error rate monitors of MTP2."

M2PA uses a SCTP/IP transport that can involve shared network resources and for which available capacity can vary dramatically over time. The SCTP/IP network quality of service characteristics can be such that, without an error rate monitor function, a M2PA signaling link will enter service and yet operate unreliably, possibly entering congestion or failing intermittently due to timer T7 expiration.

MTP2 standards provide an Alignment Error Rate Monitor (AERM) function used to determine the viability of a transport prior to placing a signaling link in service. M2PA should provide an equivalent for this function, one that is aware of the characteristics of the SCTP/IP transport.

MTP2 standards provide a Signaling Unit Error Rate Monitor (SUERM) function used to detect degraded transport operation and fail a signaling link when transport quality of service is not sufficient. M2PA should provide an equivalent for this function, one that is aware of the characteristics of the SCTP/IP transport.

2.1.2. Solution Description

Add new text that describes the M2PA Alignment Error Rate Monitor and the M2PA Signaling Unit Error Rate Monitor.

2.1.3. Text Changes

Original Text (RFC4165 section 4.1.1)

Since SCTP uses a checksum to detect transmission errors, there is no need for an M2PA checksum, as is needed in MTP2. This also eliminates the need for the error rate monitors of MTP2.

New Text (RFC4165 section 4.1.1)

Since SCTP uses a checksum to detect transmission errors, there is no need for an M2PA checksum, as is needed in MTP2.

New Text (RFC4165 section 4.2.4)

4.2.4. Alignment Error Rate Monitor (AERM)

The MTP2 standards, e.g. [Q.703] section 10.3, describe the alignment error rate monitor.

If M2PA uses a SCTP implementation that provides the ability to query round-trip-time and retransmit data for a particular association, it is RECOMMENDED that M2PA use those interfaces during proving (timer T4 is running) to ensure that the transport meets implementation-dependent quality of service requirements. An example set of quality of service requirements is: average RTT during proving must be less than or equal to 100ms, and the maximum rate of retransmissions allowed during proving is 1 per 1000 messages transmitted. It is RECOMMENDED that M2PA allow the proving state quality of service parameters to be administered by the user.

If M2PA determines that quality of service requirements are not met, and the proving procedure has been aborted less than four times, then M2PA SHOULD abort the current proving period, count the aborted

proving period, and restart proving. If M2PA determines that quality of service requirements are not met, and proving has already been aborted four times, then M2PA SHOULD move the signaling link to the out of service state.

New Text (RFC4165 section 4.2.5)

4.2.5. Signaling Unit Error Rate Monitor (SUERM)

The MTP2 standards, e.g. [Q.703] section 10.2, describe the signaling unit error rate monitor.

If M2PA uses a SCTP implementation that provides the ability to query round-trip-time and retransmit data for a particular association, it is RECOMMENDED that M2PA use those interfaces while the signaling link is in service to ensure that the transport meets implementation-dependent quality of service requirements. An example set of quality of service requirements is: average RTT while in service must be less than or equal to 100ms, and the maximum rate of retransmissions allowed while in-service is 1 in 1000. It is RECOMMENDED that M2PA allow the in service state quality of service parameters to be administered by the user.

If M2PA determines that quality of service requirements are not met, then M2PA SHOULD move the signaling link to the out of service state.

New Text (RFC4165 section 4.3.2)

4.3.2. SCTP Quality of Service Statistics

M2PA relies upon SCTP to provide a reliable communication transport, and so quality of service statistics such as round-trip time and retransmission counts are properly kept by SCTP.

It is RECOMMENDED that M2PA use an SCTP implementation that has the following characteristics:

- provides an interface for the user to obtain the last measured network round-trip-time (time from when a packet containing user data was sent to the time reception was acknowledged) for a particular association

- provides an interface for the user to obtain the number of retransmissions that have been performed for a particular association

An SCTP implementation that provides M2PA visibility into quality of service statistics for an association allows M2PA to implement the alignment error rate monitor and signaling unit error rate monitor.

2.2. Clarify Initial FSN and BSN Values

2.2.1. Problem Description

RFC4165 does not state what the M2PA initial FSN and BSN must be, though it cites applicable MTP2 standards. M2PA implementers have chosen different initial FSN values, leading to M2PA interoperability problems.

2.2.2. Solution Description

Add new text that describes the initial FSN and BSN values for two common MTP2 variants.

2.2.3. Text Changes

Original Text (RFC4165 section 2.2.2)

This is the M2PA sequence number of the User Data message being sent.

The FSN and BSN values range from 0 to 16,777,215.

New Text

This is the M2PA sequence number of the User Data message being sent.

The FSN and BSN values range from 0 to 16,777,215.

An M2PA that conforms to either Q.703 [Q.703] or T1.111.3 [T1.111] SHALL set the FSN value of the first transmitted non-empty User Data message to 0. An M2PA that conforms to either Q.703 [Q.703] or T1.111.3 [T1.111] SHALL, if no non-empty User Data messages have been received, set the BSN value of the first transmitted user data message to 16,777,215.

2.3. Clarify Local Busy Procedure

2.3.1. Problem Description

RFC1645 does not state that M2PA MUST buffer non-empty user data messages received after a Link Status Busy message has been sent and before Link Status Busy Ended has been sent. The RFC does not describe the treatment of the buffered messages when a primitive, such as Flush or Clear Buffers is received from MTP3. This ambiguity may lead to M2PA busy procedure interoperability problems.

2.3.2. Solution Description

Revise the text such that it states that M2PA MUST buffer received non-empty user data messages during the local busy condition and that it MUST discard the buffered messages if it receives a Flush or Clear Buffers primitive from MTP3.

2.3.3. Text Changes

Original Text (section 4.1.5)

The Link Status Busy message replaces the SIB message of MTP2. The message SHOULD NOT be transmitted continuously. M2PA SHALL send a Link Status Busy message to its peer at the beginning of a receive congestion condition where MTP2 would send SIB. M2PA MAY send additional Link Status Busy messages as long as that condition persists. When the condition ends, M2PA SHALL send a Link Status Busy Ended message to its peer.

M2PA SHALL continue transmitting messages while it is in receive congestion, but MUST NOT acknowledge the message that triggered the sending of the Link Status Busy message, nor any messages received before the sending of Link Status Busy Ended.

New Text

Upon receiving a User Data message that causes implementation-dependent onset of receive congestion, M2PA SHALL NOT acknowledge the message, SHALL mark the local congestion condition, and SHALL send a Link Status Busy message to the peer. The Link Status Busy message replaces the SIB message of MTP2.

While in the local congestion condition:

(a) M2PA SHALL, if other conditions allow, continue transmitting messages.

(b) M2PA MUST NOT acknowledge and MUST buffer any received non-empty User Data message.

(c) Upon receiving a Flush command from MTP3, M2PA SHALL discard any received messages that were buffered and unacknowledged during the local congestion condition, send a Link Status Busy Ended message to the peer, and clear the local congestion condition.

(d) Upon detecting that the receive congestion has abated, M2PA SHALL send a Link Status Busy Ended message to the peer, clear the local congestion condition, and process and acknowledge any received messages that had been buffered and unacknowledged during the local busy condition.

(e) M2PA MAY periodically send, but MUST NOT continuously send, Link Status Busy messages to the peer.

2.4. Clarify Remote Busy Procedure

2.4.1. Problem Description

It is not clear in RFC4165 section 4.1.5 that timer T6 MUST be running while remote congestion status is in effect.

RFC4165 does not clearly distinguish empty and non-empty User Data messages when it states that M2PA MUST NOT send User Data messages to a busy peer. M2PA SHOULD send empty User Data messages to a busy peer in order to acknowledge received messages.

2.4.2. Solution Description

Revise the text such that it states that M2PA SHALL mark remote congestion status and start T6 if and only if it receives Link Status Busy when T6 is not running, and the RTB contains one or more messages.

Revise the text such that it states that M2PA SHOULD acknowledge messages received from a busy peer but MUST NOT send non-empty user data messages to a busy peer.

Clarify that M2PA SHOULD cancel remote congestion status if its RTB becomes empty.

2.4.3. Text Changes

Original Text (section 4.1.5)

When the peer M2PA receives the first Link Status Busy message, it SHALL start the Remote Congestion timer T6 if there are messages in the retransmission buffer awaiting acknowledgement (i.e., T7 is running). M2PA SHALL stop the T7 timer if it is running. Additional Link Status Busy messages received while T6 is running do not cause T6 to be reset and do not cause T7 to be started. While T6 is running, T7 SHALL NOT be started.

When the peer M2PA receives the Link Status Busy Ended message and T6 has not expired, it SHALL stop T6 (if T6 is running) and start T7 (if there are messages awaiting acknowledgement in the retransmission buffer).

The peer M2PA SHOULD continue receiving and acknowledging messages while the other end is busy, but MUST NOT send User Data messages after receiving Link Status Busy and before receiving Link Status Busy Ended.

New Text

Upon receiving a Link Status Busy message, if the remote congestion condition is not already in effect, and the RTB contains at least one message, then M2PA SHALL mark the remote congestion condition, start the Remote Congestion timer T6, and stop a running timer T7. M2PA SHALL ignore a Link Status Busy message received when either the RTB is empty or the remote congestion condition is in effect.

While the remote congestion condition is in effect:

(a) M2PA timer T6 is running (or expired).

(b) M2PA MUST NOT send non-empty User Data messages.

(c) M2PA MUST NOT start timer T7.

(d) M2PA SHOULD continue receiving and acknowledging messages.

(e) M2PA SHALL ignore received Link Status Busy messages.

(f) M2PA SHALL, upon receiving a Link Status Busy Ended message,

stop timer T6, cancel the remote congestion condition, and start timer T7 if the RTB contains one or more messages requiring acknowledgement.

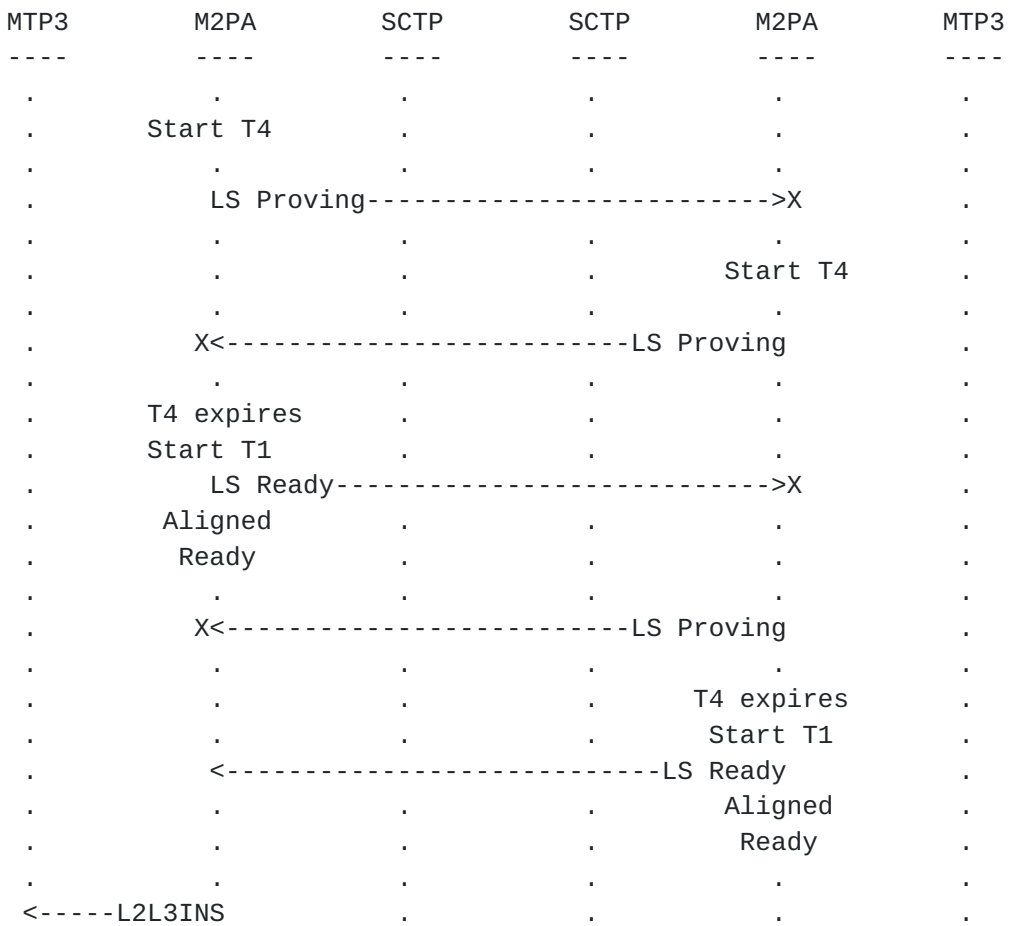
(g) M2PA SHOULD, upon receiving an acknowledgement that causes the RTB to become empty, stop timer T6 and cancel the remote congestion condition.

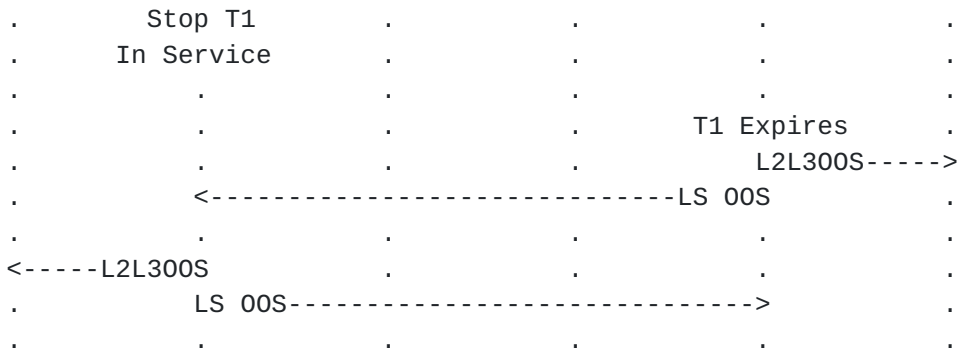
2.5. Link Status Ready Message Received During Proving

2.5.1. Problem Description

The M2PA link could fail to go in service if M2PA ignores Link Status Ready messages received while timer T4 is running. Consider the example depicted by Figure 1 where 'X<--' and '-->X' indicate a message received and ignored:

(For the diagram key, see 2.8.3).





Because the right side M2PA ignored the Link Status Ready message that it received while its timer T4 was running, it started its timer T1, and its timer T1 later expired because no subsequent Link Status Ready message was received. There would be a similar result if the left side M2PA had sent a Link Status Processor Outage message that was received during proving and ignored by the right side M2PA.

2.5.2. Solution Description

Revise the RFC to state that M2PA SHOULD mark Link Status Ready message received or Link Status Processor outage message received while timer T4 is running, and that upon T4 expiration, if either message had been received, then M2PA SHOULD NOT start timer T1, and instead SHOULD operate as if the message was received immediately after it would normally have started timer T1.

2.5.3. Text Changes

Original Text (4.1.3. Link Alignment)

The Link Status Ready message replaces the FISU of MTP2 that is sent at the end of the proving period. The Link Status Ready message is used to verify that both ends have completed proving. When M2PA starts timer T1, it SHALL send a Link Status Ready message to its peer in the case where MTP2 would send a FISU after proving is complete. If the Link Status Ready message is sent, then M2PA MAY send additional Link Status Ready messages while timer T1 is running. These Link Status Ready messages are sent on the Link Status stream.

In the case that MTP2 sends an MSU or SIPO message at the end of proving, M2PA SHALL send (respectively) a User Data or Link Status Processor Outage message.

New Text

The Link Status Ready message replaces the FISU of MTP2 that is sent at the end of the proving period. Unlike MTP2, the message SHOULD NOT be transmitted continuously. M2PA SHALL, when no local processor outage is in effect, use the Link Status Ready message to signal to the peer that it has completed proving.

The Link Status Processor Outage message replaces the SIPO message of MTP2. Unlike MTP2, the message SHOULD NOT be transmitted continuously. M2PA SHALL use the Link Status Processor Outage message to signal to the peer that it has completed proving and that local processor outage is in effect. The Link Status Processor Outage message SHALL be sent on the User Data stream.

While timer T4 is running, M2PA SHOULD mark reception of either a Link Status Ready message or a Link Status Processor Outage message. Upon expiration of timer T4, M2PA SHOULD check whether a Link Status Ready message or a Link Status Processor Outage message was received, and if either message was received, M2PA SHOULD NOT start timer T1 and instead SHOULD operate as if the message had been received after timer T1 had been started. If neither message was received during proving, then M2PA SHOULD start timer T1 and proceed according to the applicable MTP2 standard.

Upon expiration of timer T4, M2PA SHALL send a Link Status Ready message to its peer in the case where MTP2 would send a FISU and SHALL send a Link Status Processor Outage message to its peer in the case where MTP2 would send a SIPO. In the case that MTP2 sends a MSU at the end of proving, M2PA SHALL send a User Data message.

If a Link Status Ready message is sent, then M2PA MAY send additional Link Status Ready messages while timer T1 is running. These Link Status Ready messages are sent on the Link Status stream.

If a Link Status Processor Outage message is sent, then M2PA MAY send additional Link Status Processor Outage messages while timer T1 is running. These Link Status Processor Outage messages are sent on the User Data stream.

2.6. Clarify Processor Outage

2.6.1. Problem Description

RFC4165 summarizes a processor outage procedure that deviates significantly from the cited MTP2 standards. The lack of detail in the RFC's description of the procedure has led to implementation difficulty and to interoperability problems. One example of ambiguity is that it is not clear that the meaning of a received Link Status Ready message depends upon the association stream used to transport the message, as well as upon local and remote processor outage status. Another example of ambiguity is that it is unclear whether Link Status Processor Recovered or Link Status Ready is used to synchronize sequence numbers when a remote processor recovers.

A summary of M2PA processor outage procedure deviations from MTP2 follows:

- The Link Status Processor Outage message replaces the MTP2 SIPO, is not sent continuously, and is not required to be sent more than once.

- The Link Status Processor Recovered message replaces the MTP2 FISU or MSU sent after local processor outage condition clears and is not sent continuously.

- The Link Status Ready message sent on the User Data stream replaces the MTP2 FISU or MSU sent after remote processor outage clears and is not sent continuously. The Link Status Ready message sent on the Link Status stream indicates the end of proving with no local processor outage condition in effect. During alignment M2PA MUST distinguish the meaning of received Link Status Ready based upon the association stream upon which it is received.

- While in local processor outage, M2PA buffers received User Data messages without acknowledgement, whereas MTP2 discards received MSUs.

- M2PA must not transmit User Data messages after it sends Link Status Processor Recovered until it receives a Link Status Ready message on the User Data stream. Bounding the Link Status Ready waiting period is implementation dependent. MTP2 variants, such as [T1.111], are able to send either a FISU (sent continuously) or a MSU to indicate processor recovered status.

- The M2PA procedure for synchronizing sequence numbers when local processor outage clears is unbounded, unless M2PA implements a timer to establish a maximum time to wait for received Link Status Ready on the user data stream.

2.6.2. Solution Description

Clarify the M2PA processor outage procedure deviations from the cited MTP2 standards. Separate discussion of the local and remote processor outage procedures. Clarify that the M2PA experiencing remote processor outage uses the BSN of the received Link Status Processor Recovered message to synchronize its FSN. Clarify that the M2PA experiencing local processor outage uses the BSN of the Link Status Ready message received on the User Data stream to synchronize its FSN. Clarify treatment of buffered and unacknowledged non-empty User Data messages. Clarify treatment of received Link Status Processor Recovered when M2PA is experiencing both local and remote processor outage conditions. Clarify that treatment of received Link Status Ready message is based upon the association stream upon which it is received, as well as local and remote processor outage status.

2.6.3. Text Changes

Original Text (4.1.4. Processor Outage)

The Link Status Processor Outage message replaces the SIPO message of MTP2. Unlike MTP2, the message SHOULD NOT be transmitted continuously. M2PA SHALL send a Link Status Processor Outage message to its peer at the beginning of a processor outage condition where MTP2 would send SIPO. M2PA MAY send additional Link Status Processor Outage messages as long as that condition persists. The Link Status Processor Outage message SHALL be sent on the User Data stream.

While in a local processor outage (LPO) condition:

(a) Any User Data messages received from the peer MUST NOT be acknowledged and MUST be buffered.

(b) M2PA SHOULD continue to acknowledge User Data messages received and accepted by MTP3 before the local processor outage.

(c) M2PA SHOULD continue to transmit messages that have been sent by its upper layer MTP3.

While there is a remote processor outage (RPO) condition:

(a) M2PA SHOULD continue to acknowledge User Data messages received and accepted by MTP3, regardless of the remote processor outage.

(b) If any User Data messages received from the peer after the Link Status Processor Outage cannot be delivered to MTP3, then these messages MUST NOT be acknowledged and MUST be

buffered.

If M2PA receives a Flush command from MTP3,

(a) M2PA SHALL discard any incoming messages that were queued and unacknowledged during the processor outage condition.

(b) M2PA SHALL discard messages in the transmit and retransmit queues as required by MTP2.

If M2PA receives a Continue command from MTP3, M2PA SHALL begin processing the incoming messages that were queued and unacknowledged during the processor outage condition.

When the local processor outage condition ends, M2PA SHALL send a Link Status Processor Recovered message to its peer on the User Data stream. This message is used to signal the end of the processor outage condition, instead of an MSU or FISU, as is used in MTP2. The BSN in the Link Status Processor Recovered message is set to the FSN of the last User Data message received (and not discarded) from the peer M2PA. M2PA SHALL cease transmitting User Data messages after sending the Link Status Processor Recovered message, until it has received the Link Status Ready message(see below).

Upon receiving the Link Status Processor Recovered message, the M2PA in RPO SHALL respond with a Link Status Ready message on the User Data stream. The BSN in the Link Status Ready message is set to the FSN of the last User Data message received (and not discarded) from the peer M2PA.

Upon receiving the Link Status Ready message, the M2PA formerly in LPO SHALL respond with a Link Status Ready message on the User Data stream. The BSN in the Link Status Ready message is set to the FSN of the last User Data message received (and not discarded) from the peer M2PA.

M2PA (at both the LPO and RPO ends) uses the BSN value in the received Link Status Ready message to resynchronize its sequence numbers, if this is required by MTP2. M2PA SHALL NOT resume transmitting User Data messages until it has sent the Link Status Ready message.

During resynchronization, M2PA SHALL NOT discard any received User Data messages that were sent after the processor outage ended.

When M2PA experiences a local processor outage, it MAY put the link out of service by sending a Link Status Out of Service message, if this is allowed by the applicable MTP2 standard (e.g., [Q.2140]).

In other respects, M2PA SHOULD follow the same procedures as MTP2 in processor outage.

New Text

MTP2 standards, e.g. [Q.703] chapter 8, describe processor outage. M2PA's support for the procedure differs somewhat from that of MTP2.

The Link Status Processor Outage message replaces the SIPO message of MTP2. Unlike MTP2, the message SHOULD NOT be transmitted continuously. M2PA SHALL send a Link Status Processor Outage message to its peer when conditions would cause MTP2 to send SIPO. M2PA SHALL send the Link Status Processor Outage message on the User Data stream.

The Link Status Processor Recovered message replaces the FISU or MSU that MTP2 sends to indicate an end to a local processor outage condition, and the message SHOULD NOT be transmitted continuously. M2PA SHALL send the Link Status Processor Recovered message on the User Data stream. M2PA SHALL set the BSN in the Link Status Processor Recovered message to the FSN of the last non-empty User Data message received and not discarded from the peer M2PA. M2PA SHALL use the BSN of a received Link Status Processor Recovered message to synchronize its FSN.

The Link Status Ready message sent on the User Data stream replaces the FISU or MSU that MTP2 sends to acknowledge the end of a remote processor outage condition. M2PA SHALL set the BSN in the Link Status Ready message to the FSN of the last non-empty User Data message received and not discarded from the peer M2PA. M2PA SHALL use the BSN of a Link Status Ready message received on the User Data stream to synchronize its FSN.

When M2PA is notified of a local processor outage, it MAY put the link out of service by sending a Link Status Out of Service message, if this is allowed by the applicable MTP2 standard (e.g. [Q.2140]).

While in a local processor outage (LPO) condition:

(a) M2PA MUST buffer and MUST NOT acknowledge non-empty User Data messages received from the peer.

(b) M2PA SHOULD continue to acknowledge non-empty User Data messages received and accepted by MTP3 before the local processor outage.

(c) M2PA SHOULD continue to transmit messages that have been sent by its upper layer MTP3.

(d) M2PA MAY periodically send Link Status Processor Outage messages.

(e) Upon receiving a primitive from MTP3 that would cause MTP2 to send either a FISU or MSU, M2PA SHALL send a Link Status Processor Recovered message on the User Data stream, MAY start a Waiting for Link Status Ready timer (Tsync), and SHALL NOT transmit non-empty User Data messages.

(f) Upon receiving a Link Status Ready message on the User Data stream after sending a Link Status Processor Recovered message, M2PA SHALL synchronize its FSN using the message's BSN, and SHOULD stop a running Waiting for Link Status Ready timer (Tsync).

(g) Upon receiving a Link Status Processor Recovered message on the User Data stream after sending a Link Status Processor Recovered message, M2PA SHALL synchronize its FSN using the message's BSN, and SHOULD stop a running Waiting for Link Status Ready timer (Tsync).

(h) Upon expiration of the Waiting for Link Status Ready timer (Tsync), M2PA SHOULD take the link out of service.

(i) Upon receiving a primitive from MTP3 that would cause MTP2 to clear its RTB, M2PA SHOULD discard the non-empty User Data messages that it received and buffered without acknowledgement.

(j) Upon receiving primitive from MTP3 that would cause MTP2 to cancel local processor outage without clearing its buffers, M2PA SHALL begin processing the received non-empty User Data messages that were buffered and unacknowledged.

While in a remote processor outage (RPO) condition:

(a) M2PA SHOULD continue to acknowledge non-empty User Data messages received and accepted by MTP3.

(b) M2PA MUST buffer and MUST NOT acknowledge any received non-empty User Data messages that cannot be delivered to MTP3.

(c) Upon receiving a Link Status Processor Recovered message after T4 expiration, M2PA SHALL synchronize its FSN using the message BSN, and SHALL notify MTP3 of remote processor recovery. Upon receiving a Link Status Processor Recovered message while T4 is running, M2PA SHOULD cancel its remote processor outage status.

(d) Upon receiving a primitive from MTP3 that would cause MTP2 to send either a FISU or MSU, M2PA SHALL cancel remote processor outage condition and SHALL send a Link Status Ready message on the User Data stream.

(e) Upon receiving a primitive from MTP3 that would cause MTP2 to clear its RTB, M2PA SHOULD discard the non-empty User Data messages that it received and buffered without acknowledgement.

(f) Upon receiving primitive from MTP3 that would cause MTP2 to cancel remote processor outage without clearing its buffers, M2PA SHALL begin processing the received non-empty User Data messages that were buffered and unacknowledged.

M2PA SHALL ignore a Link Status Ready message received on the User Data stream after sequence number synchronization due to local processor recovery completion, i.e. the Waiting for Link Status Ready timer (Tsync) is not running.

2.7. Clarify Treatment of Received FSN

2.7.1. Problem Description

The RFC4165 text that describes the treatment of FSN in received messages does not distinguish empty and non-empty User Data messages and could be clearer in its description of the treatment of FSN for Link Status messages.

2.7.2. Solution Description

Revise the text to distinguish empty and non-empty User Data messages. Revise the text to state that M2PA SHALL ignore the FSN of received Link Status messages and empty User Data messages.

2.7.3. Text Changes

Original Text (4.1.5. Level 2 Flow Control)

For message types other than User Data, the Forward Sequence Number is set to the FSN of the last User Data message sent.

If M2PA receives a User Data message with an FSN that is out of

order, M2PA SHALL discard the message.

New Text

For message types other than Non-Empty User Data, M2PA SHALL set the Forward Sequence Number (FSN) to be equal to the FSN of the last Non-Empty User Data message sent.

M2PA SHALL ignore the FSN of received Link Status messages and Empty User Data messages.

M2PA SHALL discard and not acknowledge a received Non-Empty User Data message having a FSN that is out of order.

2.8. Provide More M2PA Procedure Examples

2.8.1. Problem Description

RFC4165 cites MTP2 standards as the basis for its procedures and also specifies significant deviations from those standards. Some deviations from MTP2 are:

- M2PA peer-to-peer primitives have names that differ from their MTP2 counterparts and some M2PA primitives serve multiple purposes, e.g. User Data and LS Ready message.

- M2PA has peer-to-peer primitives that MTP2 lacks, e.g. LS Busy Ended and LS Processor Recovery.

- M2PA does not send Link Status Alignment messages continuously, while MTP2 sends SIO continuously. M2PA does not require that more than one Link Status Alignment message be sent.

- M2PA does not send Link Status Ready messages and Empty User Data messages continuously, while MTP2 sends FISU continuously. M2PA does not require that more than one Link Status Ready message be sent.

- M2PA does not send Link Status Processor Outage messages continuously, while MTP2 sends SIPO continuously. M2PA does not require that more than one Link Status Processor Outage message be sent.

- M2PA buffers User Data messages received during local

processor outage, unlike MTP2, which discards them.

- o M2PA does not retransmit User Data messages, unlike MTP2.
- o M2PA does not send Link Status Busy messages continuously, while MTP2 sends SIB continuously. M2PA does not require that more than one Link Status Busy message be sent.

Some of the RFC's procedure examples lack detail, such as the communication of primitives between M2PA and MTP3 and between implementation-dependent management components and M2PA, and this lack of detail has made some M2PA procedures difficult to implement and has led to M2PA interoperability problems.

2.8.2. Solution Description

Provide more examples of M2PA procedures and provide examples that include the communication of primitives between MTP3 and M2PA and between implementation-dependent management components and M2PA.

2.8.3. Terminology and Tokens Used in Example Figures

The following summarizes the terms and tokens used in the procedure example figures:

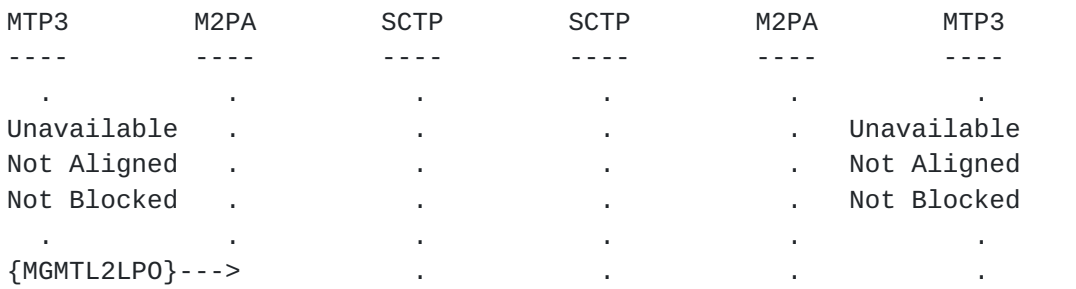
Vertical Axis	indicates from top to bottom non-linear increasing time
Horizontal Axis	distinguishes the protocol layers of the peers
event->	indicates primitive sent from left to right; destination reached
<-event	indicates primitive sent from right to left; destination reached
event->X	indicates primitive received and ignored
X<-event	indicates primitive received and ignored
event->>	indicates primitive in flight, not at destination
<<-event	indicates primitive in flight, not at destination
(number)	indicates a SCTP stream ID for Link Status messages and FSN of the M2PA User Data message from which a MSU was extracted.

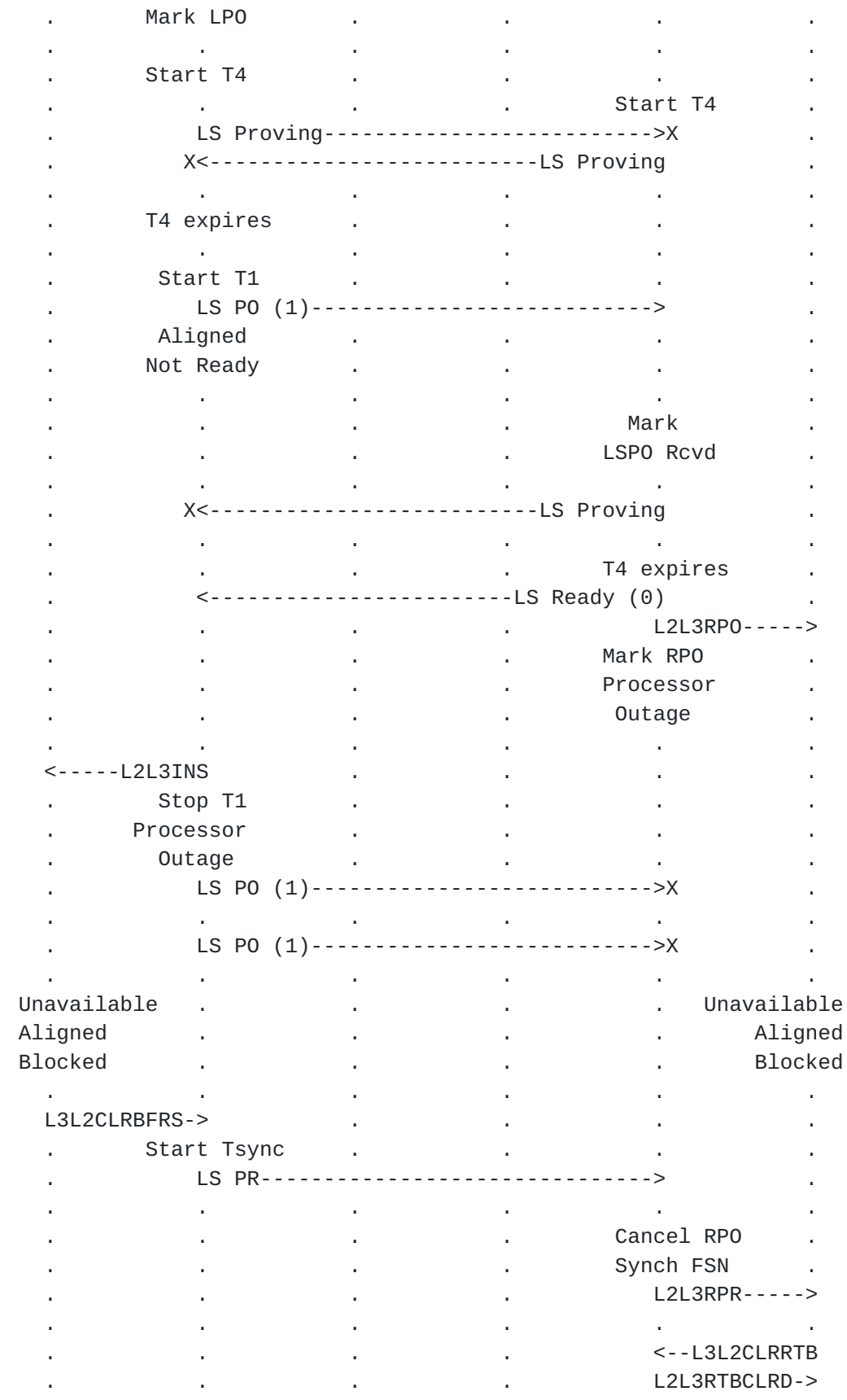
{event}	indicates an event generated by an implementation-dependent component.
Busy RB	is the M2PA busy receive buffer in which non-empty User Data messages are stored after being received and not acknowledged during a local busy condition.
Last_FSN	records the FSN of the last transmitted non-empty User Data message.
Next_BSN	records the FSN of the last user data message received from the peer requiring acknowledgement.
PO_RB	is the M2PA processor outage receive buffer in which user data messages are stored after being received and not acknowledged during processor outage condition.
RTB	is the M2PA retransmit buffer containing MSUs or user data messages that have been transmitted.
TB	is the M2PA transmit buffer containing MSUs received from MTP3 for transmission.

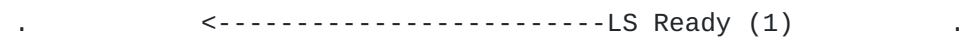
One should note that the diagrams are simplified in the sense that the communication of primitives is often depicted as instantaneous, i.e. the arrow resides on one horizontal line and crosses multiple protocol layers. In reality, the communication always involves the passage of time, and one should keep this in mind when interpreting the diagrams.

2.8.4. Aligned Not Ready Example 1 for [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the onset of local processor outage during link alignment prior to proving. The left side M2PA processor outage condition clears after a Link Status Ready message is received from the right side M2PA.




Mark LPO
Start T4
LS Proving
LS Proving
Start T4
T4 expires
Start T1
LS PO (1)
Aligned
Not Ready
Mark
LSPO Rcvd
LS Proving
T4 expires
LS Ready (0)
L2L3RPO
Mark RPO
Processor
Outage
L2L3INS
Stop T1
Processor
Outage
LS PO (1)
LS PO (1)
Unavailable
Aligned
Blocked
Unavailable
Aligned
Blocked
L3L2CLRBFRS
Start Tsync
LS PR
Cancel RPO
Synch FSN
L2L3RPR
L3L2CLRRTB
L2L3RTBCLRD


. <----------LS Ready (1) .

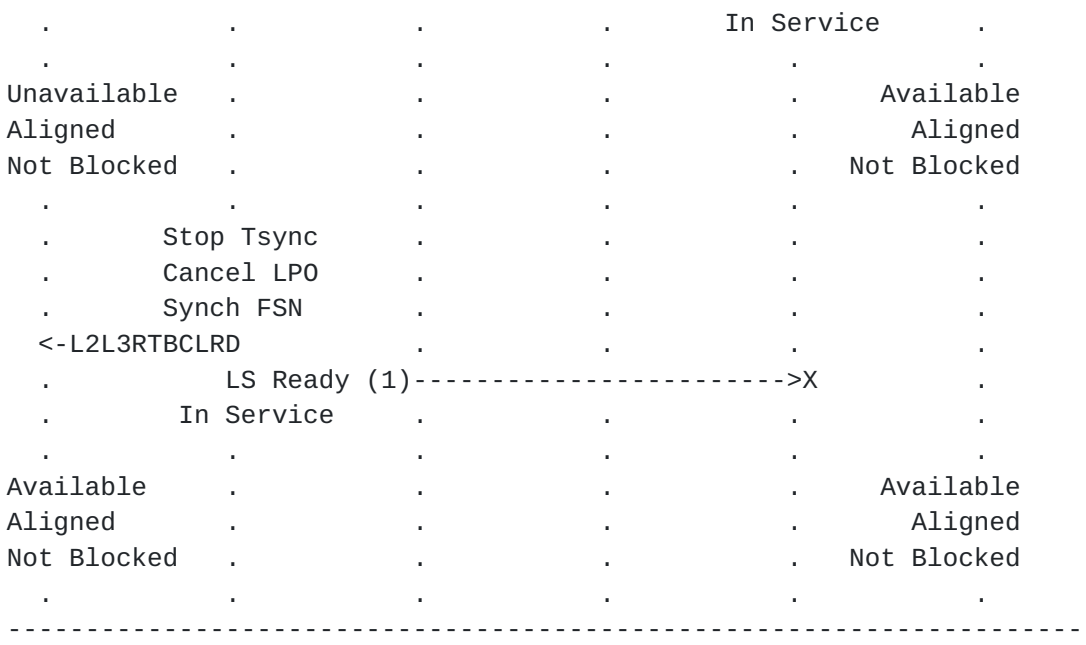
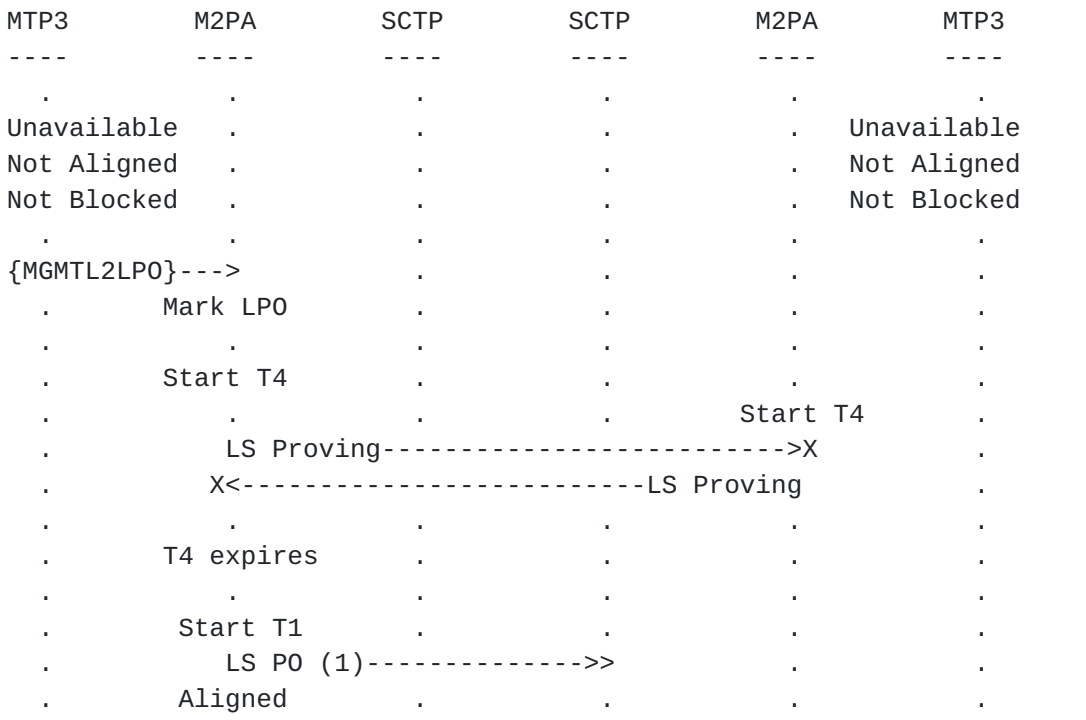
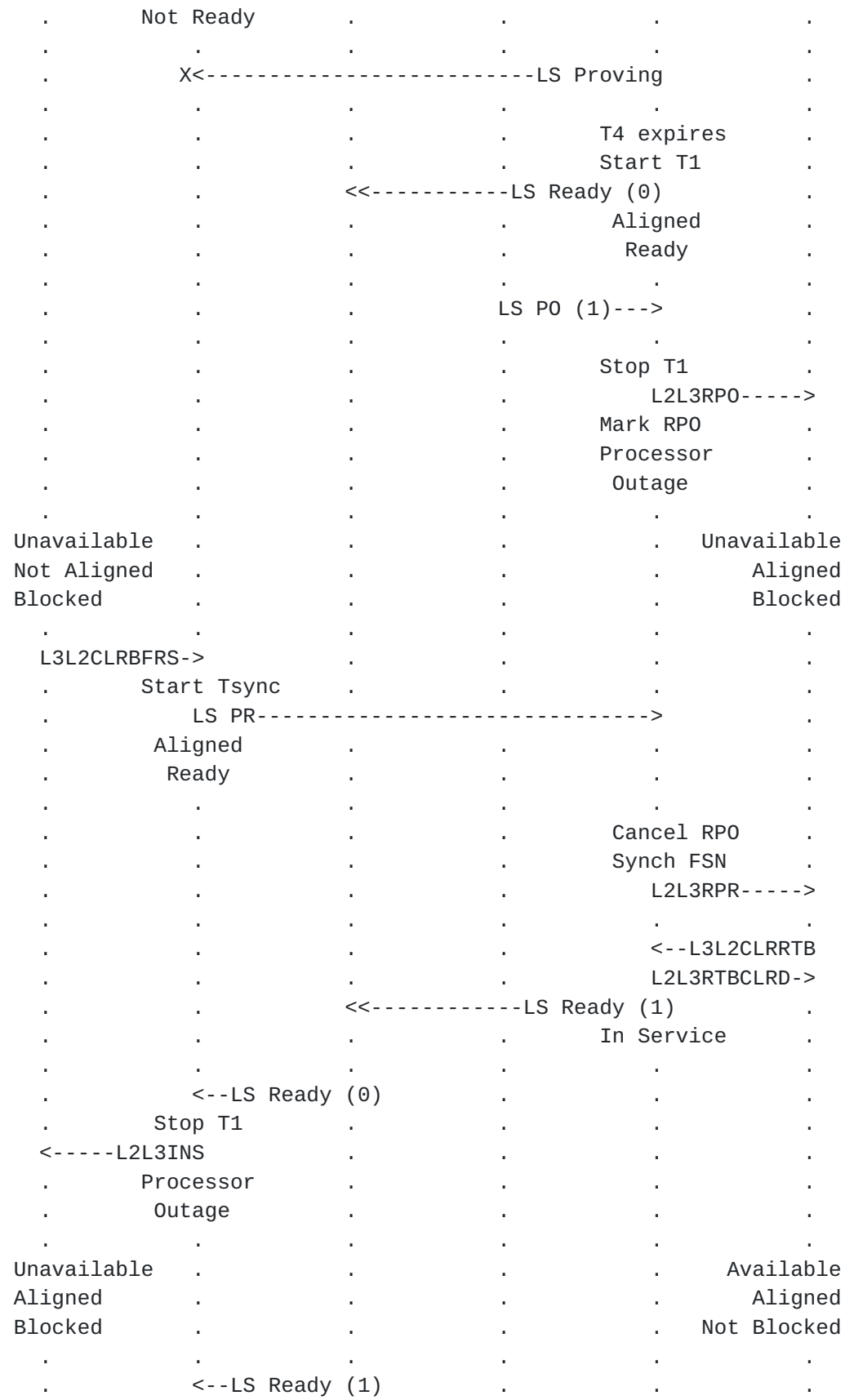


Figure 2. Aligned Not Ready Example 1 for [T1.111] Variant

2.8.5. Aligned Not Ready Example 2 for [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the onset of local processor outage during link alignment prior to proving. The left side M2PA processor outage condition clears before a Link Status Ready message is received from the right side M2PA.




Not Ready
LS Proving
T4 expires
Start T1
LS Ready (0)
Aligned
Ready
LS PO (1)--->
Stop T1
L2L3RPO----->
Mark RPO
Processor
Outage
Unavailable
Not Aligned
Blocked
Unavailable
Aligned
Blocked
L3L2CLRBFRS->
Start Tsync
LS PR----------
Aligned
Ready
Cancel RPO
Synch FSN
L2L3RPR----->
<--L3L2CLRRTB
L2L3RTBCLRD->
<<----------LS Ready (1)
In Service
<--LS Ready (0)
Stop T1
<-----L2L3INS
Processor
Outage
Unavailable
Aligned
Blocked
Available
Aligned
Not Blocked
<--LS Ready (1)

. . . . . .

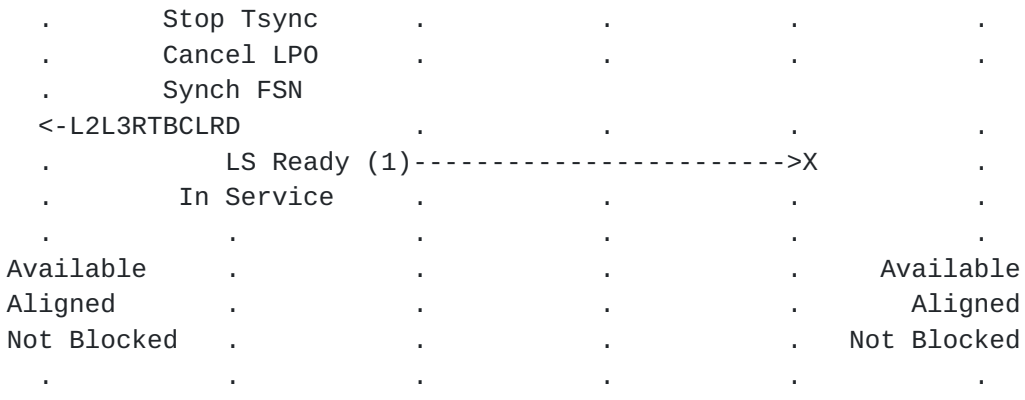
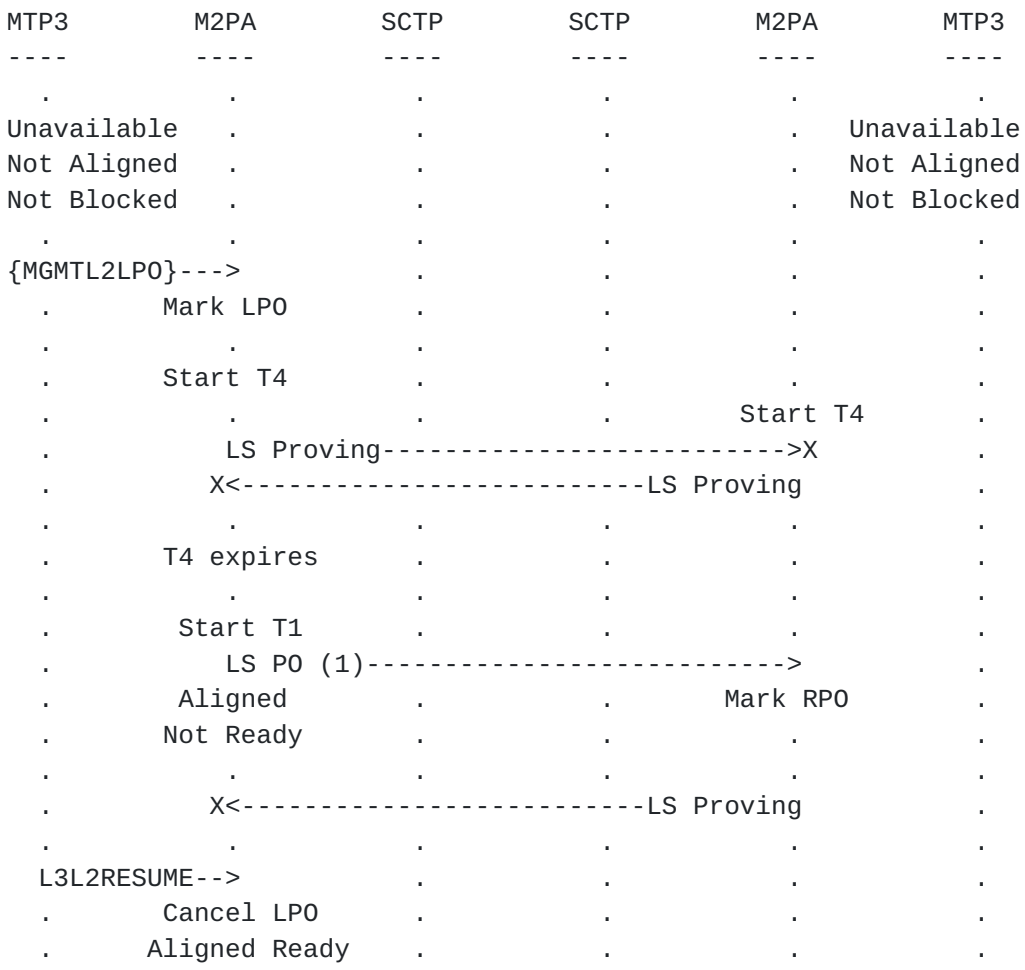


Figure 3. Aligned Not Ready Example 2 for [T1.111] Variant

2.8.6. Aligned Not Ready Example 3 for [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the onset of local processor outage during link alignment prior to proving. The left side M2PA local processor outage condition clears via a L3 resume primitive before the right side M2PA finishes proving.



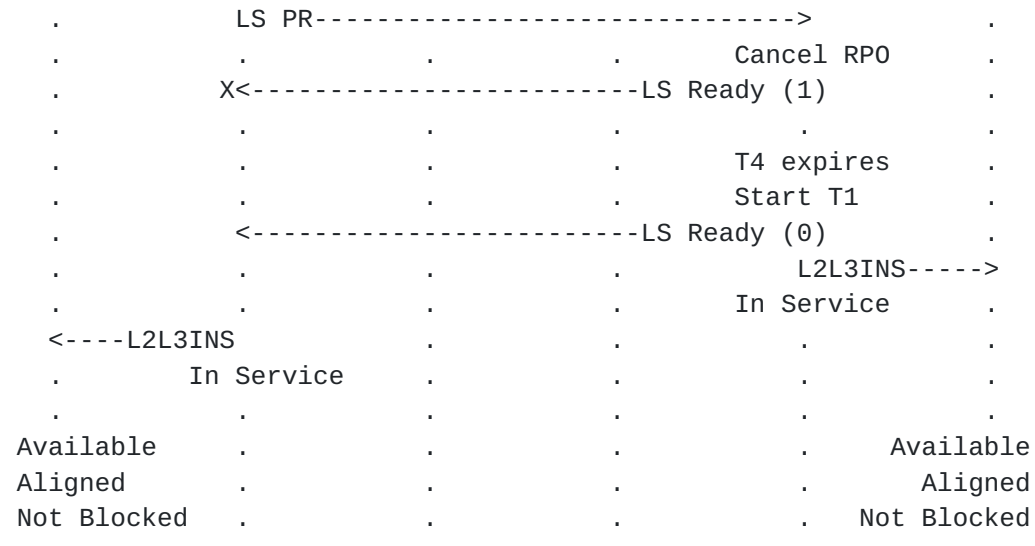
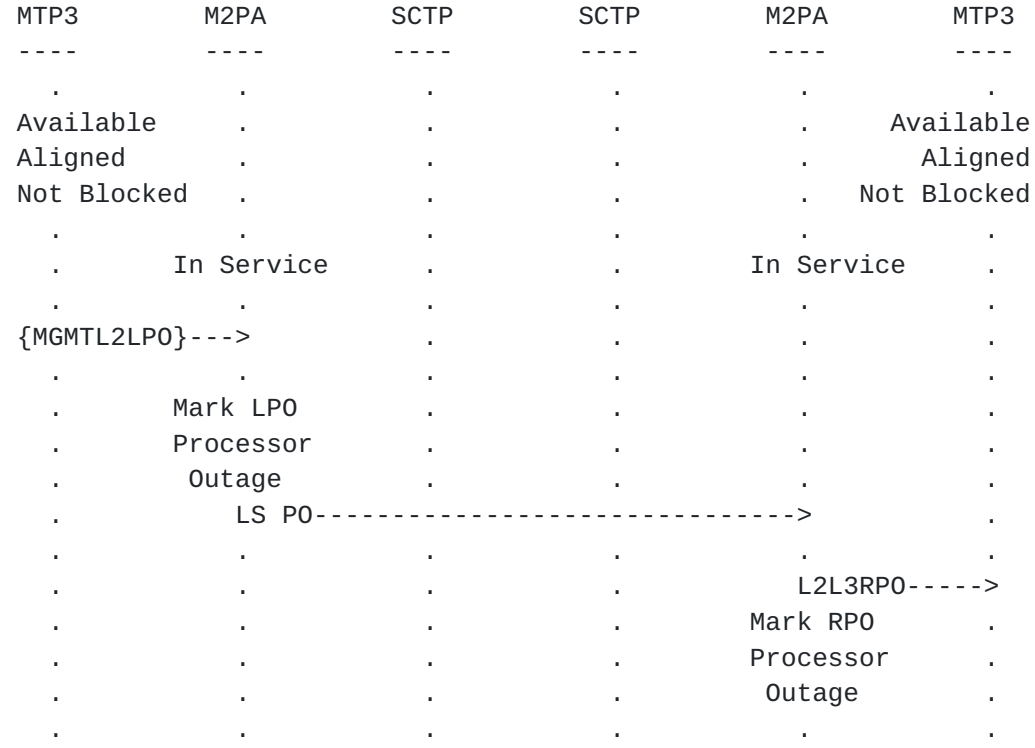


Figure 4. Aligned Not Ready Example 3 for [T1.111] Variant

2.8.7. PO Example (Clear Buffers) [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the onset and abatement of local processor outage while the link is in service. This example assumes that MTP3 issues the Clear Buffers primitive and that no User Data messages are transmitted during the period depicted.



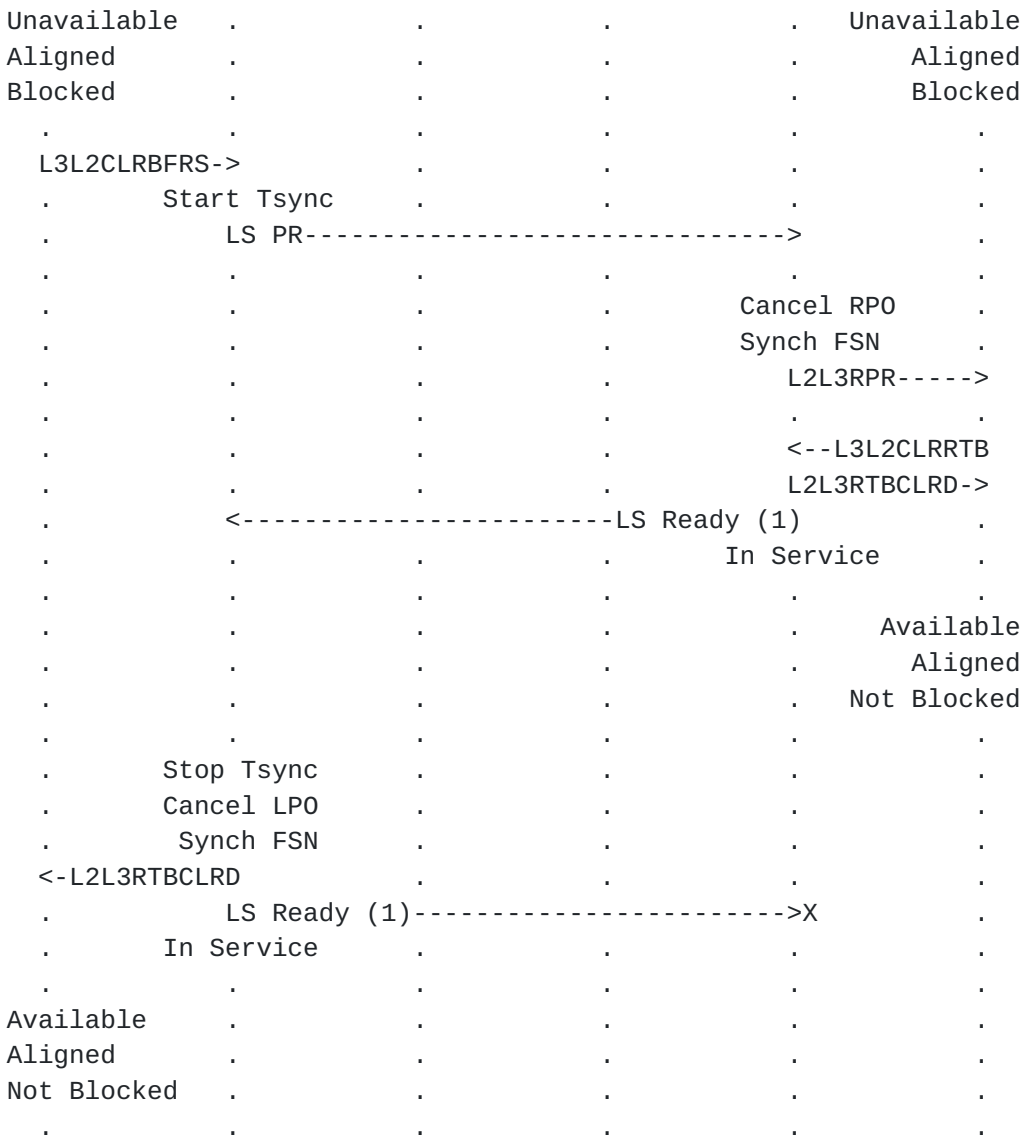
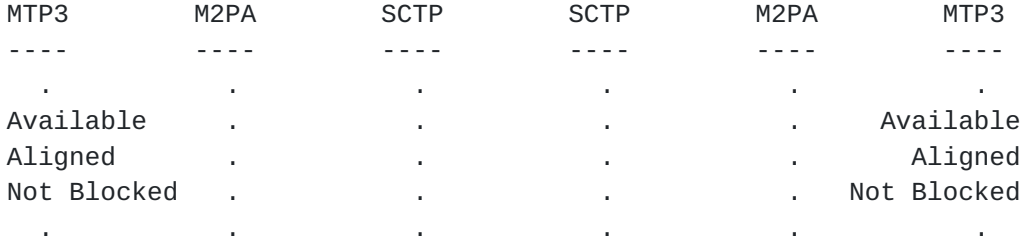
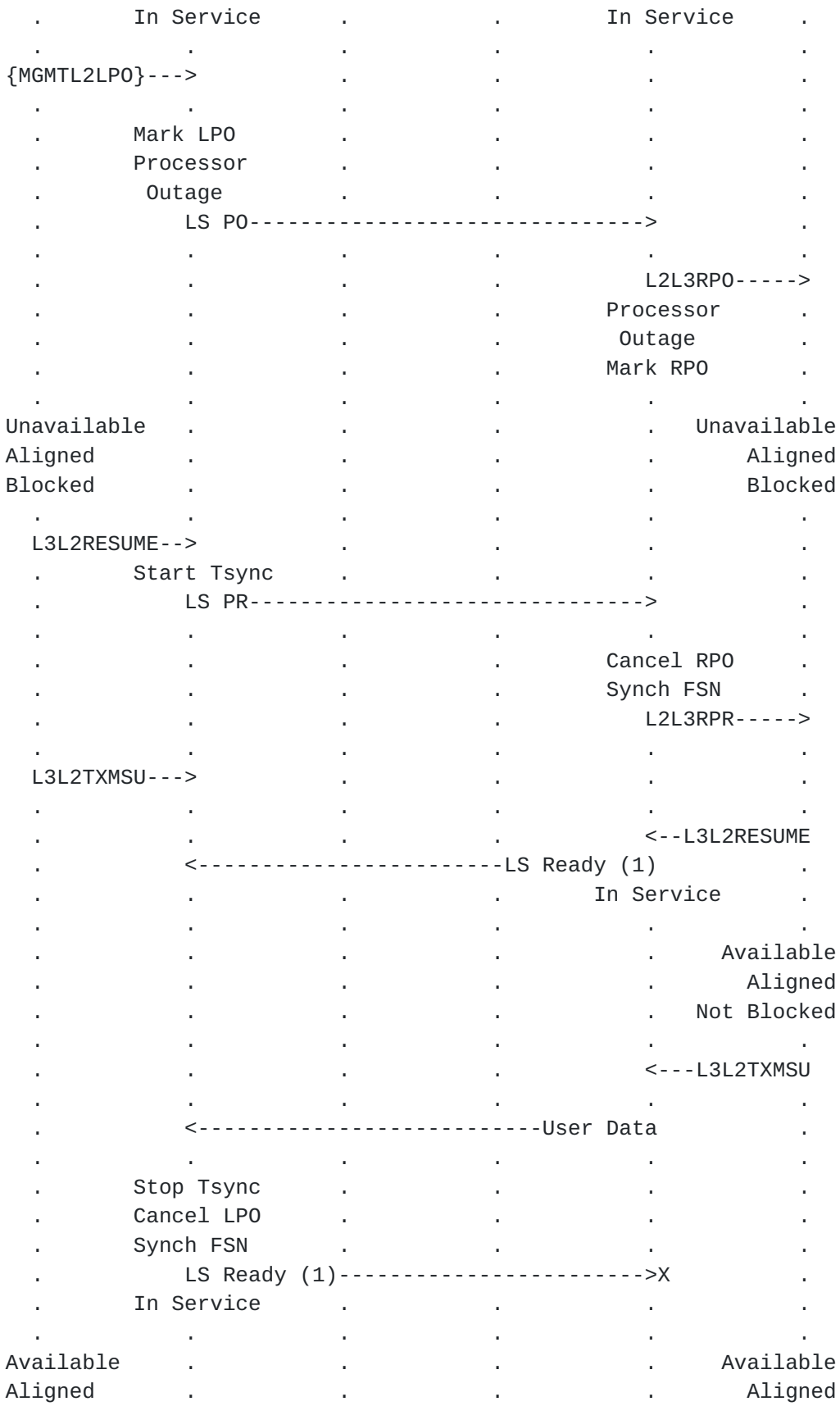


Figure 5. PO Example (Clear Buffers) [T1.111] Variant

2.8.8. PO Example (Resume) [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the onset and abatement of local processor outage while the link is in service. This example assumes that MTP3 issues a resume primitive.




In Service
In Service
{MGMTL2LPO}--->
Mark LPO
Processor
Outage
LS PO
L2L3RPO
Processor
Outage
Mark RPO
Unavailable
Aligned
Blocked
Unavailable
Aligned
Blocked
L3L2RESUME-->
Start Tsync
LS PR
Cancel RPO
Synch FSN
L2L3RPR
L3L2TXMSU--->
<--L3L2RESUME
<LS Ready (1)
In Service
Available
Aligned
Not Blocked
<---L3L2TXMSU
<User Data
Stop Tsync
Cancel LPO
Synch FSN
LS Ready (1)>X
In Service
Available
Aligned
Available
Aligned

Not Blocked . . . . Not Blocked

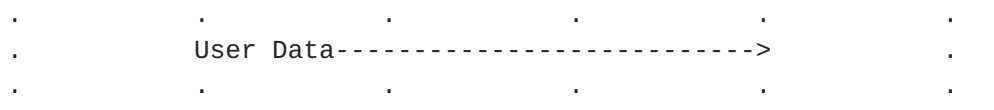
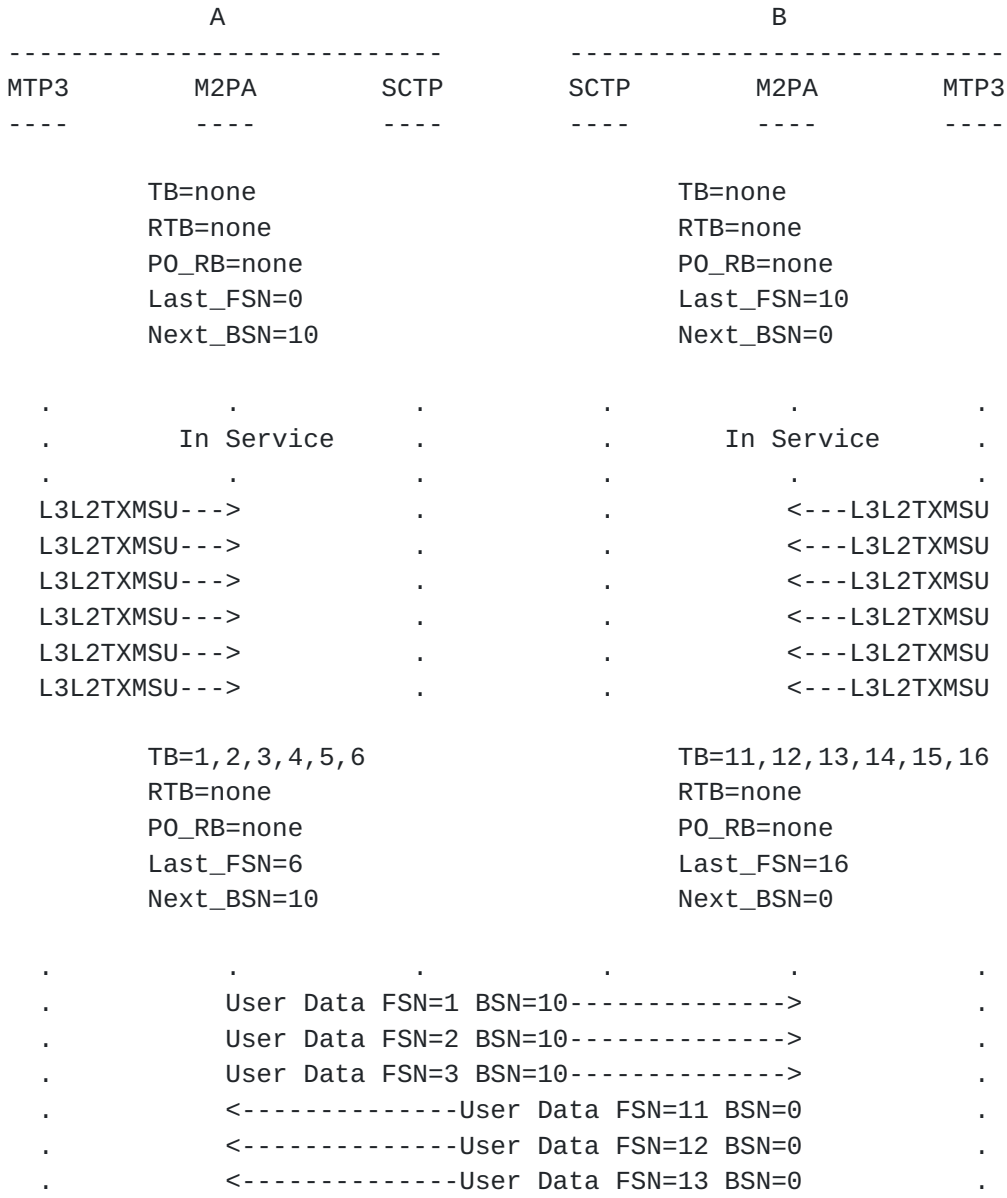
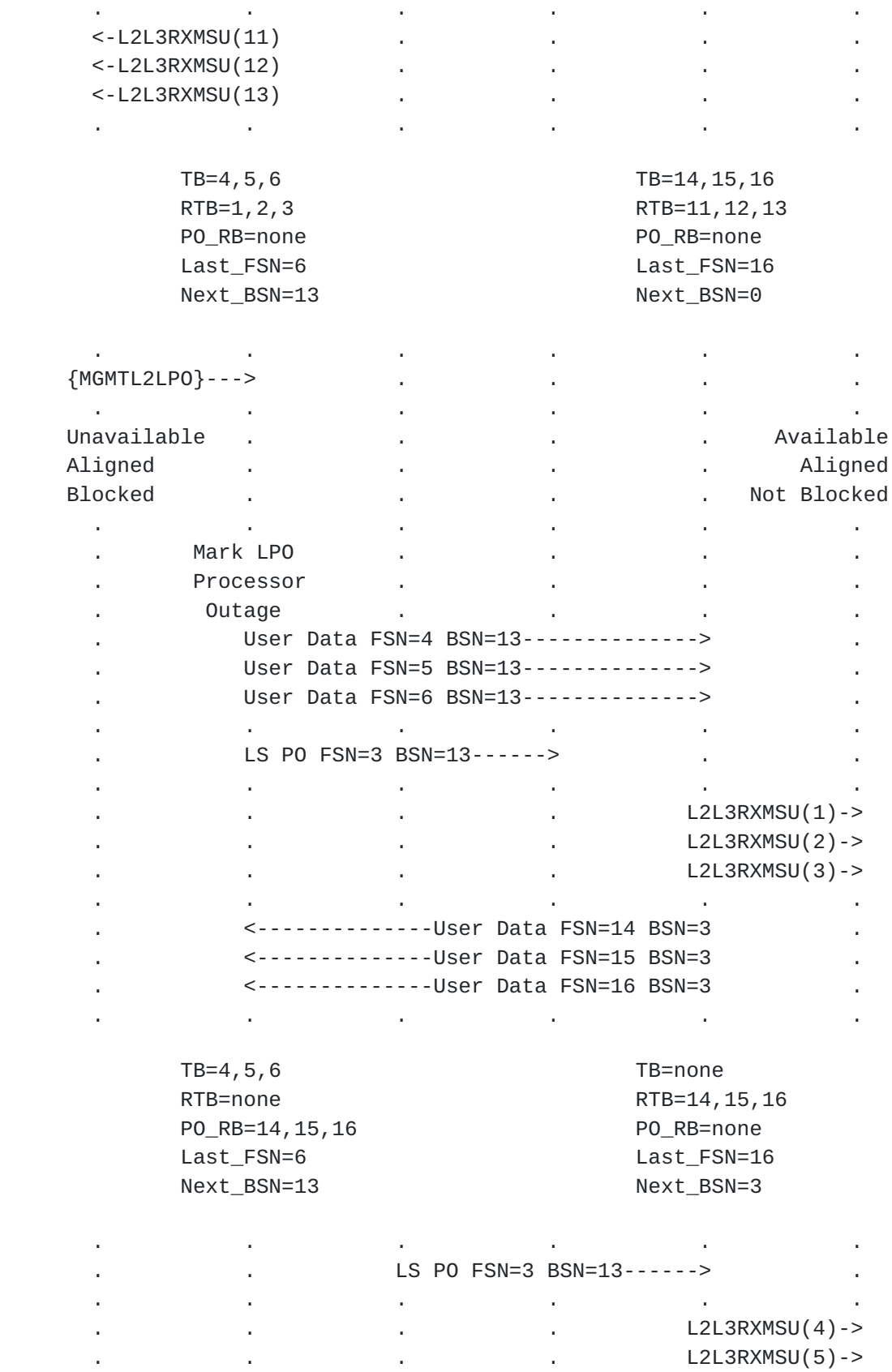


Figure 6. PO Example (Clear Buffers) [T1.111] Variant

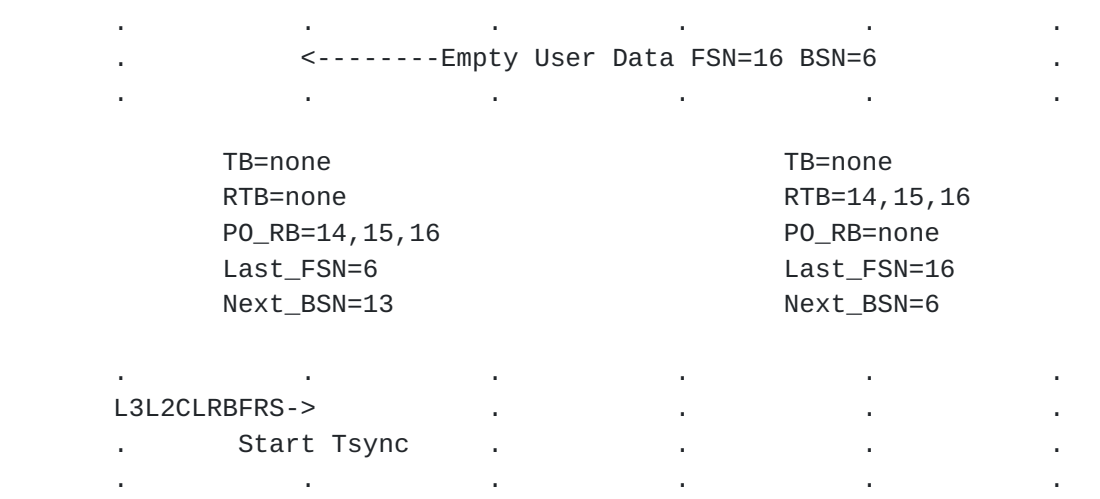
2.8.9. PO Example (Synchronization) [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the onset and abatement of local processor outage while the link is in service. This example assumes that MTP3 issues the Clear Buffers primitive, and the example provides details depicting sequence number synchronization.

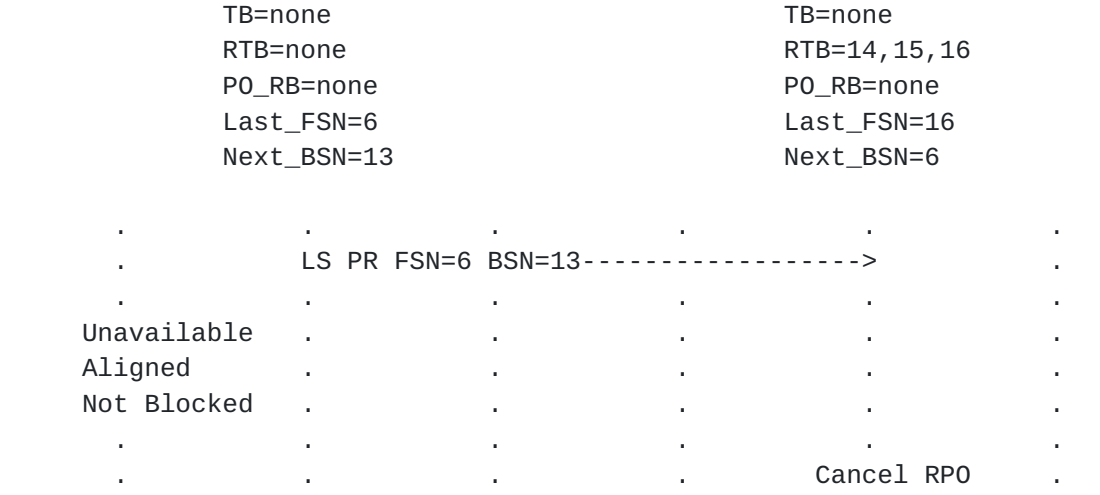


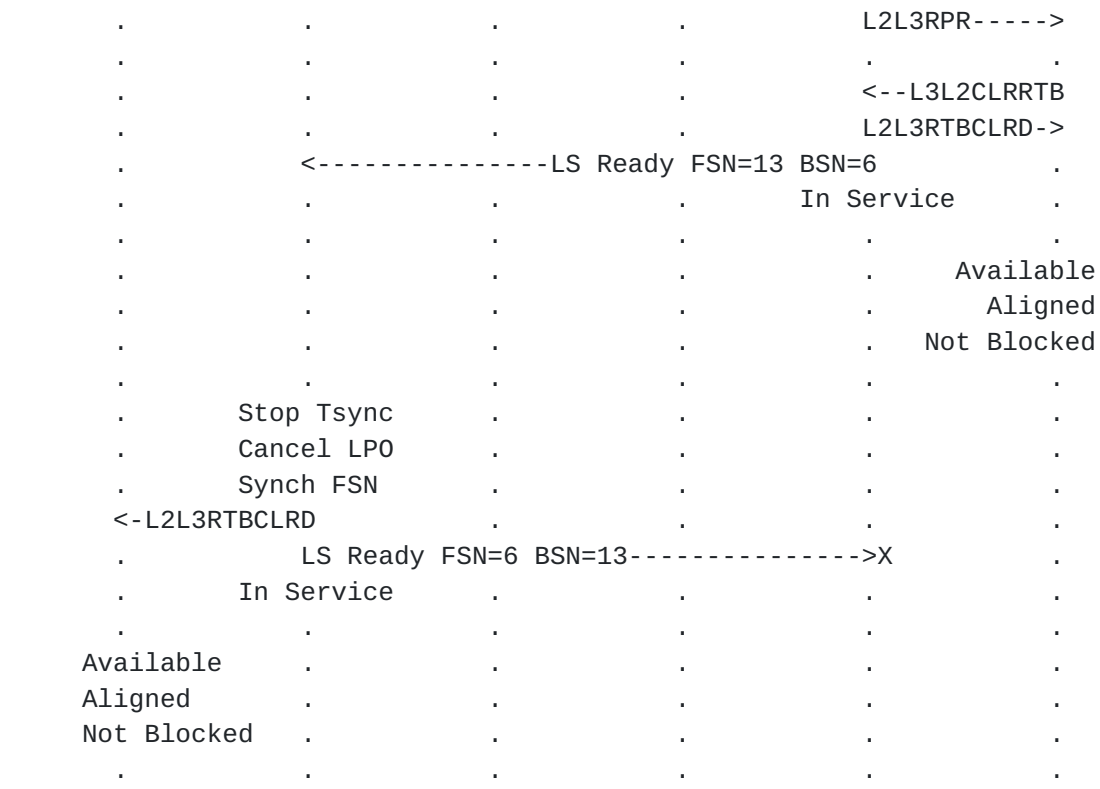

<-L2L3RXMSU(11)
<-L2L3RXMSU(12)
<-L2L3RXMSU(13)
TB=4,5,6
RTB=1,2,3
PO_RB=none
Last_FSN=6
Next_BSN=13
TB=14,15,16
RTB=11,12,13
PO_RB=none
Last_FSN=16
Next_BSN=0
{MGMTL2LPO}--->
Unavailable
Aligned
Blocked
Available
Aligned
Not Blocked
Mark LPO
Processor
Outage
User Data FSN=4 BSN=13-------------->
User Data FSN=5 BSN=13-------------->
User Data FSN=6 BSN=13-------------->
LS PO FSN=3 BSN=13------>
L2L3RXMSU(1)->
L2L3RXMSU(2)->
L2L3RXMSU(3)->
<--------------User Data FSN=14 BSN=3
<--------------User Data FSN=15 BSN=3
<--------------User Data FSN=16 BSN=3
TB=4,5,6
RTB=none
PO_RB=14,15,16
Last_FSN=6
Next_BSN=13
TB=none
RTB=14,15,16
PO_RB=none
Last_FSN=16
Next_BSN=3
LS PO FSN=3 BSN=13------>
L2L3RXMSU(4)->
L2L3RXMSU(5)->

While in LPO, (A) must buffer messages 14-16 without acknowledging them.

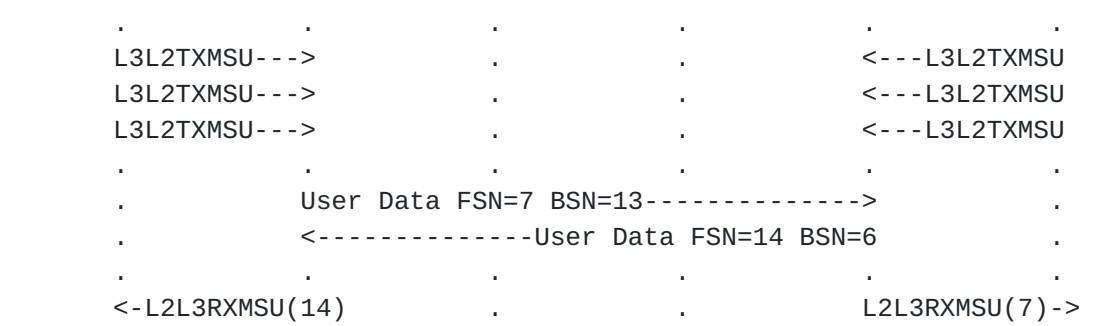


LPO ends at (A). (A) M2PA clears its PO_RB, TB, and RTB, but does not notify its MTP3 of clear completion. This prevents(A) MTP3 from delivering new MSUs for transmission until sequence number synchronization is complete.





(B) M2PA receives LS PR, uses the BSN to reset its Last FSN and notifies its MTP3. (B) MTP3 sends a Clear RTB primitive to M2PA, and M2PA clears its RTB and PO_RB. Upon completion of the clear, (B) M2PA notifies MTP3 and sends a synchronization LS Ready on the User Data stream. (A) receives the LS Ready, uses the BSN to reset its Last FSN, and then notifies its MTP3. (A) M2PA sends a LS Ready message on the User Data stream, and (B) M2PA receives it and ignores it. The peer sequence numbers are synchronized, and both peers are in service.



. . . . . .

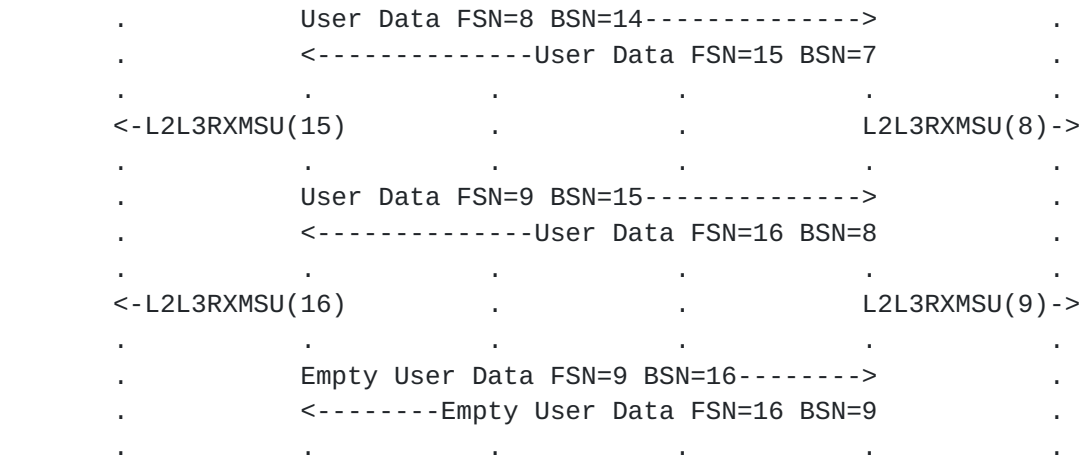
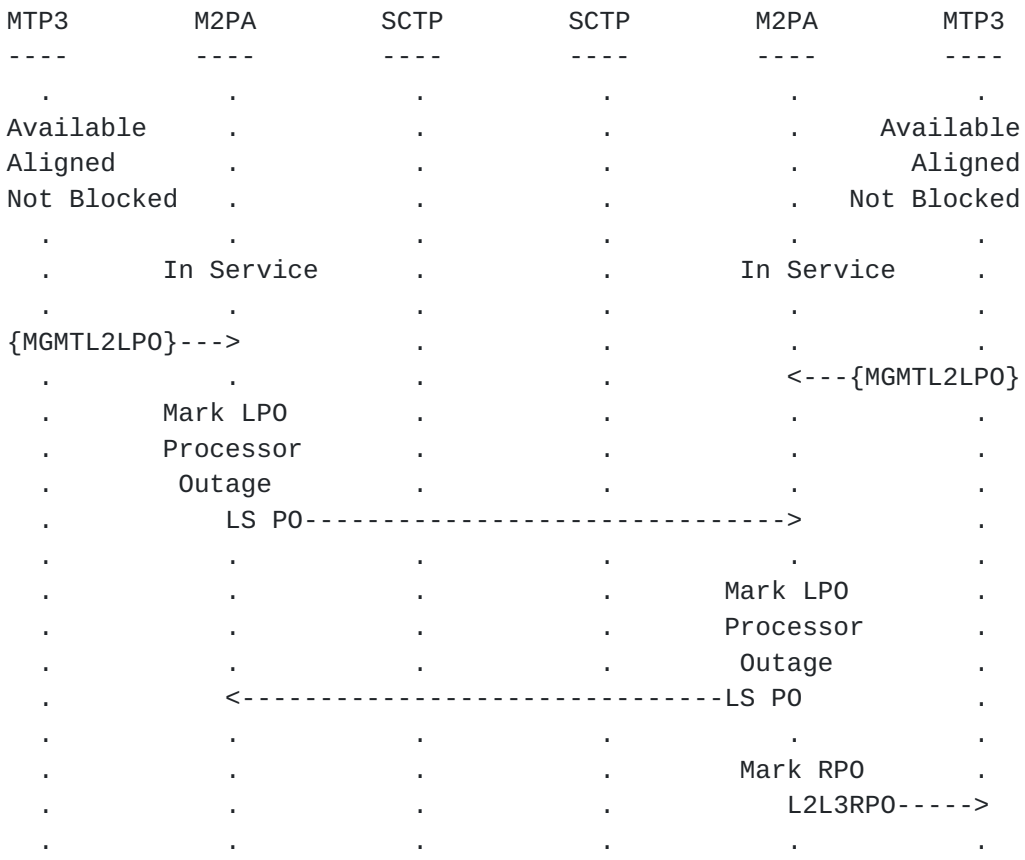


Figure 7. PO Example (Synchronization) [T1.111] Variant

2.8.10. PO Example (LPO and RPO onset) [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the nearly simultaneous onset of local processor outage on each peer while the link is in service. The processor outage condition clears only on the left side. This example assumes that MTP3 issues the Clear Buffers primitive.



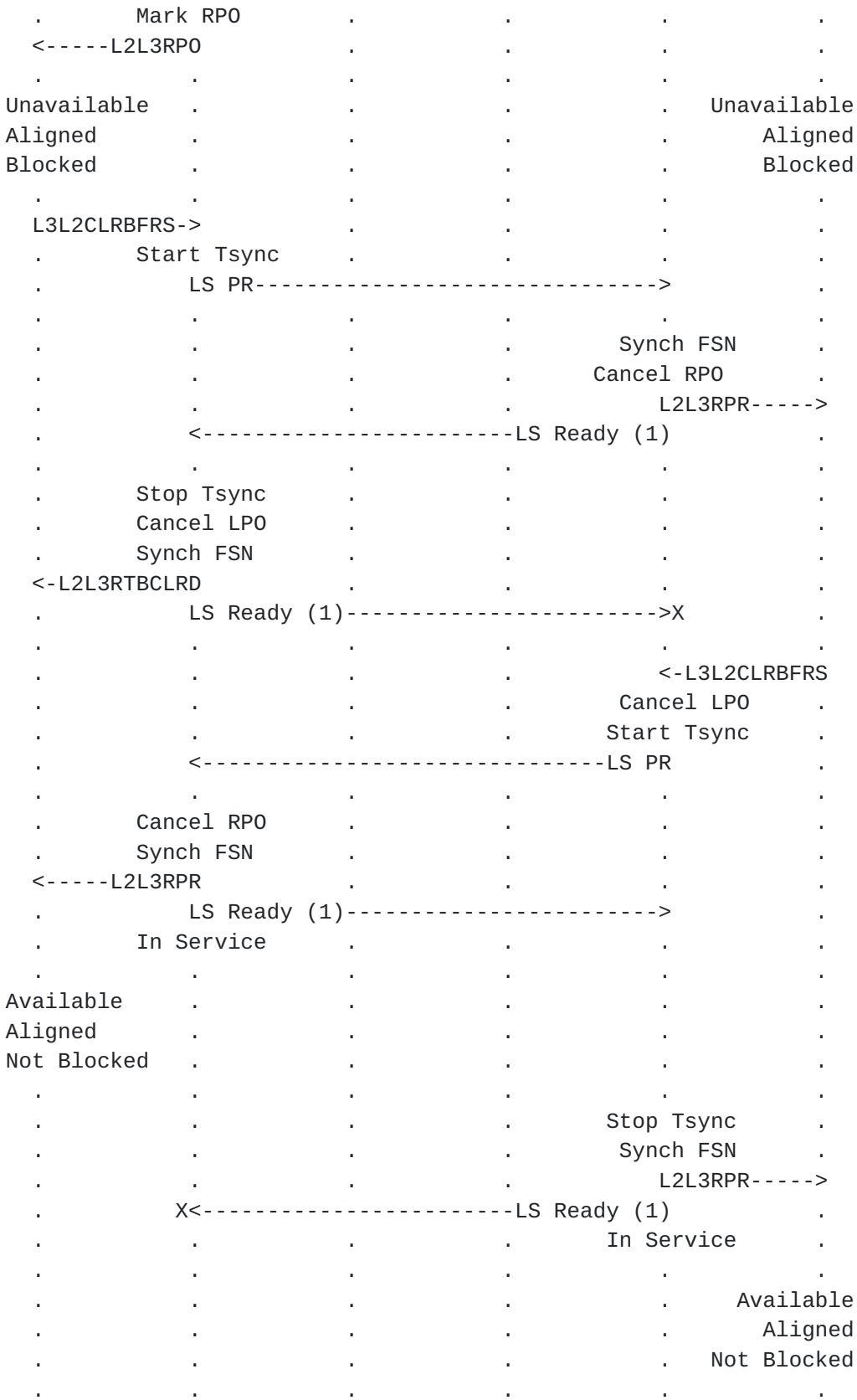
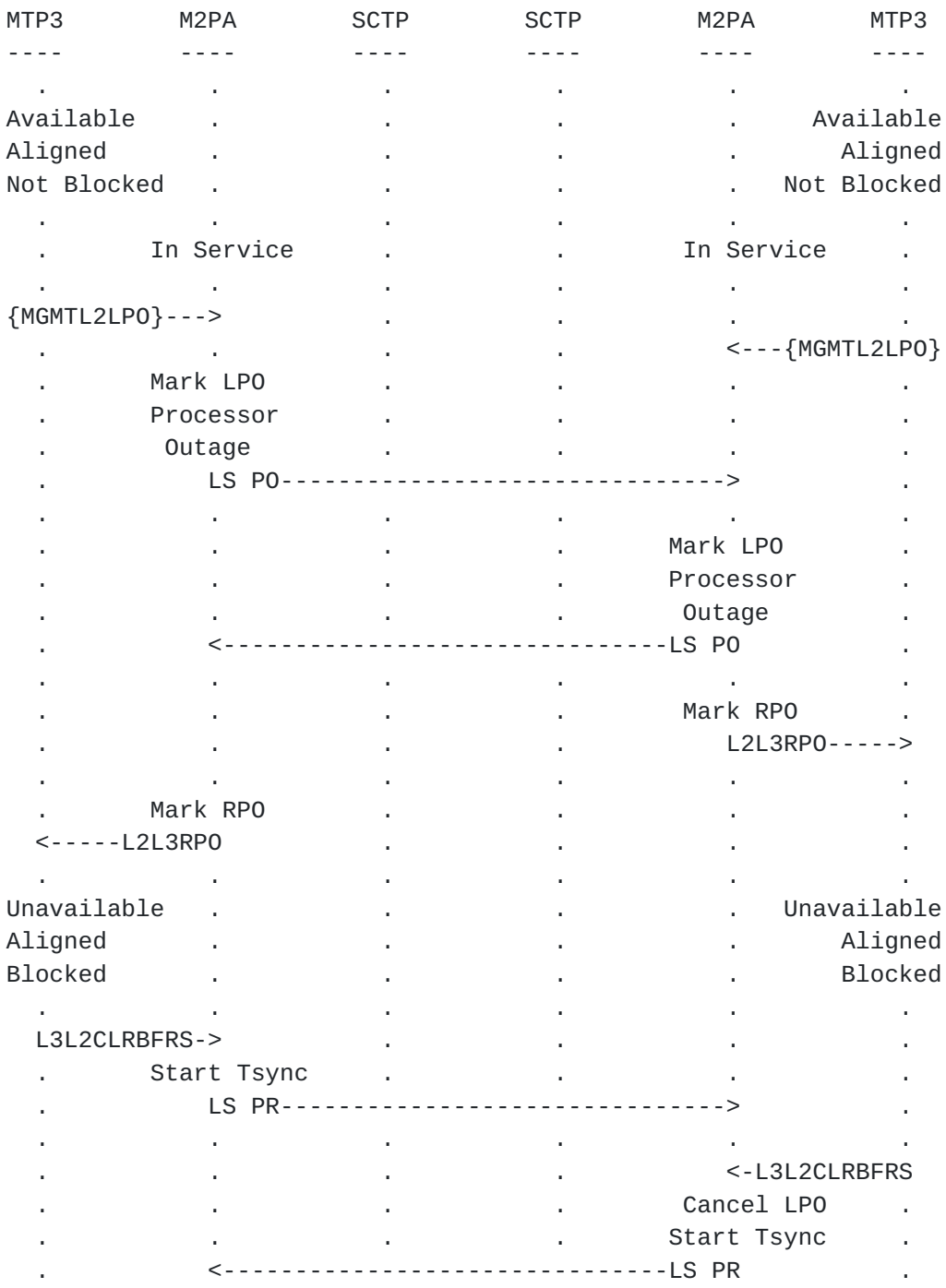


Figure 8. PO Example (LPO and RPO onset) [T1.111] Variant

2.8.11. PO Example (LPO and RPO abate) [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the nearly simultaneous onset and abtement of local processor outage on each peer while the link is in service. This example assumes that MTP3 issues the Clear Buffers primitive.



. . . . . .

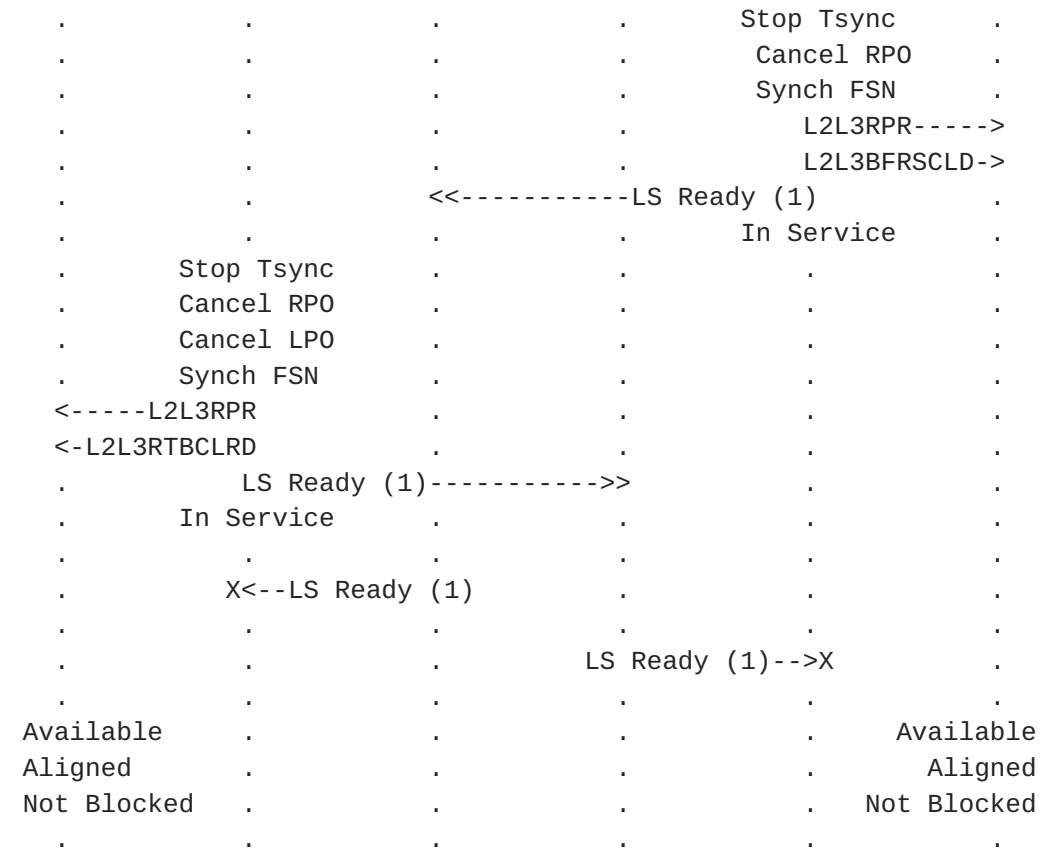
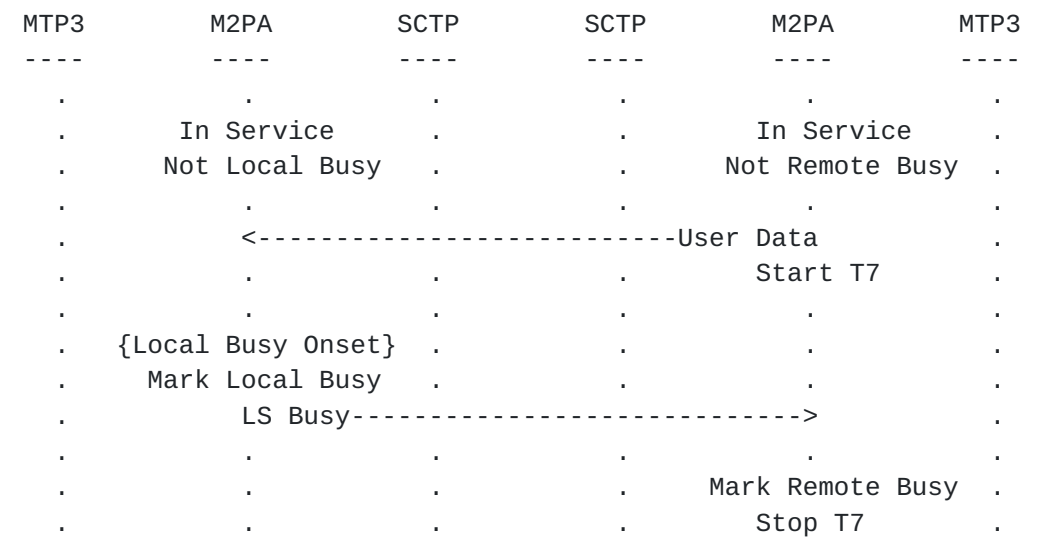


Figure 9. PO Example (LPO and RPO abate) [T1.111] Variant

2.8.12. L2 Busy Example (Simple) [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the onset of a local busy condition on one peer.



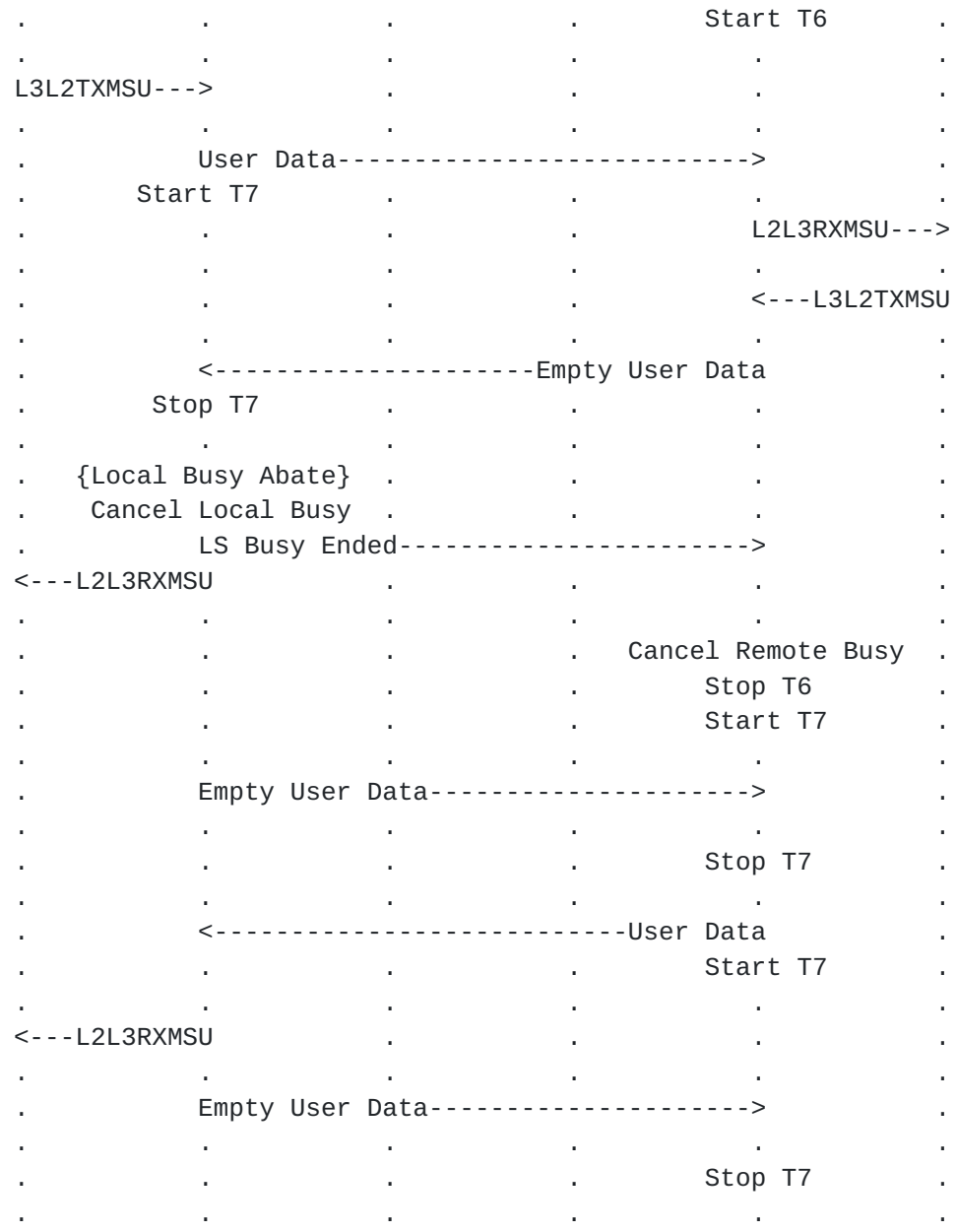
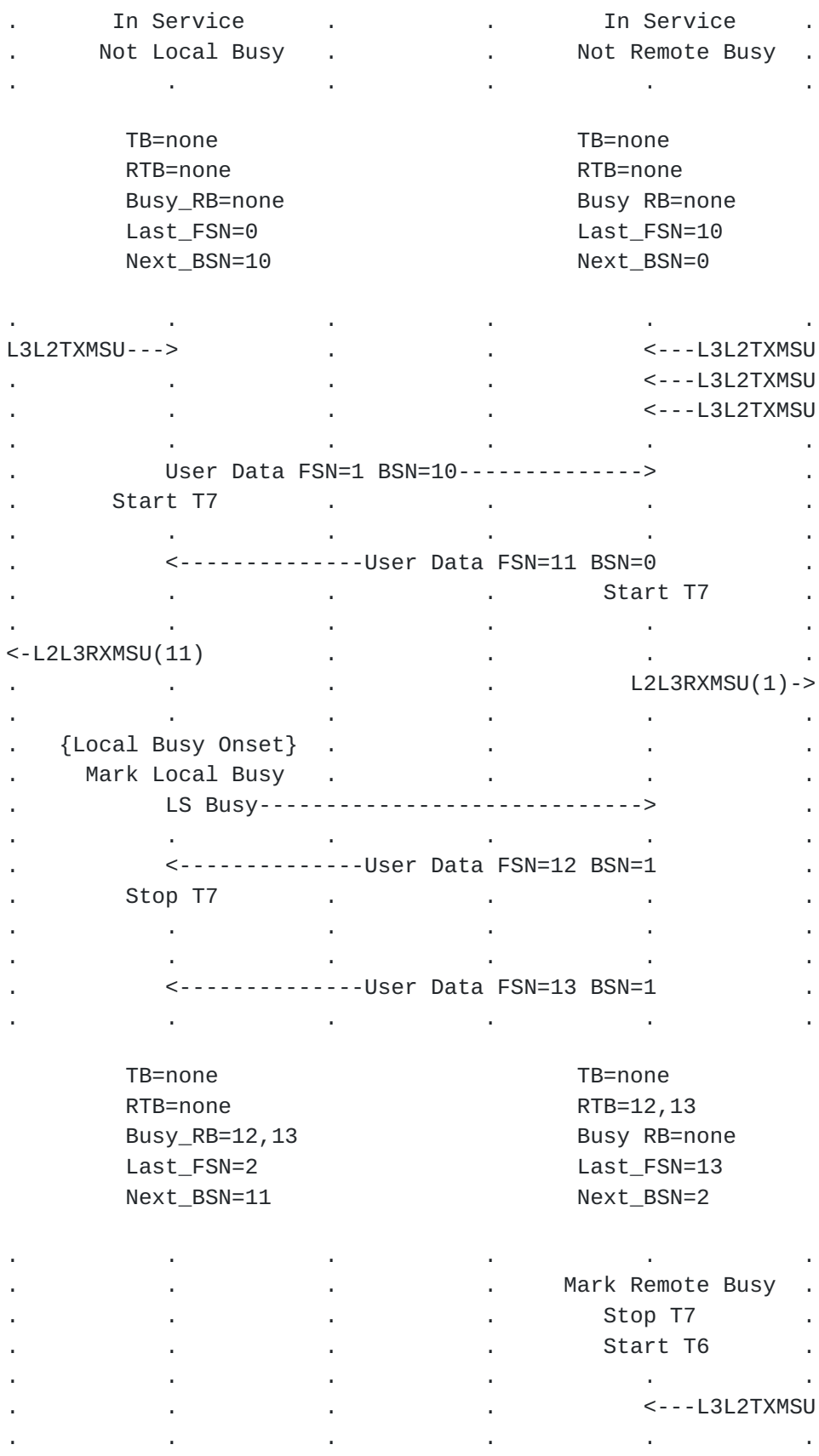


Figure 10. L2 Busy Example (One Side, Simple) [T1.111] Variant

2.8.13. L2 Busy Example (With FSN) [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the onset of a local busy condition on one peer and highlights the buffering of received non-empty User Data.


In Service
Not Local Busy
In Service
Not Remote Busy
TB=none
RTB=none
Busy_RB=none
Last_FSN=0
Next_BSN=10
TB=none
RTB=none
Busy RB=none
Last_FSN=10
Next_BSN=0
L3L2TXMSU--->
<---L3L2TXMSU
<---L3L2TXMSU
<---L3L2TXMSU
User Data FSN=1 BSN=10
Start T7
User Data FSN=11 BSN=0
Start T7
<-L2L3RXMSU(11)
L2L3RXMSU(1)->
{Local Busy Onset}
Mark Local Busy
LS Busy
User Data FSN=12 BSN=1
Stop T7
User Data FSN=13 BSN=1
TB=none
RTB=none
Busy_RB=12,13
Last_FSN=2
Next_BSN=11
TB=none
RTB=12,13
Busy RB=none
Last_FSN=13
Next_BSN=2
Mark Remote Busy
Stop T7
Start T6
<---L3L2TXMSU

```
     L3L2TXMSU--->           .           .           .           .
```

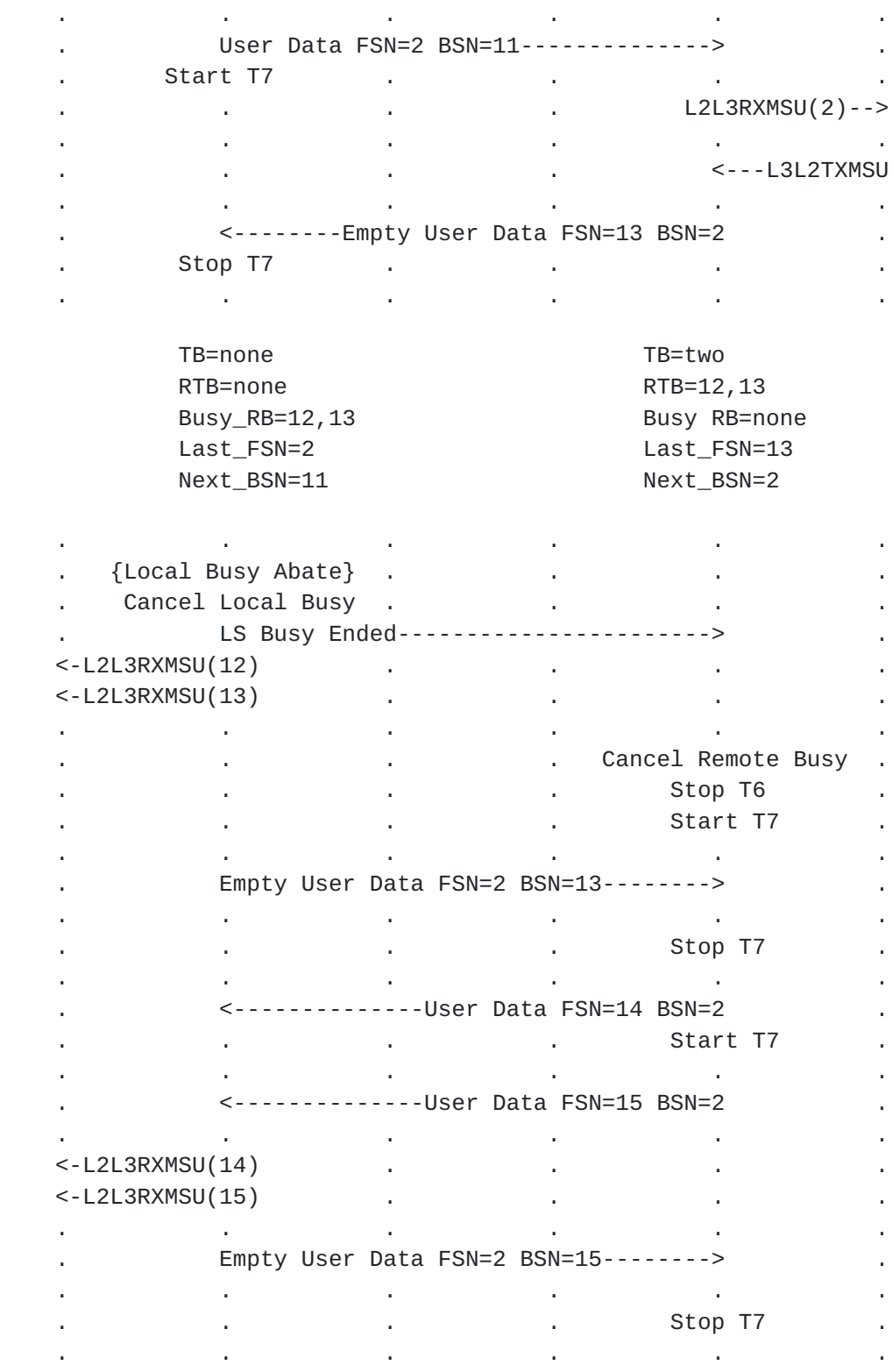
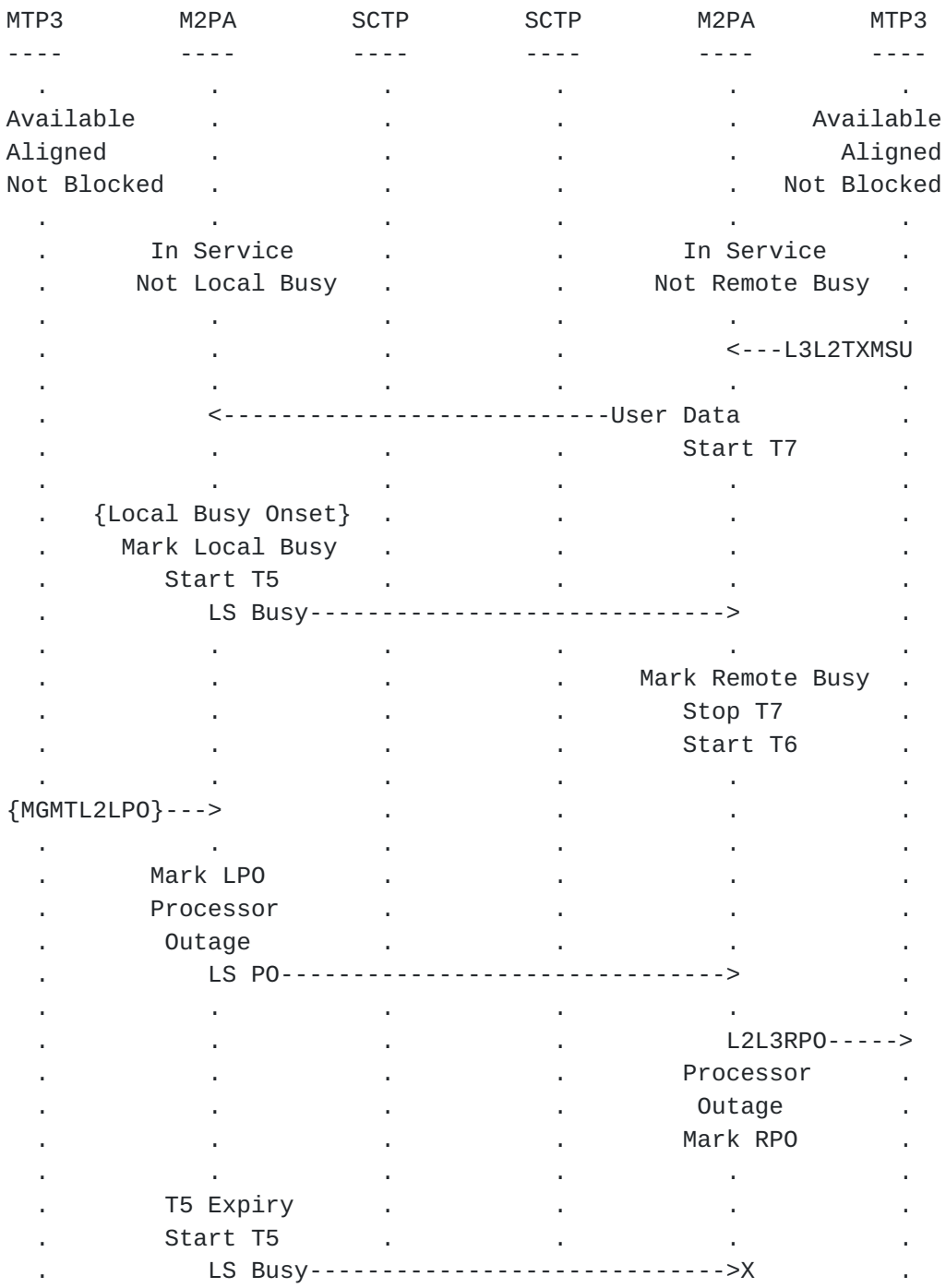


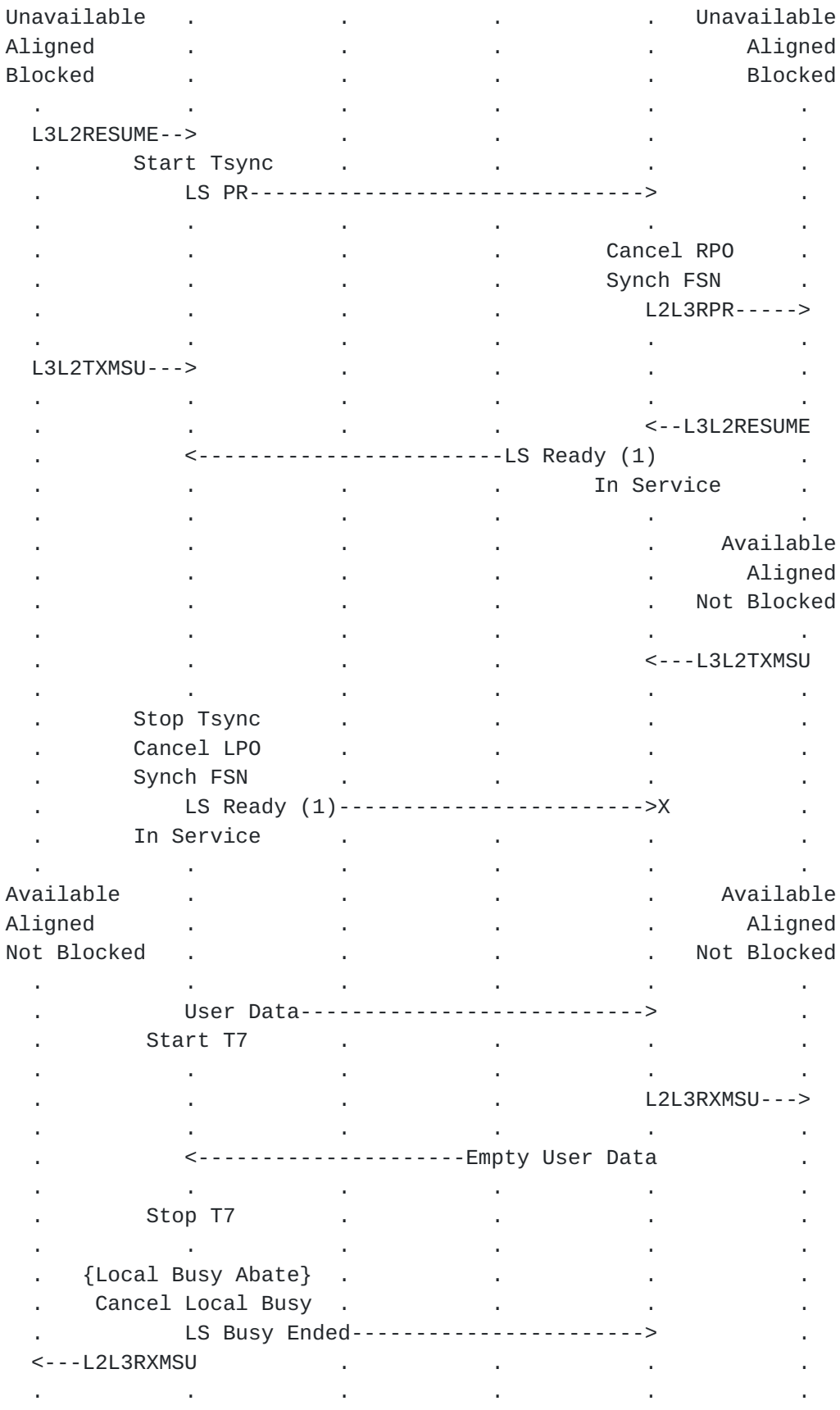
Figure 11. L2 Busy Example (One Side, With Data) [T1.111] Variant

2.8.14. Simultaneous Busy and PO Example [T1.111] Variant

This example is for a [T1.111] variant M2PA and depicts the onset of local congestion, followed by the onset of local processor outage, then the abatement of local processor outage, and finally the abatement of local busy, all occurring while the link is in service. This example assumes that MTP3 issues the Resume primitive, and that timer T6 does not expire.



. . . . . .


Unavailable
Aligned
Blocked
Unavailable
Aligned
Blocked
L3L2RESUME-->
Start Tsync
LS PR
Cancel RPO
Synch FSN
L2L3RPR----->
L3L2TXMSU--->
<--L3L2RESUME
<----------LS Ready (1)
In Service
Available
Aligned
Not Blocked
<---L3L2TXMSU
Stop Tsync
Cancel LPO
Synch FSN
LS Ready (1)----------X
In Service
Available
Aligned
Not Blocked
Available
Aligned
Not Blocked
User Data---------->
Start T7
L2L3RXMSU--->
<----------Empty User Data
Stop T7
{Local Busy Abate}
Cancel Local Busy
LS Busy Ended---------->
<---L2L3RXMSU

. . . . Cancel Remote Busy .

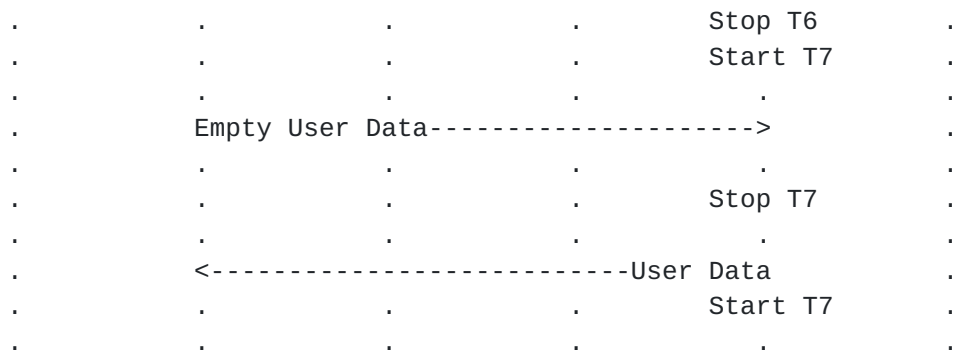


Figure 12. Simultaneous Busy and PO Example [T1.111] Variant

3. Security Considerations

No new threats have been identified in M2PA [RFC4165].

4. Acknowledgments

The author thanks Mark Davidson and many others for their invaluable comments.

5. References

[RFC4165] George, T., Bidulock, B., Dantu, R., Schwarzbauer, H., Morneault, K., "Signaling System 7 (SS7) Message Transfer Part 2 (MTP2) - User Peer-to-Peer Adaptation Layer (M2PA)", RFC 4165, September 2005.

[Q.703] ITU, "Signaling System No. 7 - Signaling Link," ITU-T Recommendation Q.703, ITU-T Telecommunication Standardization Sector of ITU (July 1996).

[Q.2140] ITU, "B-ISDN ATM Adaptation Layer - Service Specific Coordination Function for Signaling at the Network Node Interface (SSCF at NNI)," ITU-T Recommendation Q.2140, ITU-T Telecommunication Standardization Sector of ITU (February 1995).

[T1.111] ANSI, "American National Standard for Telecommunications - Signaling System Number 7 (SS7) - Message Transfer Part (MTP)," ANSI T1.111-2001, American National Standards Institute (February 2001).

Appendix A. Changes Control

Version	Summary of Changes
00	Document created.

Author's Address

Jeffrey Craig
Tekelec Inc.
5200 Paramount Parkway Phone: 1-919-380-3870
Morrisville, NC USA 27560 Email: jeffrey.craig@tekelec.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.