PPP Interface

Teldat-Dm 710-I

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents

Teldat Dm708-I PPoE Interface

Teldat Dm715-I BRS

Teldat Dm717-I Bridge

Teldat Dm727-I Backup WAN Reroute

Teldat Dm732-I Dial Profile

Teldat Dm733-I RADIUS Protocol

Teldat Dm744-I Dial-Routing

Teldat Dm760-I L2TP: Layer 2 Tunneling Protocol

Teldat Dm772-I Common Configurations for Interfaces

Teldat Dm800-I AAA Feature

Chapter 1 PPP Interface

1.1 Description

Point-to-Point Protocol (PPP), as described in RFC 1661, provides a mechanism for transporting multi-protocol datagrams over point-to-point links. PPP supports both synchronous and asynchronous data transmission and provides the following services:

- A *Link Control Protocol* (LCP) to establish, configure and test the data link connection.
- A method of encapsulating datagrams over links.
- Authentication protocols to validate device identity at the remote end of a PPP link before transmitting data. The current implementation supports *Password Authentication Protocol* (PAP), described in RFC 1172, and *Challenge Handshake Authentication Protocol* (CHAP), described in RFC 1994.
- PPP Multilink, as described in RFC 1990, to split, recombine and sequence datagrams across multiple data links.
- *Network control protocols* (NCP) for establishing and configuring different network layer protocols supported by PPP.

Note

PPP implementation in **Teldat Router**s supports the following network protocols: Internet Protocol Control Protocol (IPCP) described in RFC 1332, Bridging Control Protocol (BCP) described in RFC 2878, Compression Control Protocol (CCP) described in RFC 1962, and Internet Protocol Version 6 Control Protocol (IPV6CP) described in RFC 5072.

To set up a point-to-point connection, the originating PPP sends LCP frames to configure and test the data link. After the link has been established, the originating PPP sends NCP frames, corresponding to the supported protocols (in this case IPCP, BCP, CCP and IPV6CP), to choose and configure the link's network layer protocols. When the network layer protocols have been configured, packets from each protocol can be sent over the link.

The conversion to asynchronous PPP is performed in accordance with RFC 1662.

There are several possibilities in the current implementation:

(1) Define a PPP interface over a synchronous/asynchronous serial link to connect to another endpoint through a cable, external device, modem or terminal adapter.

(2) Define a PPP interface over a serial link as AT command interface, to connect to another endpoint through an external modem. In this case, data transmission is asynchronous.

(3) Define a PPP interface over a basic access (BRI) to connect via an ISDN line.

(4) Define a PPP interface over an ATM (PPPoA/PPPoE) subinterface.

(5) Define a PPP interface over a G.703 interface: in a channel or channel aggregation.

(6) Define a PPP interface over an L2TP virtual interface.

(7) Define a PPP interface over a Frame Relay (PPPoFR) PVC.

(8) Define a PPP interface over an internal AT command interface (GPRS, modem, UMTS, PCMCIA-serial, etc.).

(9) Define a PPP interface over an HSSI interface.

1.2 PPP frame structure

PPP frames have the same format as HDLC frames. PPP uses the bit-oriented synchronous transmission method with the following frame structure:

FLAG	ADDRESS	CONTROL	PROTOCOL	INFORMATION	FCS	FLAG

FLAG

A single byte that indicates the beginning and end of a frame and consists of a unique binary sequence: 01111110. It enables frame synchronization. Transparency (*bit-stuffing*) is applied to the rest of the data in the frame to ensure that this pattern can never occur in normal data. Transparency consists of the transmitter inserting a zero after five consecutive 1s. Upon receiving the frame, this 0 is eliminated.

ADDRESS

HDLC frame address field. In PPP, this field always contains the binary sequence 11111111. PPP does not assign

individual addresses.

CONTROL

Control field. All PPP frames are HDLC frames containing unnumbered information (UI). This field is set to 00000011.

PROTOCOL

This 2-byte field is used to identify the various protocols transported over the point-to-point link. Protocol field values in the Cxxx range correspond to link configuration protocols (LCP, PAP, CHAP) while values in the 8xxx range correspond to network control protocols (IPCP, BCP, CCP, IPV6CP). Field values in the 0xxx range identify the network protocol of specific datagrams.

INFORMATION

Zero or more bytes that contain the datagram for the transported protocol. If it is an LCP or NCP protocol, this field will contain parameters for link configuration.

FCS

Frame Check Sequence: field used to include an error detection mechanism, in this case a Cyclic Redundancy Code (CRC) for 2-byte error detection.

1.2.1 Asynchronous PPP adaptation

Asynchronous data transmission uses the same frame structure as synchronous transmission (as described in RFC 1662). The transmitted bytes are encapsulated within the frame described in the previous section but the 0x7D transparency character is added to implement *bit-stuffing*.

After calculating the FCS number, the frame is examined. Each flag, control character (below 0x20), escape character and any character defined in the *Asynchronous Control Character Map* (ACCM) is replaced by the transparency character and an additional character that is the result of performing an exclusive OR operation on the original character with 0x20.

At the receiving end, the frame bytes are checked prior to calculating the FCS. If the transparency character appears, it is removed and the next character is replaced by its exclusive OR with 0x20.

The calculation and subsequent cyclic redundancy check (CRC) of the frame's FCS field is performed using a polynomial algorithm defined in RFC 1662, and its subsequent transmission is also affected by transparency.

Thus, for example, the 0x11 (XON) character is encoded within the transmission frame as 0x7D + 0x31. At the receiving end, 0x7D is deleted and the next byte is replaced by its exclusive OR with 0x20, resulting in 0x11 again.

1.3 LCP: Link Control Protocol

The PPP *Link Control Protocol* (LCP) provides a method for establishing, configuring, maintaining, and terminating a point-to-point connection. There are four main phases to this process:

(1) Before any IP datagrams are exchanged over the link, LCP exchanges *configure-request* packets between the two endpoints to open the connection. Once the configuration has been accepted by means of *configure-ack* packets, the link enters an OPEN state. Note that both ends send *configure-request* packets and wait to receive *configure-ack* packets in order to confirm that the remote end has accepted the sent configuration. Only when both ends have received an *ack* packet does the link go into the OPEN state.

(2) Once LCP has determined that the link is OPEN, it checks whether the link quality is sufficient to bring up network layer protocols. If required, link authentication is carried out during this process.

(3) After determining that the link quality is sufficient, the PPP interface is declared UP and network control protocols (NCP) are brought up.

(4) Finally, after establishing the NCPs, LCP will maintain the link by transmitting *echo-request* and *echo-reply* packets. Say you want to terminate the connection because, for example, the idle time has expired, LCP will terminate the connection by transmitting *terminate-request* and *terminate-ack* packets.

1.4 LCP packet format

LCP packets have the same format as described above. The protocol field is set to 0xC021 for LCP and the packet type and associated data are encoded in the information field:

CODE	IDENT	SIZE	DATA

CODE

This is a one-byte field that identifies the kind of LCP packet, i.e., configure-request, configure-ack, and so on, as per the following table:

CODE	LCP PACKET TYPE
1 (0x01)	CONFIGURE-REQUEST (Establishment)
2 (0x02)	CONFIGURE-ACK (Establishment)
3 (0x03)	CONFIGURE-NAK (Establishment)
4 (0x04)	CONFIGURE-REJECT (Establishment)
5 (0x05)	TERMINATE-REQUEST (Termination)
6 (0x06)	TERMINATE-ACK (Termination)
7 (0x07)	CODE-REJECT (Maintenance)
8 (0x08)	PROTOCOL-REJECT (Maintenance)
9 (0x09)	ECHO-REQUEST (Maintenance)
10 (0x0A)	ECHO-REPLY (Maintenance)
11 (0x0B)	DISCARD-REQUEST (Maintenance)
12 (0x0C)	IDENTIFICATION (Informative)

IDENTIFIER

This is a one-byte field that aids in matching requests and replies over a link.

SIZE

This is a two-byte field that indicates the overall length of the LCP packet. If you are using asynchronous mode, it does not include any transparency bits in the frame.

DATA (optional)

This field consists of zero or more bytes. The format of the Data field depends on the type of LCP packet.

As seen in the above table, LCP packets can be grouped into four types:

- **Link establishment packets**:
  - **CONFIGURE REQUEST**

    This packet is sent to open a link and contains the configuration options for the link. The receiving end must send an appropriate response to the remote end, either accepting or rejecting the proposal by means of one of the packets described below.
  - **CONFIGURE-ACK**

    The configuration options are acceptable. The frame identifier field must match that of the accepted *configure-request* message. Once the two ends have received the *ack* message from the remote end, the link goes into an OPEN state.
  - **CONFIGURE-NAK**

    Some of the configuration options are not acceptable but the remote end returns a value that it can accept. When a *nak* is received, the receiver must generate another *configure-request* message containing the acceptable values.
  - **CONFIGURE-REJECT**

    The configuration options are not acceptable/recognizable. When a *reject* message is received, the receiver must generate a new *configure-request* message that does not contain the rejected values.
- **Link termination packets**:
  - **TERMINATE-REQUEST**

    This packet is sent when you want to close a link.
  - **TERMINATE-ACK**

    This packet is sent in response to a *terminate-request* message. If an unscheduled *terminate-ack* message is received, it indicates that the connection

has been dropped.

- **Link maintenance packets**:
  - **CODE-REJECT**

    This packet is sent in response to an incomplete or uninterpretable LCP packet. The link will close if the packet is sent repeatedly.
  - **PROTOCOL-REJECT**

    This packet is sent in response to a PPP packet with an unknown protocol. Upon receipt of this frame, the receiving end must cease to send the protocol.
  - **ECHO-REQUEST, ECHO-REPLY**

    This provides a link maintenance mechanism. An *echo-request* message is sent at regular intervals and an *echo-reply* message must be sent in response.
  - **DISCARD-REQUEST**

    This provides a mechanism for discarding and eliminating frames and is used for testing.
- **Informative packets**:
- **IDENTIFICATION**

  This type of packet allows a device to identify itself by means of a text string (this may contain information about the manufacturer, version number, etc.). This kind of message is not negotiated and there is no clearly defined response.

1.5 Authentication protocols

PPP has a series of protocols to authenticate and verify a data link. The link will only be established if the remote end of the connection is among the authorized users. This method is usually used in links in which routers connect to a network via switched circuits (ISDN or PSTN), but it can also be used in point-to-point circuits.

This check is performed prior to establishing the network control protocols (NCP). If authentication is required and fails, the link-establishment phase is terminated.

RFC 1334 describes two authentication methods:

1.5.1 Password Authentication Protocol (PAP)

PAP provides a simple method for authenticating the data link based on a 2-way handshake:

(1) When the LCP negotiation reaches the OPEN state, the end that must be authenticated sends a username (*login*) and password to the authenticator.

(2) The end that receives the authentication request checks that it is valid and sends an appropriate response indicating whether or not it accepts or rejects the connection.

PAP is a weak authentication method as the username and password are sent in clear text over the link and are thus vulnerable to errors or other kinds of attacks. This problem is resolved by the other authentication method, CHAP.

1.5.1.1 PAP packet format

PAP packets have the same format as PPP frames. The protocol field value (which is set to 0xC023) identifies the PAP packets. The type of packet and associated data are encoded in the Information field.

CODE	IDENT	SIZE	DATA

CODE

The code field is one byte in length and identifies the type of PAP packet, based on the following codes:

CODE	PAP PACKET TYPE
1	AUTHENTICATE-REQUEST
2	AUTHENTICATE-ACK
3	AUTHENTICATE-NAK

IDENTIFIER

This field is one byte, and aids in matching requests and replies.

SIZE

This field is two bytes. It indicates the overall length of the PAP frame.

DATA (optional)

This field consists of zero or more bytes. The format of the Data field depends on the type of PAP packet.

As seen in the above table, there are three different types of PAP packets:

- **AUTHENTICATE-REQUEST**

  This packet is sent to authenticate the link following a request for authentication credentials from the remote end. The authenticate-request packet is used to transmit an identifier (*login*) and password to the remote endpoint. The remote endpoint, having received the authenticate-request packet, must send a valid reply in the form of one of the following packets.

- **AUTHENTICATE-ACK**

  The received values are acceptable (the remote end is authorized to establish the link). The identifier field must match the identifier field in the received *authenticate-request* packet. Once the *ack* packet is received from the authenticator, the network control protocols (NCP) can be established.

- **AUTHENTICATE-NAK**

  The received values are not acceptable (the username/password pair are not among those authorized to establish the link). The end that has to authenticate must send another *authenticate-request* packet with appropriate values or close the link.

1.5.2 Challenge Handshake Authentication Protocol (CHAP)

CHAP provides a secure method to authenticate the data link based on a 3-way handshake.

(1) Once the LCP negotiation is complete (i.e., LCP reaches the *open* state), the authenticator sends a password, called a *Challenge* packet, to the end it wants to connect to. The password value is variable in all connections; it is usually a random value and its length depends on the hash-algorithm used for the subsequent encryption. In this case, the MD5 algorithm defined in RFC 1321 (which defines a 16-byte length for the *challenge*) is used.

(2) The end receiving the *Challenge* packet encrypts the received key with a common secret key and returns the encrypted value back to the authenticator in a response packet. The encryption function is defined by the MD5 algorithm and is the same at both ends.

(3) Upon receiving the response, the authenticator checks that the result in the packet is what it expects and then either allows (*success*) or does not allow (*failure*) the NCP establishment phase to go ahead.

The security of this method depends on a secret key known to both ends that never travels across the network in clear.

Additionally, this method repeatedly authenticates the link even after the network protocols (for example, IP) have been established.

1.5.2.1 CHAP packets format

CHAP packets have the same format as PPP frames. The protocol field value (which is set to 0xC223) identifies the PAP packets. The type of packet and associated data are encoded in the Information field.

CODE	IDENT	SIZE	DATA

CODE

The code field is one byte in length and identifies the type of CHAP packet, based on the following codes:

CODE	CHAP PACKET TYPE
1	CHALLENGE
2	RESPONSE
3	SUCCESS
4	FAILURE

IDENTIFIER

This field is one byte, and aids in matching requests and replies.

SIZE

This field is two bytes. It indicates the overall length of the CHAP frame.

DATA (optional)

This field consists of zero or more bytes. The format of the Data field depends on the type of CHAP packet.

As seen in the above table, there are four different types of CHAP packets:

- **CHALLENGE**

  This packet is sent by the authenticator to authenticate a link. It contains the encryption key and may also include the name of the network to access in clear text. This value can be used to ensure that the endpoint that has to send the response packet provides an appropriate value in the event that it can connect to more than one network (i.e., different keys may be programmed depending on the network you want to access).

- **RESPONSE**

  This is the packet sent by the endpoint which contains the password encrypted with the received key.

- **SUCCESS**

  The transmitted value is accepted by the authenticator. Once a *success* message has been received from the authenticator, the NCP establishment phase can go ahead.

- **FAILURE**

  The received value is not accepted by the authenticator. The end that has to authenticate itself in the link must send another response message with the appropriate values or close the link.

1.6 Network Control Protocol (NCP)

PPP has a series of network control protocols (NCP) to select and configure different network protocols encapsulated over PPP. Each protocol's corresponding NCP is in charge of configuring, enabling and disabling the network protocols between both ends of a link.

The network protocols currently implemented in the router include *Internet Protocol Control Protocol* (IPCP), described in RFC 1332, *Bridging Control Protocol* (BCP), described in RFC 2878, *Compression Control Protocol* (CCP), described in RFC 1962, and *Internet Protocol Version 6 Control Protocol* (IPV6CP), described in RFC 5072.

IPCP allows you to specify whether Van Jacobson compression is used and provides a mechanism for interchanging IP addresses between both endpoints or the dynamic IP number assignment, required for Internet connections. It is also possible to assign and/or learn the DNS servers, and/or assign the NetBIOS servers that the remote end should use.

BCP is able to set the link operating mode according to the configuration of the *bridge port* involved. In addition, it allows you to configure the use of frames with the *tinygram-compression* feature.

CCP allows you to establish the use of various algorithms for compressing data sent over the link.

IPV6CP allows you to establish and configure IPv6 over PPP. One of the options that can be configured is the interface identifier that each endpoint uses, through the negotiation of 64-bit interface identifiers. The interface identifier is unique in the link and is used to form the link-local address and address autoconfiguration in the PPP interface.

1.7 References

RFC 1661: The Point-to-Point protocol, W. Simpson, July-1994

RFC 1662: PPP in HDLC-Like Framing, W. Simpson, July-1994

RFC 1618: PPP in ISDN, W.Simpson, May-1994

RFC 1570: PPP LCP extensions, W. Simpson, January-1994

RFC 1332: PPP Internet control protocol, G. McGregor, May-1992

RFC 1334: PPP Authentication protocols, B. Lloyd, October-1992

RFC 1172: Point-to-Point Protocol (PPP) initial configuration options. D. Perkins, R. Hobby. July-1990

RFC 1994: PPP Challenge Handshake Authentication Protocol (CHAP). W. Simpson. August-1996

RFC 1321: The MD5 message-digest Algorithm. R. Rivest, April-1992

RFC 1700: Assigned numbers, IETF, October-1994

RFC 1471: The Definitions of Managed Objects for the Link Control Protocol of the Point-to-Point Protocol, F. Kastenholz, August-1993

RFC 1990: The PPP Multilink Protocol (MP), Network Working Group, August-1996

RFC 1962: The PPP Compression Control Protocol, Network Working Group, June-1996

RFC 2878: PPP Bridging Control Protocol (BCP). M. Higashiyama, F. Baker. July-2000

RFC 2686: The Multi-Class Extension to Multi-Link PPP. C. Bormann. September-1999

RFC 5072: IP Version 6 over PPP

Chapter 2 Interface Configuration

2.1 Configuring a PPP interface

To create PPP interfaces, type the following command from the general configuration menu:

add device ppp <PPP interface identifier>

```
Config>add device ppp 1
Config>
```

You can check whether the interface has been successfully added by listing the device's interfaces:

```
Config>list devices
Interface            Connector     Type of interface
ethernet0/0          LAN1          Quicc Ethernet
serial0/0            SERIAL0/WAN1  Synchronous Serial Line
atm0/0               DSL1          ATM
bri0/0               BRI/ISDN1     ISDN Basic Rate Int
x25-node             ---           Router->Node
atm0/0.1             ---           ATM subinterface
ppp1                 ---           Generic PPP
Config>
```

2.2 Deleting a PPP interface

To remove an existing PPP interface, use the following command from the general configuration menu:

no device <interface PPP>

Syntax:

```
Config>no device <interface_name>
```

- *<interface_name>* name of the interface to be deleted (*pppX*, X=interface identifier).

Example:

```
*config
Config>no device ppp1
Config>
```

You can check whether the interface has been removed by listing the device's interfaces:

```
Config>list devices
Interface            Connector     Type of interface
ethernet0/0          LAN1          Quicc Ethernet
serial0/0            SERIAL0/WAN1  Synchronous Serial Line
atm0/0               DSL1          ATM
bri0/0               BRI/ISDN1     ISDN Basic Rate Int
x25-node             ---           Router->Node
atm0/0.1             ---           ATM subinterface
ppp1                 ---           Generic PPP
Config>no device ppp1
Config>list devices
Interface            Connector     Type of interface
ethernet0/0          LAN1          Quicc Ethernet
serial0/0            SERIAL0/WAN1  Synchronous Serial Line
atm0/0               DSL1          ATM
bri0/0               BRI/ISDN1     ISDN Basic Rate Int
x25-node             ---           Router->Node
atm0/0.1             ---           ATM subinterface
Config>
```

2.3 Configuring a PPP interface

To access the configuration menu of a PPP interface, type **network <PPP interface>** from the general configuration menu. For example, if you want to access the ppp2 interface, type:

```
Config>network ppp2
-- Generic PPP User Configuration --
ppp2 config>
```

All the interfaces on the device support certain common commands. These are described in Teldat Dm772-I Commom Configurations for Interfaces.

Specific commands from the PPP interface configuration menu include:

```
pppX config>?
  base-interface    Enter the Base Interface configuration menu
  l2tp              Enter the L2TP configuration menu
  ppp               Enter the PPP parameters configuration menu
  pppoe             Enter the PPPoE configuration menu
  exit
pppX config>
```

Command	Function
? (HELP)	Displays the available commands or options.
BASE-INTERFACE	Accesses the configuration menu of the base interfaces associated with the PPP interface.
L2TP	Allows you to configure the PPP interface parameters when used over an L2TP virtual interface.
PPP	Accesses the menu that allows you to configure PPP parameters.
PPPOE	Accesses the configuration menu for PPPoE parameters.
EXIT	Exits the PPP interface configuration.

Configuring a PPP interface consists of the following main tasks:

- Specifying the base interfaces on which PPP will be negotiated.
- Configuring PPP parameters (negotiation, authentication, etc.).
- Sometimes you will have to modify the global profiles of the LCP options associated with the base interfaces.
- Where appropriate, specify PPPoE/L2TP protocol configurations.
- Configuring the parameters of the network protocols you want to support (IP address for IPCP, bridge port for BCP, etc.).
- If you want to display traces or receive notifications in the form of SNMP traps or syslog messages (in order to monitor/control the operation of the protocol), you must also configure the PPP events system.

2.3.1 Configuring the base interfaces on the PPP interface

To access the configuration menu for the base interfaces, type the **base-interface** command in the PPP interface configuration menu.

```
Config>network pppX
-- Generic PPP User Configuration --
pppX config>base-interface

-- Base Interface Configuration --
pppX Base IFC config>
```

The commands available in this configuration menu are:

```
pppX Base IFC config>?
  base-interface    Specify a base interface
  list              List current configuration
  mode              Defines how to handle multiple base interfaces
  no                Negates a command or sets its defaults
  exit
pppX Base IFC config>
```

Command	Function
BASE-INTERFACE	Allows you to specify the base interfaces over which the PPP connection is estab-

	lished.
LIST	Displays the base interfaces that are linked to the PPP interface.
MODE	Defines how to handle the base interfaces that are linked to the PPP interface.

BASE-INTERFACE

The **base-interface** command allows you to associate a specific base interface and, if necessary, certain DIAL process parameters (call) with the PPP interface.

The syntax of the **base-interface** command is usually as follows:

```
pppX Base IFC config>base-interface <interface> [<circuit id>] <options>
  link                 Add this interface to the dial group
  number-of-circuits   Number of circuits to request
  profile              Dial profile to use with this interface
```

<interface>	Name of the base interface.
[<circuit id>]	Circuit/channel identifier (for interfaces with multiple channels/circuits).
link	Adds a base interface to the PPP.
number-of-circuits	Number of base circuits to use (valid on switch interfaces when the circuit identifier is **255**).
profile	Call profile used by the base interface (switch interfaces only).

Depending on the base interface, a number of different cases are described below.

2.3.1.1 Permanent base interface with a single channel/circuit.

You only have to indicate the base interface.

To configure this type of base interface:

```
pppX Base IFC config>base-interface <interface> link
```

Example:

```
ppp1 Base IFC config>base-interface atm0/0.1 link
ppp1 Base IFC config>list

      Base Interface        Profile Name     Base Circuit Id  Number of circuits
-------------------------  ------------     ---------------  ------------------
       atm0/0.1   subatm/0                          255               1

ppp1 Base IFC config>
```

To remove this type of base interface:

```
pppX Base IFC config>no base-interface <interface>
```

This type of configuration applies to the following base interfaces:

- Serial (synchronous/asynchronous).
- ATM subinterface (PPPoA/PPPoE).
- HSSI interface.
- L2TP virtual interface.
- Ethernet interface/subinterface (PPPoE).

2.3.1.2 Permanent base interface with several channels/circuits

In this case, you need to specify the specific channel/circuit or DLCI, as well as the base interface.

To configure this type of base interface:

```
pppX Base IFC config>base-interface <interface> <id circuit> link
```

Example:

Add the fr1 Frame Relay interface DLCI 21:

```
ppp1 Base IFC config>base-interface fr1 21 link
ppp1 Base IFC config>list
```

```
      Base Interface        Profile Name     Base Circuit Id  Number of circuits
-------------------------  ------------     ---------------  ------------------
            fr1       fr/0                          21                1

ppp1 Base IFC config>
```

To remove this type of base interface:

```
pppX Base IFC config>no base-interface <interface> <circuit id>
```

This type of configuration applies to the following base interfaces:

- G.703 (either a specific channel or a group of channels).
- Frame Relay (PPPoFR).

2.3.1.3 Switched base interface with a single channel/circuit

As well as specifying the base interface, you also need to specify the DIAL profile that contains the call parameters (additional information on configuring Call Profiles is provided in the Teldat Dm732-I Dial Profile manual).

To configure this type of base interface:

```
pppX Base IFC config>base-interface <interface> link
pppX Base IFC config>base-interface <interface> profile <dial-profile>
```

Example:

Assuming that the serial0/1 interface is configured as an AT commands interface:

```
ppp1 Base IFC config>base-interface serial0/1 link
ppp1 Base IFC config>base-interface serial0/1 profile atPSTN
ppp1 Base IFC config>list
      Base Interface        Profile Name     Base Circuit Id  Number of circuits
-------------------------  ------------     ---------------  ------------------

      serial0/1       at/0    atPSTN                1                 1

ppp1 Base IFC config>
```

To remove this type of base interface:

```
pppX Base IFC config>no base-interface <interface>
```

This type of configuration applies to the following base interfaces:

- AT commands interface on a serial line (external modem).
- Internal AT commands interface (GPRS, modem, UMTS, PCMCIA-serial, etc).

Important

If you fail to provide a valid DIAL profile, the base interface will not be able to establish the link because it will not have the call parameters.

2.3.1.4 Switched base interface with several channels/circuits

In this case you must specify the base interface, the channel/circuit and the profile containing the call parameters - the DIAL profile (for more information on configuring Call Profiles, please refer to the Teldat Dm732-I Dial Profile manual).

When the base interface is a switch interface, the value of the channel/circuit identifier must be **255** to indicate that you do not want to specify a particular circuit and that any available channel should be used.

In particular cases where you have a permanent point-to-point link over a switched network (e.g., an ISDN basic access B channel), you must specify the circuit identifier in the same way as in section **b** (in the ISDN/BRI example, the value of the circuit identifier can be 1 or 2 to refer to channels B1 and B2 respectively).

To configure this type of base interface:

```
pppX Base IFC config>base-interface <interface> <circuit id> link
pppX Base IFC config>base-interface <interface> <circuit id> profile <dial-profile>
```

When the PPP Multilink Protocol (MP) is enabled in this type of base interface (with the circuit identifier set to **255**), you may also have the option of specifying the number of channels/circuits to use.

```
pppX Base IFC config>base-interface <interface> <circuit id> number-of-circuits <n>
```

Example:

To configure PPP Multilink (MP) on two ISDN/BRI B channels:

```
ppp1 Base IFC config>base-interface bri0/0 255 link
ppp1 Base IFC config>base-interface bri0/0 255 profile mpISDN
ppp1 Base IFC config>base-interface bri0/0 255 number-of-circuits 2
ppp1 Base IFC config>list
      Base Interface        Profile Name     Base Circuit Id  Number of circuits
-------------------------  ------------     ---------------  ------------------

         bri0/0      bri/0    mpISDN                255               2

ppp1 Base IFC config>
```

To remove this type of base interface:

```
pppX Base IFC config>no base-interface <interface> <circuit id>
```

This type of configuration applies to the following base interfaces:

- ISDN basic access (BRI).

Important

If you fail to provide a valid DIAL profile, the base interface will not be able to establish the link because it will not have the call parameters.

You can associate several base interfaces (regardless of type) with the same PPP.

This example shows how to configure Multilink PPP on two synchronous serial interfaces:

```
ppp1 Base IFC config>base-interface serial0/0 link
ppp1 Base IFC config>base-interface serial0/1 link
ppp1 Base IFC config>list

      Base Interface        Profile Name     Base Circuit Id  Number of circuits
-------------------------  ------------     ---------------  ------------------
      serial0/0     sync/0                          1                1
      serial0/1     sync/1                          1                1

ppp1 Base IFC config>
```

Note

Adding more than one base interface to a PPP interface makes sense if you are going to enable either PPP Multilink (use base interfaces simultaneously) or enable scan mode (use only the first base interface that allows PPP negotiation).

LIST

This command will show you the base interfaces associated with the PPP interface.

Example:

```
ppp1 Base IFC config>list

      Base Interface        Profile Name     Base Circuit Id  Number of circuits
-------------------------  ------------     ---------------  ------------------
         bri0/0      bri/0    mpISDN                255               2

ppp1 Base IFC config>
```

Base Interface	Base interface associated with the PPP interface.
Profile Name	Name (identifier) of the DIAL profile to be used by the base interface (this only applies to switched interfaces).

Base Circuit Id	Circuit identifier.
Number of circuits	The total number of circuits configured for the base interface.

MODE

This command allows you to define how to to use the base interfaces associated with the PPP interface.

Syntax:

```
ppp1 Base IFC config>mode <options>
  scan              Base interfaces are scanned
```

default	Base interfaces are used simultaneously in order to set up a Multilink PPP connection.
scan	Base interfaces are used sequentially, retaining the first one that allows a PPP negotiation. (This facilitates configurations that support different scenarios at once).

Note

Currently, the scan mode is only available for PPPoE.

Command history:

Release	Modification
11.01.06	This command is available as of version 11.01.06.

2.3.2 Configuring PPP parameters on the PPP interface

To set the PPP parameters, use the **ppp** command in the PPP interface configuration menu.

```
Config>network ppp1
-- Generic PPP User Configuration --
ppp1 config>ppp
-- PPP Configuration --
ppp1 PPP config>
```

The PPP parameter configuration is explained in detail in chapter 3.

2.3.3 Configuring global PPP profiles

Use the **global-profiles ppp command** to access the global PPP profile configuration menu. See chapter 4 for a detailed explanation on how to configure global PPP profiles.

Note

Global PPP profiles are common to all PPP interfaces available on the device, meaning any changes you make to them can affect other PPP interfaces.

Command history:

Release	Modification
11.00.03	This command is obsolete as of version 11.00.03. The **global-profile ppp**menu is no longer supported. See section 2.3.4 for information about the new configuration menu.

2.3.4 Configuring PPP options per interface

PPP parameters are located in each interface's configuration menu. Thus, to configure PPP in the ethernet0/0 interface, with randomly-chosen values from valid ranges, enter the following:

```
network ethernet0/0
ppp lcp-options mru 987
ppp lcp-options pfc
ppp lcp-options acfc
ppp lcp-options accm abcdef
```

```
ppp lcp-options no-magic-number
```

Note

The MRU value must be set on a per-interface basis. Typical MRU values are 1492 for Ethernet and 2048 for Serial.

Command history:

Release	Modification
11.00.03	This command is available as of version 11.00.03. See Section 5 for information about this new configuration menu.

2.3.5 Configuring PPPoE on the PPP interface

This protocol is explained in detail in the Teldat Dm708-I PPoE Interface manual.

2.3.6 Configuring L2TP parameters on the PPP interface

This protocol is explained in detail in the Teldat Dm760-I L2TP: Layer 2 Tunneling Protocol manual.

2.3.7 Configuring the network protocol parameters

IP is one of the network protocols supported by PPP interfaces. To enable the protocol on a PPP interface, the interface needs an IP address. This operation must be performed regardless of whether or not the PPP interface is going to receive a dynamic IP address. You can only have one IP address per PPP interface because IPCP only supports one IP address per interface.

Note

Given that IPCP only allows you to configure one IP address, no additional addresses should be added to the PPP interfaces.

Important

Whenever a PPP interface is involved, the IP address must be assigned to the PPP interface and NEVER to the base interface on which the protocol is negotiated.

You can also assign an **unnumbered** address to the PPP interface.

The following example shows you how to do this:

```
*config
Config>network ppp1
-- Generic PPP User Configuration --
ppp1 config>ip address 192.168.5.1 255.255.255.0
ppp1 config>exit
Config>
```

Bridge is another network protocol supported. To activate this functionality in the PPP link, simply define a bridge port in the corresponding PPP interface from the ASRT protocol configuration menu.

The following example shows you how to do this:

```
*config
Config>protocol asrt
-- ASRT Bridge user configuration --
ASRT config>bridge
ASRT config>port ppp1 2
ASRT config>exit
Config>
```

For additional information on the *bridge* functionality, see the Teldat Dm717-I Bridge manual.

2.3.8 Configuring PPP events

PPP interfaces have a specific global events menu. Use the **ppp** command in the event configuration menu to access the PPP event configuration menu:

```
Config>event
-- ELS Config --
ELS config>ppp
--  PPP Events Configuration  --
PPP Events config>
```

The options available in the PPP event configuration menu are:

```
PPP Events config>?
  disable    Disables PPP events
  enable     Enables PPP events
  list       List configuration
  mode       Configures the mode for PPP events viewing
  exit
PPP Events config>
```

ENABLE/DISABLE

This command enables/disables PPP events.

Syntax:

```
PPP Events config>enable | disable <options>
  all               All the PPP events
  authentication    PPP events related to authentication
  error             PPP events related to errors
  information       PPP events related to general information
  negotiation       PPP events related to negotiation
  packet            PPP events related to PPP packets
```

all	Enable/disable all PPP events.
authentication	Enable/disable authentication-related PPP events.
error	Enable/disable error-related PPP events.
information	Enable/disable general information-related PPP events.
negotiation	Enable/disable negotiation-related PPP events (both LCP and the different NCPs).
packet	Enable/disable PPP events related to PPP packet transmission and reception.

LIST

Displays configuration information about PPP events on the device, indicating which types of events are enabled and disabled. It also indicates the event display mode selected: user or expert.

```
PPP Events config>list
+------------------------------------------------------------------------+
+ EVENT Configuration                                                    +
+------------------------------------------------------------------------+
  PPP Authentication Events: Disabled
  PPP Error Events.........: Disabled
  PPP Information Events...: Disabled
  PPP Negotiation Events...: Disabled
  PPP Packet Events........: Disabled

  PPP Events Mode..........: User

PPP Events config>
```

MODE

Selects the PPP event display mode.

Syntax:

```
PPP Events config>mode <options>
  expert    Selects the expert mode for PPP events viewing
  user      Selects the user mode for PPP events viewing
```

expert	Selects *expert* mode when viewing PPP events. This mode allows you to display the events from the PPP interface with a high level of detail.
user	Selects *user* mode when viewing PPP events. This mode displays the main events from the PPP interface without going into much detail.

EXIT

Exits the PPP event configuration menu and returns to the general event configuration menu.

```
PPP Events config>exit
ELS config>
```

Chapter 3 PPP Configuration Parameters

3.1 PPP configuration parameters

To set PPP parameters for negotiation, authentication and various associated features, access the PPP parameter menu using the **ppp** command in the PPP interface configuration menu.

```
Config>network pppX
-- Generic PPP User Configuration --
pppX config>ppp
-- PPP Configuration --
pppX PPP config>
```

The following commands are available in the PPP parameter configuration menu:

```
pppX PPP config>?
  authentication       Set authentication parameters
  authorization        Set authorization parameters
  backup               Set backup parameters
  bcp                  Set BCP parameters
  ccp                  Set CCP parameters
  chap                 Set CHAP parameters
  clone-from           Clone configuration from interface/template
  dial-routing-pool    Enable dial-routing and set pool
  ip                   Set IP configuration
  ipcp                 Set IPCP negotiation parameters
  lcp                  Set PPP LCP parameters
  list                 Show PPP configuration
  multilink            Multilink PPP options
  ncp                  Set PPP NCP parameters
  no                   Negate a command or set its defaults
  pap                  Set PAP parameters
  radius               Set RADIUS authentication
  rip-no-dial          RIP packets do not trigger dialing
  exit
pppX PPP config>
```

Command	Function
? (HELP)	Displays the available commands or options.
AUTHENTICATION	Configures PPP link authentication parameters.
AUTHORIZATION	Configures PPP link authorization parameters.
BACKUP	Configures NCP backup parameters.
BCP	Configures BCP (Bridging Control Protocol) parameters.
CCP	Configures CCP (Compression Control Protocol) parameters.
CHAP	Configures CHAP (Challenge Handshake Authentication Protocol) parameters.
CLONE-FROM	Clones another PPP interface.
DIAL-ROUTING-POOL	Configures DIAL-ROUTING options.
IP	Configures IP (Internet Protocol) parameters.
IPCP	Configures IPCP (Internet Protocol Control Protocol) parameters.
LCP	Configures LCP (Link Control Protocol) parameters.
LIST	Displays PPP interface configuration parameters.
MULTILINK	Configures PPP Multilink parameters.
NCP	Configures options common to NCPs (Network Control Protocol).
NO	Sets the default value of a given option, disables parameters or deletes previously added configuration items.
PAP	Configures PAP (Password Authentication Protocol) parameters.
RADIUS	Enables authentication using the Radius protocol.
RIP-NO-DIAL	Prevents RIP packets from generating calls in switched interfaces.
EXIT	Exits the PPP profile configuration.

3.1.1 Configuring authentication parameters

This command allows you to configure the authentication options for the PPP link.

PPP link authentication can take place in both directions, i.e., the router can ask the remote end to authenticate itself and/or the remote end can request authentication from the router.

The following options are available with this command:

```
pppX PPP config>authentication <options>
  allowed-user           Allowed remote peer (permitted inbound user)
  chap                   CHAP authentication
  none                   Disable authentication
  pap                    PAP authentication
  sent-user              Set outbound user to authenticate itself to a remote peer
  multiple-sent-users    Enable multiple sent-user entries to be used sequentially
```

Command history:

Release	Modification
11.01.06	The option "multiple-sent-users" is available as of version 11.01.06.

3.1.1.1 Remote end authentication

To enable remote end authentication during the link negotiation phase, and control the devices allowed to establish a PPP session, you need to:

- Enable an authentication protocol (PAP/CHAP).
- Define the remote users that are allowed to set up the link (or enable RADIUS or the AAA feature).

The authentication protocols supported by the **Teldat Router** include *Password Authentication Protocol* (PAP) and *Challenge Authentication Protocol* (CHAP). By default, no authentication type is enabled.

Syntax:

Enabling remote endpoint authentication via PAP:

```
pppX PPP config>authentication pap [<listname>]
```

<listname>:	Authentication method list identifier (this is only available when AAA is enabled).

Enabling remote endpoint authentication via CHAP:

```
pppX PPP config>authentication chap [<listname>]
```

<listname>:	Authentication method list identifier (only available when AAA is enabled).

Disabling remote endpoint authentication (you can also use the no form of the command to do this).

```
pppX PPP config>authentication none
```

You use the **authentication allowed-user** command to define the remote users who are allowed to set up the PPP link.

Use the **empty-password** option to configure a zero-length password.

Syntax:

```
pppX PPP config>authentication allowed-user <user> {password <password>|empty-password}
```

If different PAP/CHAP passwords are used, then use:

```
pppX PPP config>authentication allowed-user <user> {password <pap-pwd>|empty-password} chap-password <chap-pwd>
```

Example:

Authenticating the remote endpoint using CHAP and giving multiple users permission to set up the PPP link.

```
ppp1 PPP config>authentication chap
ppp1 PPP config>authentication allowed-user user1 password pass1
ppp1 PPP config>authentication allowed-user user2 password pass2
ppp1 PPP config>authentication allowed-user user3 password pass3
ppp1 PPP config>
```

3.1.1.2 Local end authentication

The remote endpoint can request authentication from the router.

You use the **authentication sent-user** command to define the user/password that the router uses to identify itself when the remote end requests authentication.

Use the **empty-password** option to configure a zero-length password.

Syntax:

```
pppX PPP config>authentication sent-user <user> {password <password>|empty-password}
```

If different PAP/CHAP passwords are used, then use:

```
pppX PPP config>authentication sent-user <user> {password <pap-pwd>|empty-password}  chap-password <chap-pwd>}
```

Example:

```
ppp1 PPP config>authentication sent-user john password doe
ppp1 PPP config>list

[...]
AUTHENTICATION:
  CHAP authentication using local database
  Sent user (local): john
  Allowed user (remote peer): user1
  Allowed user (remote peer): user2
  Allowed user (remote peer): user3
[...]
ppp1 PPP config>
```

Note

When the router is acting as a remote end authenticator, it does not send a name in the "Name" field of a CHAP challenge packet unless the **authentication sent-user** command has been configured. When this command is configured, the local user name is sent.

When several users/passwords must be configured (configurations that work in several scenarios), use the **authentication multiple-sent-users** command.

The different user/password entries are used sequentially until a successful authentication is achieved.

Syntax:

```
pppX PPP config>authentication multiple-sent-users
```

3.1.2 Configuring authorization parameters

This command associates a list of network authorization methods defined using the AAA feature. This allows the PPP link to apply the authorization methods from the list when authorization is required during link establishment.

Syntax:

```
<pppN> PPP config>authorization network <listname>
```

<listname>:	The name of a list of authorization methods.

Example:

```
ppp1 PPP config>authorization network AuthorNet
ppp1 PPP config>
```

This example configures the device to use the AuthorNet method list when authorization is required during link establishment.

Method lists can only be applied if the AAA feature is enabled. Therefore, once you have configured AAA, you must enable it in order to apply the lists to the different services. Information on how to set up the AAA feature is provided in the Teldat Dm800-I AAA Feature manual.

3.1.3 Configuring backup parameters

Use the **backup** command to configure the backup parameters.

Syntax:

```
pppX PPP config>backup <options>
```

The backup parameters are only meaningful when the PPP interface is configured as the main interface within the WAN Reroute feature and the associated base interface is a switch interface. For additional information on the WAN Reroute backup feature, see the Teldat Dm727-I Backup WAN Reroute manual.

The options available are:

```
ppp1 PPP config>backup ?
  dial-test-time         Time between dial recovery tests (min)
  maximum-backup-time    Maximum backup time (min)
  ncp-timeout            NCP timeout before backup (sec)
  retries                Call retries before backup
```

dial-test-time	Time between test calls. When the backup interface is enabled, periodic calls are made to reestablish the connection through the main (switched) interface. If set to 0, no test calls will be made to the main interface. The default value of this parameter is 0 (test calls disabled).
maximum-backup-time	Maximum backup time. The main interface is placed into an UP state after this time and an attempt is made to reconnect through the interface. If set to 0, the backup state is maintained until such time as the secondary link goes down. The default value of this parameter is 0.
ncp-timeout	Time limit for NCP negotiation. If, after this time, the NCP has not been negotiated, the PPP interface will be placed into a DOWN state. If the WAN Reroute feature is enabled, the secondary or backup interface will be activated. The default value of this parameter is 30 seconds.
retries	Number of failed calls attempted through the main interface before switching to backup. Default is 5 retries.

Example:

```
ppp1 PPP config>backup retries 3
ppp1 PPP config>backup ncp-timeout 1m
ppp1 PPP config>backup maximum-backup-time 60:00
ppp1 PPP config>backup dial-test-time 00:02
ppp1 PPP config>list
[...]
BACKUP:
  Call retries 3, NCP timeout 1m0s
  Time between test calls 2m, Max backup time 60h0m
[...]
ppp1 PPP config>
```

Call retries	Number of call attempts before switching to backup.
NCP timeout	NCP time limit before switching to backup.
Time between test calls	Time between call retries.
Max backup time	Maximum backup time.

3.1.4 Configuring BCP parameters

Use the **bcp** command to configure the bridge control protocol (BCP) parameters.

Syntax:

```
pppX PPP config>bcp <options>
  tinygram-compress    Tinygram compression
```

tinygram-compress	Enables Tinygram compression. This allows you to remove the padding at the end of short frames encapsulated with Ethernet. This is disabled by default. It is only recommended for use on very slow lines (see BCP RFC).

Example:

```
ppp1 PPP config>bcp tinygram-compress
```

```
ppp1 PPP config>list
[...]
BCP:
  Tinygram Compression enabled
[...]
ppp1 PPP config>
```

3.1.5 Configuring CCP parameters

Use the **ccp** command to configure the *compression control protocol* (CCP) parameters.

Syntax:

```
pppX PPP config>ccp <options>
```

The options available are:

```
pppX PPP config>ccp ?
  enable     Enable CCP negotiation
  lzs-dcp    LZS compression algorithm
  none       No compression
  stac       STAC compression algorithm
```

enable	Enables CCP negotiation. CCP is disabled by default. Use the **no ccp enable** command to disable CCP negotiation
lzs-dcp	Indicates that the LZS-DCP (RFC 1967) compression algorithm will be used.
none	Indicates that you do not want to receive compressed information from the remote end. This is the default.
stac	Indicates that the STAC Electronics LZS (RFC 1974) compression algorithm will be used.

3.1.5.1 LZS-DCP

LZS-DCP compression algorithm configuration (RFC 1967) has the following syntax:

```
pppX PPP config>ccp lzs-dcp [
                    <check-mode> [process-uncompressed] history-count {0|1}]
```

<check-mode>	Check mode. Valid values are: *none* (no check), *lcb, seg* and *seg-lcb* (a combination of *seg* and *lcb*). The default value for the LZS-DCP algorithm is *seg-lcb*.
process-uncompressed	Indicates that any uncompressed packets received are processed by the CCP to update their status (non-compressed frame processing mode). If this option is disabled, then uncompressed packets are not taken into account when updating the CCP status. This parameter is disabled by default.
history-count	Indicates whether records can be maintained for decompression. That is, packet decompression takes into account previously processed data sent in previous packets. Valid values are: 0 (do not keep records) and 1 (keep one record). The default value of this parameter is 1 (keep one record).

Example:

```
ppp1 PPP config>ccp enable
ppp1 PPP config>ccp lzs-dcp seq-lcb history-count 1
ppp1 PPP config>list
[...]
CCP:
  CCP negotiation enabled
  LZS-DCP checkmode Sequence+LCB, process None, history-count 1
[...]
ppp1 PPP config>
```

Note

Not all **<check-mode>**, **process-uncompressed** and **history-count** combinations are allowed. The device applies certain restrictions to avoid illegal combinations.

3.1.5.2 STAC

The configuration of the STAC Electronics LZS compression algorithm (RFC 1974) has the following syntax:

```
pppX PPP config>ccp stac [<check-mode> history-count {0|1}]
```

<check-mode>	Check mode. Valid values are: *none* (no check), *lcb, seg* and *extended*. The default value is *seg*.
history-count	Indicates whether records can be maintained for decompression. That is, packet decompression takes into account previously processed data sent in previous packets. Valid values are: 0 (do not keep records) and 1 (keep one record). The default value of this parameter is 1 (keep one record).

Example:

```
ppp1 PPP config>ccp enable
ppp1 PPP config>ccp stac lcb history-count 1
ppp1 PPP config>list
[...]
CCP:
  CCP negotiation enabled
  STAC, checkmode LCB, history-count 1
[...]
ppp1 PPP config>
```

Note

Not all **<check-mode>** and **history-count** combinations are allowed. The device applies certain restrictions to avoid illegal combinations.

3.1.6 Configuring CHAP parameters

Use the **chap** command to configure the Challenge Handshake Authentication Protocol (CHAP) parameters.

Syntax:

```
pppX PPP config>chap <options>
  refuse    Refuse to authenticate using CHAP
```

The only thing you can do here is to specify that the router must refuse to answer CHAP authentication challenges from the remote end. This means that, if the remote end were to request CHAP authentication during LCP negotiation, it would be refused.

3.1.7 Configuring CLONE-FROM

Sometimes you may want to configure several identical PPP interfaces (for example, in the case of an LNS-L2TP server with several tunnels to be served, each requiring a corresponding PPP interface). To copy the configuration from another PPP interface, you can use the **clone-from** command.

Syntax:

```
pppX PPP config>clone-from interface <interface>
```

If you do configure this parameter, you will not be able to make any other changes to the interface's PPP configuration (except to undo cloning). Any changes must be made in the original interface (i.e., the one being cloned).

3.1.8 Configuring DIAL-ROUTING-POOL

The **dial-routing-pool** command configures the PPP interface as a member of a dial-routing interface pool. For additional information, see the Teldat Dm744-I Dial-Routing manual.

Syntax:

```
pppX PPP config>dial-routing-pool <pool name>
```

Example:

```
ppp1 PPP config>dial-routing-pool my-dr-pool
ppp1 PPP config>list
```

```
[...]
Dial Routing Pool Name: my-dr-pool
[...]
ppp1 PPP config>
```

3.1.9 Configuring IP parameters

Use the **ip** command to configure the Internet Protocol (IP) parameters.

Syntax:

```
pppX PPP config>ip <options>
  header-compression    Set IP Header Compression (IPHC) options
```

3.1.9.1 IP HEADER-COMPRESSION

This command lets you enable and configure IP header compression using the IPHC protocol.

IPHC compression is another negotiable option of the IPCP protocol and, to work properly, it must be enabled at both ends of the PPP link. IPHC can compress both TCP and UDP headers (and RTP).

To configure IPHC, you need to perform the following tasks:

- Enable IPHC.
- Set the header-compression options and operating parameters.

Syntax:

```
pppX PPP config>ip header-compression <options>
  disable-feedback          Disable context status mechanism
  iphc-compatible-format    Enable IPHC using compatible format
  iphc-ietf-format          Enable IPHC using IETF format
  max-header                Largest header size that may be compressed
  max-period                Maximum interval between full headers
  max-time                  Maximum time interval between full headers
  non-tcp                   Non-TCP specific parameters
  tcp                       TCP specific parameters
  <cr>
```

Options to enable IPHC

iphc-ietf-format	Enables the IPHC header-compression negotiation (as per RFC 2507).
<cr>	Enables the IPHC header-compression negotiation in the same way as *iphc-ietf-format*.
iphc-compatible-format	Enables the IPHC header-compression negotiation so that it is compatible with devices that do not negotiate *tcp-space* and *nontcp-space* parameters (as per RFC 2507).

Options and operating parameters

disable-feedback	Disables *CONTEXT_STATE* feedback messages.
max-header	Specifies the maximum size of a compressed IP header.
max-period	Maximum interval between uncompressed headers. No more compressed headers than those specified by the *max-period* value should be sent between uncompressed headers.
max-time	Maximum time interval between uncompressed headers. Compressed headers must be sent before *max-time* (in seconds) after sending an uncompressed header.
non-tcp	Allows you to configure non-TCP-specific options: - *connections*, maximum number of compressed connections. - *disable-refresh*, disable periodic context refresh. - *disable-udp-checksum*, disable UDP checksum. - *rtp*, RTP parameter configuration.
tcp	Allows you to configure TCP-specific options:

- *connections*, maximum number of compressed connections.

- *disable-refresh*, disable periodic context refresh.

3.1.10 Configuring IPCP parameters

You use the **ipcp** command to configure Internet Protocol Control Protocol (IPCP) parameters.

Syntax:

```
pppX PPP config>ipcp <options>
```

The options available include:

```
pppX PPP config>ipcp ?
  distance        Set the administrative distance for routes learnt through
                  ipcp
  dns             DNS servers
  local           Local IPCP parameters
  nbns            NetBIOS name servers
  peer-route      Route to remote peer
  remote          Remote IPCP parameters
  van-jacobson    Van-Jacobson compression
pppX PPP config>
```

distance	Specifies the administrative distance of the path to the remote end learned during IPCP negotiation. The default value is 60.
dns	Options in the primary and secondary DNS server negotiation. - *<a.b.c.d>:* Specifies the primary and secondary DNS servers sent to the remote end if requested in the negotiation. - *request:* Indicates that the remote end has requested the primary and secondary DNS servers.
local	Configures the local end's IP address. This value can be **fixed**, if you want the local end's IP address to be the one the user configured for the PPP interface, **negotiable**, if the remote end is allowed to decide which IP address to use for the local end, or **assigned**, if you want the remote end to assign the local end's IP address. Default is fixed.
nbns	Specifies the primary and secondary NetBIOS servers that are sent to the remote end if requested during negotiation.
peer-route	Specifies whether the device must add a host path to the remote end through the PPP interface once the IPCP negotiation is complete. This option is enabled by default. If you do not want to add this route to the remote end, you should use the **no ipcp peer-route** command.
remote	Configures the remote end's IP address. This value can be either **fixed**, if you want to assign a fixed IP address to the remote end, **pool, and a name**, if you want to choose one from a specific address pool, **same-subnet**, if you want to assign a local subnet IP address to the remote end, or **reject**, if you wish to reject the remote address requests. The remote end is not assigned an IP address by default (unless it requests one, in which case it will be given one from the device's address pool).
van-jacobson	Enables the Van Jacobson header-compression negotiation at the local end. This parameter is disabled by default. IPHC must be disabled before Van-Jacobson compression can be enabled.

Example:

```
ppp1 PPP config>ipcp local address fixed
ppp1 PPP config>ipcp remote address pool TELDAT
ppp1 PPP config>ipcp van-jacobson
ppp1 PPP config>list
[...]
```

```
IPCP:
  Local IP address fixed
  Remote IP address pool TELDAT
  Van Jacobson compression enabled
  Route to remote peer enabled
```

```
  DNS servers none
  NBNS servers none
[...]
ppp1 PPP config>
```

Command history:

Release	Modification
10.09.27	The **reject** option for the **ipcp remote address** command was introduced as of version 10.09.27.
11.00.06	The **reject** option for the **ipcp remote address** command was introduced as of version 11.00.06.
11.01.02	The **reject** option for the **ipcp remote address** command was introduced as of version 11.01.02.

3.1.11 Configuring LCP parameters

Use the **lcp** command to configure the Link Control Protocol (LCP) parameters.

Syntax:

```
pppX PPP config>lcp <options>
  conf-req             Configure-Request tries
  conf-nak             Configure-Nak tries
  term-req             Terminate-Request tries
  timeout-retry        Time between tries
  echo-req             Number of Echo-Request
  negotiation-delay    Delay before starting LCP negotiation
```

conf-req — Specifies the number of times an *LCP configure-request* message is sent to establish the PPP link. The valid range is 1 to 100. Default is 10.

conf-nak — Maximum number of *configure-nak* rejections sent during link establishment before the link is closed because no matching configuration exists between the two ends. The valid range is 1 to 100. Default is 10.

term-req — Number of times a *terminate-request* frame is sent to close a link in an orderly fashion without receiving a *terminate-ack* response. The valid range is 1 to 20. Default is 10.

timeout-retry — Number of seconds to wait for an appropriate response to *LCP negotiation* packets before retrying. The valid range is 1 to 30 seconds. Default is 3 seconds.

echo-req — Number of times an *echo-request* frame is sent without receiving a response from the remote end (*echo-reply*) before considering the link lost. Valid numerical values range from 2 to 30 and you can use the **off** option to disable *echo-request* frame transmissions. The default is 6. By default, *echo-request* frames are sent every 10 seconds. You can configure a different time interval after the number of echo-request packets. This interval can range from 1 to 65535 seconds.

Note

If you disable the *LCP echo-request* feature, the PPP interface will remain permanently active as long as the base interface stays up. (CAUTION: some base interfaces are always active).

negotiation-delay — Amount of time the device waits from the moment the link is considered active to the moment it begins the LCP negotiation phase. Acceptable values range from 0 (negotiation starts immediately) to 65535 seconds (18h12m15s). Default is 0.

Example:

```
ppp1 PPP config>lcp conf-req 4
ppp1 PPP config>lcp conf-nak 4
ppp1 PPP config>lcp term-req 2
ppp1 PPP config>lcp timeout-retry 5
ppp1 PPP config>lcp echo-req 3 5
ppp1 PPP config>lcp negotiation-delay 2
ppp1 PPP config>list
[...]
LCP parameters:
  Conf-Req 4, Conf-Nak 4, Term-Req 2, Timeout 5s
```

```
  Echo-Req 3, time interval 5s
  Delay 2s before beginning the LCP negotiation
[...]
ppp1 PPP config>
```

3.1.12 Configuring PPP Multilink

Use the **multilink** command to enable and configure the PPP Multilink (MP) parameters.

Syntax:

```
pppX PPP config>multilink <options>
  activation           Options to activate new links
  call-out             Add new links just for outgoing calls
  deactivation         Options to deactivate links
  enable               Enable Multilink PPP negotiation
  endpoint             Endpoint discriminator
  fragmentation        Forced MP fragmentation size
  interleave           Enable real-time interleaving
  link-reorders        MP fragments may be received out-of-order within a link
  links-per-bundle     Maximum number of links per bundle
  mrru                 MRRU size in bytes
  multiclass           Enable Multiclass Multilink
  queue                MP queue parameters
  ssn-header-format    Short sequence number header format
  traffic-load         Traffic load calculation
  interoperable-mtu    Decrease the bundle MTU by the protocol field 2-bytes
  mtu                  Forced MTU size in bytes
```

3.1.12.1 multilink activation

Configures link activation options in a multilink bundle. The mechanism is controlled using two parameters: the activation interval and the activation threshold. If, during the activation interval, the average occupancy of the PPP Multilink exceeds the activation threshold, an additional PPP link is brought into the multilink bundle.

Syntax:

```
pppX PPP config>multilink activation <options>
  interval     Measurement interval
  threshold    Activity threshold in percentage
```

interval	Activation interval for new links. This value can range from 4 to 1800 seconds (30 minutes). Default is 120 seconds (2 minutes).
threshold	Activation threshold (%). Default is 90 %.

Note

Only switched links can be dynamically enabled/disabled in a PPP Multilink bundle.

3.1.12.2 multilink call out

When you enable this option, new PPP links can only be added if the first (switched) link originated with an outgoing call. It is disabled by default.

Syntax:

```
pppX PPP config>multilink call-out
```

3.1.12.3 multilink deactivation

Configures link deactivation options in a multilink bundle. The mechanism is controlled using two parameters: the deactivation interval and the deactivation threshold. If, during the deactivation interval, the average occupancy of the total capacity of the PPP Multilink falls below the deactivation threshold, one of the PPP links in the multilink bundle is dropped.

Syntax:

```
pppX PPP config>multilink deactivation <options>
  interval     Measurement interval
```

```
  threshold    Activity threshold in percentage
```

interval	Link deactivation interval. The valid range is from 4 to 1800 seconds (30 minutes). Default is 300 seconds (5 minutes).
threshold	Deactivation threshold (%). Default is 50 %.

Note

Only switched links can be dynamically activated/deactivated in a PPP Multilink bundle.

3.1.12.4 multilink enable

This enables the PPP Multilink (MP) negotiation with the remote end.

Syntax:

```
pppX PPP config>multilink enable
```

3.1.12.5 multilink endpoint

This specifies the *Endpoint Discriminator* to use in the multilink bundle.

Syntax:

```
pppX PPP config>multilink endpoint <options>
  ethernet    Ethernet MAC address
  ip          IP address
  null        Null endpoint discriminator class
  psndn       Public Switched Network Directory Number
```

ethernet	Ethernet MAC address (the ethernet0/0 interface MAC address is used by default).
ip	IP address.
null	Null class, this has no associated value.
psndn	*Public Switched Network Directory Number.* E.164 address (packet-switched public network number).

RFC 1990 defines two other types of *Endpoint-Discriminator*: *locally-assigned* and*magic-number* (ranging from 4 to 20 bytes). You are not allowed to configure these values because their use is not recommended.

The default value is *null*.

3.1.12.6 multilink fragmentation

Configures the forced fragmentation size of PPP Multilink frames. This parameter is usually used when you want to force a smaller frame size than the value negotiated in the MRRU negotiation.

Valid fragmentation values range from 0 (fragmentation disabled) to 1500 bytes. By default, forced fragmentation is disabled.

Syntax:

```
pppX PPP config>multilink fragmentation <forced-fragmentation-size>
```

To disable this option, you can either set the value to 0 or use the **no multilink fragmentation** command.

3.1.12.7 multilink interleave

This command enables the interleaving of packets that have been classified as real-time class traffic using the BRS functionality. By enabling the interleaving of real-time class traffic, packets from that class are sent by the various bundle links without the PPP Multilink header, thus minimizing delay. For additional information on real-time traffic classification, please see the Teldat Dm715-I – BRS manual. This functionality is disabled with the **no multilink interleave** command. This command has no effect if the **multilink multiclass** functionality is configured.

Syntax:

```
pppX PPP config>multilink interleave
```

Note

By not including the PPP Multilink header, the receiver lacks the information needed to reorder the real-time class packets from the different links of the multilink bundle.

3.1.12.8 multilink link reorders

This option causes the router to reorder incoming packets for each link individually instead of directly discarding them (the rule is that within the same multilink bundle link, fragments must arrive in increasing monotonic order). This option is disabled by default.

Syntax:

```
pppX PPP config>multilink link-reorders
```

3.1.12.9 multilink links-per-bundle

Maximum number of PPP links that can be associated with a PPP Multilink bundle.Valid parameters range from 1 to 31. The default value is 2.

Syntax:

```
pppX PPP config>multilink links-per-bundle <num-max-links-per-bundle>
```

3.1.12.10 multilink mrru

Maximum PPP Multilink frame size permitted at reception (*Max-Receive-Reconstructed-Unit*). Valid parameters range from 1 to 4088 bytes. Default is 1524 bytes.

Syntax:

```
pppX PPP config>multilink mrru <mrru-size>
```

3.1.12.11 multilink multiclass

Enables the *multilink header format* option to be included in the LCP negotiation with the remote end. If the negotiation is successful, real-time traffic is encapsulated with Multilink Protocol headers rather than being sent in interleaving mode.

Syntax:

```
pppX PPP config>multilink multiclass
```

If you enable this option, PPP Multilink traffic will be classified into two classes: class **0** for normal traffic and class **1** for real-time traffic.

3.1.12.12 multilink queue

Configures parameters relating to the PPP Multilink fragment reception queues.

Syntax:

```
pppX PPP config>multilink queue <options>
  items-per-link    Queue items per link
  max-items         Maximum items within a bundle
```

items-per-link	Number of fragments saved per bundle link. Valid parameters range from 2 to 32. Default is 5.
max-items	Maximum number of fragments saved for reconstruction before being considered invalid and discarded. Valid parameters range from 8 to 128, or 0 (unlimited). Default is 0.

3.1.12.13 multilink ssn-header-format

The Multilink Short Sequence Number Header Format allows devices to negotiate a shorter sequence number field for PPP Multilink frames. This parameter is disabled by default.

Syntax:

```
pppX PPP config>multilink ssn-header-format
```

3.1.12.14 multilink traffic-load

This indicates the direction of traffic taken into account for the purpose of calculating average channel load. There are three possible values: *inbound* (from the external network to the device), *outbound* (from the device to the external network) and *either* (both types of traffic).

In normal circumstances of access to an external network like the Internet, where traffic is mainly incoming, we recommend setting the *inbound* value. The default value for this parameter is *either*.

Syntax:

```
pppX PPP config>multilink traffic-load {inbound | outbound | either}
```

3.1.12.15 multilink interoperable-mtu

This allows the Teldat router's MTU size in the MP link to be reduced by two bytes in relation to the MRRU advertised by the remote end.

This parameter causes the maximum number of information bytes in packets reassembled at the remote end to be reduced by two bytes, thus allowing interoperability with other manufacturer's MP implementations.

Syntax:

```
pppX PPP config>multilink interoperable-mtu
```

3.1.12.16 multilink mtu

Sets the MTU value of the PPP Multilink interface. This represents the maximum number of layer 3 information bytes that the router is going to deliver to the PPP Multilink interface. At the remote end, it corresponds to the maximum number of information bytes in reassembled packets.

Its value must be between 0 and 2040 bytes. Default is 0, meaning the MTU value of the PPP Multilink link is automatically calculated from the MRRU value advertised by the remote end.

If the remote end does not send an MRRU value, the normal MTU value is 1524 bytes.

Syntax:

```
pppX PPP config>multilink mtu <mtu-size>
  <0..2040>    MTU
```

Multilink PPP configuration example

```
ppp1 PPP config>multilink enable
ppp1 PPP config>multilink activation interval 60s
ppp1 PPP config>multilink activation threshold 33
ppp1 PPP config>multilink deactivation interval 600s
ppp1 PPP config>multilink deactivation threshold 33
ppp1 PPP config>multilink traffic-load inbound
ppp1 PPP config>list
[...]
Multilink PPP parameters:
  Multilink PPP negotiation enabled
  MRRU 1524, SSN Header disabled, Max links-per-bundle 2
  Endpoint [MAC address] 00-A0-26-5C-7C-2A (ethernet0/0)
  Activation 1m0s/33%, Deactivation 10m0s/33%, Load: Inbound
  Fragmentation 0 (disabled), Link aggregation unrestricted
  Real-time interleaving disabled
  Queue items per-link 5, Max queue items per-bundle 0 (unlimited)
[...]
ppp1 PPP config>
```

3.1.13 Configuring NCP parameters

Use the **ncp** command to configure the Network Control Protocol (NCP) parameters.

These parameters apply to the negotiation of all supported NCPs (IPCP, BCP and CCP).

Syntax:

```
pppX PPP config>ncp <options>
```

```
  conf-req         Configure-Request tries
  conf-nak         Configure-Nak tries
  term-req         Terminate-Request tries
  timeout-retry    Time between tries
```

conf-req	Indicates how many times an *NCP configure-request* will be sent to try to establish the network protocol. Valid parameters range from 1 to 100. Default is 10.
conf-nak	Maximum number of *configure-nak* rejections sent during link establishment before the link is closed because no matching configuration exists between the two ends. Valid parameters range from 1 to 100. Default is 10.
term-req	Number of times a *terminate-request* frame is sent to finalize the network protocol in an orderly fashion without receiving a *terminate-ack* response. Valid parameters range from 1 to 20. Default is 10.
timeout-retry	Number of seconds to wait for an appropriate response to *NCP negotiation* frames before retrying. Valid parameters range from 1 to 30 seconds. Default is 3 seconds.

Example:

```
ppp1 PPP config>ncp conf-req 5
ppp1 PPP config>ncp term-req 3
ppp1 PPP config>ncp timeout-retry 2s
ppp1 PPP config>list
[...]
NCP parameters:
  Conf-Req 5, Conf-Nak 10, Term-Req 3, Timeout 2s
[...]
ppp1 PPP config>
```

3.1.14 Configuring PAP parameters

Use the **PAP** command to configure Password Authentication Protocol (PAP) parameters.

Syntax:

```
ppp1 PPP config>pap <options>
  refuse    Refuse to authenticate using PAP
```

The only thing you can do here is to specify that the router must refuse to answer PAP authentication challenges from the remote end. So, if the remote end were to request PAP authentication in LCP negotiation, it would be refused.

3.1.15 Configuring RADIUS parameters

Use the **radius** command to configure Remote Access Dial-In User Server (RADIUS) parameters.

Syntax:

```
pppX PPP config>radius <options>
  enable    Enable RADIUS authentication
```

All you can do here is enable the RADIUS protocol (which is deactivated by default). If enabled, validation of the remote end authentication and the PPP link configuration are accomplished using a RADIUS server containing the relevant information, rather than using the router's allowed user list. In order to enable this feature, you must first configure the RADIUS protocol in the router and globally enable it through the associated configuration environment. For additional information on how to configure the RADIUS protocol, see the Teldat Dm733-I manual. Default is disabled.

To disable the RADIUS feature in the PPP interface, use the **no radius enable** command.

Example:

```
ppp1 PPP config>radius enable
ppp1 PPP config>list
[...]
RADIUS authentication enabled
[...]
ppp1 PPP config>
```

3.1.16 Configuring Rip-No-Dial

The **rip-no-dial** command prevents RIP packets from causing and maintaining calls on switched interfaces.

When configuring a *non permanent* base interface to make outgoing calls, the following problem occurs if the RIP protocol is also enabled on the PPP interface that is configured on top of the base interface: RIP starts sending IP packets over the PPP interface and this traffic causes a call to be created. Furthermore, and because RIP sends messages periodically, the call is never released due to a lack of traffic. The **rip-no-dial** option has been implemented to correct this problem in cases where dynamic routing is desired. If enabled, RIP packets do not generate a call and are not taken into account when maintaining a previously established call (i.e., if the only IP traffic existing in the link is due to RIP, the call is dropped when the no-traffic timer expires). Default is disabled.

Syntax:

```
pppX PPP config>rip-no-dial
```

Use the **no rip-no-dial** command to disable RIP-NO-DIAL.

Example:

```
ppp1 PPP config>rip-no-dial
ppp1 PPP config>list
[...]
RIP-no-dial enabled
[...]
ppp1 PPP config>
```

3.1.17 Viewing configured PPP parameters

The **list** command displays all PPP interface configuration parameters.

Syntax:

```
pppX PPP config>list
```

Example:

```
ppp1 PPP config>list
AUTHENTICATION:
  PAP authentication using local database
  Sent user (local): localuser
  Allowed user (remote peer): remoteuser
  Allowed user (remote peer): alternativeremote
BACKUP:
  Call retries 5, NCP timeout 30s
  Time between test calls 0m, Max backup time 0m
BCP:
  Tinygram Compression disabled
CCP:
  CCP negotiation disabled
  No compression algorithm
IPCP:
  Local IP address assigned
  Remote IP address none (unspecified)
  Van Jacobson compression disabled
  Route to remote peer enabled
  DNS servers none
  NBNS servers none
IPHC:
  IPHC negotiation enabled (IETF format), feedback on
  MaxPeriod 256, MaxTime 5, MaxHeader 168
  NonTCP connections 16, refresh on, UDP checksum enabled, no RTP-list
  TCP connections 16, refresh on
LCP parameters:
  Conf-Req 10, Conf-Nak 10, Term-Req 10, Timeout 3s
  Echo-Req 6, time interval 10s
  Delay 0s before beginning the LCP negotiation
Multilink PPP parameters:
  Multilink PPP negotiation enabled
```

```
  MRRU 1524, SSN Header disabled, Max links-per-bundle 2
  Endpoint [MAC address] 00-A0-26-5C-7C-2A (ethernet0/0)
  Activation 2m0s/90%, Deactivation 5m0s/50%, Load: Inbound & Outbound
  Fragmentation 0 (disabled), Link aggregation unrestricted
  Real-time interleaving disabled
NCP parameters:
  Conf-Req 10, Conf-Nak 10, Term-Req 10, Timeout 3s
No member of any Dial Routing Pool
RADIUS authentication disabled
RIP-no-dial disabled
L2TP (LAC) is disabled

ppp1 PPP config>
```

Chapter 4 PPP Global Profile Configuration

Command history:

Release	Modification
11.00.03	This command is obsolete as of version 11.00.03. The PPP global profile menu is no longer supported. See Section 2.3.4 for information about the new configuration menu.

4.1 Configuring PPP global profiles

Global profiles are configured in the same way in all PPP interfaces. To access the PPP global profile configuration menu, use the **global-profiles ppp** command in the general configuration menu:

```
Config>global-profiles ppp
-- PPP Profiles Configuration --
PPP Profiles config>
```

The PPP profile configuration menu options are as follows:

```
PPP Profiles config>?
  description    User descriptions
  lcp-options    LCP per-interface options profile
  list           Display PPP profiles configuration
  no             Negate a command or set its defaults
  exit
PPP Profiles config>
```

Command	Function
? (HELP)	Displays the available commands or options.
DESCRIPTION	Adds a description to the PPP global profile configuration.
LCP-OPTIONS	Configures an LCP option profile (per interface).
LIST	Displays the configured profiles.
NO	Removes a preconfigured profile.
EXIT	Exits the PPP profile configuration.

4.1.1 Configuring LCP option profiles

Use the **lcp-options** command to configure a Link Control Protocol (LCP) option profile dependent on the base interface.

Syntax:

```
PPP Profiles config>lcp-options <interface base> <options>
  accm             Asynchronous Control Character Map
  acfc             Address Control Field Compression
  default          Set LCP-options profile default configuration
  magic-number     Magic number
  mru              Maximum Receive Unit
  no               Negate a command or set its defaults
  pfc              Protocol Field Compression
```

The LCP option profile is associated with the base interface.

To remove an LCP option profile, use the **no lcp-options <base interface>** command.

accm	Configures the mask to be applied if transmission transparency is used on control characters in the case of asynchronous PPP. When these characters are included within the frame, they do not activate flow control processes in the modems/adapters used for the connection. The default value is 0xFFFFFFFF (hexadecimal). Typical negotiated values are 0x00000000 for links that can handle any data, and 0x000A0000 for links with XON/XOFF software flow control.
acfc	Enables *Address Control Field Compression* negotiation. This allows devices to receive PPP HDLC frames without the *Address* and *Control* fields after the link has been established. Default is disabled. We recommend enabling *acfc* on asynchronous interfaces.

default	Creates an LCP option profile with the default parameters or restores the default parameters of an existing profile.
magic-number	Enables the *magic-number* option when establishing the link. Default is enabled. This helps detecting whether the link is in a looped-back condition by transmitting a random number between the two ends based on the system clock and the number of times the clock has restarted.
mru	Maximum PPP frame size accepted by the receiving end. Valid parameters range from 576 to 4096 bytes. Default is 1500 bytes.
pfc	Enables *Protocol Field Compression* negotiation. This gives devices the ability to receive the protocol field of PPP frames in compressed format, provided they are not of control protocols (LCP, IPCP etc). This option is disabled by default. We recommend enabling *pfc* on asynchronous interfaces.

Example:

```
PPP Profiles config>lcp-options subatm1 default
PPP Profiles config>lcp-options subatm1 mru 1524
PPP Profiles config>lcp-options bri0/0 default
PPP Profiles config>lcp-options bri0/0 mru 1524
PPP Profiles config>lcp-options bri0/0 pfc
PPP Profiles config>lcp-options serial0/0 default
PPP Profiles config>lcp-options serial0/0 accm 00000000
PPP Profiles config>lcp-options serial0/0 acfc
PPP Profiles config>lcp-options serial0/0 pfc
PPP Profiles config>list lcp-options
+------------------------------------------------------------------------+
+ LCP OPTIONS Profiles                                                   +
+------------------------------------------------------------------------+
***  Base Interface: atm0/0.1 ***
  Interface MRU (bytes)              : 1524
  Magic Number                       : YES
  Asynchronous Control Character Map : ffffffff
  Protocol Field Compression         : NO
  Address Control Field Compression  : NO

***  Base Interface: bri0/0 ***
  Interface MRU (bytes)              : 1524
  Magic Number                       : YES
  Asynchronous Control Character Map : ffffffff
  Protocol Field Compression         : YES
  Address Control Field Compression  : NO

***  Base Interface: serial0/0 ***
  Interface MRU (bytes)              : 1500
  Magic Number                       : YES
  Asynchronous Control Character Map : 00000000
  Protocol Field Compression         : YES
  Address Control Field Compression  : YES

PPP Profiles config>show menu
; Showing Menu Configuration for access-level 15 ...

   lcp-options atm0/0.1 default
   lcp-options atm0/0.1 mru 1524
;
   lcp-options bri0/0 default
   lcp-options bri0/0 mru 1524
   lcp-options bri0/0 pfc
;
   lcp-options serial0/0 default
   lcp-options serial0/0 acfc
   lcp-options serial0/0 pfc
   lcp-options serial0/0 accm 0
;
PPP Profiles config>
```

Chapter 5 PPP Options Per Interface

Command history:

Release	Modification
11.00.03	This command is available as of version 11.00.03.

5.1 Configuring PPP options per interface

PPP parameter configuration is not shared between different physical interfaces. This is to allow different configurations between interfaces working together as a PPP multilink.

To enter the configuration menu, first enter the interface menu:

```
Config>network XXXX
-- XXXX Interface User Configuration --
XXXX config>
```

The PPP parameter menu is as follows:

```
XXXX config>ppp lcp-options ?
  accm               Asynchronous Control Character Map
  acfc               Address Control Field Compression
  no-magic-number    Disable Magic Number
  mru                Maximum Receive Unit
  pfc                Protocol Field Compression
XXXX config>
```

accm	Configures the mask to apply if transmission transparency is used on control characters in the case of asynchronous PPP. When these characters are included within the frame, they do not activate flow control processes in the modems/adapters used for the connection. Default is 0xFFFFFFFF (hexadecimal). Typical negotiated values are 0x00000000 for links that can handle any data, and 0x000A0000 for links with XON/XOFF software flow control.
acfc	Enables *Address Control Field Compression* negotiation. This allows devices to receive PPP HDLC frames without *Address* and *Control* fields after the link is established. Default is disabled. We recommend enabling *acfc* on asynchronous interfaces.
no-magic-number	Disables the *magic-number* option when establishing the link. Default is enabled. This helps detecting whether the link is in a looped-back condition by transmitting a random number between the two ends based on the system clock and the number of times the clock has restarted
mru	Maximum PPP frame size accepted by the receiving end. Valid parameters range from 576 to 4096 bytes. Default is 1500 bytes.
pfc	Enables *Protocol Field Compression* negotiation. This gives devices the ability to receive the protocol field of PPP frames in compressed format, provided they are not of control protocols (LCP, IPCP etc). This option is disabled by default. We recommend enabling *pfc* on asynchronous interfaces.

Chapter 6 PPP Interface Monitoring

6.1 Monitoring the PPP interface

This section summarizes and details PPP interface monitoring commands.

To access the PPP interface monitoring menu, use the **network <PPP Interface>** command in the general monitoring menu:

```
*monitor
Console Operator
+network pppX
-- Generic PPP Console --
pppX+
```

Command	Function
? (HELP)	Lists the commands or options.
BASE-INTERFACE	Accesses the monitoring menu of the base interfaces associated with the PPP interface.
PPP	Accesses the PPP parameter monitoring menu.
PPPOE	Accesses the PPPoE parameter monitoring menu.
EXIT	Exits PPP interface monitoring.

6.2 Monitoring the PPP Interface's base interfaces

Use the **base-interface** command in the PPP interface monitoring menu to access the menu that allows you to monitor the PPP interface's base interfaces:

```
pppX+base-interface
-- Base Interface Console --
pppX Base IFC+
```

The options available in this menu are:

```
pppX Base IFC+?
  list    Display base interface parameters
  exit
pppX Base IFC+
```

6.2.1 LIST

Displays the base interface parameters associated with a particular PPP interface. If you have associated more than one base interface with the PPP interface, the value of these parameters is shown for each link.

Syntax:

```
pppX Base IFC+list
```

Example:

```
ppp1 Base IFC+list
Profile name           : briConn
Destination address    : 987654321
Local address          : 123456789
Base interface         : bri0/0
Circuit id request     : 255
Dial circuit status    : OPENING
Circuit id assigned    : NOT ASSIGNED

ppp1 Base IFC+
```

Profile name	Name of the *Dial* profile used for the link (only for those base interfaces that need to use this type of profile).
Destination address	Remote address used (destination number in switched interfaces).
Local address	Displays the local address used (local number in switched interfaces).

Base interface	Base interface.
Circuit id request	Identifier for the circuit requested in the configuration.
Dial circuit status	Current status of the Dial circuit (base interface status).
Circuit id assigned	Assigned circuit identifier.

6.2.2 EXIT

This command allows you to exit PPP base interface monitoring and return to the general PPP monitoring menu.

Syntax:

```
pppX Base IFC+exit
```

Example:

```
ppp1 Base IFC+exit
ppp1+
```

6.3 PPP monitoring of the PPP interface

Use the **PPP** command in the PPP interface monitoring menu to access the PPP parameter monitoring menu.

```
pppX+ppp
-- PPP Console --
pppX PPP+
```

You will see the following commands in the PPP monitoring menu:

```
pppX PPP+?
  clear          Initialize all the statistics
  list           Monitoring information relative to PPP parameters
  reset-link     Reset the PPP connection
  exit
pppX PPP+
```

6.3.1 CLEAR

Allows you to erase (initialize) all the statistics associated with the PPP interface in question.

Syntax:

```
pppX PPP+clear
```

Example:

```
ppp1 PPP+clear
ppp1 PPP+
```

6.3.2 LIST

You will see the following set of options when you use the **list** command:

```
pppX PPP+list ?
  all                          All the corresponding monitoring information
  bcp                          Bridging Control Protocol
  ccp                          Compression Control Protocol
  control                      Negotiation process results
  ipcp                         IP control Protocol
  iphc-compression             IP Header Compression
  ipv6cp                       IPv6 Control Protocol
  lcp                          Link Control Protocol
  multilink-ppp                Multilink-PPP
  van-jacobson-compression     Van Jacobson compression statistics
pppX PPP+
```

6.3.2.1 LIST ALL

Displays all monitoring information related to PPP parameters.

Syntax:

```
pppX PPP+list all
```

Example:

```
ppp1 PPP+list all
Version             : 3.0.0

Base Interface      : 3  bri0/0
Base Circuit Id     : 255
LCP State           : OPENED
LCP Options           Local          Remote
-----------           -----          ------
Max Receive Unit:     1500           1500
Async Char Mask :     0xffffffff     0xffffffff
Authentication  :     PAP            PAP
Magic Number    :     0xa8fa3240     0x27690269
Prot Field Compr:     NO             NO
Addr/Ctrl Compr :     NO             NO
32-Bit Checksum :     NO             NO

BCP State           : INITIAL
BCP Options               Local             Remote
-----------------     ---------------   ---------------
Line Id         :     0000              0000
802.3 Frames    :     YES               YES
802.5 Frames    :     YES               YES
FDDI  Frames    :     YES               YES
Tinygram        :     NO                NO
STP 802.1D      :     NO                NO
STP IBM SRB     :     NO                NO

IPCP State          : OPENED
IPCP Options              Local             Remote
-----------------     ---------------   ---------------
IP Address      :     192.168.1.1       192.168.1.2
Van Jacobson Cmp:     NO                NO
IPHC Compression:     YES               YES
       tcp-space:     16                16
   non-tcp-space:     16                16
      max-period:     256               256
        max-time:     5                 5
      max-header:     168               168
       suboption:     RTP-Compression   RTP-Compression
Primary DNS     :     ----              0.0.0.0
Secondary DNS   :     ----              0.0.0.0
Primary NBNS    :     ----              0.0.0.0
Secondary NBNS  :     ----              0.0.0.0

Base Interface      : 3  bri0/0
Base Circuit Id     : 255
LCP Statistics      Received     Sent
-----------------   ----------   ----------
Frames.........:            89           95
Bytes..........:           912         1186
Config. Request:             2            6
Config. Ack....:             2            2
Config. Nak....:             0            0
Config. Reject.:             1            0
Termin. Request:             0            1
Termin. Ack....:             0            0
Echo Request...:            42           44
Echo Reply.....:            42           42

BCP Statistics      Received     Sent
-----------------   ----------   ----------
Config. Request:             0            0
```

```
Config. Ack....:             0            0
Config. Nak....:             0            0
Config. Reject.:             0            0
Termin. Request:             0            0
Termin. Ack....:             0            0

IPCP Statistics     Received     Sent
-----------------   ----------   ----------
Config. Request:             2            2
Config. Ack....:             2            2
Config. Nak....:             0            0
Config. Reject.:             0            0
Termin. Request:             0            0
Termin. Ack....:             0            0

Van Jacobson Statistics
-----------------------
outbound TCP/IP packets.............:          0
outbound TCP/IP compressed packets.:          0
searches for connection state......:          0
times couldn't find conn. state....:          0
inbound TCP/IP uncompressed packets:          0
inbound TCP/IP compressed packets..:          0
inbound TCP/IP unknown type packets:          0
inbound TCP/IP packets tossed error:          0

IPHC Statistics
---------------
IPHC Statistics not available

IPV6CP Statistics   Received     Sent
-----------------   ----------   ----------
Config. Request:             1            1
Config. Ack....:             1            1
Config. Nak....:             0            0
Config. Reject.:             0            0
Termin. Request:             0            0
Termin. Ack....:             0            0

--- Multilink PPP: NEGOTIATED ---

MP Options:
  Local MRRU..........: 1524
  Local SSN...........: NO
  Local EndPoint Disc.: Class 3
  ED value (MAC Address): 00-A0-26-00-90-B0

  Remote MRRU..........: 1524
  Remote SSN...........: NO
  Remote EndPoint Disc.: Class 1
  ED value (Locally Assigned Address): santacruz

  Active links in bundle: 1
  Current Fragment Queue Length: 0
  Relaxed lost fragment detection enabled

MP Statistics:
--------------
  Tx. Frames......: 643
  Tx. Bytes.......: 17765
  Rx. Frames......: 698
  Rx. Bytes.......: 17192
  Enqueued Frames.: 237
  Error Frames....: 0
  Lost Frames.....: 2

CCP Statistics       Received     Sent
```

```
-----------------   ----------   ----------
Config. Request:             1            2
Config. Ack....:             1            1
Config. Nak....:             0            0
Config. Reject.:             0            0
Termin. Request:             0            0
Termin. Ack....:             0            0
Processed Frms.:            85          100
Processed Bytes:          6025       120832
Dec/Enc Frames.:            85           84
Dec/Enc Bytes..:        120162         7919
Uncompress Frms:             0            0
Error Frames...:             0            0
Reset Request..:             0            0
Reset Ack......:             0            0
Out of Sequence:             0            0
LCB Error......:             0            0
Header Error...:             0            0

CCP Enabled            CCP State          : OPENED

O(PPP): ENCODE LZS-DCP 1 SeqNum+LCB ProcUncomp
          Session:1    Sequence number:0
I(PPP): DECODE STAC 1 Extended
          Session:2    Sequence number:0
ppp1 PPP+
```

The following sections explain the different parameters shown in PPP monitoring.

6.3.2.2 LIST BCP

Displays monitoring information for the *Bridging Control Protocol* (BCP).

Syntax:

```
pppX PPP+list bcp
```

Example:

```
ppp1 PPP+list bcp
BCP Statistics      Received     Sent
-----------------   ----------   ----------
Config. Request:             0            0
Config. Ack....:             0            0
Config. Nak....:             0            0
Config. Reject.:             0            0
Termin. Request:             0            0
Termin. Ack....:             0            0
ppp1 PPP+
```

6.3.2.3 LIST CCP

Displays statistics associated with the CCP protocol.

Syntax:

```
pppX PPP+list ccp
```

Example:

```
ppp1 PPP+list ccp
CCP Statistics       Received     Sent
-----------------   ----------   ----------
Config. Request:             1            2
Config. Ack....:             1            1
Config. Nak....:             0            0
Config. Reject.:             0            0
Termin. Request:             0            0
Termin. Ack....:             0            0
Processed Frms.:            85          100
```

```
Processed Bytes:          6025       120832
Dec/Enc Frames.:            85           84
Dec/Enc Bytes..:        120162         7919
Uncompress Frms:             0            0
Error Frames...:             0            0
Reset Request..:             0            0
Reset Ack......:             0            0
Out of Sequence:             0            0
LCB Error......:             0            0
Header Error...:             0            0
ppp1 PPP+
```

Config. Request	Number of received/sent CCP *Configure Request* frames.
Config. Ack	Number of received/sent CCP *Configure Ack* frames.
Config. Nak	Number of received/sent CCP *Configure Nak* frames.
Config. Reject	Number of received/sent CCP *Configure Reject* frames.
Termin. Request	Number of received/sent CCP *Terminate Request* frames.
Termin. Ack	Number of received/sent CCP *Terminate Ack* frames.
Processed Frms	Number of processed frames received for decompressing/ compressing.
Processed Bytes	Number of processed bytes received for decompressing/compressing.
Dec/Enc Frames	Number of decompressed/compressed frames.
Dec/Enc Bytes	Number of decompressed/compressed bytes.
Uncompress Frms	Number of unprocessed frames received for decompressing/ compressing.
Error Frames	Number of frames with errors during compression/decompression.
Reset Request	Number of received/sent CCP *Reset Request* frames.
Reset Ack	Number of received/sent CCP *Reset Ack* frames.
Out of Sequence	Number of sequence errors detected during reception/transmission.
LCB Error	Number of LCB errors detected during reception/transmission.
Header Error	Number of header errors.

6.3.2.4 LIST CONTROL

You will see the following set of options when you use the **list control** command:

```
pppX PPP+list control ?
  bcp              Bridging Control Protocol
  ccp              Compression Control Protocol
  ipcp             IP control Protocol
  lcp              Link Control Protocol
  multilink-ppp    Multilink-PPP
pppX PPP+
```

LIST CONTROL BCP

Displays the options negotiated in establishing the *Bridging Control Protocol* (BCP) network layer protocol.

Syntax:

```
pppX PPP+list control bcp
```

Example:

```
ppp1 PPP+list control bcp
BCP State           : INITIAL
BCP Options               Local             Remote
-----------------     ---------------   ---------------
Line Id         :     0000              0000
802.3 Frames    :     YES               YES
802.5 Frames    :     YES               YES
FDDI  Frames    :     YES               YES
Tinygram        :     NO                NO
STP 802.1D      :     NO                NO
STP IBM SRB     :     NO                NO
ppp1 PPP+
```

Line Id	Line identifier configured for the SRB (*Source-Route Bridge*) at the local and re-

	mote ends.
802.3 Frames	Indicates whether the local and remote ends can receive 802.3 frames.
802.5 Frames	Indicates whether the local and remote ends can receive 802.5 frames.
FDDI Frames	Indicates whether the local and remote ends can receive *Fiber Distributed Data Interface* (FDDI) frames.
Tinygram	Indicates whether *tinygram-compression* has been selected at the local and remote ends.
STP 802.1D	Indicates whether the local and remote ends can receive 802.1D *Spanning-Tree Protocol* (STP) frames.
STP IBM SRB	Indicates whether the local and remote ends can receive IBM SRB STP frames.

LIST CONTROL CCP

Displays the negotiated CCP encryption/decryption control parameters, the current CCP status, and whether the CCP protocol is enabled or disabled. If CCP has not been negotiated, the configuration values are displayed.

Syntax:

```
pppX PPP+list control ccp
```

Example:

```
ppp1 PPP+list control ccp
CCP Enabled            CCP State          : OPENED
O(PPP): ENCODE LZS-DCP 1 SeqNum+LCB ProcUncomp
          Session:1    Sequence number:0
I(PPP): DECODE STAC 1 Extended
          Session:2    Sequence number:0
ppp1 PPP+
```

CCP State	CCP state.
ENCODE	Negotiated encryption options.
DECODE	Negotiated decryption options.

LIST CONTROL IPCP

Shows the options negotiated in establishing the IPCP network layer protocol. If dynamic IP address assignment is being used, you will be able to view the assigned IP addresses. You will also be able to view the remote end IP address if it has been received.

Syntax:

```
pppX PPP+list control ipcp
```

Example:

```
ppp1 PPP+list control ipcp
IPCP State          : OPENED
IPCP Options              Local            Remote
-----------------     ---------------   ---------------
IP Address      :     192.168.1.1       192.168.1.2
Van Jacobson Cmp:     NO                NO
IPHC Compression:     YES               YES
       tcp-space:     16                16
   non-tcp-space:     16                16
      max-period:     256               256
        max-time:     5                 5
      max-header:     168               168
       suboption:     RTP-Compression   RTP-Compression
Primary DNS     :     ----              0.0.0.0
Secondary DNS   :     ----              0.0.0.0
Primary NBNS    :     ----              0.0.0.0
Secondary NBNS  :     ----              0.0.0.0
ppp1 PPP+
```

IPCP State	IPCP status.
IP Address	IP addresses at both ends of the PPP interface.
Van Jacobson Cmp	Indicates whether Van Jacobson compression has been negotiated.
IPHC Compression	Indicates whether IPHC compression has been negotiated and the negotiated parameters.

tcp-space	Maximum TCP context identifier.
non-tcp-space	Maximum NO-TCP context identifier
max-period	Maximum separation between complete headers.
max-time	Maximum time interval between complete headers.
max-header	Maximum header size that can be compressed.
suboption	*RTP-Compression* (*Real Time Protocol* compression) or *none*.
Primary DNS	Primary DNS server assigned to each remote endpoint.
Secondary DNS	Secondary DNS server assigned to each remote endpoint.
Primary NBNS	Primary NetBIOS server assigned to each remote endpoint.
Secondary NBNS	Secondary NetBIOS server assigned to each remote endpoint.

LIST CONTROL LCP

This shows the options sent/received during link establishment and the current status of the LCP protocol. If more than one base interface is associated with the PPP interface, the value of these parameters is shown for each link.

Syntax:

```
pppX PPP+list control lcp
```

Example:

```
ppp1 PPP+list control lcp
Version             : 3.0.0
Base Interface      : 3  bri0/0
Base Circuit Id     : 255
LCP State           : OPENED
LCP Options           Local          Remote
-----------           -----          ------
Max Receive Unit:     1500           1500
Async Char Mask :     0xffffffff     0xffffffff
Authentication  :     PAP            PAP
Magic Number    :     0xa8fa3240     0x27690269
Prot Field Compr:     NO             NO
Addr/Ctrl Compr :     NO             NO
32-Bit Checksum :     NO             NO
ppp1 PPP+
```

Version	Always 3.0.0.
Base Interface	Base interface identifier and type.
Base Circuit Id	Base circuit identifier.
LCP State	LCP status.
Max Receive Unit	Negotiated MRU value.
Async Char Mask	Negotiated Asynchronous Control Character Map (ACCM).
Magic Number	Negotiated Magic Number.
Prot Field Compr	Indicates whether *Protocol Field Compression* (PFC) has been negotiated.
32-Bit Checksum	Always **NO**, because a 16-bit checksum is used.

LIST CONTROL MULTILINK-PPP

When you negotiate PPP multilink on a PPP interface, this command lets you display the negotiated MP options along with other parameters related to the PPP Multilink bundle.

Syntax:

```
pppX PPP+list control multilink-ppp
```

Example:

```
ppp1 ppp+list control multilink-ppp
--- Multilink PPP: NEGOTIATED ---
MP Options:
  Local MRRU..........: 1524
  Local SSN...........: NO
  Local EndPoint Disc.: Class 3
```

```
  ED value (MAC Address): 00-A0-26-00-90-B0

  Remote MRRU..........: 1524
  Remote SSN...........: NO
  Remote EndPoint Disc.: Class 1
  ED value (Locally Assigned Address): santacruz
  Multiclass Multilink: NO
  Active links in bundle: 1
  Current Fragment Queue Length: 0
  Relaxed lost fragment detection enabled
ppp1 PPP+
```

MRRU	Negotiated MRRU.
SSN	Indicates whether *Short Sequence Number* (SSN) header format has been negotiated.
EndPoint Disc	Negotiated Endpoint Discriminator class.
ED value	Negotiated Endpoint Discriminator value.
Active Links in Bundle	Number of active PPP links in the MP bundle.
Current Fragment Queue Length	The fragment queue's current length.
Relaxed lost fragment detection enabled	Indicates that the option to reorder multilink fragments is enabled when the fragments arrive out of order through a link.

6.3.2.5 LIST IPCP

This shows statistics associated with the IPCP protocol.

Syntax:

```
pppX PPP+list ipcp
```

Example:

```
ppp1 PPP+list ipcp
IPCP Statistics     Received     Sent
-----------------   ----------   ----------
Config. Request:             2            2
Config. Ack....:             2            2
Config. Nak....:             0            0
Config. Reject.:             0            0
Termin. Request:             0            0
Termin. Ack....:             0            0
ppp1 PPP+
```

Config. Request	Number of received/sent IPCP *Configure Request* frames.
Config. Ack	Number of received/sent IPCP *Configure Ack* frames.
Config. Nak	Number of received/sent IPCP *Configure Nak* frames.
Config. Reject	Number of received/sent IPCP *Configure Reject* frames.
Termin. Request	Number of received/sent IPCP *Terminate Request* frames.
Termin. Ack	Number of received/sent IPCP *Terminate Ack* frames.

6.3.2.6 LIST IPHC-COMPRESSION

This command lists statistics associated with *IP Header Compression* (IPHC). The command also includes the following subset of options:

```
pppX PPP+list iphc-compression ?
  all        All the corresponding monitoring information
  detail     Detailed monitoring information
  non-tcp    Information relative to NO TCP connections
  summary    Summarized version of the monitoring information
  tcp        Information relative to TCP connections
ppp1 PPP+
```

6.3.2.6.1 LIST IPHC-COMPRESSION ALL

Displays monitoring information related to IPHC compression.

Syntax:

```
pppX PPP+list iphc-compression all
```

Example:

```
ppp1 PPP+list iphc-compression all
IPHC Statistics
---------------
--- IPHC [%s] statistics ---

IPHC running parameters:
  IETF format, feedback ON
  TCP: periodic refresh ON
  NonTCP: periodic refresh ON, UDP checksum enabled, RTP list 101
  TX: TCP-conn 16, NonTCP-conn 16
      Max-Header 168, Max-Time 5, Max-Period 256
  RX: TCP-conn 16, NonTCP-conn 16
      Max-Header 168, Max-Time 5, Max-Period 256
TCP/IP Header Compression stats:
  Conn: 0 tx connections (free 16), 0 rx connections
  Sent: 0 total, 0 comp, 0 context-states, 0 not predicted
        0/0 bytes saved/sent
  Rcvd: 0 comp, 0 context-states, 0 error
Non-TCP/IP Header Compression stats:
  Conn: 0 tx connections (free 16), 0 rx connections
  Sent: 0 total, 0 comp, 0 context-states, 0 not predicted
        0/0 bytes saved/sent
  Rcvd: 0 comp, 0 context-states, 0 error

--- TCP/IP Header Compression detailed stats ---
TX detailed contexts:
  There are no active connections
RX detailed contexts:
  There are no active connections

--- Non-TCP/IP Header Compression detailed stats ---
TX detailed contexts:
  There are no active connections
RX detailed contexts:
  There are no active connections
ppp1 PPP+
```

6.3.2.6.2 LIST IPHC-COMPRESSION DETAIL

Displays in-depth monitoring information related to IPHC compression.

Syntax:

```
pppX PPP+list iphc-compression detail
```

Example:

```
ppp1 PPP+list iphc-compression detail
IPHC Statistics
---------------
--- TCP/IP Header Compression detailed stats ---
TX detailed contexts:
  There are no active connections
RX detailed contexts:
  There are no active connections
--- Non-TCP/IP Header Compression detailed stats ---
TX detailed contexts:
  There are no active connections
RX detailed contexts:
  There are no active connections
ppp1 PPP+
```

6.3.2.6.3 LIST IPHC-COMPRESSION NON-TCP

This command displays monitoring information relating to *no TCP* connections. Again, you have more options to fully determine the functionality of this command.

```
pppX PPP+list iphc-compression non-tcp ?
  all    All the corresponding monitoring information
  rtp    RTP traffic information
  rx     Received traffic information
  tx     Transmitted traffic information
ppp1 PPP+
```

LIST IPHC-COMPRESSION NON-TCP ALL

This command allows you to view all the information related to *NO TCP* connections.

Syntax:

```
pppX PPP+list iphc-compression non-tcp all
```

Example:

```
ppp1 PPP+list iphc-compression non-tcp all
IPHC Statistics
---------------
--- Non-TCP/IP Header Compression detailed stats ---
TX detailed contexts:
  There are no active connections
RX detailed contexts:
  There are no active connections
ppp1 PPP+
```

LIST IPHC-COMPRESSION NON-TCP RTP

Allows you to view RTP-traffic-related monitoring information.

Syntax:

```
pppX PPP+list iphc-compression non-tcp rtp
```

Example:

```
ppp1 PPP+list iphc-compression non-tcp rtp
IPHC Statistics
---------------
--- RTP/UDP/IP Header Compression detailed stats ---
TX detailed contexts:
  There are no active connections
RX detailed contexts:
  There are no active connections
ppp1 PPP+
```

LIST IPHC-COMPRESSION NON-TCP RX

This command displays information about received traffic for established *NO TCP* connections.

Syntax:

```
pppX PPP+list iphc-compression non-tcp rx [<connection_id>]
```

You can specify the index of the connection that you are interested in to delimit the information presented. If you do not enter this value, you will be shown the results for all connections.

Example:

```
ppp1 PPP+list iphc-compression non-tcp rx
IPHC Statistics
---------------
--- Non-TCP/IP Header Compression detailed stats ---
RX detailed contexts:
  There are no active connections
ppp1 PPP+
```

LIST IPHC-COMPRESSION NON-TCP TX

Displays information about transmitted traffic for established *NO TCP* connections.

Syntax:

```
pppX PPP+list iphc-compression non-tcp tx [<connection_id>]
```

You can specify the index of the connection that you are interested in to delimit the information presented. If you do not enter this value, you will be shown the results for all connections.

Example:

```
ppp1 PPP+list iphc-compression non-tcp tx
IPHC Statistics
---------------
--- Non-TCP/IP Header Compression detailed stats ---
TX detailed contexts:
  There are no active connections
ppp1 PPP+
```

6.3.2.6.4 LIST IPHC-COMPRESSION SUMMARY

Displays a summary of the monitoring information related to IPHC compression.

Syntax:

```
pppX PPP+list iphc-compression summary
```

Example:

```
ppp1 PPP+list iphc-compression summary
IPHC Statistics
---------------
--- IPHC [%s] statistics ---
IPHC running parameters:
  IETF format, feedback ON
  TCP: periodic refresh ON
  NonTCP: periodic refresh ON, UDP checksum enabled, RTP list 101
  TX: TCP-conn 16, NonTCP-conn 16
      Max-Header 168, Max-Time 5, Max-Period 256
  RX: TCP-conn 16, NonTCP-conn 16
      Max-Header 168, Max-Time 5, Max-Period 256
TCP/IP Header Compression stats:
  Conn: 0 tx connections (free 16), 0 rx connections
  Sent: 0 total, 0 comp, 0 context-states, 0 not predicted
        0/0 bytes saved/sent
  Rcvd: 0 comp, 0 context-states, 0 error
Non-TCP/IP Header Compression stats:
  Conn: 0 tx connections (free 16), 0 rx connections
  Sent: 0 total, 0 comp, 0 context-states, 0 not predicted
        0/0 bytes saved/sent
  Rcvd: 0 comp, 0 context-states, 0 error
ppp1 PPP+
```

6.3.2.6.5 LIST IPHC-COMPRESSION TCP

Displays statistics for TCP connections. Again, you have more options to fully determine the functionality of this command.

```
pppX PPP+list iphc-compression tcp ?
  all    All the corresponding monitoring information
  rx     Received traffic information
  tx     Transmitted traffic information
ppp1 PPP+
```

LIST IPHC-COMPRESSION TCP ALL

Allows you to view all information regarding TCP connections.

Syntax:

```
pppX PPP+list iphc-compression tcp all
```

Example:

```
ppp1 PPP+list iphc-compression tcp all
IPHC Statistics
---------------
--- TCP/IP Header Compression detailed stats ---
TX detailed contexts:
  There are no active connections
RX detailed contexts:
  There are no active connections
ppp1 PPP+
```

LIST IPHC-COMPRESSION TCP RX

Displays information about received traffic for established TCP connections.

Syntax:

```
pppX PPP+list iphc-compression tcp rx [<connection_id>]
```

You can specify the index of the connection that you are interested in to delimit the information presented. If you do not enter this value, you will be shown the results for all connections.

Example:

```
ppp1 PPP+list iphc-compression tcp rx
IPHC Statistics
---------------
--- TCP/IP Header Compression detailed stats ---
RX detailed contexts:
  There are no active connections
ppp1 PPP+
```

LIST IPHC-COMPRESSION TCP TX

Displays information about transmitted traffic for established TCP connections.

Syntax:

```
pppX PPP+list iphc-compression tcp tx [<connection_id>]
```

You can specify the index of the connection that you are interested in to delimit the information presented. If you do not enter this value, you will be shown the results for all connections.

Example:

```
ppp1 PPP+list iphc-compression tcp tx
IPHC Statistics
---------------
--- TCP/IP Header Compression detailed stats ---
TX detailed contexts:
  There are no active connections
ppp1 PPP+
```

6.3.2.7 LIST IPV6CP

This command lists statistics associated with IPV6CP.

Syntax:

```
pppX PPP+list ipv6cp
```

Example:

```
ppp1 PPP+list ipv6cp
IPV6CP Statistics   Received     Sent
-----------------   ----------   ----------
Config. Request:             1            1
Config. Ack....:             1            1
Config. Nak....:             0            0
Config. Reject.:             0            0
```

```
Termin. Request:             0            0
Termin. Ack....:             0            0
ppp1 PPP+
```

6.3.2.8 LIST LCP

Displays the LCP statistics for each link. If the PPP interface has been associated with more than one base interface, the statistics value for each link is shown.

Syntax:

```
pppX PPP+list lcp
```

Example:

```
ppp1 PPP+list lcp
Base Interface      : 3  bri0/0
Base Circuit Id     : 255
LCP Statistics      Received     Sent
-----------------   ----------   ----------
Frames.........:            89           95
Bytes..........:           912         1186
Config. Request:             2            6
Config. Ack....:             2            2
Config. Nak....:             0            0
Config. Reject.:             1            0
Termin. Request:             0            1
Termin. Ack....:             0            0
Echo Request...:            42           44
Echo Reply.....:            42           42

ppp1 PPP+
```

Base Interface	Base interface number and associated descriptor.
Base Circuit Id	Identifier of the base circuit to which these counters are associated.
Frames	Number of received/transmitted frames.
Bytes	Number of received/transmitted bytes.
Config. Request	Number of received/sent LCP *Configure Request* frames.
Config. Ack	Number of received/sent LCP *Configure Ack* frames.
Config. Nak	Number of received/sent LCP *Configure Nak* frames.
Config. Reject	Number of received/sent LCP *Configure Reject* frames.
Termin. Request	Number of received/sent LCP *Terminate Request* frames.
Termin. Ack	Number of received/sent LCP *Terminate Ack* frames.
Echo Request	Number of received/sent LCP *Echo Request* frames.
Echo Reply	Number of received/sent LCP *Echo Reply* frames.

6.3.2.9 LIST MULTILINK-PPP

This command displays statistics associated with PPP Multilink.

Syntax:

```
pppX PPP+list multilink-ppp
```

Example:

```
ppp1 PPP+list multilink-ppp
MP Statistics:
--------------
  Tx. Frames......: 643
  Tx. Bytes.......: 17765
  Rx. Frames......: 698
  Rx. Bytes.......: 17192
  Enqueued Frames.: 237
  Error Frames....: 0
  Lost Frames.....: 2
```

```
MP Real-Time Statistics:
--------------
  Tx. Frames......: 1983
  Tx. Bytes.......: 42399
  Rx. Frames......: 1719
  Rx. Bytes.......: 39519
  Enqueued Frames.: 905
  Error Frames....: 0
  Lost Frames.....: 0

ppp1 PPP+
```

MP Statistics	Statistics for normal data traffic (class 0).
MP Real-Time Statistics	Statistics for real-time data traffic (class 1).
Tx. Frames	Number of transmitted MP frames.
Tx. Bytes	Number of transmitted bytes.
Rx. Frames	Number of received MP frames.
Rx. Bytes	Number of received bytes.
Enqueued Frames	Number of enqueued frames.
Error Frames	Number of frames dropped due to errors.
Lost Frames	Number of frames dropped due to the sequence number.

6.3.2.10 LIST VAN-JACOBSON-COMPRESSION

This allows you to view the statistics related to the Van Jacobson compression.

Syntax:

```
pppX PPP+list van-jacobson-compression
```

Example:

```
ppp1 PPP+list van-jacobson-compression
Van Jacobson Statistics
-----------------------
outbound TCP/IP packets:                    0
outbound TCP/IP compressed packets:         0
searches for connection state:              0
times couldn't find conn. state:            0
inbound TCP/IP uncompressed packets:        0
inbound TCP/IP compressed packets:          0
inbound TCP/IP unknown type packets:        0
inbound TCP/IP packets tossed error:        0
ppp1 PPP+
```

outbound TCP/IP packets	Number of outgoing TCP/IP packets.
outbound TCP/IP compressed packets	Number of outgoing TCP/IP compressed packets.
searches for connection state	Number of searches for connection status.
times couldn't find conn. State	Number of times the connection status could not be found.
inbound TCP/IP uncompressed packets	Number of inbound TCP/IP uncompressed packets.
inbound TCP/IP compressed packets	Number of inbound TCP/IP compressed packets.
inbound TCP/IP unknown type packets	Number of inbound TCP/IP packets with unknown type discarded.
inbound TCP/IP packets tossed error	Number of erroneous inbound TCP/IP packets discarded.

6.3.2.11 RESET-LINK

Restarts the PPP connection. If the PPP interface has negotiated the PPP Multilink protocol and has multiple PPP links in a multilink bundle, all links associated with that PPP interface are closed.

Syntax:

```
pppX PPP+reset-link
```

Example:

```
ppp1 PPP+reset-link
ppp1 PPP+
```

6.3.2.12 EXIT

This command allows you to exit PPP parameter monitoring and return to the PPP general monitoring menu.

Syntax:

```
pppX PPP+exit
```

Example:

```
ppp1 PPP+exit
ppp1+
```

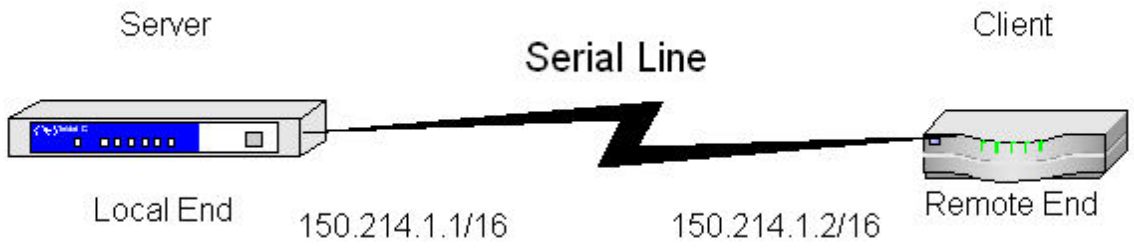
Chapter 7 PPP Interface Configuration Examples

7.1 PPP Interface over serial line

Release	Modification
11.00.02	This example uses commands available as of version 11.00.02. Additional information can be found in sections 2.3.3, 2.3.4 and chapters 5 and 6.

7.1.1 Description

In this example, we configure a PPP interface over a synchronous serial line. The PPP interface is configured so that the local end behaves as server and the remote as client. That is, the local end will use a pre-configured fixed IP address and assign one to the remote end during IPCP negotiation. In addition, the remote end will be authenticated via PAP during link establishment, using the allowed user list configured at the local end. Lastly, the LCP options associated with the serial interface are configured to use a 2048-byte MRU and request, in the LCP negotiation, that the remote end performs protocol field compression (deletion).



7.1.2 Configuration

The first step is to assign the interfaces. This involves assigning a synchronous serial interface to the device's WAN connector (if there is more than one WAN connector, you can choose the one you want) and creating the PPP interface.

```
*config
Config>set data-link sync serial0/0
Config>add device ppp 1
Config>
```

The next step is to assign an IP address to the PPP interface you created. To do this, access the configuration menu and use the **ip address** command, indicating the desired IP address and its mask.

```
Config>network ppp1
-- Generic PPP User Configuration --
ppp1 config>ip address 150.214.1.1 255.255.0.0
ppp1 config>exit
Config>
```

The next step is to indicate that the PPP interface you created is going to be mounted over the WAN1 synchronous serial line. To do this, access the configuration menu of the base interfaces associated with the PPP1 interface and use the **base-interface** command indicating the synchronous serial interface and the option for associating the interface with PPP.

```
Config>network ppp1
-- Generic PPP User Configuration --
ppp1 config>base-interface
-- Base Interface Configuration --
ppp1 Base IFC config>base-interface serial0/0 link
ppp1 Base IFC config>exit
ppp1 config>exit
Config>
```

Then, you need to configure the actual PPP parameters. First of all, specify that the remote end must authenticate

using PAP and establish the list of authorized users. This is a list of usernames and associated passwords that the remote end must present before the link can be established.

```
Config>network ppp1
-- Generic PPP User Configuration --
ppp1 config>ppp
-- PPP Configuration --
ppp1 PPP config>authentication pap
ppp1 PPP config>authentication allowed-user REMOTEUSER password remotekeyword
ppp1 PPP config>
```

The next thing to do is to allow the local end to assign an IP address (specifically 150.214.1.2) to the remote end, while the local end uses the one that has been configured in the PPP interface.

```
ppp1 PPP config>ipcp local address fixed
ppp1 PPP config>ipcp remote address fixed 150.214.1.2
ppp1 PPP config>exit
ppp1 config>exit
Config>
```

You then configure the LCP option profile. Here, you indicate that a 2048-byte Maximum Receive Unit (MRU) is used, that magic number and protocol field compression are enabled, and that HDLC address and control field compression is not enabled.

```
network serial0/0
ppp lcp-options mru 2048
ppp lcp-options pfc
ppp lcp-options acfc
```

Once you have carried out the above configuration steps, all that remains is to save the configuration and restart the device.

```
Config>save
Save configuration (Yes/No)? yes
Building configuration as text... OK
Writing configuration... OK on Flash
Config>                                                    press <ctrl-p>
*restart
Are you sure to restart the system(Yes/No)? yes
  Done
Restarting. Please wait ..............................................
```

The complete configuration for this example is as follows:

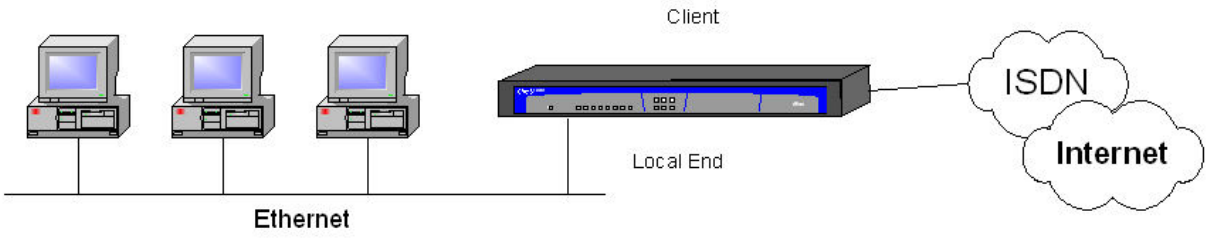
```
; Showing System Configuration for access-level 15 ...
; C4i SNA IPSec VoIP CR Router 1 125 Version 10.7.0 TM
log-command-errors
no configuration
add device ppp 1
set data-link sync serial0/0
;
;
;
network serial0/0
ppp lcp-options mru 2048
ppp lcp-options pfc
ppp lcp-options acfc
;
;
;
;
network x25-node
; -- X25-node interface configuration --
   no ip address
;
exit
;
;
network ppp1
```

```
; -- Generic PPP User Configuration --
   ip address 150.214.1.1 255.255.0.0
;
;
;
;
   ppp
; -- PPP Configuration --
      authentication pap
      authentication allowed-user REMOTEUSER ciphered-pwd 0x6858752A4CB715311D07
338A5477E997
      ipcp remote address fixed 150.214.1.2
   exit
;
   base-interface
; -- Base Interface Configuration --
      base-interface serial0/0 link
;
   exit
;
exit
;
;
;
dump-command-errors
end
; --- end ---
```

7.2 Internet access via ISDN

7.2.1 Description

In this example, we are going to configure a PPP interface to access Internet through an Integrated Service Digital Network (ISDN) basic access. The PPP interface is configured so the local end behaves as client and the remote end as server. That is, the local end will request an IP address from the remote end during the IPCP negotiation. In addition, during link establishment, the remote end will authenticate the local end, which will send the username and password associated with the account it has open in the server. Meanwhile, we will enable Multilink in the PPP interface to give us an access speed of up to 128 Kbps. This speed is achieved using the two basic access B channels simultaneously.



7.2.2 Configuration

The first step is to create the PPP interface that is responsible for providing the device with Internet access.

```
*config
Config>add device ppp 2
Config>
```

The next step is to assign an IP address to the newly created PPP interface to enable IP on that interface. Since the IP address assigned by the remote end is not known, an unnumbered IP address is used. To do this, access the PPP interface configuration menu and use the **ip address** command to assign an unnumbered address to the interface.

```
Config>network ppp2
-- Generic PPP User Configuration --
ppp2 config>ip address unnumbered
```

```
ppp2 config>exit
Config>
```

You also need to select the IP interface as the default route. This is done from the actual IP protocol menu.

```
Config>protocol ip
-- Internet protocol user configuration --
IP config>route 0.0.0.0 0.0.0.0 ppp2
IP config>exit
Config>
```

Next, the Call Profile is configured to define all the parameters related to the call. These will be referred to later when we configure the base interface. Additional information on Call Profiles is available in the Teldat Dm732-I Dial Profile manual.

```
Config>global-profiles dial
-- Dial Profiles Configuration --
Dial Profiles config>profile ISDN default
Dial Profiles config>profile ISDN dialout
Dial Profiles config>profile ISDN remote-address 384020
Dial Profiles config>profile ISDN idle-time 60
Dial Profiles config>exit
Config>
```

Then you need to specify that the created PPP interface will be mounted over the ISDN basic access. In addition, you indicate that you want both B channels for PPP Multilink. You also associate the call profile that you configured earlier.

```
Config>network ppp2
-- Generic PPP User Configuration --
ppp2 config>base-interface
-- Base Interface Configuration --
ppp2 Base IFC config>base-interface bri0/0 255 link
ppp2 Base IFC config>base-interface bri0/0 255 number-of-circuits 2
ppp2 Base IFC config>base-interface bri0/0 255 profile ISDN
ppp2 Base IFC config>exit
ppp2 config>exit
Config>
```

The next step is to configure the PPP parameters. First you specify that the local end requests an IP address from the remote end.

```
Config>network ppp2
-- Generic PPP User Configuration --
ppp2 config>ppp
-- PPP Configuration --
ppp2 PPP config>ipcp local address assigned
ppp2 PPP config>
```

You also need to configure the user and password that are sent to the remote end during link establishment for the purpose of authenticating the remote end.

```
ppp2 PPP config>authentication sent-user MYUSER password mykeyword
ppp2 PPP config>
```

In addition, you must enable the PPP Multilink protocol negotiation and configure a series of parameters related to the functionality. These include specifying a 1750-byte desired MRRU, a 90-second activation interval, a 180-second deactivation interval, an activation threshold of 80% and a deactivation threshold of 40%. You also need to indicate that outbound traffic will be taken into account to calculate the average channel load.

```
ppp2 PPP config>multilink enable
ppp2 PPP config>multilink mrru 1750
ppp2 PPP config>multilink activation interval 90s
ppp2 PPP config>multilink activation threshold 80
ppp2 PPP config>multilink deactivation interval 180s
ppp2 PPP config>multilink deactivation threshold 40
ppp2 PPP config>multilink traffic-load outbound
ppp2 PPP config>
```

Once you have completed the above configuration steps, all that remains is to save the configuration and restart the device.

```
Config>save
Save configuration (Yes/No)? yes
Building configuration as text... OK
Writing configuration... OK on Flash
Config>                                                          press <ctrl-p>

*restart
Are you sure to restart the system(Yes/No)? yes
  Done
Restarting. Please wait ..............................................
```

The complete configuration for this example is as follows:

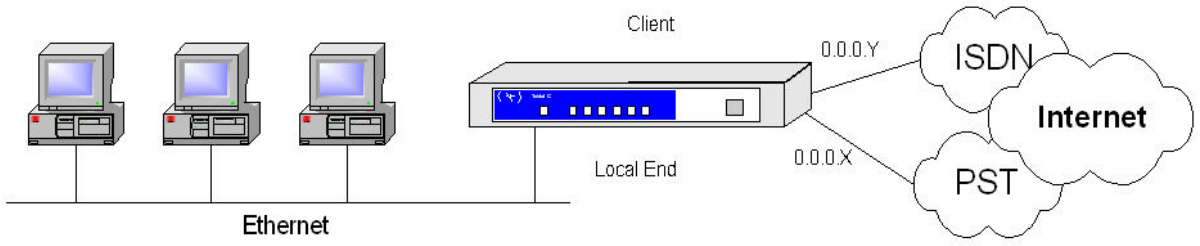
```
; Showing System Configuration for access-level 15 ...
; C4i SNA IPSec VoIP CR Router 1 125 Version 10.7.0 TM
log-command-errors
no configuration
add device ppp 2
set data-link at serial0/0
global-profiles dial
; -- Dial Profiles Configuration --
   profile ISDN default
   profile ISDN dialout
   profile ISDN remote-address 384020
   profile ISDN idle-time 60
;
exit
;
;
;
network ethernet0/0
; -- Ethernet Interface User Configuration --
   no ip address
;
exit
;
;
;
;
network x25-node
; -- X25-node interface configuration --
   no ip address
;
exit
;
;
network ppp2
; -- Generic PPP User Configuration --
   ip address unnumbered
;
;
;
;
   ppp
; -- PPP Configuration --
      authentication sent-user MYUSER ciphered-pwd 0xCBF511457AFAD51ADC189EA2BD7
67FE4
      ipcp local address assigned
      multilink enable
      multilink mrru 1750
      multilink activation interval 1m30s
      multilink activation threshold 80
      multilink deactivation interval 3m
      multilink deactivation threshold 40
      multilink traffic-load outbound
   exit
;
   base-interface
```

```
; -- Base Interface Configuration --
      base-interface bri0/0 255 link
      base-interface bri0/0 255 profile ISDN
      base-interface bri0/0 255 number-of-circuits 2
;
   exit
;
exit
;
;
protocol ip
; -- Internet protocol user configuration --
   route 0.0.0.0 0.0.0.0 ppp2
;
;
exit
;
;
dump-command-errors
end
; --- end ---
Config>
```

7.3 Internet access via ISDN with PSTN backup

7.3.1 Description

In this example, we want Internet access through the Integrated Service Digital Network (ISDN) and, should this fail, a secondary access through the Public Switched Telephone Network (PSTN). To do this, you configure a PPP interface over an ISDN basic access as the main interface, and a PPP interface over an AT commands interface as the secondary interface. Under normal conditions, the PPP interface over ISDN will work correctly and all traffic will be routed through that interface. When the main interface goes down and/or calls cannot be made through ISDN, or the IPCP negotiation timer expires on this interface, the secondary or backup interface will be activated and will be in charge of all the interface traffic that used to travel through ISDN basic access. The configuration of both PPP interfaces is similar to the previous example. The difference here consists of configuring a backup profile in the main interface. You also need to configure the WAN Reroute backup feature.



7.3.2 Configuration

The first step is to assign the interfaces.

```
*config
Config>set data-link at serial0/0
Config>add device ppp 3
Config>add device ppp 4
Config>
```

The next step is to assign IP addresses to the PPP interfaces that you created to enable the IP protocol on those interfaces. Since the IP address assigned by the remote end is not known, unnumbered IP addresses are used.

```
Config>network ppp3
-- Generic PPP User Configuration --
ppp3 config>ip address unnumbered
ppp3 config>exit
Config>network ppp4
```

```
-- Generic PPP User Configuration --
ppp4 config>ip address unnumbered
ppp4 config>exit
Config>
```

In addition, you need to add two default routes, the lowest cost associated with the main link and the higher cost associated with the secondary or backup link. If you want devices connected to a LAN to have Internet access, you will need to configure one of the network's IP addresses in the corresponding interface.

```
Config>protocol ip
-- Internet protocol user configuration --
IP config>route 0.0.0.0 0.0.0.0 ppp3 1
IP config>route 0.0.0.0 0.0.0.0 ppp4 2
IP config>exit
Config>
```

Then you need to configure the call profiles. Additional information on configuring Call Profiles can be found in manual Dm732-I.

```
Config>global-profiles dial
-- Dial Profiles Configuration --
Dial Profiles config>profile ISDN default
Dial Profiles config>profile ISDN dialout
Dial Profiles config>profile ISDN remote-address 384020
Dial Profiles config>profile ISDN idle-time 60
Dial Profiles config>profile AT default
Dial Profiles config>profile AT remote-address 974000
Dial Profiles config>profile AT idle-time 120
Dial Profiles config>exit
Config>
```

Next, you indicate the base interface associated with each of the created PPP interfaces. Since the lower cost default route that you configured earlier corresponds to the PPP3 interface, it must be mounted over the ISDN basic access, constituting the main interface. In this way, the PPP4 interface is mounted over the AT commands interface, constituting the secondary interface.

```
Config>network ppp3
-- Generic PPP User Configuration --
ppp3 config>base-interface
-- Base Interface Configuration --
ppp3 Base IFC config>base-interface bri0/0 255 link
ppp3 Base IFC config>base-interface bri0/0 255 profile ISDN
ppp3 Base IFC config>exit
ppp3 config>exit
Config>network ppp4
-- Generic PPP User Configuration --
ppp4 config>base-interface
-- Base Interface Configuration --
ppp4 Base IFC config>base-interface serial0/0 link
ppp4 Base IFC config>base-interface serial0/0 profile AT
ppp4 Base IFC config>exit
ppp4 config>exit
Config>
```

Next, you need to configure the actual PPP parameters.

First of all, set the backup-related parameters. Indicate a switch to the backup interface when 3 unsuccessful calls have been made or when the NCP negotiation takes longer than 30 seconds. Maximum backup time must establish that, after 60 minutes have elapsed since the backup started, the main link status changes to UP and a further 3 calls are made to see whether the link has recovered. If it has not recovered, it switches to the backup interface again.

```
Config>network ppp3
-- Generic PPP User Configuration --
ppp3 config>ppp
-- PPP Configuration --
ppp3 PPP config>backup retries 3
ppp3 PPP config>backup ncp-timeout 30s
ppp3 PPP config>backup maximum-backup-time 01:00
ppp3 PPP config>exit
ppp3 config>exit
```

```
Config>
```

The parameters related to IPCP are identical in both PPP interfaces: the local IP address is assigned by the remote end.

```
Config>network ppp3
-- Generic PPP User Configuration --
ppp3 config>ppp
-- PPP Configuration --
ppp3 PPP config>ipcp local address assigned
ppp3 PPP config>exit
ppp3 config>exit
Config>net
Config>network ppp4
-- Generic PPP User Configuration --
ppp4 config>ppp
-- PPP Configuration --
ppp4 PPP config>ipcp local address assigned
ppp4 PPP config>exit
ppp4 config>exit
Config>
```

As for the users to send for link authentication from the remote end, assume that there are two different access accounts, using a different user and password depending on the access used (ISDN or PSTN).

```
Config>network ppp3
-- Generic PPP User Configuration --
ppp3 config>ppp
-- PPP Configuration --
ppp3 PPP config>authentication sent-user ISDN_USER password isdnkeyword
ppp3 PPP config>exit
ppp3 config>exit
Config>network ppp4
-- Generic PPP User Configuration --
ppp4 config>ppp
-- PPP Configuration --
ppp4 PPP config>authentication sent-user AT_USER password atkeyword
ppp4 PPP config>exit
ppp4 config>exit
Config>
```

Lastly, configure WAN ReRoute backup. Additional information about this feature is available in the Teldat Dm727-I Backup WAN Reroute manual.

```
Config>feature wrr-backup-wan
-- WAN Reroute Backup user configuration --
Backup WRR>enable
Backup WRR>pair 1 primary interface ppp4
Backup WRR>pair 1 secondary interface ppp3
Backup WRR>exit
Config>
```

Once you have completed the configuration steps detailed above, all that remains is to save the configuration and restart the device.

```
Config>save
Save configuration (Yes/No)? yes
Building configuration as text... OK
Writing configuration... OK on Flash
Config>                                                    press <ctrl-p>
*restart
Are you sure to restart the system(Yes/No)? yes
  Done
Restarting. Please wait ................................................
```

The complete configuration for this example is as follows:

```
; Showing System Configuration for access-level 15 ...
; C4i SNA IPSec VoIP CR Router 1 125 Version 10.7.0 TM
log-command-errors
no configuration
```

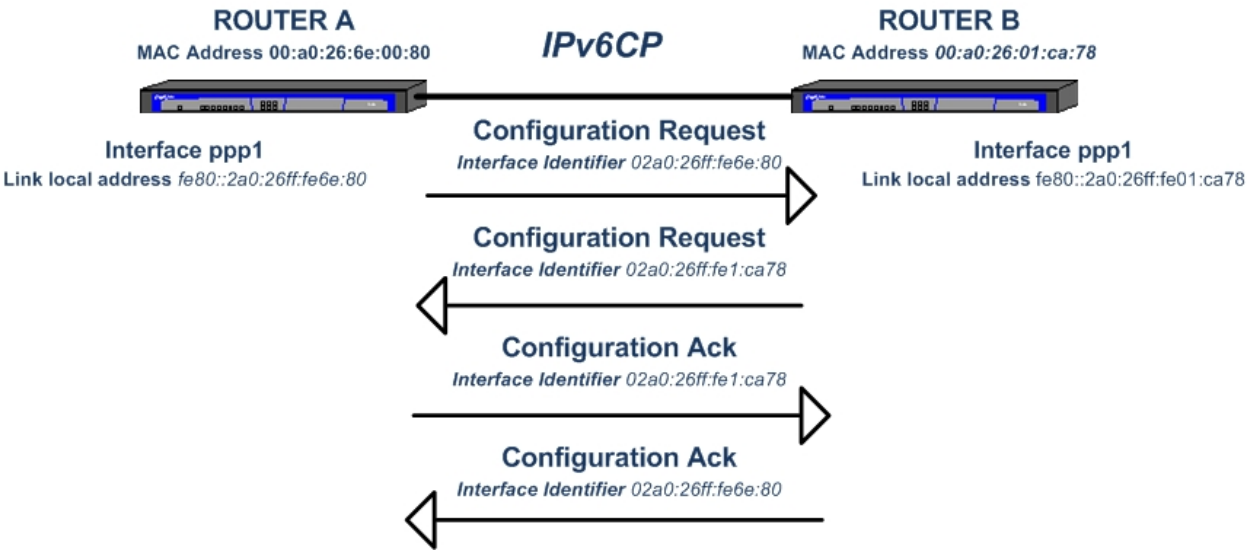
```
add device ppp 3
add device ppp 4
set data-link at serial0/0
global-profiles dial
; -- Dial Profiles Configuration --
   profile ISDN default
   profile ISDN dialout
   profile ISDN remote-address 384020
   profile ISDN idle-time 60
;
   profile AT default
   profile AT dialout
   profile AT remote-address 974000
   profile AT idle-time 120
;
exit
;
;
;
network ethernet0/0
; -- Ethernet Interface User Configuration --
   no ip address
;
exit
;
;
;
;
network x25-node
; -- X25-node interface configuration --
   no ip address
;
exit
;
;
network ppp3
; -- Generic PPP User Configuration --
   ip address unnumbered
;
;
;
;
   ppp
; -- PPP Configuration --
      authentication sent-user ISDN_USER ciphered-pwd 0x0AC96B8369E47627CA65D4F4
7B51A1D0
      backup retries 3
      backup maximum-backup-time 01:00
      ipcp local address assigned
   exit
;
   base-interface
; -- Base Interface Configuration --
      base-interface bri0/0 255 link
      base-interface bri0/0 255 profile ISDN
;
   exit
;
exit
;
;
network ppp4
; -- Generic PPP User Configuration --
   ip address unnumbered
;
;
;
```

```
;
   ppp
; -- PPP Configuration --
      authentication sent-user AT_USER ciphered-pwd 0xA88421D445C4D0210CEFB668B5
4B187B
      ipcp local address assigned
   exit
;
   base-interface
; -- Base Interface Configuration --
      base-interface serial0/0 link
      base-interface serial0/0 profile AT
;
   exit
;
exit
;
;
protocol ip
; -- Internet protocol user configuration --
   route 0.0.0.0 0.0.0.0 ppp3
   route 0.0.0.0 0.0.0.0 ppp4 2
;
;
exit
;
;
feature wrr-backup-wan
; -- WAN Reroute Backup user configuration --
   pair 1 primary interface ppp3 secondary interface ppp4
;
   enable
exit
;
dump-command-errors
end
; --- end ---
```

7.4 IPv6 over PPPoE

In the following scenario, a PPPv6 connection is established between router A and router B. NCP **IPV6CP** is performed over the connection, allowing each end's interface identifier to be negotiated.

The interface identifier negotiation is shown in the following figure:



Each end behaves as follows:

(1) A *Configure Request* message containing a tentative 64-bit interface identifier is sent to the other endpoint.

When the message is received, the receiving endpoint compares the received interface identifier with the one it has.

(2) If the interface identifiers are not the same and are also non-zero, a *Configure Ack* message is sent containing the received interface identifier, to indicate that the receiving endpoint accepts the choice made by the other end.

(3) When the *Configure Ack* is received, the link local address is generated on the PPP interface and the 64-bit identifier is used to form the auto-configured IPv6 addresses in the PPP interface, from the prefixes received in the Router Advertisements.

Configuration

The necessary configurations for each router are shown below:

Router A

```
Config$show config
   log-command-errors
   no configuration
   add device ppp 1
   add device eth-subinterface ethernet0/0 107
   set data-link at cellular0/0
   set data-link at cellular0/1
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.213.231 255.255.254.0
;
      ipv6 enable
   exit
;
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address unnumbered
;
      ipv6 enable
      ipv6 address autoconfig
      ppp
; -- PPP Configuration --
         authentication sent-user teldat ciphered-pwd 0x81B6E08DB61092DB
         ipcp local address assigned
      exit
;
      base-interface
; -- Base Interface Configuration --
         base-interface ethernet0/0.107 link
;
      exit
;
      pppoe
; -- PPPoE User Configuration --
         enable pppoe
         ac-name acteldat
      exit
;
   exit
;
   network ethernet0/0.107
; -- Ethernet Subinterface Configuration --
      ipv6 enable
      encapsulation dot1q 107
;
;
;
;
   exit
;
   event
```

```
; -- ELS Config --
      enable trace subsystem PPP ALL
   exit
;
   protocol ipv6
; -- IPv6 user configuration --
      route ::/0 interface ppp1
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 ppp1
;
   exit
;
;
   dump-command-errors
   end
```

Router B

```
Config$show config
   log-command-errors
   no configuration
   add device ppp 1
   add device eth-subinterface ethernet0/0 107
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.213.230 255.255.254.0
;
      ipv6 enable
   exit
;
;
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address 10.10.10.1 255.255.255.0
;
      ipv6 enable
      ipv6 address 5555::1/64
      ppp
; -- PPP Configuration --
         authentication pap
         authentication allowed-user teldat ciphered-pwd 0x81B6E08DB61092DB
         ipcp remote address fixed 10.10.10.2
      exit
;
      base-interface
; -- Base Interface Configuration --
         base-interface ethernet0/0.107 link
;
      exit
;
      pppoe
; -- PPPoE User Configuration --
         enable pppoe
         ac-name acteldat
         server
      exit
;
   exit
;
   network ethernet0/0.107
; -- Ethernet Subinterface Configuration --
      ipv6 enable
```

```
      encapsulation dot1q 107
;
;
;
;
   exit
;
   event
; -- ELS Config --
      enable trace subsystem PPP ALL
   exit
;
   protocol ipv6
; -- IPv6 user configuration --
      route ::/0 interface ppp1
      unicast-routing
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 ppp1
;
   exit
;
;
   dump-command-errors
   end
```

Monitoring the PPP interface

Router A

The 64-bit interface identifier negotiated for Router A's PPP interface is *02a0:26ff:fe6e:80*. Since this interface identifier is unique in the link, it is used to form the PPP interface link local address, in this case: *fe80::2a0:26ff:fe6e:80.*

When the prefix announced by Router B is received through a Router Advertisement, an IPv6 address is automatically configured in the PPP interface (5555::2a0:26ff:fe6e:80), by combining the received prefix and the previously negotiated interface identifier.

```
*monitor
Console Operator
+protocol ipv6
-- IPv6 protocol monitor --
IPv6+list interface ppp1
Interface ppp1:
-----------------------------
IPv6 is Enabled
Link-local address is:  fe80::2a0:26ff:fe6e:80 [PERM]
Global unicast address(es):
  5555::2a0:26ff:fe6e:80/64 ra-auto [UP] valid lifetime 2591963s, preferred lifetime 604763s

Joined group address(es):
  ff02::1:ff6e:80
  ff02::1
MTU is 1492 bytes
ICMP error messages limited to one every 1000 milliseconds
ICMP redirects are enabled
ICMP unreachables are sent
ND DAD is enabled and if it fails with link-local address IPv6 will be disabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
Default router is fe80::2a0:26ff:fe01:ca78 on ppp1

IPv6 Prefix Advertisements ppp1
Codes: A - Address, P - Prefix-Advertisement, N - Not advertised,
       [L] - On-link, [A] - Autonomous
AN  5555::/64  [LA]  Valid lifetime: 2591963, preferred lifetime: 604800
IPv6+
```

Router B

The 64-bit interface identifier negotiated for Router B's PPP interface is *02a0:26ff:fe1:ca78*. Since this interface identifier is unique in the link, it is used to form the PPP interface link local address, in this case: *fe80::2a0:26ff:fe01:ca78.*

```
+protocol ipv6
-- IPv6 protocol monitor --
IPv6+list interface ppp1
Interface ppp1:
-----------------------------
IPv6 is Enabled
Link-local address is:  fe80::2a0:26ff:fe01:ca78 [PERM]
Global unicast address(es):
  5555::1/64 cfg [PERM/UP]
Joined group address(es):
  ff02::1:ff00:0
  ff02::1:ff00:1
  ff02::1:ff01:ca78
  ff02::2
  ff02::1
MTU is 1492 bytes
ICMP error messages limited to one every 1000 milliseconds
ICMP redirects are enabled
ICMP unreachables are sent
ND DAD is enabled and if it fails with link-local address IPv6 will be disabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds (unspecified)
ND advertised retransmit interval is 0 milliseconds (unspecified)
ND router advertisements are sent every 200 to 600 seconds
ND router advertisements live for 1800 seconds
ND advertised default router preference is Medium
IPv6 Prefix Advertisements ppp1
Codes: A - Address, P - Prefix-Advertisement, N - Not advertised,
       [L] - On-link, [A] - Autonomous
A  5555::/64  [LA]  Valid lifetime: 2592000, preferred lifetime: 604800
IPv6+
```