Message Sequence Charts

PPP

Document Number:	8441.208.99
Version:	006
Status:	Proposed
Approval Authority:	
Creation Date:	1999-Sep-27
Last changed:	2003-Nov-6 by Alexander Till
File Name:	ppp.doc

Important Notice

Change History

Date	Changed by	Approved by	Version	Status	Notes
1999-Sep-27	Steffen Winter		001	Proposed	1
1999-Dec-08	Steffen Winter		002	Proposed	2
2000-May-31	Steffen Winter		003	Proposed	3
2001-Sep-24	Ola Flatus		004	Proposed	4
2003-Mar-25	Steffen Winter		005	Proposed	5
2003-Nov-6	Alexander Till		006	Proposed	6

Notes:

1. Initial
2. PPP_PDP_ACTIVATE_REJ primitive added.
3. Transparent mode and client mode added, DTI interface included and Link Quality Monitoring removed.
4. DTILIB changes
5. Changes to TI Standard
6. CHAP authentication in client mode added.

Table of Contents

List of References

[1] GSM 05.02 version 8.0.0 Release 1999
Digital cellular telecommunications system (Phase 2+);
Multiplexing and multiple access on the radio path

[2] GSM 04.60 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Mobile Station (MS) - Base Station System (BSS) interface;
Radio Link Control/ Medium Access Control (RLC/MAC) protocol

[3] GSM 04.08 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Mobile radio interface layer 3 specification

[4] GSM 03.64 version 6.1.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Overall description of the GPRS radio interface; Stage 2

[5] GSM 03.60 version 6.3.1 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Service description; Stage 2

[6] GSM 04.07 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Mobile radio interface signalling layer 3; General aspects

[7] GSM 04.64 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Mobile Station - Serving GPRS Support Node (MS-SGSN)
Logical Link Control (LLC) layer specification

[8] GSM 05.08 version 6.4.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Radio subsystem link control

[9] GSM 05.10 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Radio subsystem synchronization

[10] GSM 03.20 TS 100 929: July 1998 (GSM 03.20 version 6.0.1)
Security related network functions, ETSI

[11] Draft GSM 03.22: August 1998 (GSM 03.22 version 6.1.0)
Functions related to Mobile Station (MS) in idle mode and group receive mode, ETSI

[12] GSM 04.65 V6.3.0: Subnetwork Dependant Convergence Protocol
ETSI, March 1999

[13] ITU-T V42bis ITU-T, Recommendation V.42 bis 1990

[14] GSM 09.60 GPRS Tunneling Protocol (GTP) across the Gn and Gp Interface

[15] RFC 1661 IETF STD 51 July 1994
The Point-to-Point Protocol (PPP)

[16] RFC 1662 IETF STD 51 July 1994
PPP in HDLC-like Framing

[17] RFC 1570 January 1994
PPP LCP Extensions

[18] RFC 1989 August 1996
PPP Link Quality Monitoring

[19] RFC 1332 May 1992
The PPP Internet Protocol Control Protocol (IPCP)

[20] RFC 1877 December 1995
PPP IPCP Extensions for Name Server Addresses

[21] RFC 2153 May 1997
PPP Vendor Extensions

[22] RFC 1334 October 1992
PPP Authentication Protocols (for Password Authentication Protocol only)

[23] RFC 1994 August 1996
PPP Challenge Handshake Authentication Protocol (CHAP)

[24] TIA/EIA-136-370
Packet-Data Services – Enhanced General Packet Radio for TIA/EIA-136 (EGPRS-136) - Overview, Telecommunications Industry Association

[25] TIA/EIA-136-376
Packet-Data Services – EGPRS-136 Mobility Management, Telecommunications Industry Association

[26] TIA/EIA-136-972
Packet-Data Services – Stage 2 Description, Telecommunications Industry Association

1 Overview

The Protocol Stacks are used to define the functionality of the GSM protocols for interfaces. The GSM specifications are normative when used to describe the functionality of interfaces, but the stacks and the subdivision of protocol layers does not imply or restrict any implementation.

The protocol stack for GPRS consists of several entities. Each entity has one ore more service access points, over which the entity provides a service for the upper entity.

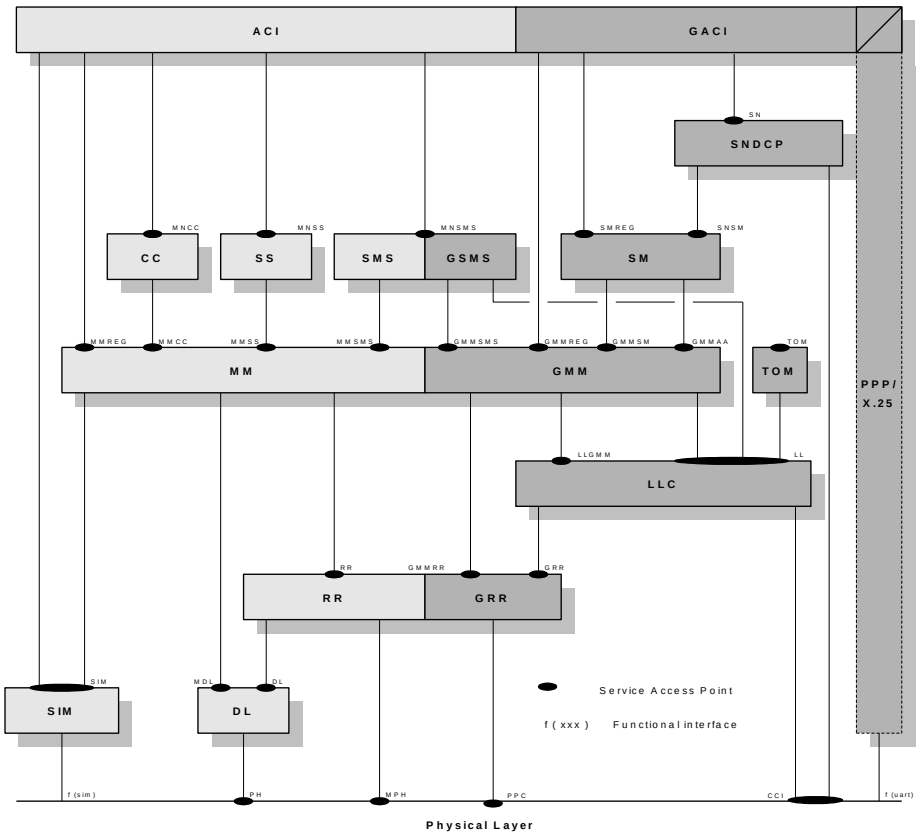


Figure 1-1: Architecture of the GSM/GPRS protocol stack

The information units passed via the SAPs are called primitives and consists of an operation code and several parameters. See the Users Guide for details.

The entities of the GPRS protocol stack are:

1.1 GRR (RLC/MAC) – Radio Link Control/Medium Access Control

This layer contains two functions: The Radio Link Control function provides a radio-solution-dependent reliable link. The Medium Access Control function controls the access signalling (request and grant) procedures for the radio channel, and the mapping of LLC frames onto the GSM physical channel.

1.2 LLC – Logical Link Control

The LLC entity provides multiple highly reliable logical links for asynchronous data transfer between the MS and the network. It supports variable-length information frames, acknowledged and unacknowledged data transfer, flow and sequence control, error detection and recovery, notification of unrecoverable errors, user identity confidentiality, and ciphering of user and signaling data.

1.3 GMM – GPRS Mobility Management

The GMM entity provides procedures for the mobility of the MS, such as informing the network of its present location, and user identity confidentiality. It manages the GMM context (attach, detach, routing area updating), supports security functions such as authentication of user and MS, controls ciphering of data, and initiates the response to paging messages.

1.4 SM – Session Management

The main function of the session management (SM) is to support PDP context handling of the user terminal. Session Management activates, modifies and deletes the contexts for packet data protocols (PDP). Session Management services are provided at the SMREG-SAP and the SNSM-SAP for anonymous and non-anonymous access. The non-anonymous and anonymous access procedures for PDP context activation and PDP context deactivation are available at the SMREG-SAP. In addition there exists a PDP context modification for non-anonymous PDP contexts.

1.5 SNDCP - Subnetwork Dependant Convergence Protocol

SNDCP carries out all functions related to transfer of Network layer Protocol Data Units (N-PDUs) over GPRS in a transparent way. SNDCP helps to improve channel efficiency by means of compression techniques. The set of protocol entities above SNDCP consists of commonly used network protocols. They all use the same SNDCP entity, which then performs multiplexing of data coming from different sources to be sent using the service provided by the LLC layer.

1.6 GACI – GPRS Application Control Interface

The GACI is the GPRS extension of the ACI. It is specified in GSM 07.07 and 07.60. It is responsible for processing of the GPRS related AT Commands to setup, activate and deactivate the PDP context parameter. It also provides functionality for the interworking between GMM/SM/SNDCP and a packet oriented protocol like PPP.

1.7 USART - Universal Synchronous Asynchronous Receiver Transmitter Driver

The USART is a hardware component that facilitates a connection between the mobile station and terminal equipment (e.g. a PC). This interface uses some of the circuits described in V.24.

The data exchange provided by this unit is serial and asynchronous (synchronous communication is not in the scope of this document). A driver that uses interrupts to manage a circular buffer for the sending and receiving direction is necessary in order to use this component in the GPRS. The driver has to be able to perform flow control.

1.8 TOM – Tunnelling of Messages

The TOM entity is present if and only if HS136 is supported (the feature flag FF_HS136 is enabled).

The main function of TOM is to tunnel non-GSM signalling messages between the MS and the SGSN. The only non-GSM signalling which is currently supported by TOM is for the EGPRS-136 system (according to TIA/EIA-136-376). Data transfer in both uplink and downlink direction is possible. Two different priorities (high, low) of signalling data transfer are supported. TOM uses the unacknowledged mode of LLC and the acknowledged mode of GRR (RLC/MAC).

2 Introduction

2.1 PPP Link Operation

In order to establish communications over a point-to-point link, each end of the PPP link must first send LCP packets to configure and test the data link. After the link has been established, the peer may be authenticated.

Then, PPP must send IPCP packets to configure the network-layer protocol IP. Once the network-layer protocol has been configured, IP datagrams can be sent over the link.

The link will remain configured for communications until explicit LCP or IPCP packets close the link down, or until some external event occurs (network administrator intervention).

In the process of configuring, maintaining and terminating the point-to-point link, the PPP link goes through several distinct phases which are specified in the following simplified state diagram:

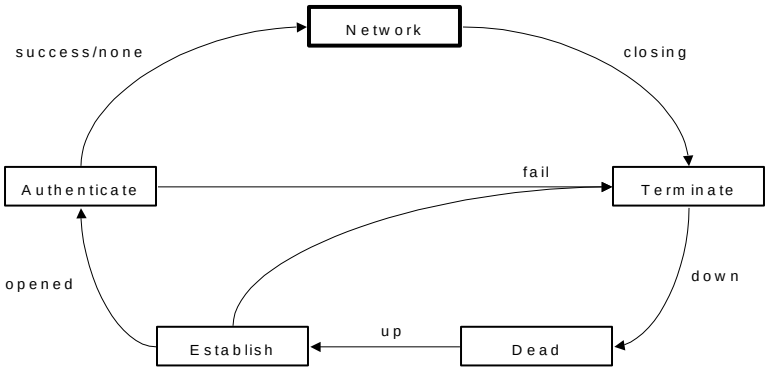


Figure 2-1: Phase Diagram

2.1.1 Link Dead (physical-layer not ready)

The link necessarily begins and ends with this phase. When an external event (such as network administrator configuration) indicates that the physical-layer is ready to be used, PPP will proceed to the Link Establishment phase.

During this phase, the LCP automaton (see The Option Negotiation Automaton) will be in the Closed state. The transition to the Link Establishment phase will signal an Open event to the LCP automaton.

PPP returns to this phase automatically after the disconnection of the link.

2.1.2 Link Establishment Phase

The Link Control Protocol (LCP) is used to establish the connection through an exchange of Configure packets. This exchange is complete, and the LCP Opened state entered, once a Configure-Ack packet (see LCP and IPCP Packet Format) has been both sent and received.

All Configuration Options are assumed to be at default values unless altered by the configuration exchange. It is important to note that only Configuration Options which are independent of particular network-layer protocols are configured by LCP. Configuration of the individual network-layer protocol is handled by the separate Network Control Protocol IPCP during the Network-Layer Protocol phase.

Any non-LCP packets received during this phase must be silently discarded.

The receipt of the LCP Configure-Request causes a return to the Link Establishment phase from the Network-Layer Protocol phase or Authentication phase.

2.1.3 Authentication Phase

On some links it may be desirable to require a peer to authenticate itself before allowing network-layer protocol packets to be exchanged.

By default, authentication is not mandatory. If the implementation desires that the peer authenticate with some specific authentication protocol, then the use of that authentication protocol is requested during Link Establishment phase.

Authentication should take place as soon as possible after link establishment.

Advancement from the Authentication phase to the Network-Layer Protocol phase will not occur until authentication has completed. If authentication fails, the implementation should proceed instead to the Link Termination phase.

Only Link Control Protocol and authentication protocol packets are allowed during this phase. All other packets received during this phase must be silently discarded.

The implementation responsible for commencing Link Termination phase is the implementation which has refused authentication to its peer.

2.1.4 Network-Layer Protocol Phase

Once PPP has finished the previous phases, the network-layer protocol (IP) must be configured by the appropriate Network Control Protocol (IPCP).

After IPCP has reached the Opened state, PPP will carry IP packets. Any IP packets received when IPCP is not in the Opened state must be silently discarded.

While LCP is in the Opened state, any protocol packet which is unsupported by the implementation must be returned in a Protocol-Reject (see LCP and IPCP Packet Format). Only protocols which are supported are silently discarded.

During this phase, link traffic consists of any possible combination of LCP, NCP, and network-layer protocol packets.

2.1.5 Link Termination Phase

PPP can terminate the link at any time. This might happen because of the loss of carrier, authentication failure, or the administrative closing of the link.

LCP is used to close the link through an exchange of Terminate packets.

The sender of the Terminate-Request should disconnect after receiving a Terminate-Ack, or after the Restart counter expires. The receiver of a Terminate-Request should wait for the peer to disconnect, and must not disconnect until at least one Restart time has passed after sending a Terminate-Ack. PPP should proceed to the Link Dead phase.

Any non-LCP packets received during this phase must be silently discarded.

2.2 PPP Components

The following figure shows the components of PPP.

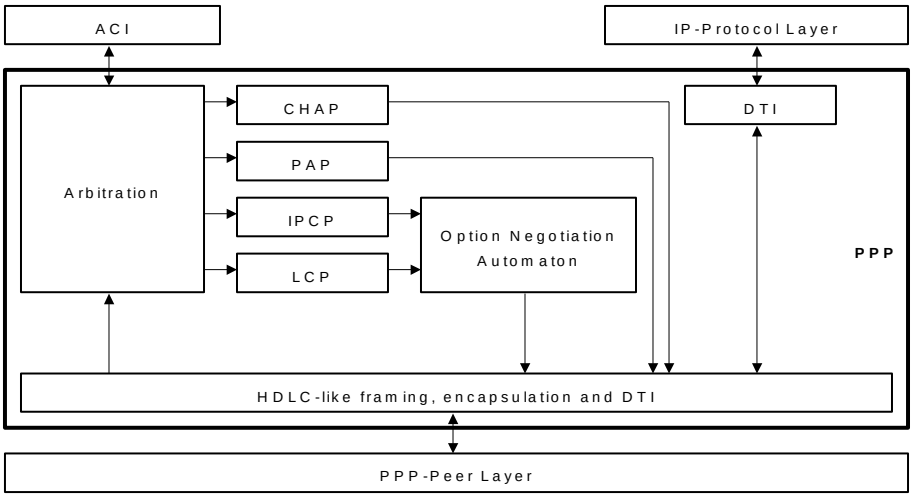


Figure 2-2: PPP Components

2.3 Frame Format

A summary of the PPP HDLC-like frame structure is shown below. This figure does not include octets inserted for transparency. The fields are transmitted from left to right.

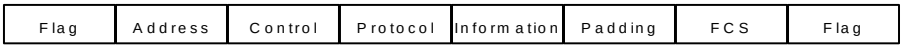


Figure 2-3: Frame Structure

2.3.1 Flag Sequence

Each frame begins and ends with a Flag Sequence, which is the binary sequence 01111110 (hexadecimal 0x7e). The implementations continuously check for this flag, which is used for frame synchronization.

Only one Flag Sequence is required between two frames. Two consecutive Flag Sequences constitute an empty frame, which is silently discarded, and not counted as a FCS error.

2.3.2 Address Field

The Address field is a single octet, which contains the binary sequence 11111111 (hexadecimal 0xff), the All-Stations address. Individual station addresses are not assigned. The All-Stations address must always be recognized and received.

Frames with unrecognized Addresses should be silently discarded.

The Address field can be omitted if Address-and-Control-Field-Compression is negotiated.

2.3.3 Control Field

The Control field is a single octet, which contains the binary sequence 00000011 (hexadecimal 0x03), the Unnumbered Information (UI) command with the Poll/Final (P/F) bit set to zero.

Frames with unrecognized Control field values should be silently discarded.

The Control field can be omitted if Address-and-Control-Field-Compression is negotiated.

2.3.4 Protocol Field

The Protocol field is one or two octets, and its value identifies the datagram encapsulated in the Information field of the packet. The field is transmitted and received most significant octet first.

The structure of this field is consistent with the ISO 3309 extension mechanism for address fields. All Protocols must be odd; the least significant bit of the least significant octet must equal "1". Also, all Protocols must be assigned such that the least significant bit of the most significant octet equals "0". Frames received which don't comply with these rules must be treated as having an unrecognized Protocol.

If the most significant octet equals to zero and Protocol-Field-Compression is negotiated, then the most significant octet can be omitted.

2.3.5 Information Field

The Information field is zero or more octets. The Information field contains the datagram for the protocol specified in the Protocol field.

The maximum length for the Information field, including Padding, but not including the Protocol field, is termed the Maximum Receive Unit (MRU), which defaults to 1500 octets. By negotiation, consenting PPP implementations may use greater values for the MRU.

2.3.6 Padding

On transmission, the Information field may be padded with an arbitrary number of octets up to the MRU. It is the responsibility of each protocol to distinguish padding octets from real information.

2.3.7 Frame Check Sequence (FCS) Field

The Frame Check Sequence field defaults to 16 bits (two octets). The FCS is transmitted least significant octet first, which contains the coefficient of the highest term.

The FCS field is calculated over all bits of the Address, Control, Protocol, Information and Padding fields, not including any octets inserted for transparency. This also does not include the Flag Sequences nor the FCS field itself. When octets are received which are flagged in the Async-Control-Character-Map, they are discarded before calculating the FCS.

The end of the Information and Padding fields is found by locating the closing Flag Sequence and removing the Frame Check Sequence field.

2.4 LCP and IPCP Packet Format

There are three classes of LCP and IPCP packets:

- Link Configuration packets used to establish and configure a link (Configure-Request, Configure-Ack, Configure-Nak and Configure-Reject).
- Link Termination packets used to terminate a link (Terminate-Request and Terminate-Ack).
- Link Maintenance packets used to manage and debug a link (Code-Reject, Protocol-Reject (LCP only), Echo-Request (LCP only), Echo-Reply (LCP only), and Discard-Request (LCP only)).

In the interest of simplicity, there is no version field in the packet. A correctly functioning implementation will always respond to unknown Protocols and Codes with an easily recognizable packet, thus providing a deterministic fallback mechanism for implementations of other versions.

Regardless of which Configuration Options are enabled, all Link Configuration, Link Termination, and Code-Reject packets (codes 1 through 7) are always sent as if no Configuration Options were negotiated. In particular, each Configuration Option specifies a default value. This ensures that such packets are always recognizable, even when one end of the link mistakenly believes the link to be open.

Exactly one LCP packet or one IPCP packet is encapsulated in the PPP Information field, where the PPP Protocol field indicates type hex c021 (Link Control Protocol) or type hex 8021 (IP Control Protocol).

A summary of the packet format is shown below. The fields are transmitted from left to right.

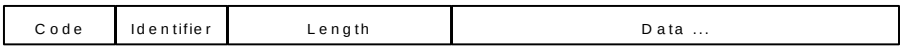


Figure 2-4: LCP and IPCP Packet Format

2.4.1 Code field

The Code field is one octet, and identifies the kind of packet. When a packet is received with an unknown Code field, a Code-Reject packet is transmitted.

This implementation supports the following values:

1 Configure-Request

2 Configure-Ack

3 Configure-Nak

4 Configure-Reject

5 Terminate-Request

6 Terminate-Ack

7 Code-Reject

8 Protocol-Reject (LCP only)

9 Echo-Request (LCP only)

10 Echo-Reply (LCP only)

11 Discard-Request (LCP only)

2.4.2 Identifier field

The Identifier field is one octet, and aids in matching requests and replies. When a packet is received with an invalid Identifier field, the packet is silently discarded without affecting the automaton.

If the Code field indicates a Request packet (codes 1, 5 and 9), then the Identifier field must be changed whenever the content of the Data field changes, and whenever a valid reply has been received for a previous request. For retransmissions, the Identifier may remain unchanged.

If the Code field indicates a reply of a Request packet (codes 2 through 4, 6 and 10), then the Identifier field is a copy of the Identifier field of the Request which caused this reply.

If the Code field indicates any other packet (codes 7, 8 and 11), then the Identifier field must be changed for each such packet sent.

2.4.3 Length field

The Length field is two octets, and indicates the length of the packet, including the Code, Identifier, Length and Data fields. The Length must not exceed the MRU of the link.

Octets outside the range of the Length field are treated as padding and are ignored on reception. When a packet is received with an invalid Length field, the packet is silently discarded without affecting the automaton.

2.4.4 Data field

The Data field is zero or more octets, as indicated by the Length field. The format of the Data field is determined by the Code field.

2.4.4.1 Options field

If the Code field indicates a Link Configuration packet (codes 1 through 4), then the Data field is shown as Options field.

The Options field is variable in length, and contains the list of zero or more Configuration Options that the sender desires to negotiate, acknowledge, Nak or reject. All Configuration Options are always negotiated, acknowledged, Nak'd or rejected simultaneously. The format of Configuration Options is protocol specific and further described in a later chapter.

2.4.4.2 Data field

If the Code field indicates a Link Termination packet (codes 5 and 6), then the Data field contains uninterpreted data for use by the sender. The data may consist of any binary value.

2.4.4.3 Rejected-Packet field

If the Code field indicates a Code-Reject packet (code 7), then the Data field is shown as Rejected-Packet field.

The Rejected-Packet field contains a copy of the LCP or IPCP packet which is being rejected. It begins with the Information field, and does not include any Data Link Layer headers nor an FCS. The Rejected-Packet must be truncated to comply with the peer's established MRU.

2.4.4.4 Rejected-Protocol

If the Code field indicates a Protocol-Reject packet (code 8), then the Data field is shown as Rejected-Protocol field and Rejected-Information field.

The Rejected-Protocol field is two octets, and contains the PPP Protocol field of the packet which is being rejected.

The Rejected-Information field contains a copy of the packet which is being rejected. It begins with the Information field, and does not include any Data Link Layer headers nor an FCS. The Rejected-Information must be truncated to comply with the peer's established MRU.

2.4.4.5 Magic-Number

If the Code field indicates a Link Maintenance Debug packet (codes 9 through 11), then the Data field is shown as Magic-Number field and Data field.

The Magic-Number field is four octets, and aids in detecting links which are in the looped-back condition. Until the Magic-Number Configuration Option has been successfully negotiated, the Magic-Number must be transmitted as zero.

The Data field is zero or more octets, and contains uninterpreted data for use by the sender. The data may consist of any binary value.

2.5 The Option Negotiation Automaton

The finite-state automaton is defined by events, actions and state transitions. Events include reception of external commands such as Open and Close, expiration of the Restart timer, and reception of packets from a peer. Actions include the starting of the Restart timer and transmission of packets to the peer.

Some types of packets -- Configure-Naks and Configure-Rejects, or Code-Rejects and Protocol-Rejects, or Echo-Requests, Echo-Replies and Discard-Requests -- are not differentiated in the automaton descriptions. As will be described later, these packets do indeed serve different functions. However, they always cause the same transitions.

Abbreviation	Description
Down	lower layer is Down
Open	administrative Open
Close	administrative Close
TO+	Timeout with counter > 0
TO-	Timeout with counter expired
RCR+	Receive-Configure-Request (acceptable)
RCR-	Receive-Configure-Request (adaption needed)
RCR--	Receive-Configure-Request (counter expired)
RCA+	Receive-Configure-Ack with counter > 0
RCA-	Receive-Configure-Ack with counter expired
RCN	Receive-Configure-Nak/Rej
RTR	Receive-Terminate-Request
RTA	Receive-Terminate-Ack
RUC	Receive-Unknown-Code
RXJ+	Receive-Code-Reject or Receive-Protocol-Reject (permitted)
RXJ-	Receive-Code-Reject or Receive-Protocol-Reject (catastrophic)
RXR	Receive-Echo-Request or Receive-Echo-Reply or Receive-Discard-Request

Table 2-1: Events of The Option Negotiation Automaton

Abbreviation	Description
tlu	This-Layer-Up
tld	This-Layer-Down
tlf	This-Layer-Finished
irc	Initialize-Restart-Count
ifc	Initialize-Failure-Count
ilc	Initialize-Loop-Count
dlc	Decrement-Loop-Count
zrc	Zero-Restart-Count
scr	Send-Configure-Request
sca	Send-Configure-Ack
scn	Send-Configure-Nak/Rej
str	Send-Terminate-Request
sta	Send-Terminate-Ack
scj	Send-Code-Reject
ser	Send-Echo-Reply

Table 2-2: Actions of The Option Negotiation Automaton

2.5.1 State Transition Table

The complete state transition table follows. States are indicated horizontally, and events are read vertically. State transitions and actions are represented in the form action/new-state. Multiple actions are separated by commas, and may continue on succeeding lines as space requires; multiple actions may be implemented in any convenient order. The dash ('-') indicates an illegal transition.

	States					
Events	**0 Closed**	**1 Closing**	**2 Req-Sent**	**3 Ack-Rcvd**	**4 Ack-Sent**	**5 Opened**
Down	0	0	0	0	0	0
Open	irc, ifc, ilc, scr/2	irc, ifc, ilc, scr/2	irc, ifc, ilc, scr/2	irc, ifc, ilc, scr/2	irc, ifc, ilc, scr/2	irc, ifc, ilc, scr/2
Close	tlf/0	1	irc, str/1	irc, str/1	irc, str/1	irc,str/1
TO+	-	str/1	scr/2	scr/2	scr/4	-
TO-	-	tlf/0	tlf/0	tlf/0	tlf/0	-
RCR+	-	1	ifc, sca/4	ifc, sca, tlu/5	ifc, sca/4	tld, scr, sca/4
RCR-	-	1	scn/2	scn/3	scn/2	tld, scr, scn/2
RCR--	-	1	irc, str/1	irc, str/1	irc, str/1	tld, scr, scn/2
RCA+	-	1	irc, dlc/3	3	irc, dlc, tlu/5	5
RCA-	-	1	irc, str/1	3	irc, str/1	5
RCN	-	1	irc, scr/2	scr/2	irc, scr/4	tld, scr/2
RTR	-	sta/1	zrc, sta/1	zrc, sta/1	zrc, sta/1	tld, zrc, sta/1
RTA	-	tlf/0	2	3	4	tld, scr/2
RUC	-	scj/1	scj/2	scj/3	scj/4	scj/5
RXJ+	-	1	2	3	4	5
RXJ-	-	tlf/0	irc, str/1	irc, str/1	irc, str/1	tld, irc, str/1
RXR	-	1	2	3	4	ser/5

Table 2-3: State Transitions

The states in which the Restart timer is running are identifiable by the presence of TO events. Only the Send-Configure-Request, Send-Terminate-Request and Zero-Restart-Count actions start or re-start the Restart timer. The Restart timer is stopped when transitioning from any state where the timer is running to a state where the timer is not running.

2.5.2 States

Following is a more detailed description of each automaton state.

2.5.2.1 Closed

In the Closed state, the lower layer is unavailable (Down), and no Open has occurred. The Restart timer is not running in the Closed state.

When an administrative Open has been initiated, a Configure-Request is sent.

2.5.2.2 Closing

In the Closing state, an attempt is made to terminate the connection. A Terminate-Request has been sent and the Restart timer is running, but a Terminate-Ack has not yet been received.

Upon reception of a Terminate-Ack, the Closed state is entered. Upon the expiration of the Restart timer, a new Terminate-Request is transmitted, and the Restart timer is restarted. After the Restart timer has expired Max-Terminate times, the Closed state is entered.

2.5.2.3 Request-Sent

In the Request-Sent state an attempt is made to configure the connection. A Configure-Request has been sent and the Restart timer is running, but a Configure-Ack has not yet been received nor has one been sent.

2.5.2.4 Ack-Received

In the Ack-Received state, a Configure-Request has been sent and a Configure-Ack has been received. The Restart timer is still running, since a Configure-Ack has not yet been sent.

2.5.2.5 Ack-Sent

In the Ack-Sent state, a Configure-Request and a Configure-Ack have both been sent, but a Configure-Ack has not yet been received. The Restart timer is running, since a Configure-Ack has not yet been received.

2.5.2.6 Opened

In the Opened state, a Configure-Ack has been both sent and received. The Restart timer is not running.

When entering the Opened state, the implementation should signal the upper layers that it is now Up. Conversely, when leaving the Opened state, the implementation should signal the upper layers that it is now Down.

2.5.3 Events

Transitions and actions in the automaton are caused by events.

2.5.3.1 Down

This event occurs when a lower layer indicates that it is no longer ready to carry packets.

It is also used by LCP to signal the authentication protocol and the IPCP that it left Opened state. That is, the This-Layer-Down action from LCP triggers the Down event in the upper layers.

2.5.3.2 Open

This event occurs when a lower layer indicates that it is ready to carry packets and the network administrator has indicated that the link is allowed to be Opened. When this event occurs the automaton attempts to send configuration packets to the peer.

It is also used by LCP to signal the authentication protocol and the IPCP that it entered the Opened state. That is, the This-Layer-Up action from LCP triggers the Open event in the upper layers.

2.5.3.3 Close

This event indicates that the link is not available for traffic; that is, the network administrator has indicated that the link is not allowed to be Opened. When this event occurs, and the link is not in the Closed state, the automaton attempts to terminate the connection.

When authentication fails, the link should be terminated. This can be accomplished by simulating a Close event to the LCP.

2.5.3.4 Timeout (TO+,TO-)

This event indicates the expiration of the Restart timer. The Restart timer is used to time responses to Configure-Request and Terminate-Request packets.

The TO+ event indicates that the Restart counter continoues to be greater than zero, which triggers the corresponding Configure-Request or Terminate-Request packet to be retransmitted.

The TO- event indicates that the Restart counter is not greater than zero, and no more packets need to be retransmitted.

2.5.3.5 Receive-Configure-Request (RCR+,RCR-,RCR--)

This event occurs when a Configure-Request packet is received from the peer. The Configure-Request packet indicates the desire to open a connection and may specify Configuration Options.

The RCR+ event indicates that the Configure-Request was acceptable, and triggers the transmission of a corresponding Configure-Ack.

The RCR- event indicates that the Configure-Request was unacceptable and the Failure counter continoues to be greater than zero, which triggers the transmission of a corresponding Configure-Nak or Configure-Reject.

The RCR-- event indicates that the Configure-Request was unacceptable and the Failure counter is not greater than zero, which triggers the termination of the connection.

These events may occur on a connection which is already in the Opened state. The implementation must be prepared to immediately renegotiate the Configuration Options.

2.5.3.6 Receive-Configure-Ack (RCA+,RCA-)

This event occurs when a valid Configure-Ack packet is received from the peer. The Configure-Ack packet is a positive response to a Configure-Request packet. An out of sequence or otherwise invalid packet is silently discarded.

The RCA+ event indicates that the Loop counter continoues to be greater than zero.

The RCA- event indicates that the Loop counter is not greater than zero, which triggers the termination of the connection.

2.5.3.7 Receive-Configure-Nak/Rej (RCN)

This event occurs when a valid Configure-Nak or Configure-Reject packet is received from the peer. The Configure-Nak and Configure-Reject packets are negative responses to a Configure-Request packet. An out of sequence or otherwise invalid packet is silently discarded.

Although the Configure-Nak and Configure-Reject cause the same state transition in the automaton, these packets have significantly different effects on the Configuration Options sent in the resulting Configure-Request packet.

2.5.3.8 Receive-Terminate-Request (RTR)

This event occurs when a Terminate-Request packet is received. The Terminate-Request packet indicates the desire of the peer to close the connection.

2.5.3.9 Receive-Terminate-Ack (RTA)

This event occurs when a Terminate-Ack packet is received from the peer. The Terminate-Ack packet is usually a response to a Terminate-Request packet.

2.5.3.10 Receive-Unknown-Code (RUC)

This event occurs when an un-interpretable packet is received from the peer. A Code-Reject packet is sent in response.

2.5.3.11 Receive-Code-Reject, Receive-Protocol-Reject (RXJ+,RXJ-)

This event occurs when a Code-Reject or a Protocol-Reject packet is received from the peer.

The RXJ+ event arises when the rejected value is acceptable, such as a Code-Reject of an extended code, or a Protocol-Reject of a NCP. These are within the scope of normal operation. The implementation must stop sending the offending packet type.

The RXJ- event arises when the rejected value is catastrophic, such as a Code-Reject of Configure-Request, or a Protocol-Reject of LCP! This event communicates an unrecoverable error that terminates the connection.

2.5.3.12 Receive-Echo-Request, Receive-Echo-Reply, Receive-Discard-Request (RXR)

This event occurs when an Echo-Request, Echo-Reply or Discard-Request packet is received from the peer. The Echo-Reply packet is a response to an Echo-Request packet. There is no reply to an Echo-Reply or Discard-Request packet.

2.5.4 Actions

Actions in the automaton are caused by events and typically indicate the transmission of packets and/or the starting or stopping of the Restart timer.

2.5.4.1 Illegal-Event (-)

This indicates an event that cannot occur in a properly implemented automaton. No transition is taken, and the implementation should not reset or freeze.

2.5.4.2 This-Layer-Up (tlu)

This action indicates to the upper layers that the automaton is entering the Opened state.

This action is used by the LCP to signal the Open event to the IPCP or Authentication Protocol, or may be used by the IPCP to indicate that the link is available for its network layer traffic.

2.5.4.3 This-Layer-Down (tld)

This action indicates to the upper layers that the automaton is leaving the Opened state.

Typically, this action is used by the LCP to signal the Down event to the IPCP or Authentication Protocol, or may be used by the IPCP to indicate that the link is no longer available for its network layer traffic.

2.5.4.4 This-Layer-Finished (tlf)

This action indicates to the lower layers that the automaton is entering the Closed states, and the lower layer is no longer needed for the link.

This action is used by the IPCP to indicate to the LCP that the link should be terminate.

2.5.4.5 Initialize-Restart-Count (irc)

This action sets the Restart counter to the appropriate value (Max-Terminate or Max-Configure). The counter is decremented for each transmission, including the first.

2.5.4.6 Initialize-Failure-Count (ifc)

This action sets the Failure counter to the appropriate value (Max-Failure).

2.5.4.7 Initialize-Loop-Count (ilc)

This action sets the Loop counter to the appropriate value (Max-Configure + 1).

2.5.4.8 Decrement-Loop-Count (dlc)

This action decrements the Loop counter.

2.5.4.9 Zero-Restart-Count (zrc)

This action sets the Restart counter to zero.

In addition to zeroing the Restart counter, the implementation must set the timeout period to an appropriate value.

2.5.4.10 Send-Configure-Request (scr)

A Configure-Request packet is transmitted. This indicates the desire to open a connection with a specified set of Configuration Options. The Restart timer is started when the Configure-Request packet is transmitted, to guard against packet loss. The Restart counter is decremented each time a Configure-Request is sent.

2.5.4.11 Send-Configure-Ack (sca)

A Configure-Ack packet is transmitted. This acknowledges the reception of a Configure-Request packet with an acceptable set of Configuration Options.

2.5.4.12 Send-Configure-Nak (scn)

A Configure-Nak or Configure-Reject packet is transmitted, as appropriate. This negative response reports the reception of a Configure-Request packet with an unacceptable set of Configuration Options.

Configure-Nak packets are used to refuse a Configuration Option value, and to suggest a new, acceptable value. Configure-Reject packets are used to refuse all negotiation about a Configuration Option, typically because it is not recognized or implemented.

2.5.4.13 Send-Terminate-Request (str)

A Terminate-Request packet is transmitted. This indicates the desire to close a connection. The Restart timer is started when the Terminate-Request packet is transmitted, to guard against packet loss. The Restart counter is decremented each time a Terminate-Request is sent.

2.5.4.14 Send-Terminate-Ack (sta)

A Terminate-Ack packet is transmitted. This acknowledges the reception of a Terminate-Request packet or otherwise serves to synchronize the automatons.

2.5.4.15 Send-Code-Reject (scj)

A Code-Reject packet is transmitted. This indicates the reception of an unknown type of packet.

2.5.4.16 Send-Echo-Reply (ser)

An Echo-Reply packet is transmitted. This acknowledges the reception of an Echo-Request packet.

2.5.5 Counters and Timers

2.5.5.1 Restart Timer

There is one special timer used by the automaton. The Restart timer is used to time transmissions of Configure-Request and Terminate-Request packets. Expiration of the Restart timer causes a Timeout event, and retransmission of the corresponding Configure-Request or Terminate-Request packet. The Restart timer must be configurable, but should default to three (3) seconds.

2.5.5.2 Max-Terminate

There is one required restart counter for Terminate-Requests. Max-Terminate indicates the number of Terminate-Request packets sent without receiving a Terminate-Ack before assuming that the peer is unable to respond. Max-Terminate must be configurable, but should default to two (2) transmissions.

2.5.5.3 Max-Configure

A similar counter is recommended for Configure-Requests. Max-Configure indicates the number of Configure-Request packets sent without receiving a valid Configure-Ack, Configure-Nak or Configure-Reject before assuming that the peer is unable to respond. Max-Configure must be configurable, but should default to ten (10) transmissions.

2.5.5.4 Max-Failure

A related counter is recommended for Configure-Nak. Max-Failure indicates the number of Configure-Nak packets sent without sending a Configure-Ack before assuming that configuration is not converging. Any further Configure-Nak packets for peer requested options are converted to Configure-Reject packets, and locally desired options are no longer appended. Max-Failure must be configurable, but should default to five (5) transmissions.

2.6 LCP Configuration Options

LCP Configuration Options allow negotiation of modifications to the default characteristics of a point-to-point link. If a Configuration Option is not included in a Configure-Request packet, the default value for that Configuration Option is assumed.

None of the Configuration Options in this specification can be listed more than once.

The end of the list of Configuration Options is indicated by the Length field of the LCP packet.

Unless otherwise specified, all Configuration Options apply in a half-duplex fashion; typically, in the receive direction of the link from the point of view of the Configure-Request sender.

The options indicate additional capabilities or requirements of the implementation that is requesting the option.

A default is specified for each option which allows the link to correctly function without negotiation of the option, although perhaps with less than optimal performance.

Except where explicitly specified, acknowledgement of an option does not require the peer to take any additional action other than the default.

It is not necessary to send the default values for the options in a Configure-Request.

A summary of the Configuration Option format is shown below. The fields are transmitted from left to right.

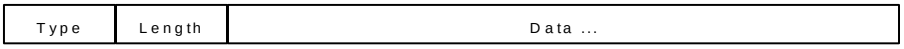


Figure 2-5: LCP Configuration Option Format

2.6.1 Type field

The Type field is one octet, and indicates the type of Configuration Option.

This implementation supports the following values:

1 Maximum-Receive-Unit

2 Async-Control-Character-Map

3 Authentication-Protocol

7 Protocol-Field-Compression

8 Address-and-Control-Field-Compression

2.6.1.1 Maximum-Receive-Unit (MRU)

This Configuration Option may be sent to inform the peer that the implementation can receive larger packets, or to request that the peer send smaller packets.

The default value is 1500 octets. If smaller packets are requested, an implementation must still be able to receive the full 1500 octet information field in case link synchronization is lost.

2.6.1.2 Async-Control-Character-Map (ACCM)

This Configuration Option provides a method to negotiate the use of control character transparency on asynchronous links.

The Configuration Option is used to inform the peer which control characters must remain mapped when the peer sends them. The default ACCM is 0xffffffff, plus the Control Escape and Flag Sequence characters themselves.

The ACCM field (Data field) is four octets, and indicates the set of control characters to be mapped. The map is sent most significant octet first. Each numbered bit corresponds to the octet of the same value. If the bit is cleared to zero, then that octet need not be mapped. If the bit is set to one, then that octet must remain mapped. For example, if bit 19 is set to zero, then the ASCII control character 19 (DC3, Control-S) may be sent in the clear. The least significant bit of the least significant octet (the final octet transmitted) is numbered bit 0, and would map to the ASCII control character NUL.

2.6.1.3 Authentication-Protocol

This Configuration Option provides a method to negotiate the use of a specific protocol for authentication. By default, authentication is not required.

An implementation must not include multiple Authentication-Protocol Configuration Options in its Configure-Request packets. Instead, it should attempt to configure the most desirable protocol first. If that protocol is Configure-Nak'd, then the implementation should attempt the next most desirable protocol in the next Configure-Request.

The implementation supports the following authentication protocols:

c023 Password Authentication Protocol

c223 Challenge Handshake Authentication Protocol

2.6.1.4 Protocol-Field-Compression (PFC)

This Configuration Option provides a method to negotiate the compression of the PPP Protocol field. By default, all implementations must transmit packets with two octet PPP Protocol fields. This Configuration Option is sent to inform the peer that the implementation can receive single octet Protocol fields.

If the most significant octet equals to zero and Protocol-Field-Compression is negotiated, then the most significant octet can be omitted.

The Protocol field is never compressed when sending any LCP packet.

2.6.1.5 Address-and-Control-Field-Compression (ACFC)

This Configuration Option provides a method to negotiate the compression of the Data Link Layer Address and Control fields. By default, all implementations must transmit frames with Address and Control fields appropriate to the link framing. This Configuration Option is sent to inform the peer that the implementation can receive compressed Address and Control fields.

If Address-and-Control-Field-Compression has been negotiated, the Address and Control fields are simply omitted.

The Address and Control fields must not be compressed when sending any LCP packet.

2.6.2 Length field

The Length field is one octet, and indicates the length of this Configuration Option including the Type, Length and Data fields.

If a negotiable Configuration Option is received in a Configure-Request, but with an invalid or unrecognized Length, a Configure-Nak should be transmitted which includes the desired Configuration Option with an appropriate Length and Data.

2.6.3 Data field

The Data field is zero or more octets, and contains information specific to the Configuration Option. The format and length of the Data field is determined by the Type and Length fields.

When the Data field is indicated by the Length to extend beyond the end of the Information field, the entire packet is silently discarded without affecting the automaton.

2.7 IPCP Configuration Options

The IPCP Configuration Option format is the same as the LCP Configuration Option format, but with a distinct set of Configuration Options.

This implementation supports the following values of the Type field:

2	IP-Compression-Protocol
3	IP-Address
129	Primary DNS Server Address
131	Secondary DNS Server Address

2.7.1 IP-Compression-Protocol

This Configuration Option provides a way to negotiate the use of a specific compression protocol. By default, compression is not enabled.

The implementation supports following value:

002d	Van Jacobson Compressed TCP/IP

2.7.2 IP-Address

This Configuration Option provides a way to negotiate the IP address to be used on the local end of the link. It allows the sender of the Configure-Request to state which IP-address is desired, or to request that the peer provide the information. The peer can provide this information by NAKing the option, and returning a valid IP-address.

By default, no IP address is assigned.

2.7.3 Primary DNS Server Address

This Configuration Option defines a method for negotiating with the remote peer the address of the primary DNS server to be used on the local end of the link. If local peer requests an invalid server address (which it will typically do intentionally) the remote peer specifies the address by NAKing this option, and returning the IP address of a valid DNS server.

By default, no primary DNS address is provided.

2.7.4 Secondary DNS Server Address

This Configuration Option defines a method for negotiating with the remote peer the address of the secondary DNS server to be used on the local end of the link. If local peer requests an invalid server address (which it will typically do intentionally) the remote peer specifies the address by NAKing this option, and returning the IP address of a valid DNS server.

By default, no secondary DNS address is provided.

3 Protocol

3.1 Flow Control

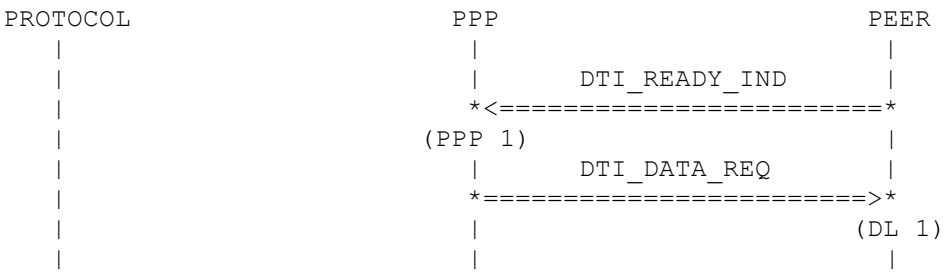
Every data transmission through PPP is controlled by flow control, so that it can be ensured that the layers are able to process the data and no buffer overflow occurs.

Before any data primitive is allowed to send, the corresponding layer has to indicate that it is ready to receive data by sending a flow control primitive. After sending a data primitive, the corresponding layer has to indicate again that it is ready to receive data befor send a new data primitive.

PPP has a special interface to transmit data to the upper and lower layer. This interface is called Data Transmission Interface (DTI). It is only used for data transmission and flow control. Any other PPP control primitives are described in the PPP interface.

The flow control primitives are only shown in this chapter. All data primitives in further chapters are shown without there corresponding flow control primitives. It is assumed that the corresponding flow control primitive was sent before sending such data primitive.

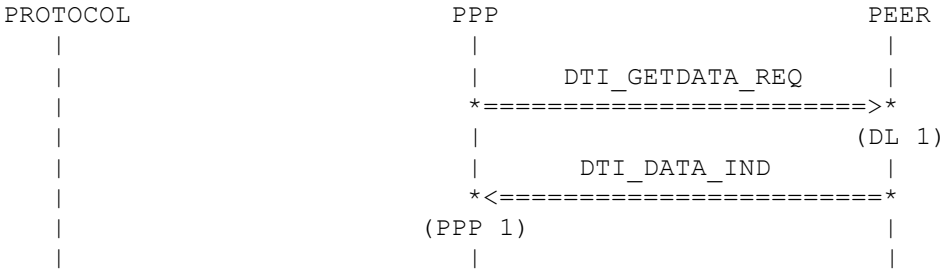
3.1.1 Downlink data transfer to PPP peer layer



(PPP 1)
PEER layer indicates to PPP that it is ready to receive a data primitive. (flow control primitive)

(DL 1)
If there is a frame to send to PEER layer then the data are delivered to the lower layer. (data primitive)

3.1.2 Uplink data transfer from PPP peer layer

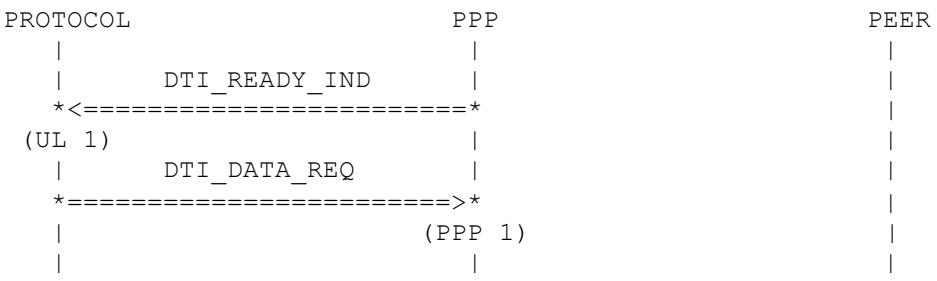


(DL 1)
PPP indicates to PEER layer that it is ready to receive a data primitive. (flow control primitive)

(PPP 1)
If there is a frame or a part of frame to send to PPP then the data are delivered to the upper layer. (data primitive)

A frame can be delivered from PEER layer to PPP in more than one data primitive. A corresponding flow control primitive must be sent for every data primitive. If a frame will be delivered by more than on data primitive then it will shown as just one data primitive in further chapters.

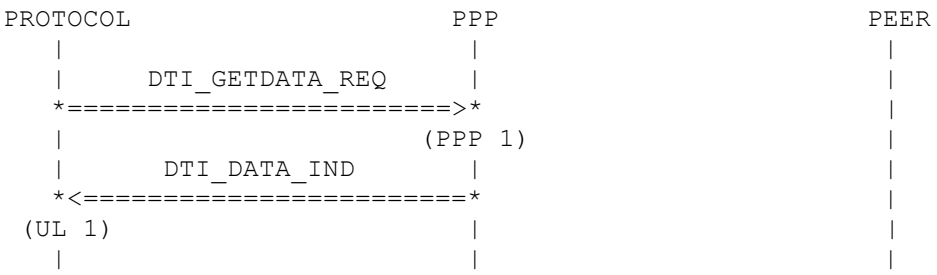
3.1.3 Downlink data transfer from PROTOCOL layer



(UL 1)
PPP indicates to PROTOCOL layer that it is ready to receive a packet. (flow control primitive)

(PPP 1)
If there is a packet to send to PPP then the data are delivered to the lower layer. (data primitive)

3.1.4 Uplink data transfer to PROTOCOL layer



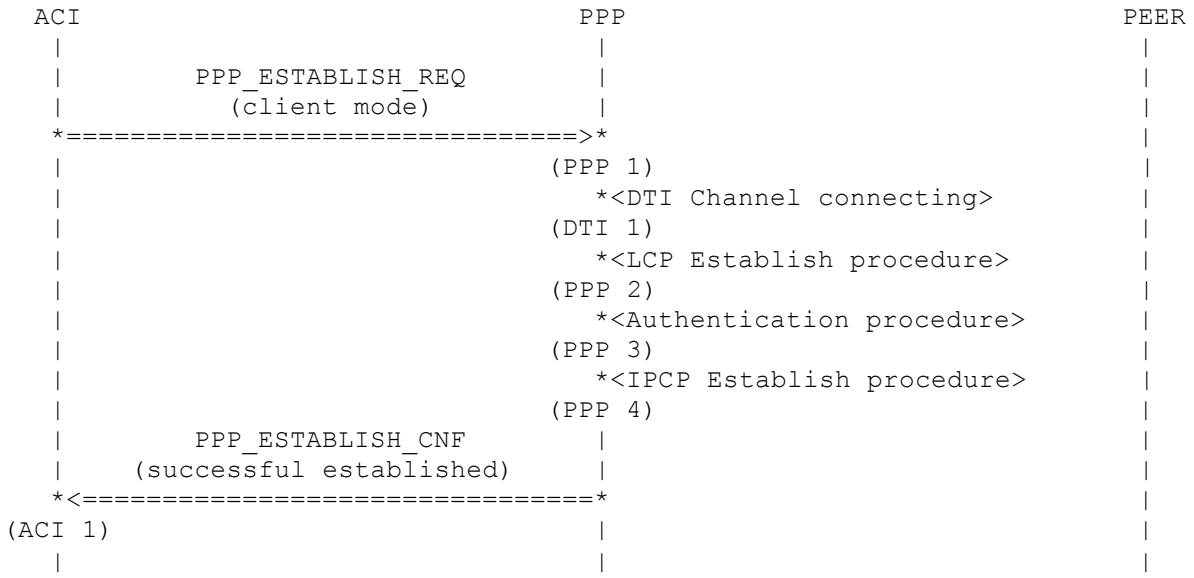
(PPP 1)
PROTOCOL layer indicates to PPP that it is ready to receive a packet. (flow control primitive)

(UL 1)
If there is a packet to send to PROTOCOL layer then the data are delivered to the upper layer. (data primitive)

3.2 Link Establishment

3.2.1 Link establishment in client mode

3.2.1.1 Usual establishment



(PPP 1)
ACI indicates to PPP that PPP has to start Client mode.

(DTI 1)
PPP connects the DTI channels to the neighbor entities. Please look to chapter 3.10.

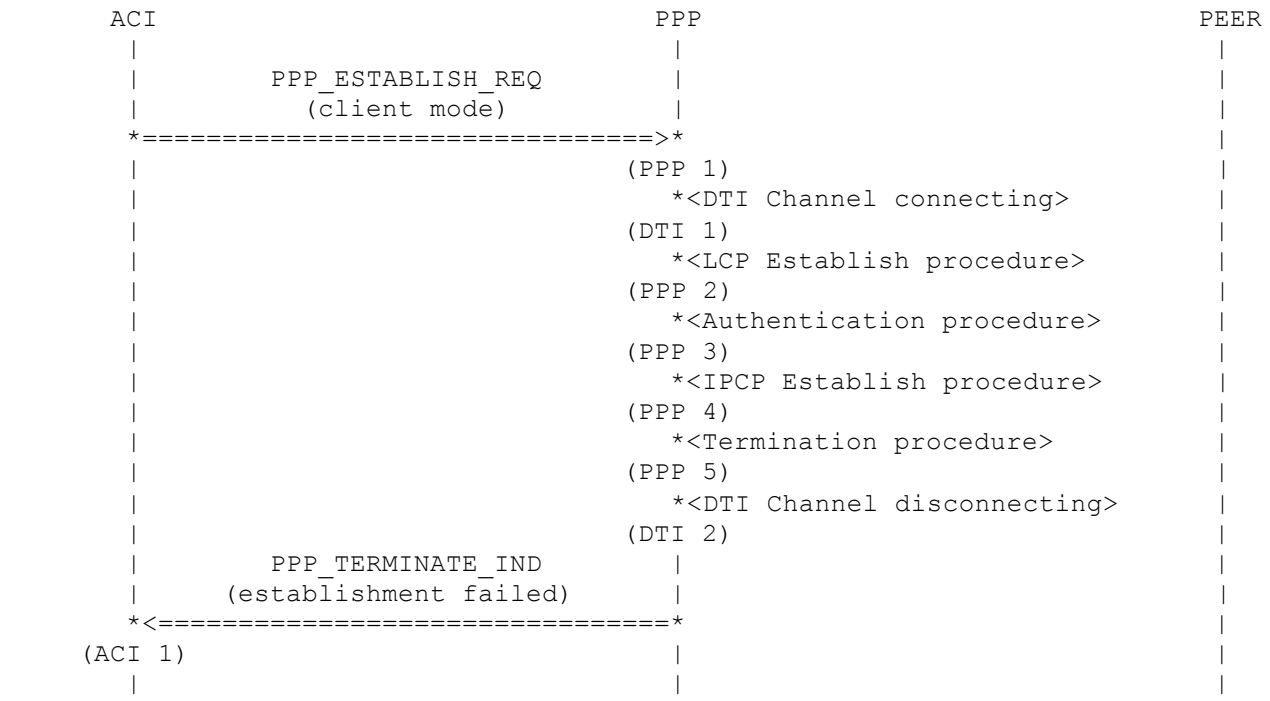
(PPP 2)
PPP proceeds to the Link Establishment phase (see 2.1.2). Within the Link Establishment phase the LCP Establish procedure (see 3.6) is running.

(PPP 3)
The Authentication phase/procedure (see 2.1.3 and 3.7) is optional and will only run if the use of an authentication protocol is negotiated during Link Establishment phase.

(PPP 4)
To finish the establishment of the PPP link the IPCP Establish procedure (see 3.6) is running.

(ACI 1)
After successful establishment PPP indicates to ACI the results of the negotiation.

3.2.1.2 IPCP Establish procedure failed



(PPP 1)
ACI indicates to PPP that PPP has to start Client mode.

(DTI 1)
PPP connects the DTI channels to the neighbor entities. Please look to chapter 3.10.

(PPP 2)
PPP proceeds to the Link Establishment phase (see 2.1.2). Within the Link Establishment phase the LCP Establish procedure (see 3.6) is running.

(PPP 3)
The Authentication phase/procedure (see 2.1.3 and 3.7) is optional and will only run if the use of an authentication protocol is negotiated during Link Establishment phase.

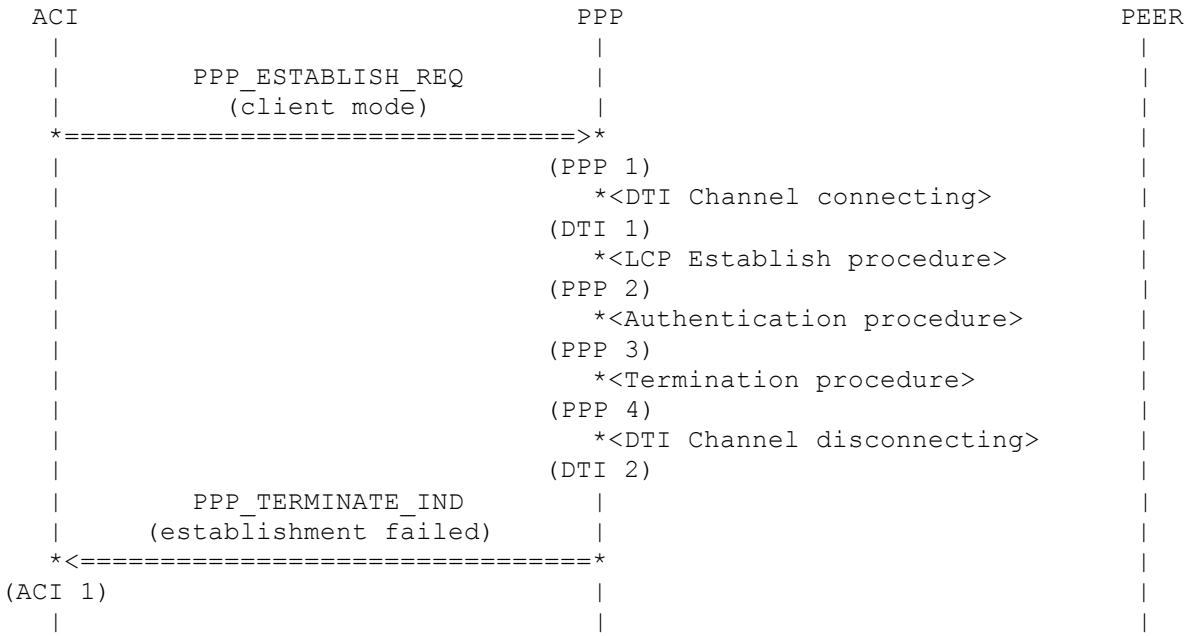
(PPP 4)
IPCP tries to negotiate the rest of parameters. If the negotiation fails PPP has to terminate the PPP link. In this case the Network-Layer Protocol phase (see 2.1.4) is finished.

(PPP 5)
If the Network-Layer Protocol phase (see 2.1.4) is finished, then the Termination procedure (see 3.9) within the Link Termination phase (see 2.1.5) will terminate the PPP link.

(DTI 2)
PPP disconnects to the neighbor entities (see 3.10.2).

(ACI 1)
PPP informs ACI that the establishment failed and the PPP link is terminated now.

3.2.1.3 Authentication procedure failed



(PPP 1)
ACI indicates to PPP that PPP has to start Client mode.

(DTI 1)
PPP connects the DTI channels to the neighbor entities. Please look to chapter 3.10.

(PPP 2)
PPP proceeds to the Link Establishment phase (see 2.1.2). Within the Link Establishment phase the LCP Establish procedure (see 3.6) is running.

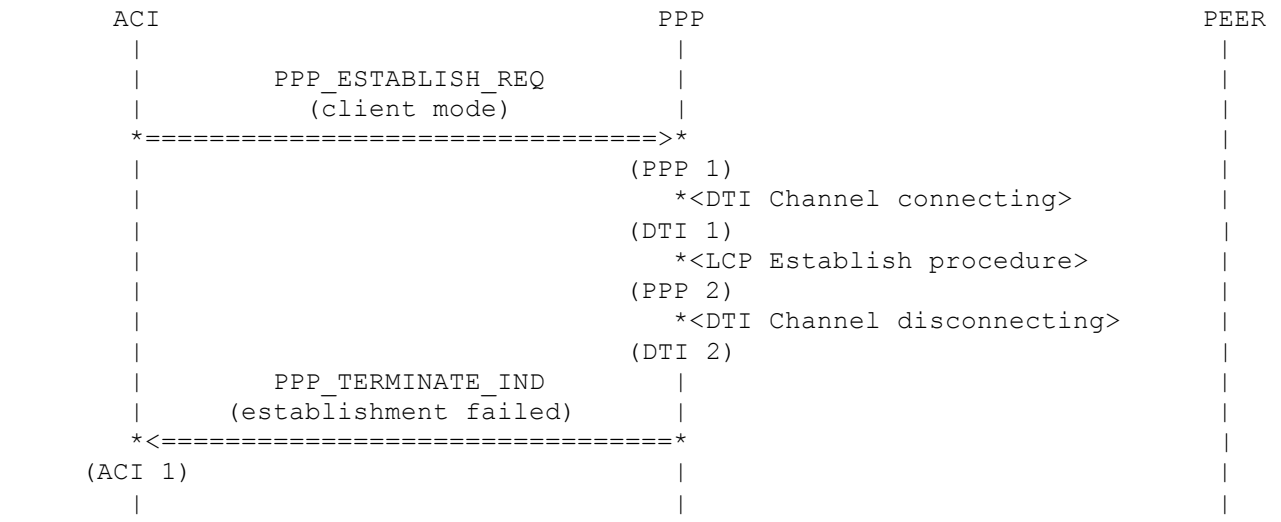
(PPP 3)
The Authentication phase/procedure (see 2.1.3 and 3.7) is optional and will only run if the use of an authentication protocol is negotiated during Link Establishment phase. If authentication fails the implementation must proceed to the Link Termination phase (see 2.1.5).

(PPP 4)
The Termination procedure (see 3.9) within the Link Termination phase (see 2.1.5) will terminate the PPP link.

(DTI 2)
PPP disconnects to the neighbor entities (see 3.10.2).

(ACI 1)
PPP informs ACI that the establishment failed and the PPP link is terminated now.

3.2.1.4 LCP Establish procedure failed



(PPP 1)
ACI indicates to PPP that PPP has to start Client mode.

(DTI 1)
PPP connects the DTI channels to the neighbor entities. Please look to chapter 3.10.

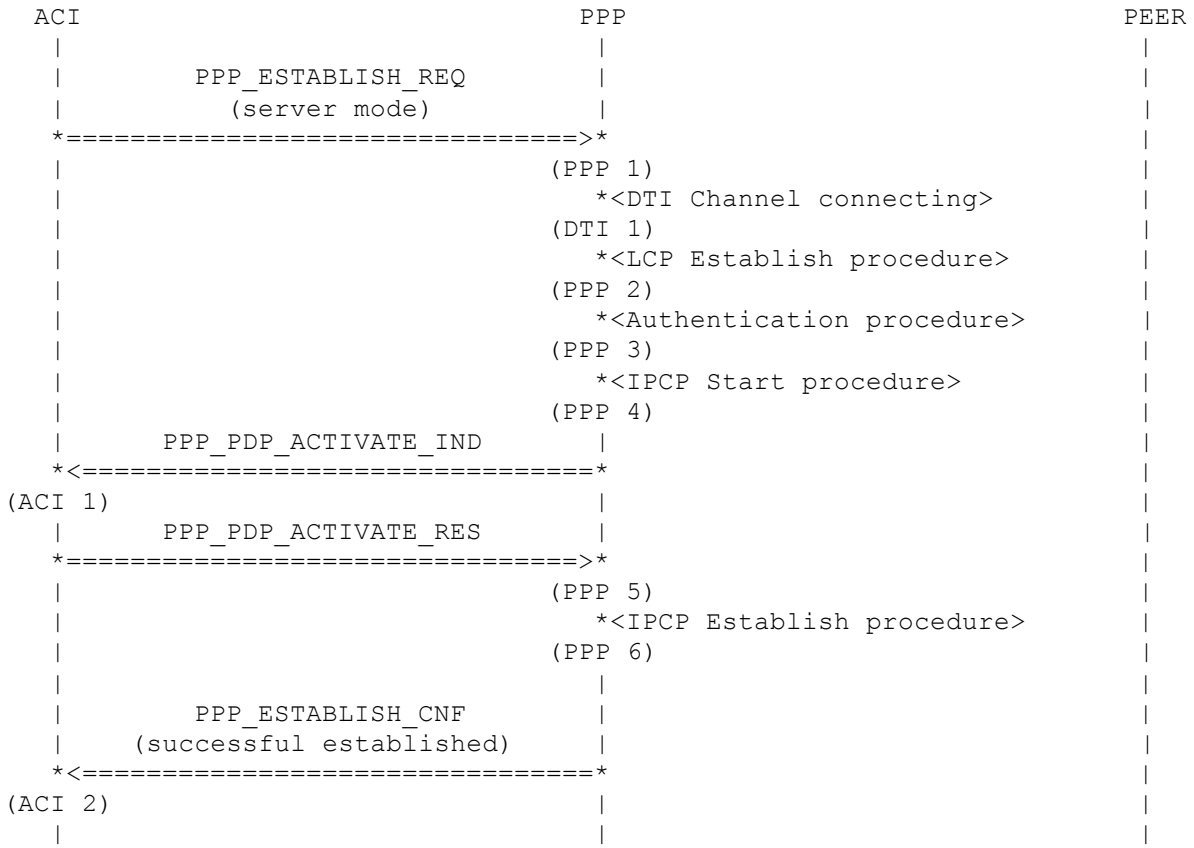
(PPP 2)
PPP proceeds to the Link Establishment phase (see 2.1.2). Within the Link Establishment phase the LCP Establish procedure (see 3.6) is running. If the LCP Establish procedure fails (this might happen because of the loss of carrier) the implementation must proceed to the Link Dead phase (see 2.1.1).

(DTI 2)
PPP disconnects to the neighbor entities (see 3.10.2).

(ACI 1)
PPP informs ACI that the establishment failed and the PPP link is terminated now.

3.2.2 Link establishment in server mode

3.2.2.1 Usual establishment



(PPP 1)
ACI indicates to PPP that PPP has to start Server mode.

(DTI 1)
PPP connects the DTI channels to the neighbor entities. Please look to chapter 3.10.

(PPP 2)
PPP proceeds to the Link Establishment phase (see 2.1.2). Within the Link Establishment phase the LCP Establish procedure (see 3.6) is running.

(PPP 3)
The Authentication phase/procedure (see 2.1.3 and 3.7) is optional and will only run if the use of an authentication protocol is negotiated during Link Establishment phase.

(PPP 4)
After authentication the implementation waits for a valid IPCP Configure-Request to fill in protocol configuration options needed for the PPP_PDP_ACTIVATE_IND primitive (see 3.8). This is also the start of the Network-Layer Protocol phase (see 2.1.4).

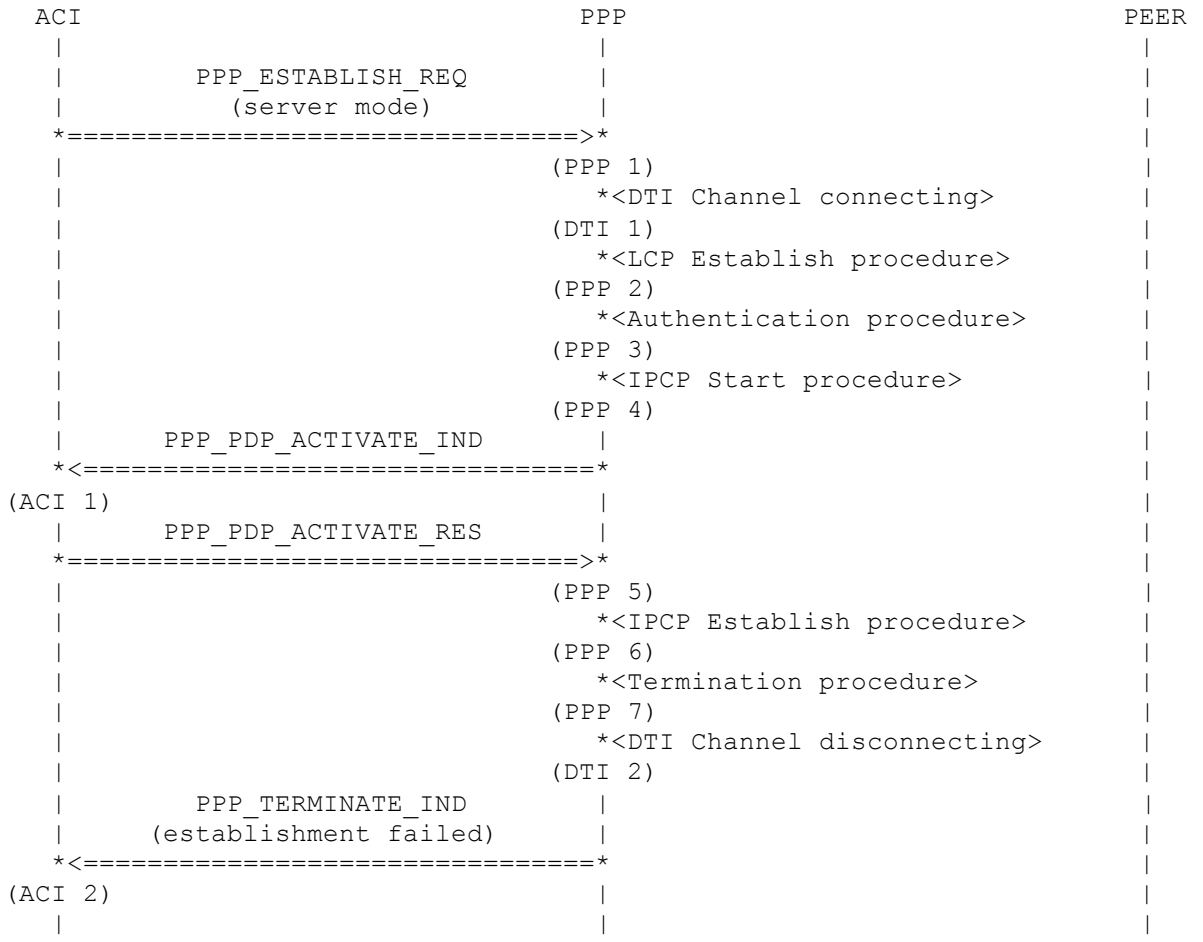
(ACI 1)
PPP sends a PPP_PDP_ACTIVATE_IND primitive to activate the PDP context.

(PPP 5)
ACI deliver all necessary information to finish the establishment of the PPP link.

(PPP 6)
To finish the establishment of the PPP link the IPCP Establish procedure (see 3.6) is running.

(ACI 2)
After successful establishment PPP indicates to ACI the results of the negotiation.

3.2.2.2 IPCP Establish procedure failed



(PPP 1)
ACI indicates to PPP that PPP has to start Server mode.

(DTI 1)
PPP connects the DTI channels to the neighbor entities. Please look to chapter 3.10.

(PPP 2)
PPP proceeds to the Link Establishment phase (see 2.1.2). Within the Link Establishment phase the LCP Establish procedure (see 3.6) is running.

(PPP 3)
The Authentication phase/procedure (see 2.1.3 and 3.7) is optional and will only run if the use of an authentication protocol is negotiated during Link Establishment phase.

(PPP 4)
After authentication the implementation waits for a valid IPCP Configure-Request to fill in protocol configuration options needed for the PPP_PDP_ACTIVATE_IND primitive (see 3.8). This is also the start of the Network-Layer Protocol phase (see 2.1.4).

(ACI 1)
PPP sends a PPP_PDP_ACTIVATE_IND primitive to activate the PDP context.

(PPP 5)
ACI deliver all necessary information to finish the establishment of the PPP link.

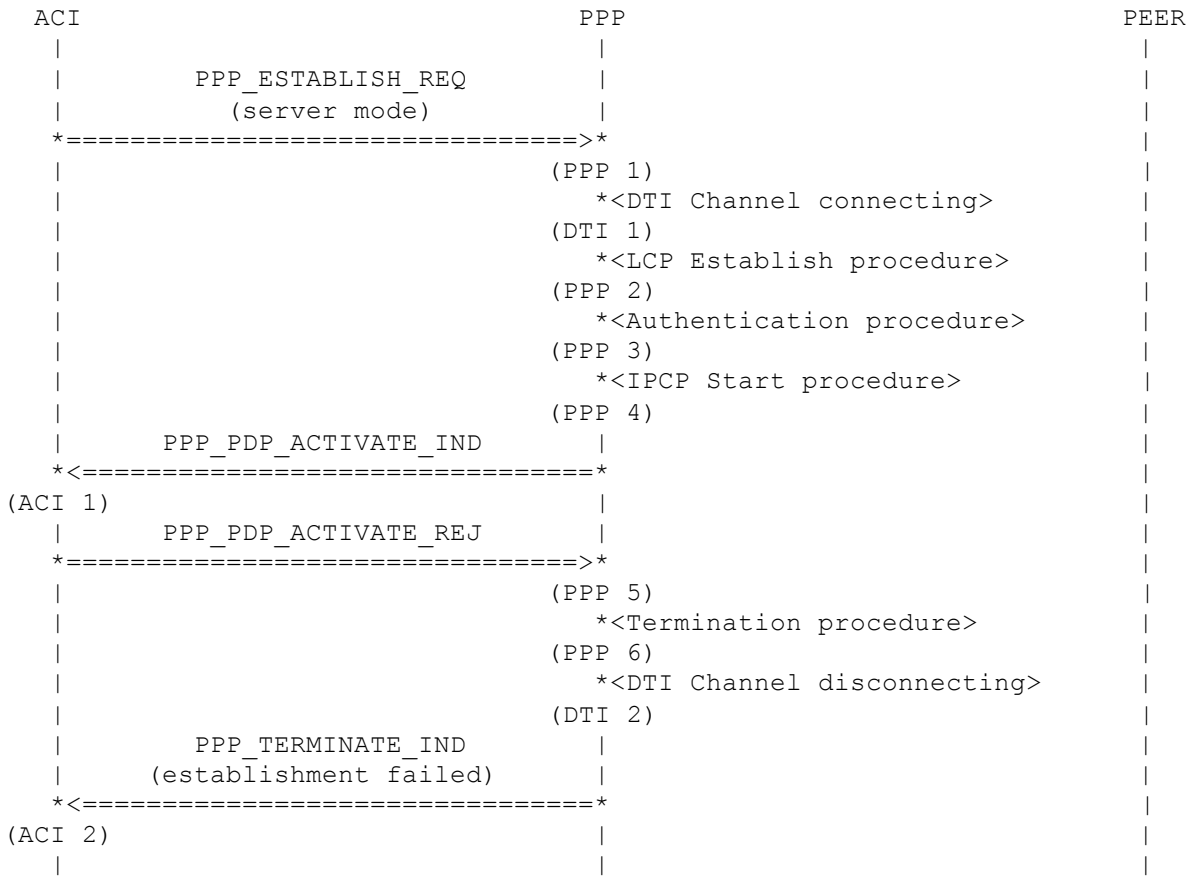
(PPP 6)
IPCP tries to negotiate the rest of parameters. If the negotiation fails PPP has to terminate the PPP link. In this case the Network-Layer Protocol phase (see 2.1.4) is finished.

(PPP 7)
If the Network-Layer Protocol phase (see 2.1.4) is finished, then the Termination procedure (see 3.9) within the Link Termination phase (see 2.1.5) will terminate the PPP link.

(DTI 2)
PPP disconnects to the neighbor entities (see 3.10.2).

(ACI 2)
PPP informs ACI that the establishment failed and the PPP link is terminated now.

3.2.2.3 Context activation failed



(PPP 1)
ACI indicates to PPP that PPP has to start Server mode.

(DTI 1)
PPP connects the DTI channels to the neighbor entities. Please look to chapter 3.10.

(PPP 2)
PPP proceeds to the Link Establishment phase (see 2.1.2). Within the Link Establishment phase the LCP Establish procedure (see 3.6) is running.

(PPP 3)
The Authentication phase/procedure (see 2.1.3 and 3.7) is optional and will only run if the use of an authentication protocol is negotiated during Link Establishment phase.

(PPP 4)
After authentication the implementation waits for a valid IPCP Configure-Request to fill in protocol configuration options needed for the PPP_PDP_ACTIVATE_IND primitive (see 3.8). This is also the start of the Network-Layer Protocol phase (see 2.1.4).

(ACI 1)
PPP sends a PPP_PDP_ACTIVATE_IND primitive to activate the PDP context.

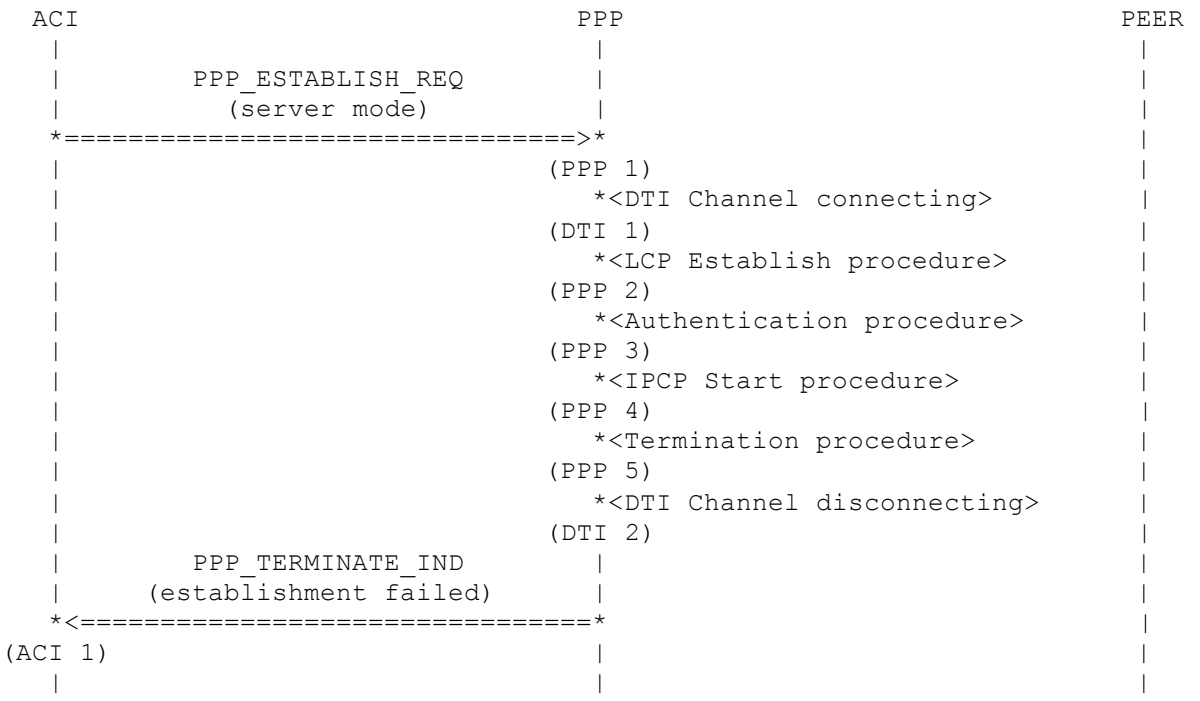
(PPP 5)
If context activation failed ACI indicates that by sending the reject primitive.

(PPP 6)
In case of context activation failure the Network-Layer Protocol phase (see 2.1.4) is finished and the Termination procedure (see 3.9) within the Link Termination phase (see 2.1.5) will terminate the PPP link.

(DTI 2)
PPP disconnects to the neighbor entities (see 3.10.2).

(ACI 2)
PPP informs ACI that the establishment failed and the PPP link is terminated now.

3.2.2.4 IPCP Start procedure failed



(PPP 1)
ACI indicates to PPP that PPP has to start Server mode.

(DTI 1)
PPP connects the DTI channels to the neighbor entities. Please look to chapter 3.10.

(PPP 2)
PPP proceeds to the Link Establishment phase (see 2.1.2). Within the Link Establishment phase the LCP Establish procedure (see 3.6) is running.

(PPP 3)
The Authentication phase/procedure (see 2.1.3 and 3.7) is optional and will only run if the use of an authentication protocol is negotiated during Link Establishment phase.

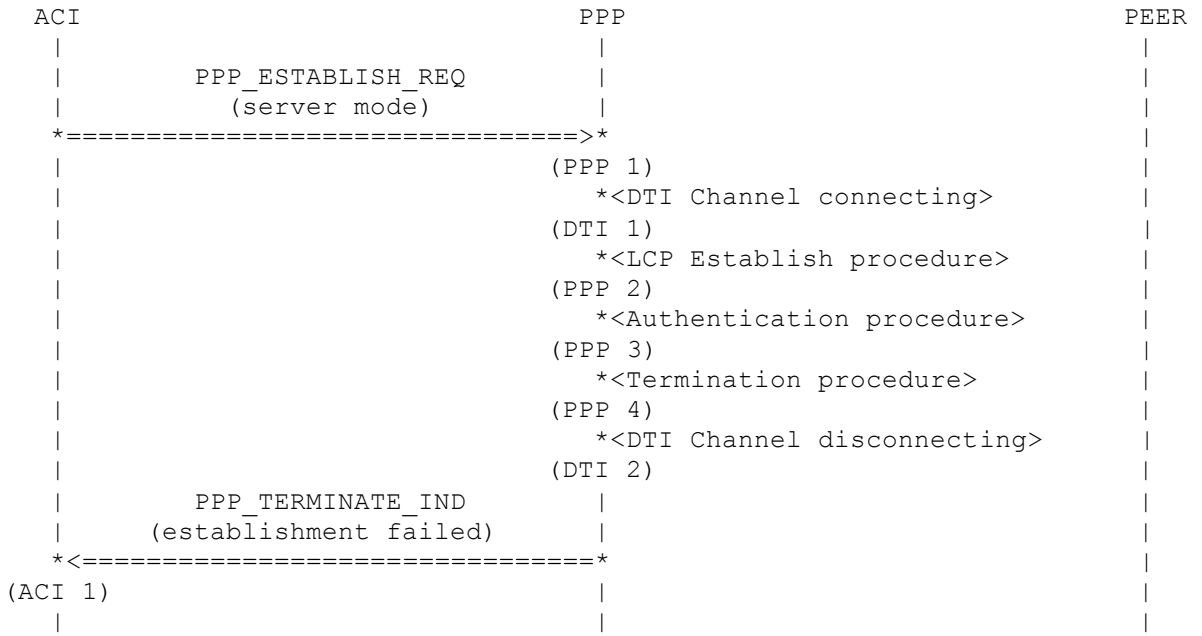
(PPP 4)
After authentication the implementation waits for a valid IPCP Configure-Request to fill in protocol configuration options needed for the PPP_PDP_ACTIVATE_IND primitive (see 3.8).

(PPP 5)
If no valid IPCP Configure-Request packet is received, then the Termination procedure (see 3.9) within the Link Termination phase (see 2.1.5) will terminate the PPP link.

(DTI 2)
PPP disconnects to the neighbor entities (see 3.10.2).

(ACI 1)
PPP informs ACI that the establishment failed and the PPP link is terminated now.

3.2.2.5 Authentication procedure failed



(PPP 1)
ACI indicates to PPP that PPP has to start Server mode.

(DTI 1)
PPP connects the DTI channels to the neighbor entities. Please look to chapter 3.10.

(PPP 2)
PPP proceeds to the Link Establishment phase (see 2.1.2). Within the Link Establishment phase the LCP Establish procedure (see 3.6) is running.

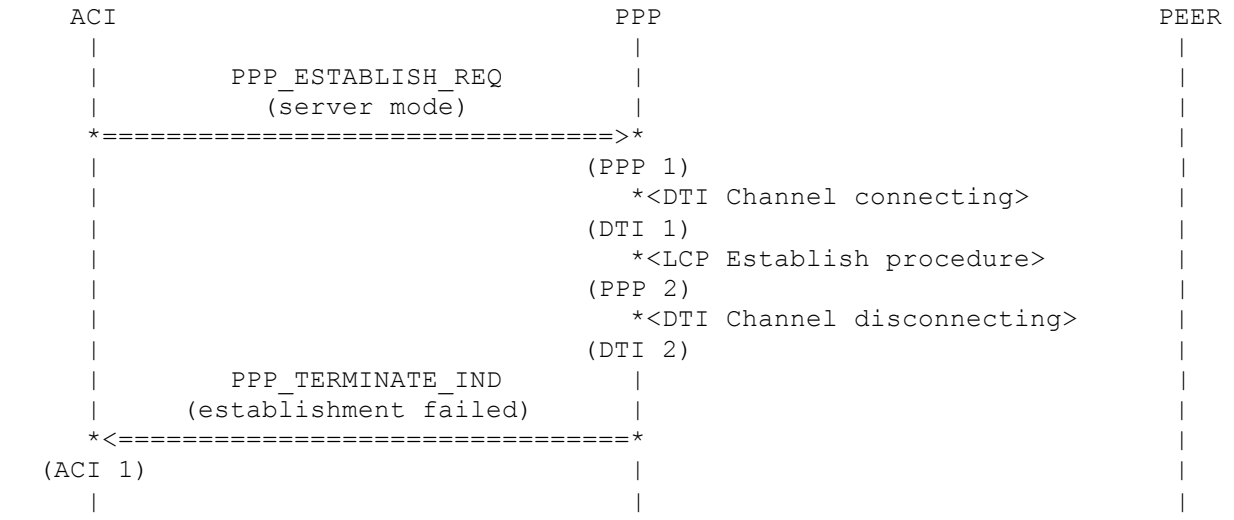
(PPP 3)
The Authentication phase/procedure (see 2.1.3 and 3.7) is optional and will only run if the use of an authentication protocol is negotiated during Link Establishment phase. If authentication fails the implementation must proceed to the Link Termination phase (see 2.1.5).

(PPP 4)
The Termination procedure (see 3.9) within the Link Termination phase (see 2.1.5) will terminate the PPP link.

(DTI 2)
PPP disconnects to the neighbor entities (see 3.10.2).

(ACI 1)
PPP informs ACI that the establishment failed and the PPP link is terminated now.

3.2.2.6 LCP Establish procedure failed



(PPP 1)
ACI indicates to PPP that PPP has to start Server mode.

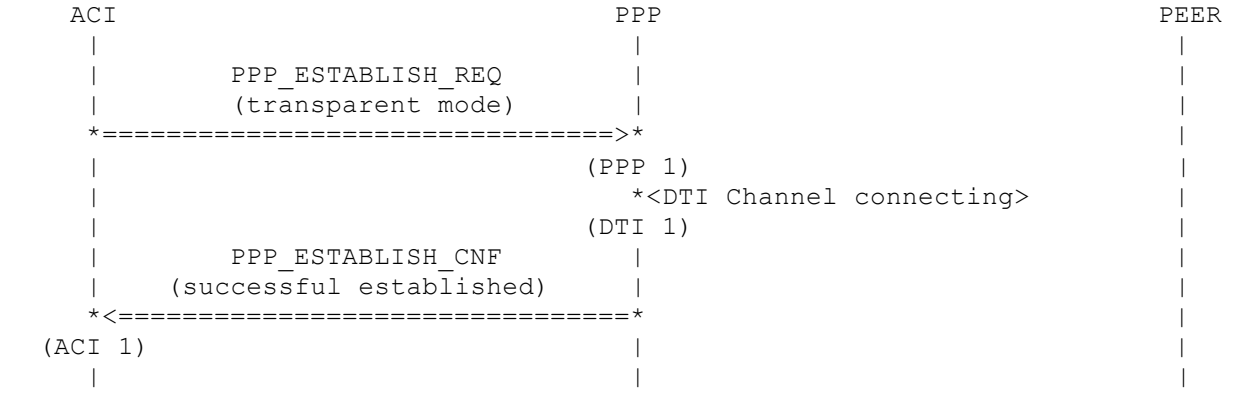
(DTI 1)
PPP connects the DTI channels to the neighbor entities. Please look to chapter 3.10.

(PPP 2)
PPP proceeds to the Link Establishment phase (see 2.1.2). Within the Link Establishment phase the LCP Establish procedure (see 3.6) is running. If the LCP Establish procedure fails (this might happen because of the loss of carrier) the implementation must proceed to the Link Dead phase (see 2.1.1).

(DTI 2)
PPP disconnects to the neighbor entities (see 3.10.2).

(ACI 1)
PPP informs ACI that the establishment failed and the PPP link is terminated now.

3.2.3 Link establishment in transparent mode



(PPP 1)
ACI indicates to PPP that PPP have to start Transparent mode.

(DTI 1)
PPP connects the DTILIB channels to the neighbor entities. Please look to chapter 3.10.

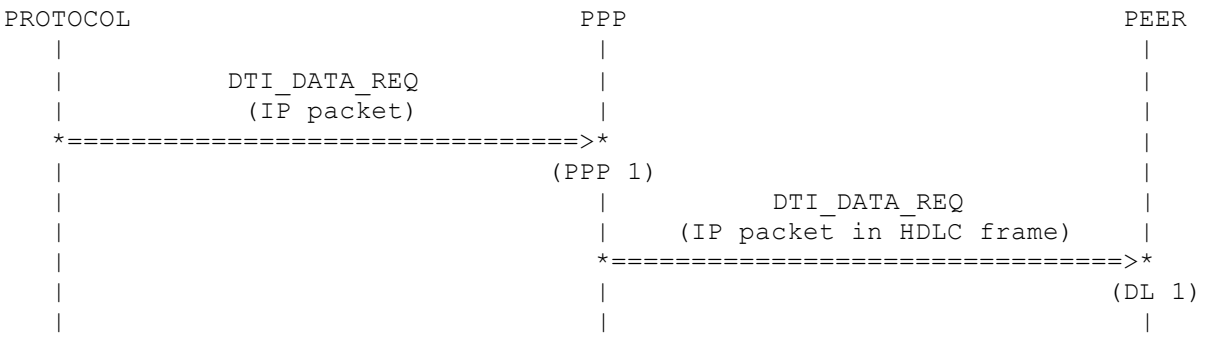
(ACI 1)
PPP indicates to ACI the results of the negotiation.

3.3 Packet transfer after establishment

3.3.1 IP packet transfer in client and server mode

Before any IP packets may be communicated, the PPP link must be established. Any IP packets received when the PPP link is not estabslished will silently discarded.

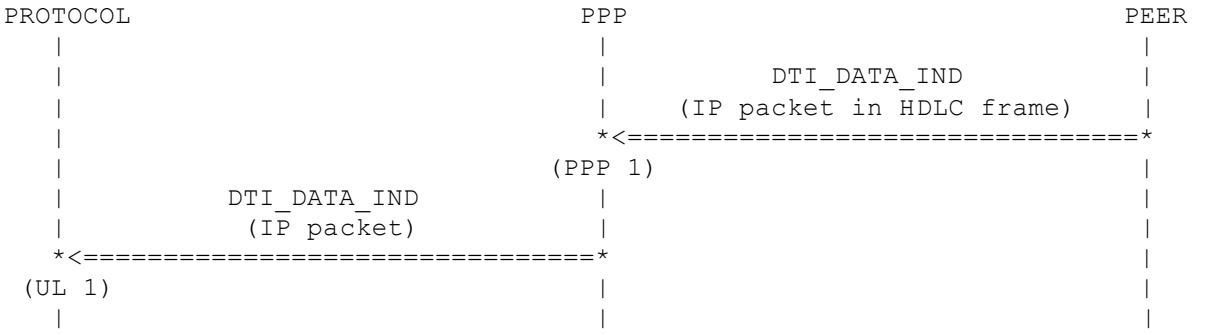
3.3.1.1 Downlink transfer



(PPP 1)
PPP gets an IP packet form PROTOCOL layer to send to PEER layer.

(DL 1)
PPP puts the IP packet into a HDLC frame and passes it on to PEER layer.

3.3.1.2 Uplink transfer



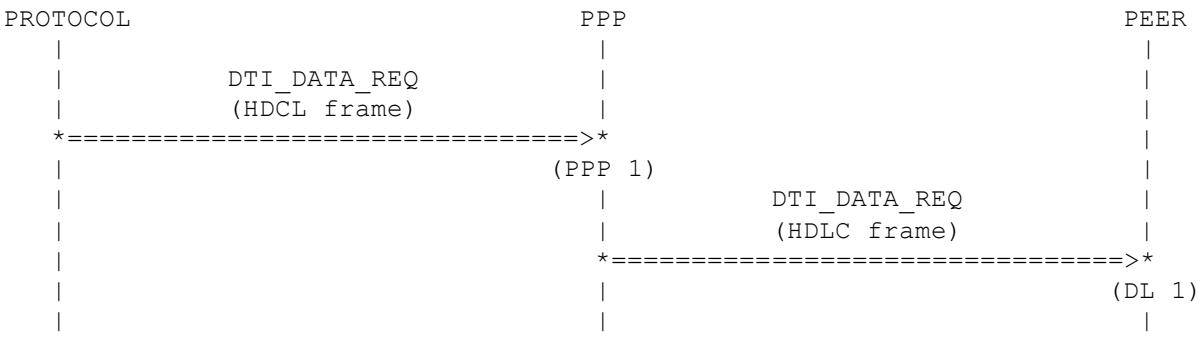
(PPP 1)
PPP gets the IP packet in a HDLC frame to send to PROTOCOL layer.

(UL 1)
PPP decapsulates the IP packet and passes it on to PROTOCOL layer.

3.3.2 HDCL frame transfer in transparent mode

In Transparent mode PPP just detect begin and end of HDLC frames and delivers complete HDLC-frames to PROTOCOL layer. This mode is used to send PPP frames over the air.

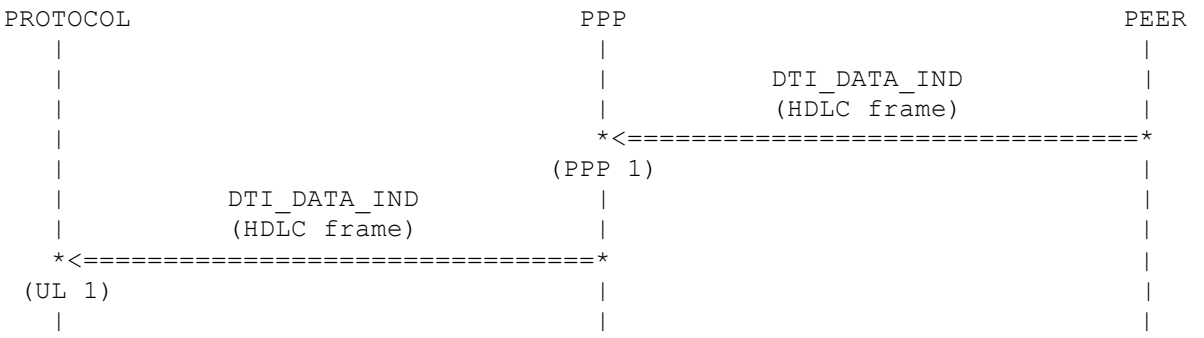
3.3.2.1 Downlink transfer



(PPP 1)
PPP gets a HDLC frame form PROTOCOL layer to send to PEER layer.

(DL 1)
PPP passes it on to PEER layer.

3.3.2.2 Uplink transfer



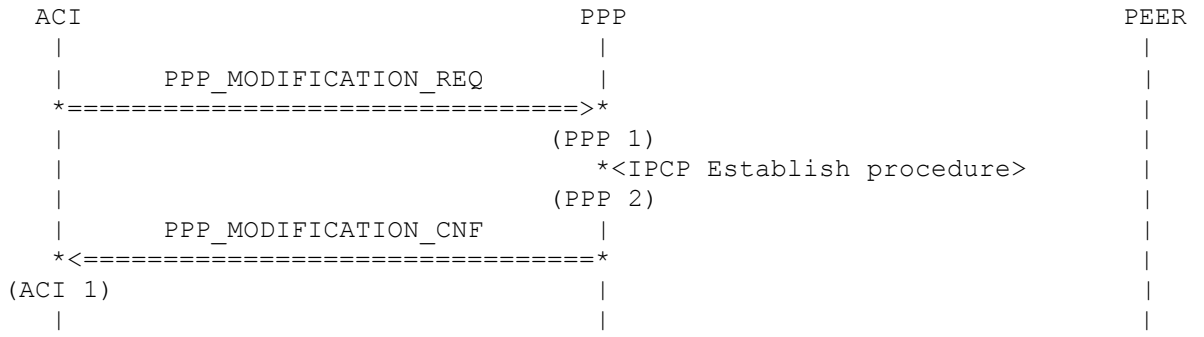
(PPP 1)
PPP gets a HDLC frame to send to PROTOCOL layer.

(UL 1)
PPP passes it on to PROTOCOL layer.

3.4 Link Modification in server mode

The PPP link may be modified at any time after Link Establishment.

3.4.1 Usual Modification

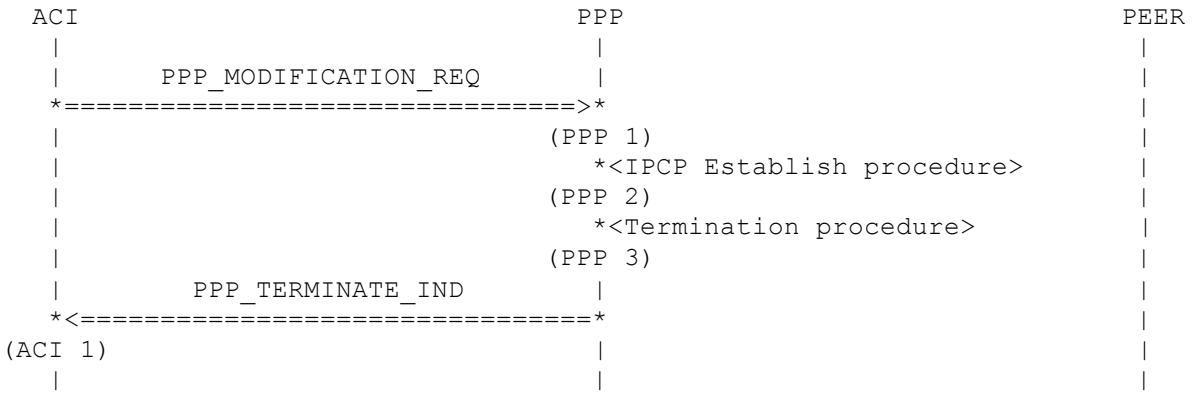


(PPP 1)
ACI requests a modification of the PPP link.

(PPP 2)
PPP tries to modify the link parameters by using the IPCP Establish procedure (see 3.6). If it is impossible to modify the parameters, then the old parameters will be negotiated.

(ACI 1)
The result of the negotiation of the link parameter is given to ACI.

3.4.2 Modification failed



(PPP 1)
ACI requests a modification of the PPP link.

(PPP 2)
PPP tries to negotiate the new link parameters by using the IPCP Establish procedure (see 3.6). If the negotiation fails, then PPP tries to negotiate the old parameters again. If this negotiation also failed PPP has to terminate the PPP link. In this case the Network-Layer Protocol phase (see 2.1.4) is finished.

(PPP 3)
If the Network-Layer Protocol phase (see 2.1.4) is finished, then the Termination procedure (see 3.9) within the Link Termination phase (see 2.1.5) will terminate the PPP link.

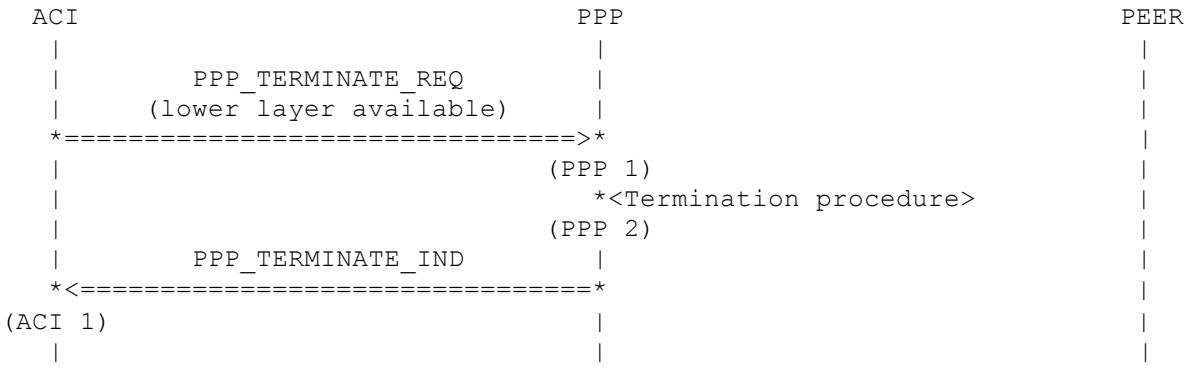
(ACI 1)
PPP informs ACI that the modification failed and the PPP link is terminated now.

3.5 Link Termination

Link termination can occur at any time including within link establishment. If link termination occurs all other activities within PPP will be stoped and termination will be finished.

3.5.1 ACI initiated Termination

3.5.1.1 Lower layer available in client and server mode

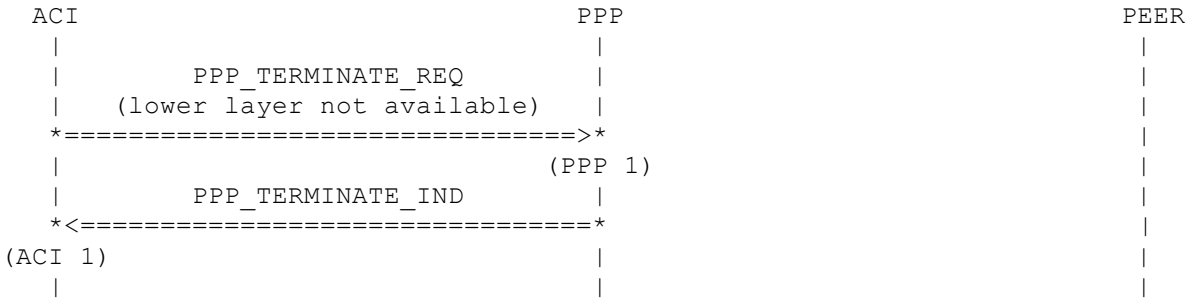


(PPP 1)
ACI indicates to PPP that PPP must terminate the link.

(PPP 2)
The Network-Layer Protocol phase (see 2.1.4) is finished now and the Termination procedure (see 3.9) within the Link Termination phase (see 2.1.5) will terminate the PPP link.

(ACI 1)
PPP informs ACI that the PPP link is terminated now.

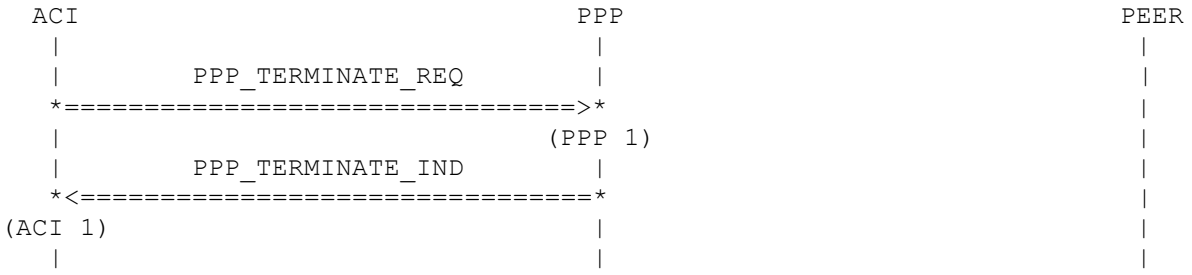
3.5.1.2 Lower layer not available in client and server mode



(PPP 1)
ACI indicates to PPP that PPP has to finish.

(ACI 1)
PPP informs ACI that PPP is terminated now.

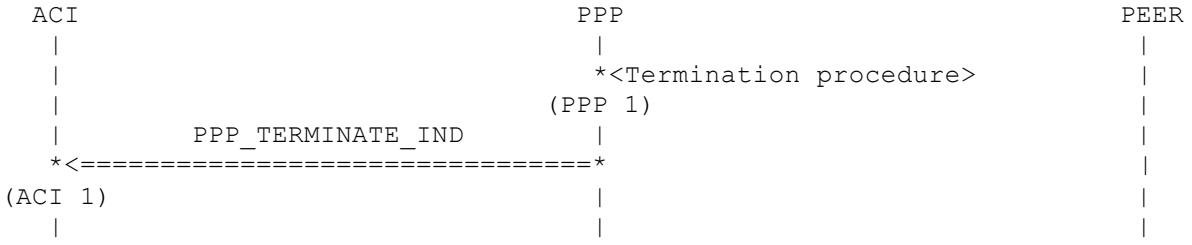
3.5.1.3 In transparent mode



(PPP 1)
ACI indicates to PPP that PPP has to finish.

(ACI 1)
PPP informs ACI that PPP is terminated now.

3.5.2 PPP peer initiated Termination in client and server mode



(PPP 1)
If the PPP peer desired to terminate the PPP link, then the Network-Layer Protocol phase (see 2.1.4) is finished and the Termination procedure (see 3.9) within the Link Termination phase (see 2.1.5) will terminate the PPP link.

(ACI 1)
PPP informs ACI that the PPP link is terminated now.

3.6 LCP and IPCP Establish procedure

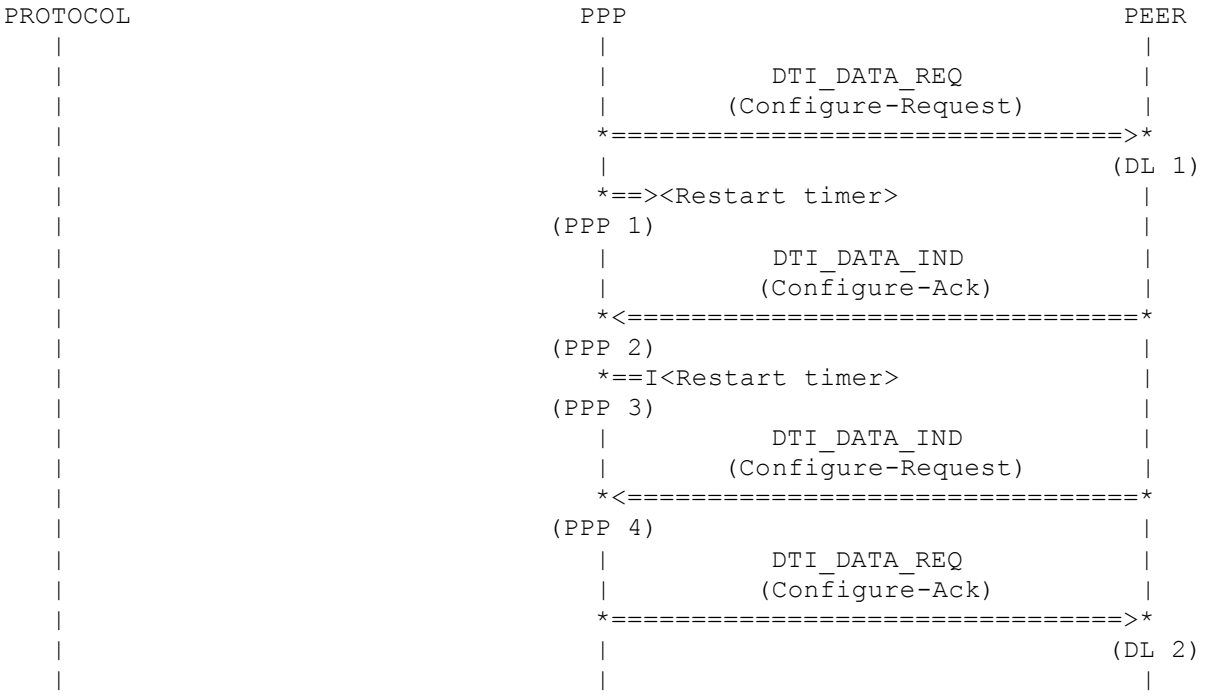
LCP and IPCP are used to establish the connection through an exchange of Configure packets. This exchange is complete, once a Configure-Ack packet has been both peers of the PPP link sent and received.

Only Configuration Options which are independent of particular network-layer protocols are configured by LCP. Configuration Options of the individual network-layer protocol IP is handled by the separate Network Control Protocol IPCP during the Network-Layer Protocol phase. Within each protocol all Configuration Options are always negotiated, acknowledged, Nak'd or rejected simultaneously.

After IPCP has reached the Opened state, PPP will carry IP packets.

3.6.1 Ideal establishment

The sequence of primitives could be different. It is only important that each peer sends a Configure-Ack when it receives a Configure-Request.



(DL 1)
PPP sends a Configure-Request to negotiate all desired Configuration Options.

(PPP 1)
The Restart timer is started to time retransmission of Configure-Request.

(PPP 2)
The PPP peer accepts all desired Configuration Options by sending a Configure-Ack.

(PPP 3)
The Restart timer is stopped because retransmission is not necessary.

(PPP 4)
The PPP peer sends a Configure-Request to negotiate all Configuration Options desired by the PPP peer.

(DL 2)
PPP accepts all desired Configuration Options by sending a Configure-Ack.

3.6.2 Get acceptable Configuration Options

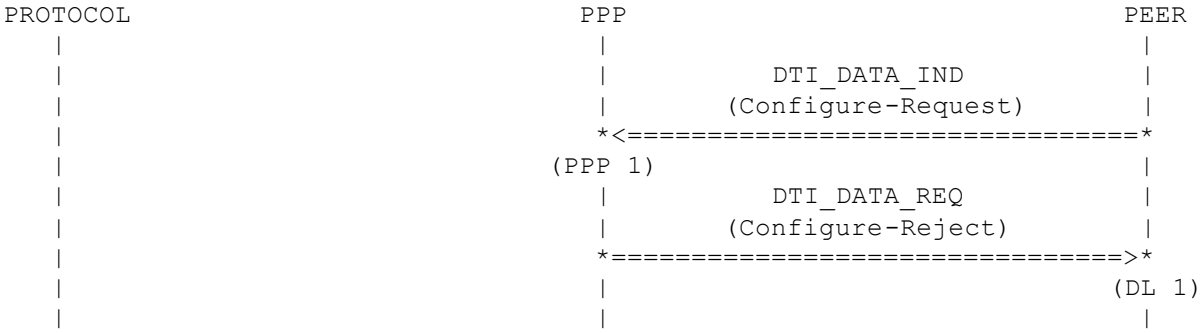
These MSCs just show the mechanism from the point of view of PPP. The same mechanism is used by the PPP peer to converge the Configuration Options requested by PPP. These two mechanisms

are work at the same time. Once both peers has been acknowledged the requested Configuration Options, the protocol is in the Opened state and the procedure is successful finished.

If a Configure-Request contains unrecognizable Configuration Options and unacceptable values, then the unrecognizable Configuration Options must be rejected.

Max-Failure indicates the number of Configure-Nak or Configure-Reject packets sent without receiving acceptable Configuration Options before assuming that configuration is not converging.

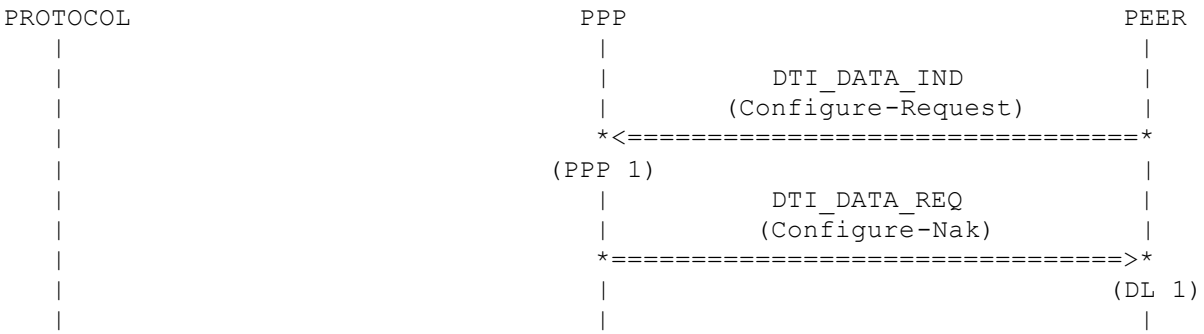
3.6.2.1 Reject unrecognizable Configuration Options



(PPP 1)
The PPP peer requests Configuration Options which are unrecognizable.

(DL 1)
PPP rejects these Configuration Options. The PPP peer must not send these Configuration Options in the next Configure-Request.

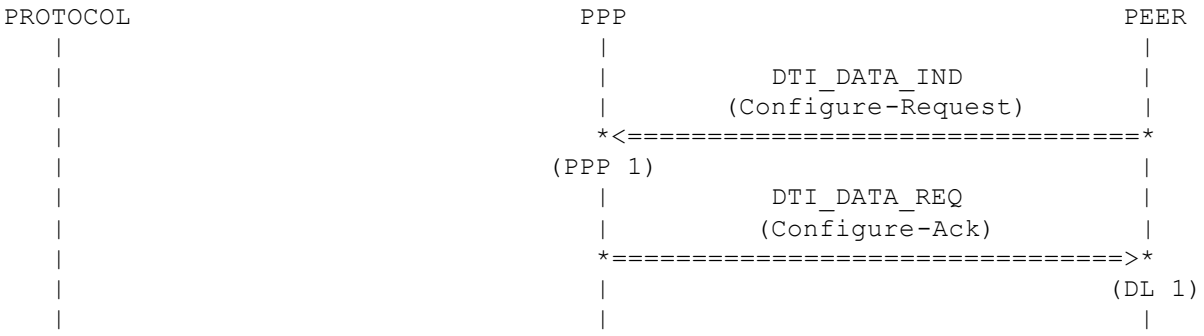
3.6.2.2 Modify unacceptable values



(PPP 1)
The PPP peer requests recognizable Configuration Options but with unacceptable values.

(DL 1)
PPP suggests acceptable values by sending a Configure-Nak. The PPP peer must modify these values in the next Confugure-Request.

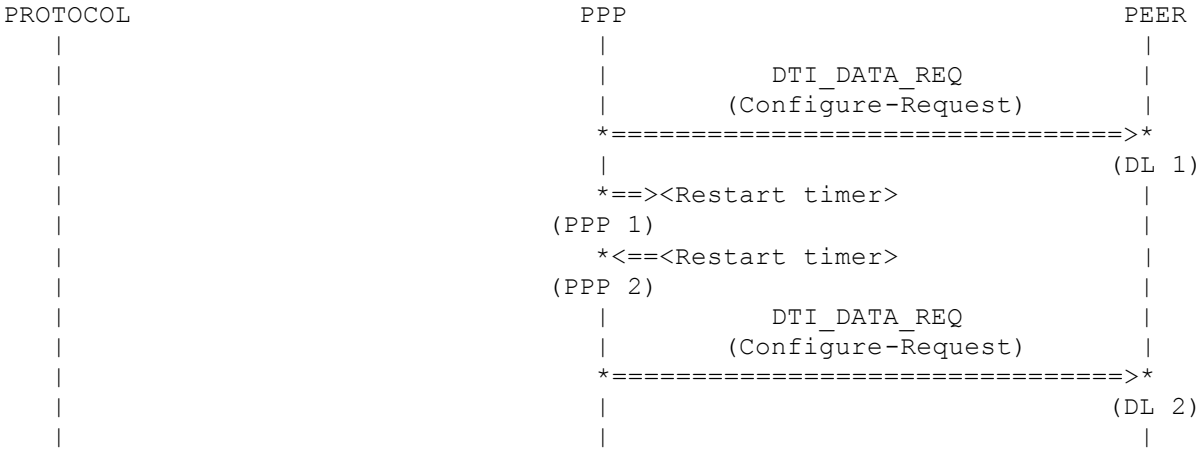
3.6.2.3 Acknowledge acceptable Configuration Options



(PPP 1)
The PPP peer only requests recognizable and acceptable Configuration Options.

(DL 1)
PPP acknowledges these Configuration Options by sending a Configure-Ack.

3.6.3 Retransmission



(DL 1)
PPP sends a Configure-Request to negotiate all desired Configuration Options.

(PPP 1)
The Restart timer is started to time retransmission of Configure-Request.

(PPP 2)
If no valid Configure-Ack, Configure-Nak or Configure-Reject is received, then the Restart timer expires.

(DL 2)
PPP sends the Configure-Request again. The Restart timer is also started again. The maximum number of retransmissions is indicated by Max-Configure.

3.7 Authentication procedure

If ACI desires authentication with some specific authentication protocol, then the use of that authentication protocol is requested during Link Establishment phase.

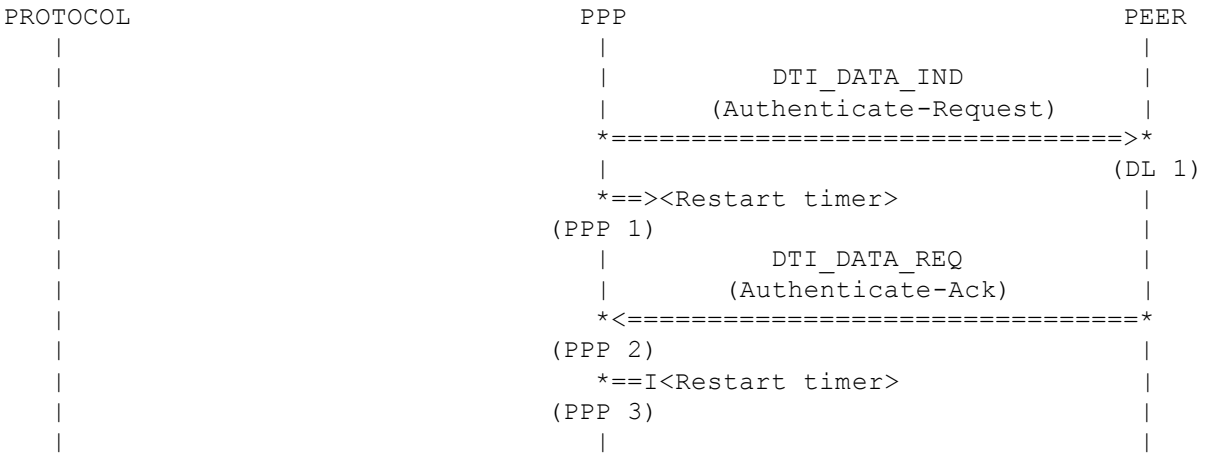
The implementation supports the following authentication protocols:

- Password Authentication Protocol (PAP) in client and server mode
- Challenge Handshake Authentication Protocol (CHAP) in server mode

In server mode PPP just stores the necessary authentication data and sends a forced positive acknowledgement of the authentication to the PPP peer. The stored authentication data will be inserted into the protocol configuration options of the PPP_PDP_ACTIVATE_IND primitive.

3.7.1 Password Authentication Protocol (PAP) in client mode

3.7.1.1 Usual authentication



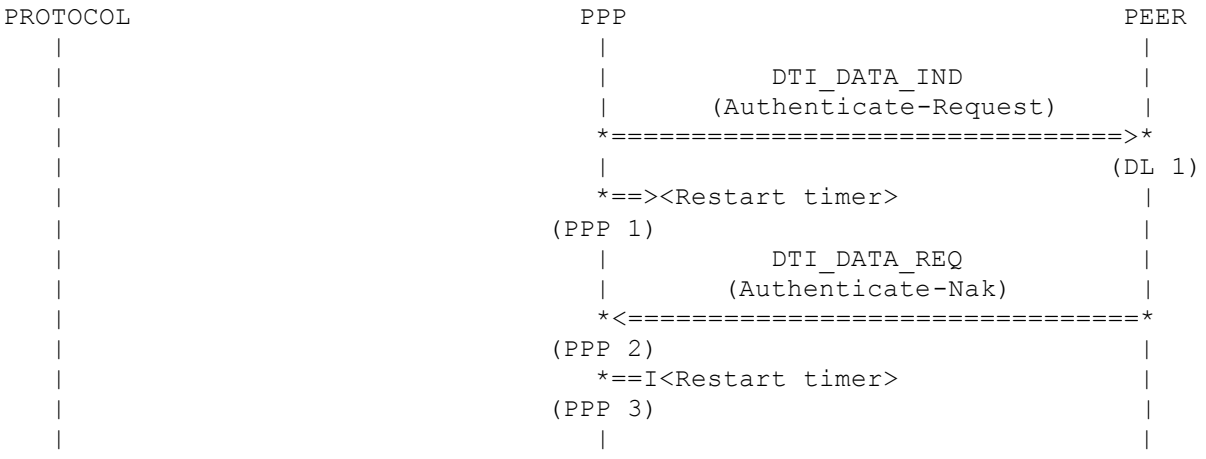
(DL 1)
PPP sends a PAP Authenticate-Request to authenticate itself at the PPP peer.

(PPP 1)
The Restart timer is started to time retransmission of Authenticate-Request.

(PPP 2)
The valid Authenticate-Ack indicates successful authentication.

(PPP 3)
The Restart timer is stopped because a valid PAP Authenticate-Ack is received.

3.7.1.2 Authentication failed



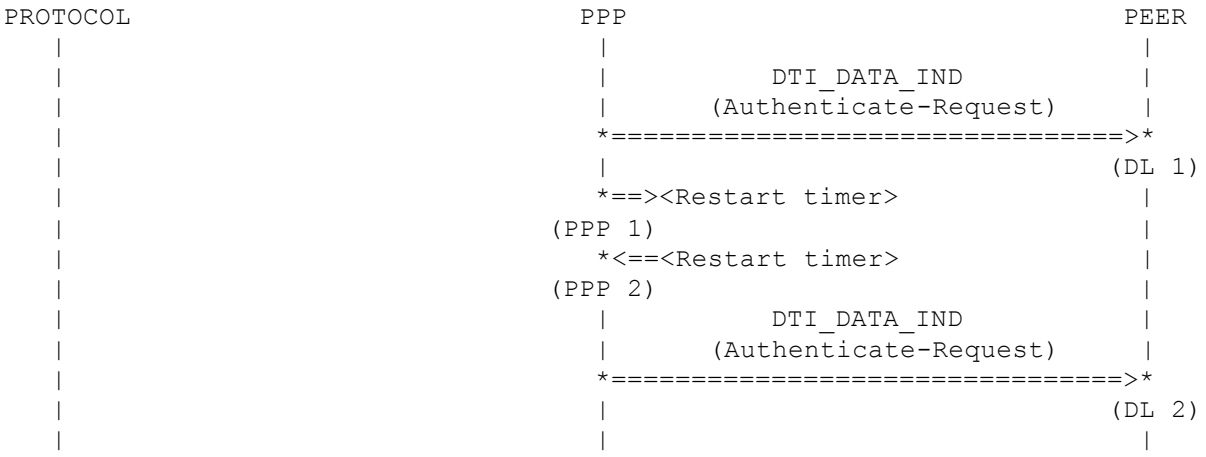
(DL 1)
PPP sends a PAP Authenticate-Request to authenticate itself at the PPP peer.

(PPP 1)
The Restart timer is started to time retransmission of Authenticate-Request.

(PPP 2)
The valid Authenticate-Nak indicates that the authentication failed.

(PPP 3)
The Restart timer is stopped because a valid PAP Authenticate-Nak is received.

3.7.1.3 Timer expiration



(DL 1)
PPP sends a PAP Authenticate-Request to authenticate itself at the PPP peer.

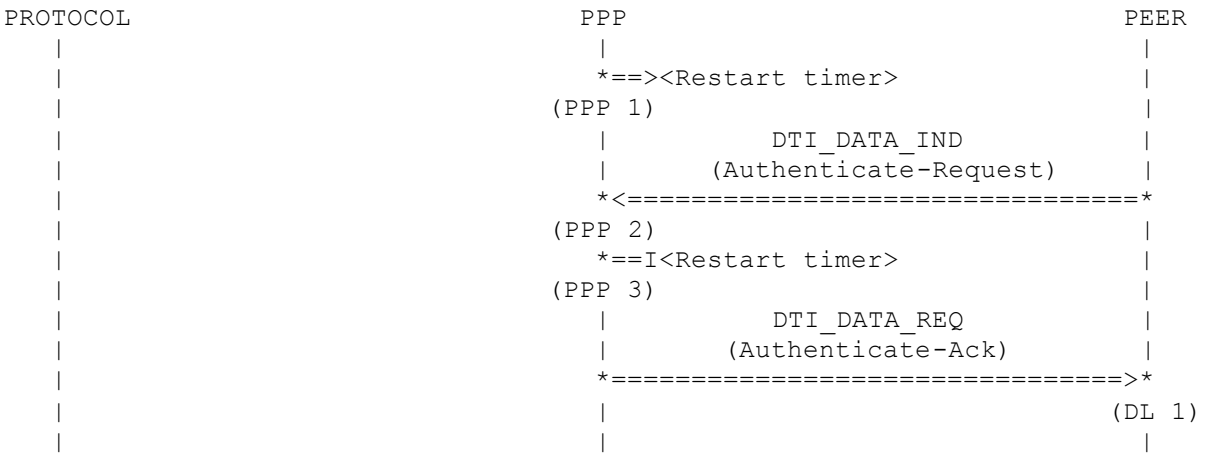
(PPP 1)
The Restart timer is started to time retransmission of Authenticate-Request.

(PPP 2)
If no valid PAP Authenticate-Ack or Authenticate-Nak is received, then the Restart timer expires.

(DL 2)
PPP sends the Authenticate-Request again. The Restart timer is also started again. The maximum number of retransmissions is indicated by Max-Configure.

3.7.2 Password Authentication Protocol (PAP) in server mode

3.7.2.1 Usual authentication



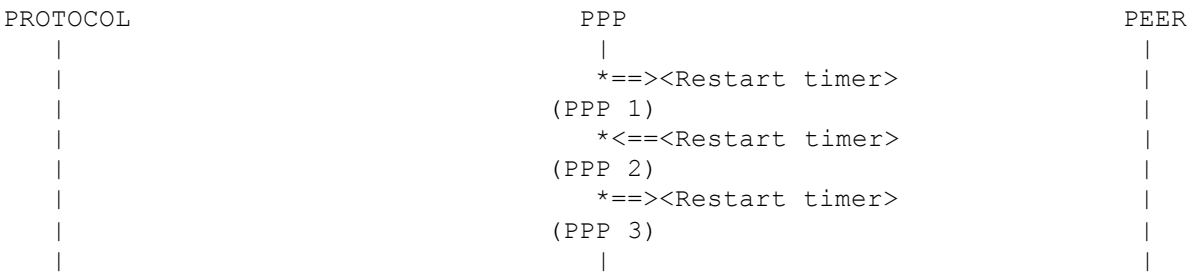
(PPP 1)
The Restart timer is started to avoid endless waiting.

(PPP 2)
PPP receives a valid PAP Authenticate-Request and stores it.

(PPP 3)
The Restart timer is stopped because a valid PAP Authenticate-Request is received.

(DL 1)
PPP does not check the received authentication data and sends always an Authenticate-Ack to the PPP peer.

3.7.2.2 Timer expiration



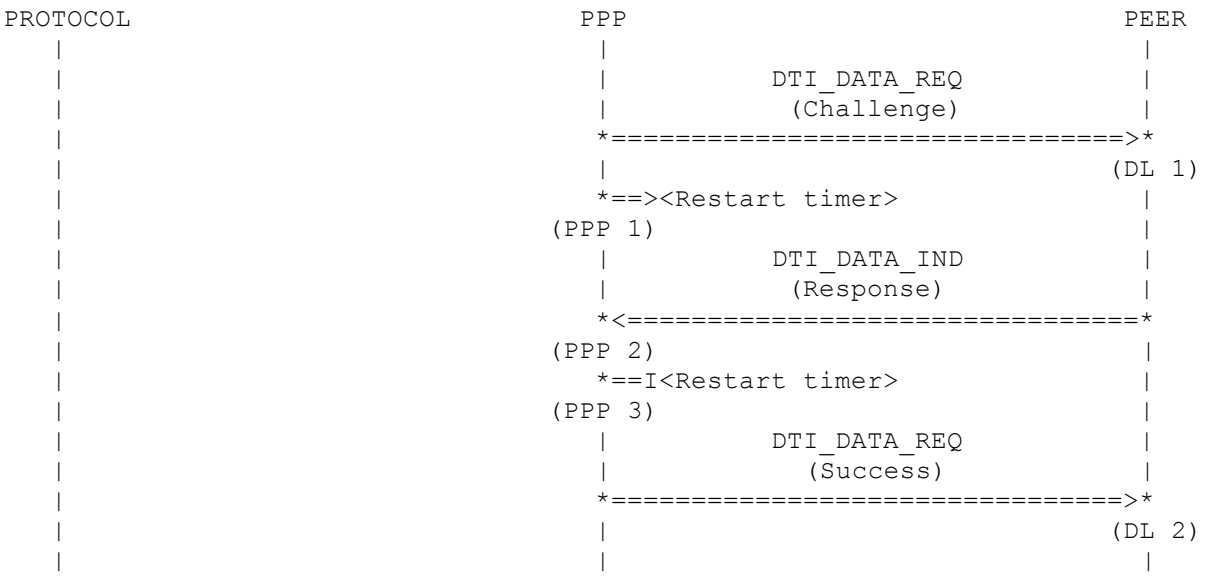
(PPP 1)
The Restart timer is started to avoid endless waiting.

(PPP 2)
If no valid PAP Authenticate-Request is received, then the Restart timer expires.

(PPP 3)
The Restart timer is started once more. The maximum number of restarts is indicated by Max-Configure.

3.7.3 Challenge Handshake Authentication Protocol (CHAP) in server mode

3.7.3.1 Usual authentication



(DL 1)
PPP sends a CHAP Challenge with an variable unique stream of octets.

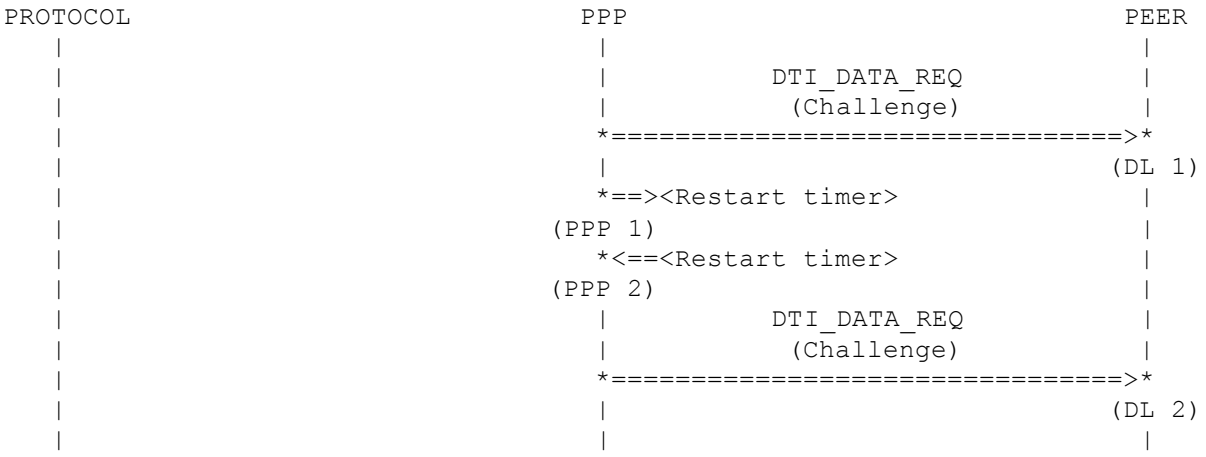
(PPP 1)
The Restart timer is started to time retransmission of CHAP Challenge.

(PPP 2)
The PPP peer sends a CHAP Response with a calculated stream of octets. PPP stores the sent and received octet stream.

(PPP 3)
The Restart timer is stopped because a valid CHAP Response is received.

(DL 2)
PPP do not check the received authentication data and sends always a CHAP Success to the PPP peer.

3.7.3.2 Retransmission



(DL 1)
PPP sends a CHAP Challenge to the PPP peer.

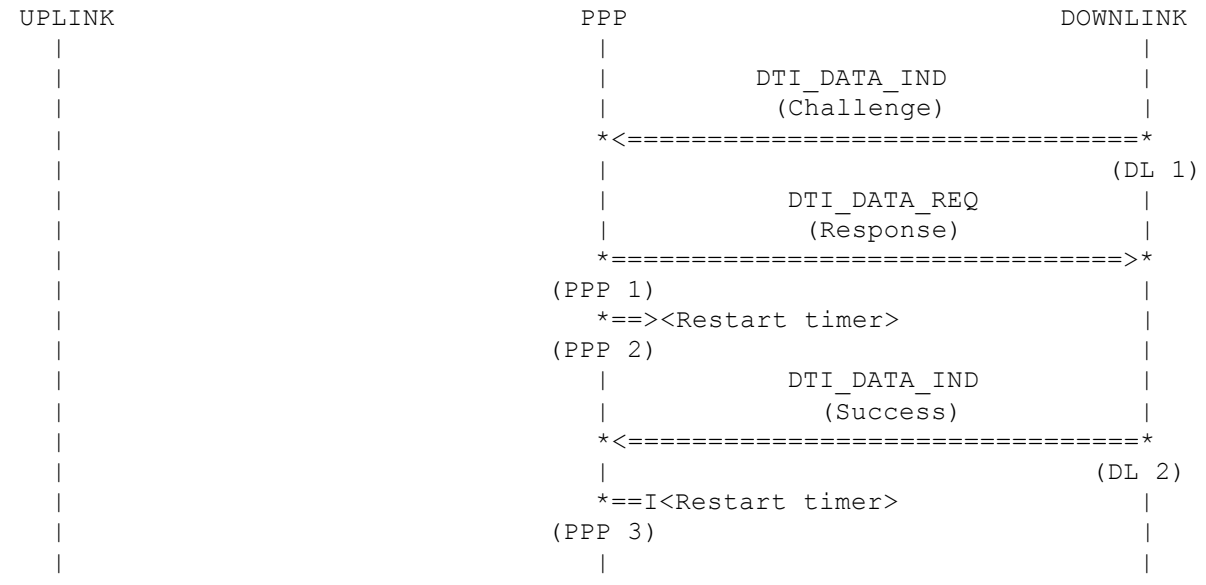
(PPP 1)
The Restart timer is started to time retransmission of CHAP Challenge.

(PPP 2)
If no valid CHAP Response is received, then the Restart timer expires.

(DL 2)
PPP sends a new CHAP Challenge. The Restart timer is started again. The maximum number of retransmissions is indicated by Max-Configure.

3.7.4 Challenge Handshake Authentication Protocol (CHAP) in client mode

3.7.4.1 Usually authentication



(DL 1)
The PPP receives a CHAP Challenge with a variable unique stream of octets. PPP stores the received octet stream.

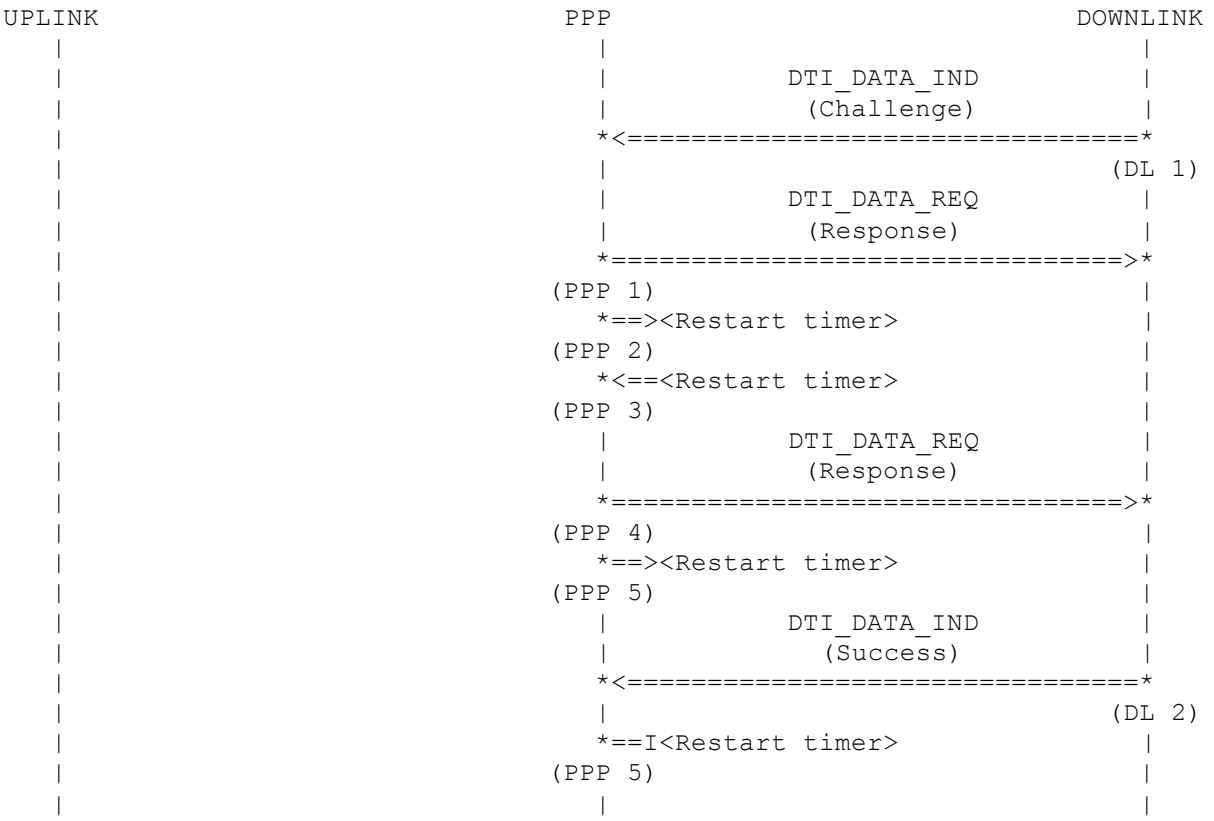
(PPP 1)
The PPP transmits a CHAP Response with a stream of octets calculated with Hash-Function (MD5 algorithm). PPP don’t save the transmitted packet. In case of retransmission, Response packet will be calculated again based on saved Challenge packet.

(PPP 2)
The Restart timer is started to time retransmission of CHAP Response.

(DL 2)
PPP peer checks the received authentication data and sends a CHAP Success to the PPP.

(PPP 3)
The Restart timer is stopped because the authentication is done.

3.7.4.2 Retransmission



(DL 1)
The PPP receives a CHAP Challenge with a variable unique stream of octets. PPP stores the received octet stream.

(PPP 1)
The PPP transmits a CHAP Response with a stream of octets calculated with Hash-Function (MD5 algorithm). PPP don't save the transmitted packet.

(PPP 2)
The Restart timer is started to time retransmission of CHAP Response.

(PPP 3)
If no valid CHAP Success is received, then the Restart timer expires.

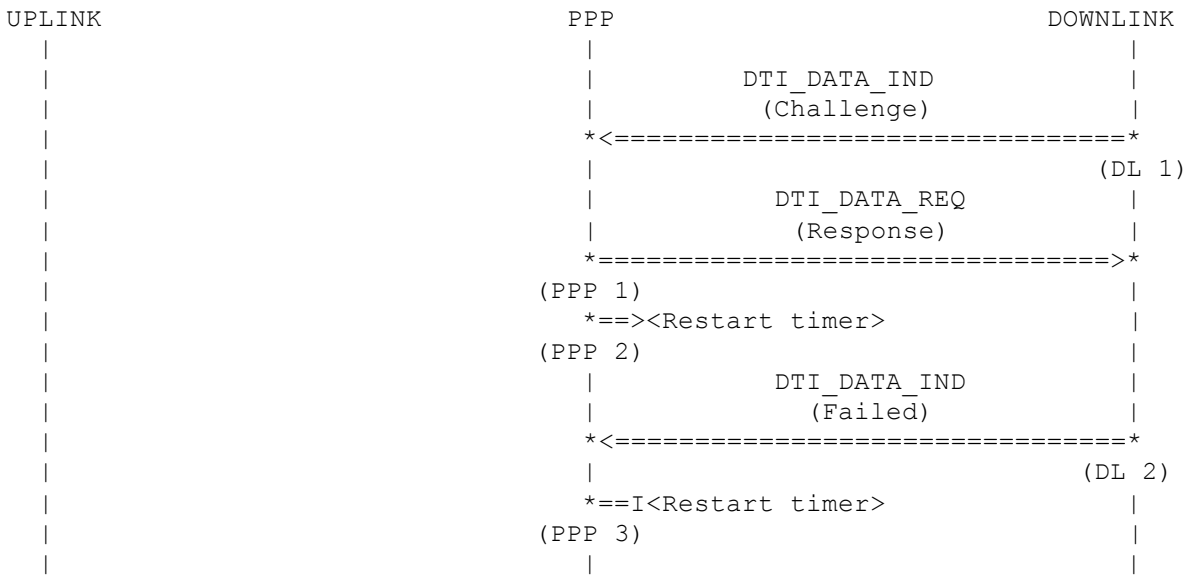
(PPP 4)
The PPP calculates a CHAP Response based on saved CHAP Challenge. The Restart timer is started again. The maximum number of retransmissions is indicated by Max-Configure.

(PPP 5)
The Restart timer is started to time retransmission of CHAP Response.

(DL 2)
The PPP peer sends CHAP Success to the PPP.

(PPP 5)
The Restart timer is stopped because the authentication is done.

3.7.4.3 Authentication failed



(DL 1)
The PPP receives a CHAP Challenge with a variable unique stream of octets. PPP stores the received octet stream.

(PPP 1)
The PPP transmits a CHAP Response with a stream of octets calculated with Hash-Function (MD5 algorithm). PPP don't save the transmitted packet. In case of retransmission, Response packet will be calculated again based on saved Challenge packet.

(PPP 2)
The Restart timer is started to time retransmission of CHAP Response.

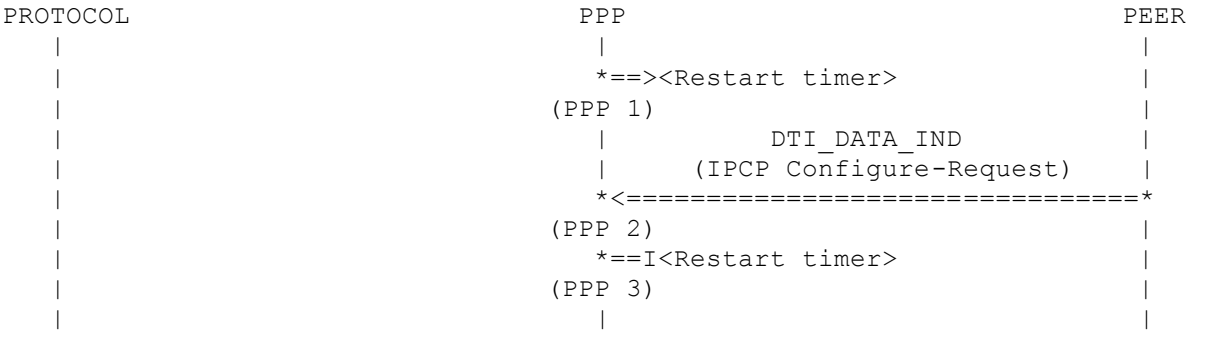
(DL 2)
PPP peer checks the received authentication data and sends a CHAP Failed to the PPP, if the value received in the Response in not equal to the expected value.

(PPP 3)
The Restart timer is stopped because the authentication is failed, the connection will be terminated, initiated by server.

3.8 IPCP Start procedure

This procedure is used to wait for a valid IPCP Configure-Request to fill in protocol configuration options needed for the PPP_PDP_ACTIVATE_IND primitive.

3.8.1 Usual IPCP Start procedure



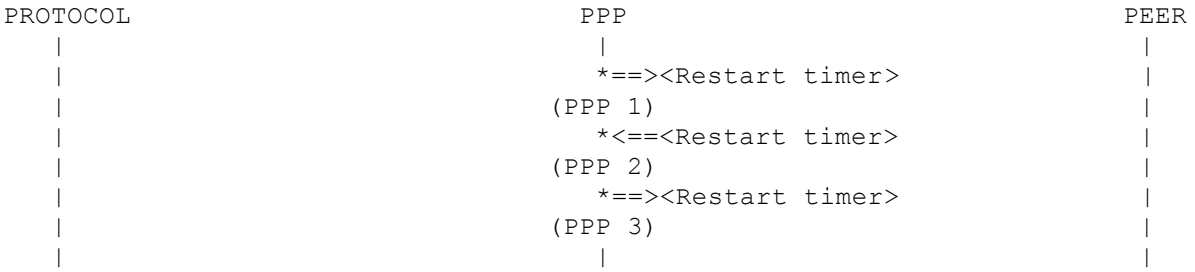
(PPP 1)
The Restart timer is started to avoid endless waiting.

(PPP 2)
PPP receives a valid IPCP Configure-Request and stores it.

(PPP 3)
The Restart timer is stopped because a valid IPCP Configure-Request is received.

3.8.2 Timer expiration

3.8.2.1 Without authentication

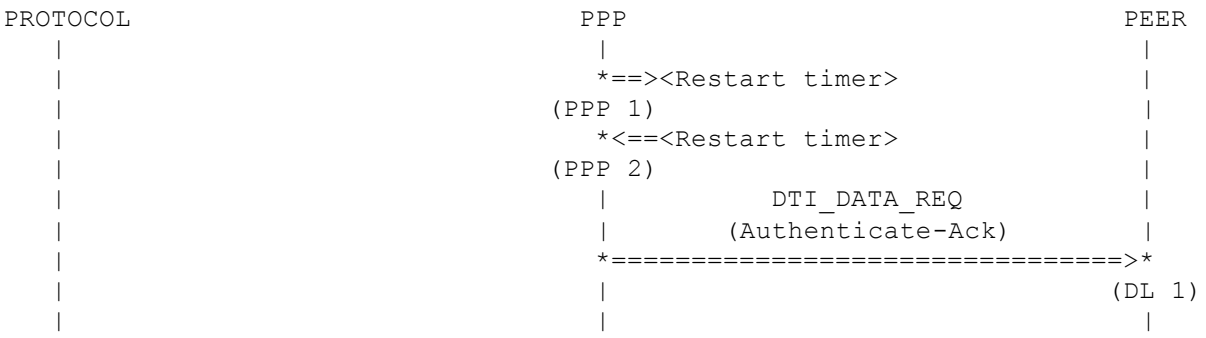


(PPP 1)
The Restart timer is started to avoid endless waiting.

(PPP 2)
If no valid IPCP Configure-Request is received, then the Restart timer expires.

(PPP 3)
The Restart timer is started once more. The maximum number of restarts is indicated by Max-Configure.

3.8.2.2 With Password Authentication Protocol (PAP)

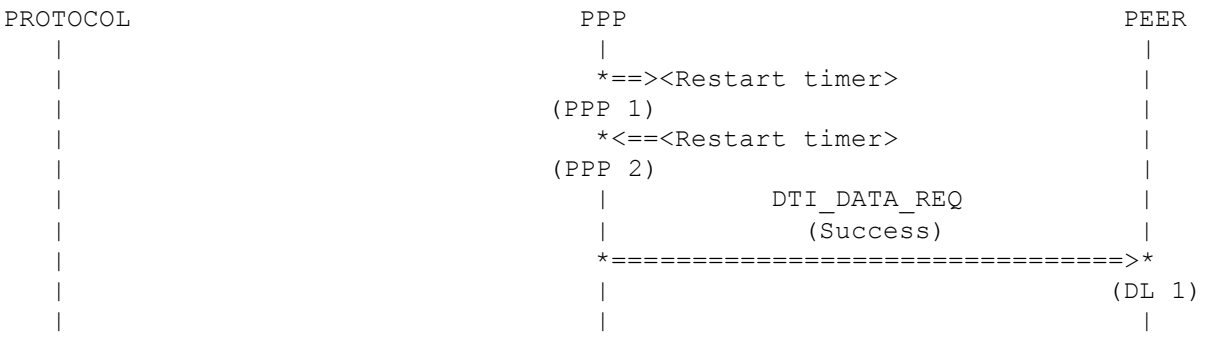


(PPP 1)
The Restart timer is started to avoid endless waiting.

(PPP 2)
If no valid IPCP Configure-Request is received, then the Restart timer expires.

(DL 1)
PPP sends the Authenticate-Ack again. The Restart timer is also started again. The maximum number of re-transmissions is indicated by Max-Configure.

3.8.2.3 With Challenge Handshake Authentication Protocol (CHAP)



(PPP 1)
The Restart timer is started to avoid endless waiting.

(PPP 2)
If no valid IPCP Configure-Request is received, then the Restart timer expires.

(DL 1)
PPP sends the CHAP Success again. The Restart timer is also started again. The maximum number of retransmissions is indicated by Max-Configure.

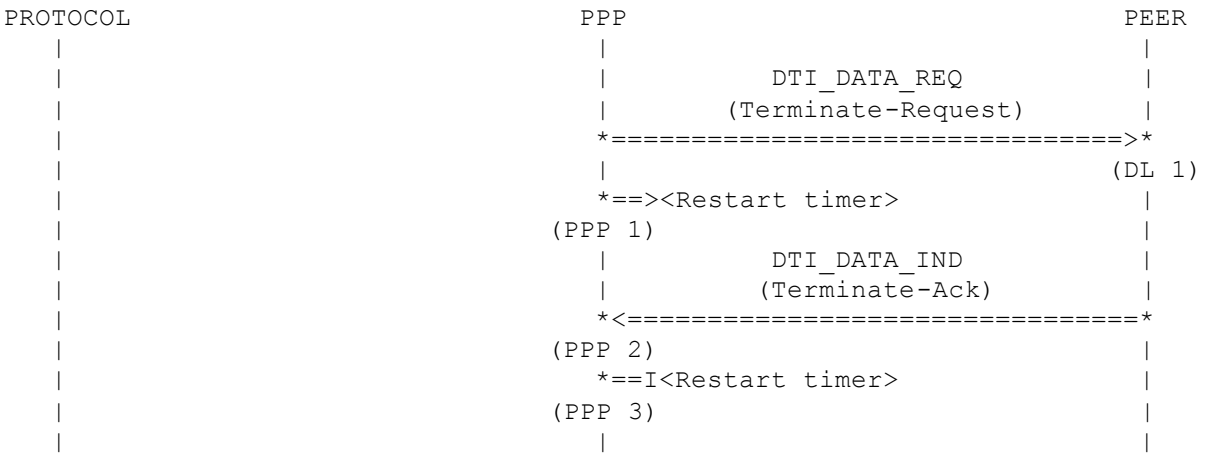
3.9 Termination procedure

PPP can terminate the link at any time. This might happen because of the loss of carrier, authentication failure or the administrative closing of the link.

LCP is used to close the link through an exchange of Terminate packets.

The sender of the Terminate-Request should disconnect after receiving a Terminate-Ack, or after the Restart counter expires. The receiver of a Terminate-Request should wait for the peer to disconnect, and must not disconnect until at least one Restart time has passed after sending a Terminate-Ack.

3.9.1 ACI/PPP initiated Termination



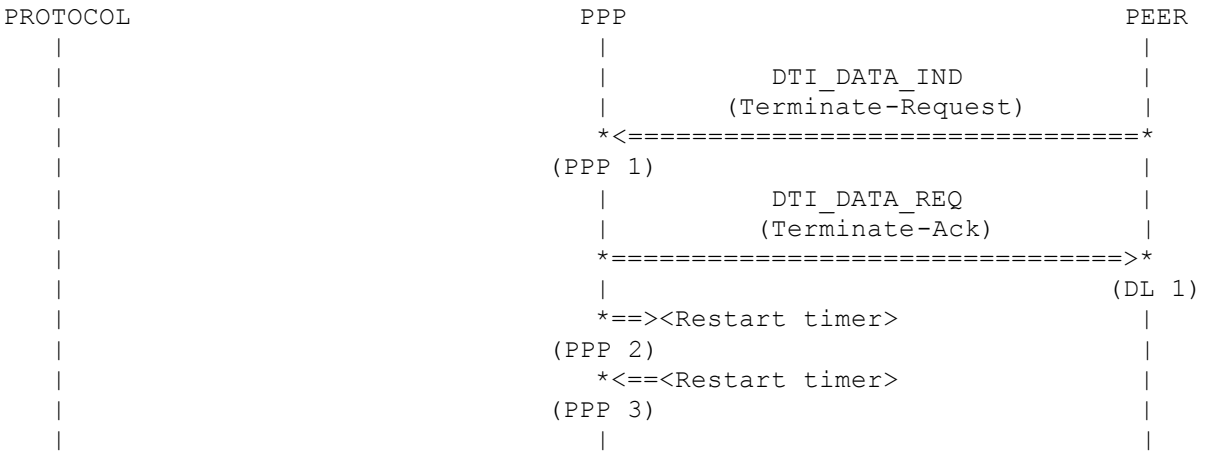
(DL 1)
PPP sends a Terminate-Request to indicate termination of the PPP link.

(PPP 1)
The Restart timer is started to time retransmission of Terminate-Request.

(PPP 2)
The PPP peer acknowledges the termination of the PPP link.

(PPP 3)
The Restart timer is stopped because a Terminate-Ack is received.

3.9.2 PPP peer initiated Termination



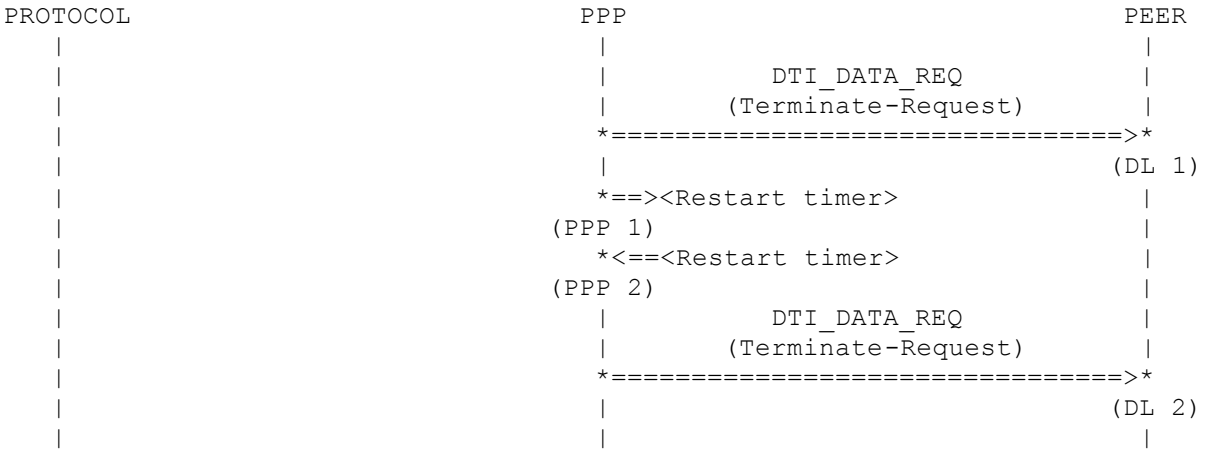
(PPP 1)
The PPP peer sends a Terminate-Request to indicate termination of the PPP link.

(DL 1)
PPP acknowledges the termination of the PPP link.

(PPP 2)
The Restart timer is started to be prepared in case of retransmission of Terminate-Request.

(PPP 3)
If no valid Terminate-Request is received again, then the Restart timer expires and PPP closes down the connection.

3.9.3 Retransmission



(DL 1)
PPP sends a Terminate-Request to indicate termination of the PPP link.

(PPP 1)
The Restart timer is started to time retransmission of Terminate-Request.

(PPP 2)
If no valid Terminate-Ack is received, then the Restart timer expires.

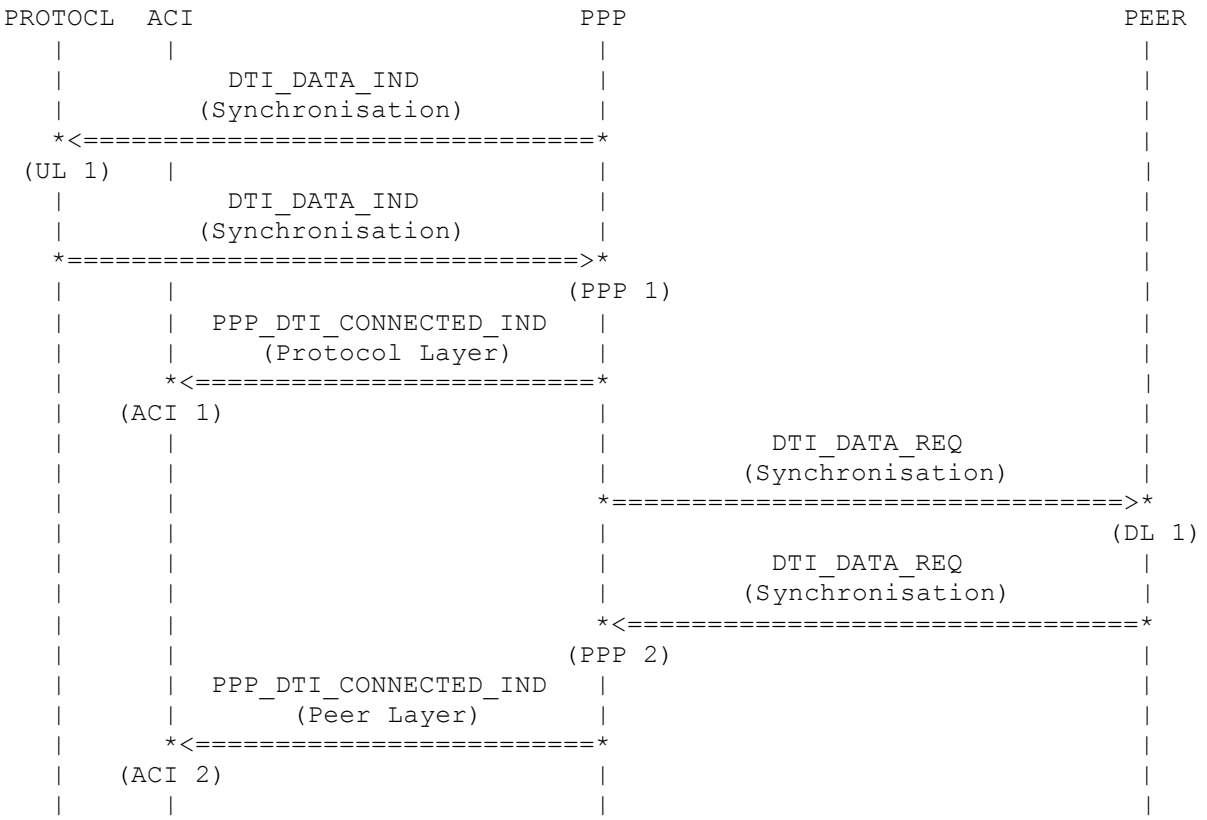
(DL 2)
PPP sends the Terminate-Request again. The Restart timer is also started again. The maximum number of retransmissions is indicated by Max-Terminate.

3.10 Handling of DTI

3.10.1 Connection of DTI channels

After receiving the primitive PPP_ESTABLISH_REQ, PPP starts to establish DTI connections to PROTOCOL layer and PEER layer.

PPP sends the primitive PPP_DTI_CONNECTED_IND for each established DTI connection.



(UL 1)
PPP sends a DTI synchronisation primitive to its PROTOCOL layer.

(PPP 1)
PPP receives a synchronisation primitive from its PROTOCOL layer.

(ACI 1)
PPP has established DTI channel connection to PROTOCOL layer, because it has send and received a synchronisation primitive.

(DL 1)
PPP sends a DTI synchronisation primitive to its PEER layer.

(PPP 2)
PPP receives a synchronisation primitive from its PEER layer.

(ACI 2)
PPP has established DTI channel connection to PEER layer, because it has send and received a synchronisation primitive.

3.10.2 Disconnection of DTI channels

There is no primitive exchange to disconnect a DTI channel. Disconnection is simply done by reseting internal states.

Appendices

A. Acronyms

ACI	Application Control Interface
AGCH	Access Grant Channel
AT	Attention sequence "AT" to indicate valid commands of the ACI
BCCH	Broadcast Control Channel
BS	Base Station
BSIC	Base Station Identification Code
C/R	Command/Response
C1	Path Loss Criterion
C2	Reselection Criterion
CBCH	Cell Broadcast Channel
CBQ	Cell Bar Qualify
CC	Call Control
CCCH	Common Control Channel
CCD	Condat Coder Decoder
CCI	Compression and Ciphering Interface
CHAP	Challenge Handshake Authentication Protocol
CKSN	Ciphering Key Sequence Number
CRC	Cyclic Redundancy Check
DCCH	Dedicated Control Channel
DCOMP	Identifier of the user data compression algorithm used for the N-DPU
DISC	Disconnect Frame
DL	Data Link Layer
DM	Disconnected Mode Frame
DTX	Discontinuous Transmission
E	Extension bit
EA	Extension Bit Address Field
EL	Extension Bit Length Field
EMMI	Electrical Man Machine Interface
F	Final Bit
FACCH	Fast Associated Control Channel
FHO	Forced Handover
GACI	GPRS Application Control Interface
GMM	GPRS Mobility Management
GP	Guard Period
GRR	GPRS RR
GSM	Global System for Mobile Communication
HDLC	High-level Data Link Control
HISR	High level Interrupt Service Routine
HPLMN	Home Public Land Mobile Network
I	Information Frame
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPCP	Internet Protocol Control Protocol
ITU	International Telecommunication Union
IWF	Interworking Function
Kc	Ciphering Key

L	Length Indicator
LAI	Location Area Information
LCP	Link Control Protocol
LISR	Low level Interrupt Service Routine
LLC	Logical Link Control
LPD	Link Protocol Discriminator
LQM	Link Quality Monitoring
M	More bit used to indicate the last segment of N-DPU
MAC	Medium Access Control
MCC	Mobile Country Code
MM	Mobility Management
MMI	Man Machine Interface
MNC	Mobile Network Code
MS	Mobile Station
MT	Mobile Termination
N(R)	Receive Number
N(S)	Send Number
NC	Network Control
NCC	National Colour Code
NCP	Network Control Protocol
NECI	New Establishment Causes included
N-PDU	Network Protocol Data Unit
NSAPI	Network Layer Service Access Point Identifier
OTD	Observed Time Difference
P	Poll Bit
P/F	Poll/Final Bit
PACCH	Packet Associated Control Channel
PAP	Password Authentication Protocol
PBCCH	Packet BCCH
PCCCH	Packet CCCH
PCOMP	Identifier of the protocol control information compression algorithm used for the N-DPU
PDCH	Packet Data Channel
PDP	Packet Data Protocol e.g. IP or X.25
PDTCH	Packet Data Traffic Channel
PRACH	Packet RACH
PSI	Packet System Information
PCH	Paging Channel
PCO	Point of Control and Observation
PDU	Protocol Data Unit
PL	Physical Layer
PLMN	Public Land Mobile Network
PPC	Packet Physical Convergence
PPP	Point-to-Point Protocol
PTP	Point to Point
QoS	Quality of Service
RACH	Random Access Channel
REJ	Reject Frame
RLC	Radio Link Control
RNR	Receive Not Ready Frame
RR	Radio Resource Management
RR	Receive Ready Frame
RTD	Real Time Difference
RTOS	Real Time Operating System
SABM	Set Asynchronous Balanced Mode
SACCH	Slow Associated Control Channel
SAP	Service Access Point

SAPI	Service Access Point Identifier
SDCCH	Stand alone Dedicated Control Channel
SDU	Service Data Unit
SGSN	Serving GPRS Support Node
SIM	Subscriber Identity Module
SM	Session Management
SMS	Short Message Service
SMSCB	Short Message Service Cell Broadcast
SNDCP	Subnetwork Dependant Convergence Protocol
SNSM	SNDCP-SM
SS	Supplementary Services
TAP	Test Application Program
TBF	Temporary Block Flow
TCH	Traffic Channel
TCH/F	Traffic Channel Full Rate
TCH/H	Traffic Channel Half Rate
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TE	Terminal Equipment - e. g. a PC
TFI	Temporary Flow Identifier
TLLI	Temporary Logical Link Identifier
TMSI	Temporary Mobile Subscriber Identity
TOM	Tunnelling of Messages
TQI	Temporary Queuing Identifier
UA	Unnumbered Acknowledgement Frame
UART	Universal Asynchronous Receiver Transmitter
UI	Unnumbered Information Frame
USF	Uplink State Flag
V(A)	Acknowledgement State Variable
V(R)	Receive State Variable
V(S)	Send State Variable
VPLMN	Visited Public Land Mobile Network

B. Terms

Entity:	Program which executes the functions of a layer
Message:	A message is a data unit which is transferred between the entities of the same layer (peer-to-peer) of the mobile and infrastructure side. Message is used as a synonym to protocol data unit (PDU). A message may contain several information elements.
Primitive:	A primitive is a data unit which is transferred between layers on one component (mobile station or infrastructure). The primitive has an operation code which identifies the primitive and its parameters.
Service Access Point:	A Service Access Point is a data interface between two layers on one component (mobile station or infrastructure).