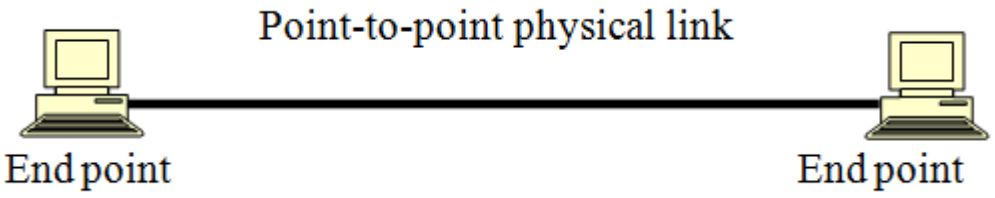
Point-to-Point Protocol (PPP)

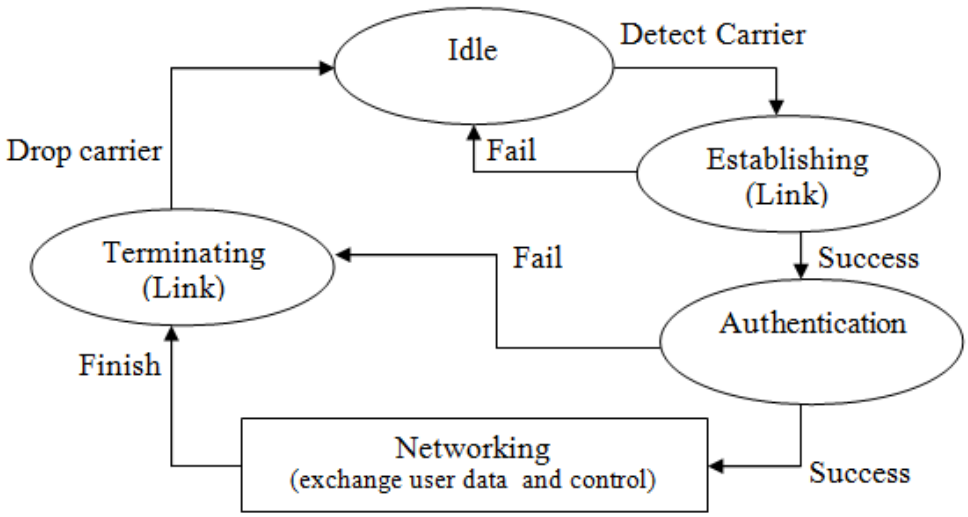
Today, millions of Internet users need to connect their home computers to the computers of an Internet provider to access the Internet. There are also a lot of individuals who need to connect to a computer from home, but they do not want to go through the Internet. The majority of these users have either a dialup or leased line. The telephone line provides a physical link, but to control and manage the transfer of data, there is a need for a point-to-point protocol. The following figure shows a physical point-to-point connection.



The first protocol devised for this purpose was **S**erial **L**ine **I**nternet **P**rotocol (SLIP). However, SLIP has some deficiencies; it does not support protocols other than Internet protocol (IP), it does not support authentication of the user. The PPP is a protocol designed to respond to these deficiencies.

Transition States:

The different phases through which a PPP connection can be described using transition state as shown in figure below:



Idle state: The idle state means that the link is not being used. There is no active carrier and the line is quite.

Establishing State: When one of the end points starts the communications, the connection goes into the establishing state. In this state, options are negotiated between the two parties. If the negotiation is successful, the system goes to the authenticating state (if authentication is required) or directly to the networking state. Several packets may be exchanged during this state.

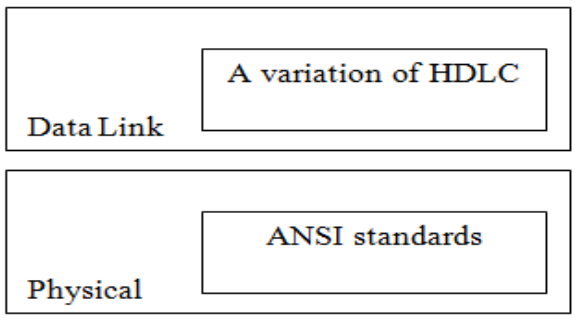
Authenticating State: The authenticating state is optional; the two end points may decide, during the establishing state, not to go through this state. However, if they decide to proceed with authentication, they send several authentication packets. If the result is successful, the connection goes to the networking state; otherwise it goes to the terminating state.

Networking State: The networking state is the heart of the transition states. When a connection reaches this state, the exchange of user control and data packets can be started. The connection remains in this state until one end of the end points wants to terminate the connection.

Terminating State: When the connection is in the terminating state, several packets are exchanged between the two ends for house cleaning and closing the link.

PPP Layers:

PPP has only physical and data link layers as shown in figure below. This means that a protocol that wants to use the services of PPP should have other layers (network, transport, and so on).



Physical Layer:

No specific protocol is defined for the physical layer in PPP. Instead, it is left to the implementer to use whatever is available. PPP supports any of the protocols recognized by ANSI.

Data Link Layer:

At the data link layer, PPP employs a version of HDLC (High level Data Link Control: is a bit oriented data link protocol designed to support both half duplex and full duplex communication over point to point and multipoint links).

The format of PPP frame is shown in figure below**:**

	Flag	Address	Control	Protocol	Data and Padding	FCS	Flag
Bytes	1	1	1	1 or 2	Variable	2 or 4	1

Flag field: The flag field identifies the boundaries of a PPP frame. Its value is 01111110.

Address field: Because PPP is used for a point to point connection; it uses the broadcast address of HDLC, 11111111, to avoid a data link address in the protocol.

Control field: The value is 11000000 to show that the frame does not contain any sequence numbers and that there is no flow and error control.

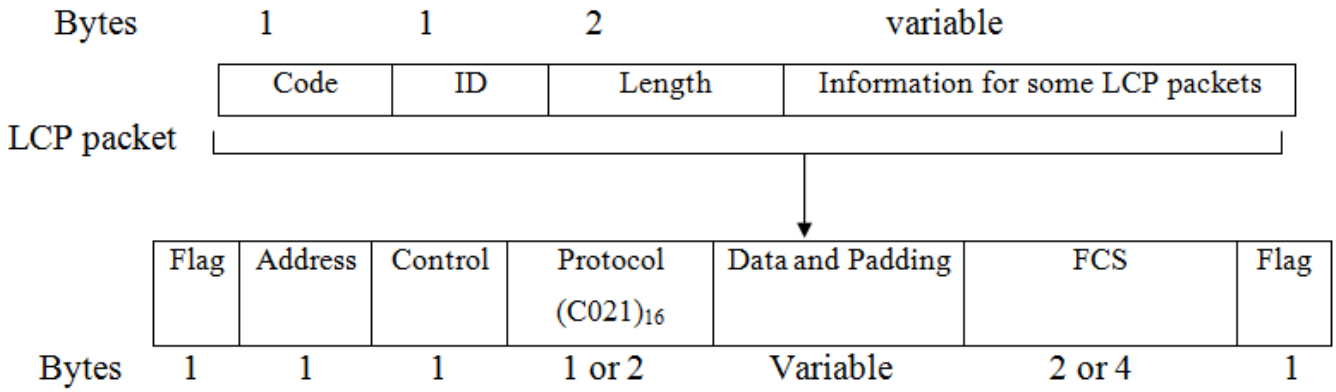
Data field: This field carries either the user data or other information.

FCS: The **F**rame **C**heck **S**equence field is simply a two byte CRC (**C**yclic **R**edundancy **C**heck).

Link Control Protocol (LCP):

LCP is responsible for establishing, maintaining, configuring, and terminating links. It also provides negotiation mechanism to set options between the two end points. Both end points of the link must reach an agreement about the options before the link can be established.

All LCP packets are carried in the payload field of the PPP frame. What defines the frame as one carrying an LCP packet is the value of the protocol field, which should set to $(C021)_{16}$. The format of the LCP packet is shown in figure below:



Code: This field defines the type of LCP packet.

ID: This field holds a value used to match a request with the replay. One end point inserts a value in this field, which will be copied in the replay packet.

Length: This field defines the length of the whole LCP packet.

Information: This field contains extra information needed for some LCP packets**.**

LCP Packets:

Code	Packet Type	Description
01_{16}	Configure-request	Contains the list of proposed options and their values
02_{16}	Configure-ack	Accepts all options proposed
03_{16}	Configure-nak	Announce that some options are not acceptable
04_{16}	Configure-reject	Announce that some options are not recognized
05_{16}	Terminate-request	Request to shut the line down

06_{16}	Terminate-ack	Accepts the shut down request
07_{16}	Code-reject	Announce an known code
08_{16}	Protocol-reject	Announce an known protocol
09_{16}	Echo-request	A type of hello message to check if the other end is alive
$0A_{16}$	Echo-replay	The response to the echo-request message
$0B_{16}$	Discard-request	A request to discard the packet

Options:

There are many options that can be negotiated between the two end points. Options are inserted in the information field of the configuration packets. Some of the most common options are listed in table below:

Options	Default
Maximum receive unit	1500
Authentication protocol	None
Protocol field compression	Off
Address and control field compression	off

Authentication:

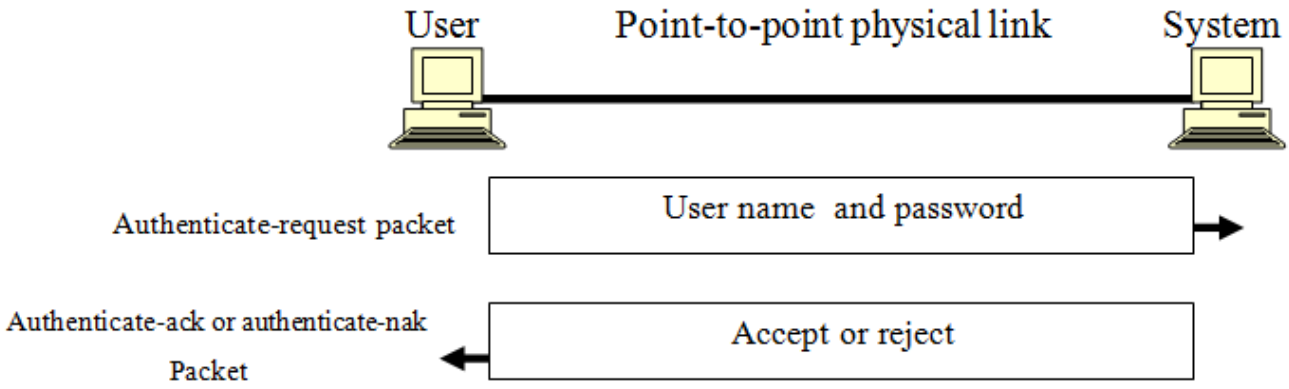
Authentication plays a very important role in PPP because PPP is designed for use over dial-up links where verification of user identity is necessary. Authentication means validating the identity of a user who needs to access a set of resources. PPP has created two protocols for authentication:

PAP:

The Password Authentication Protocol (PAP) is a simple authentication procedure with a two step process:

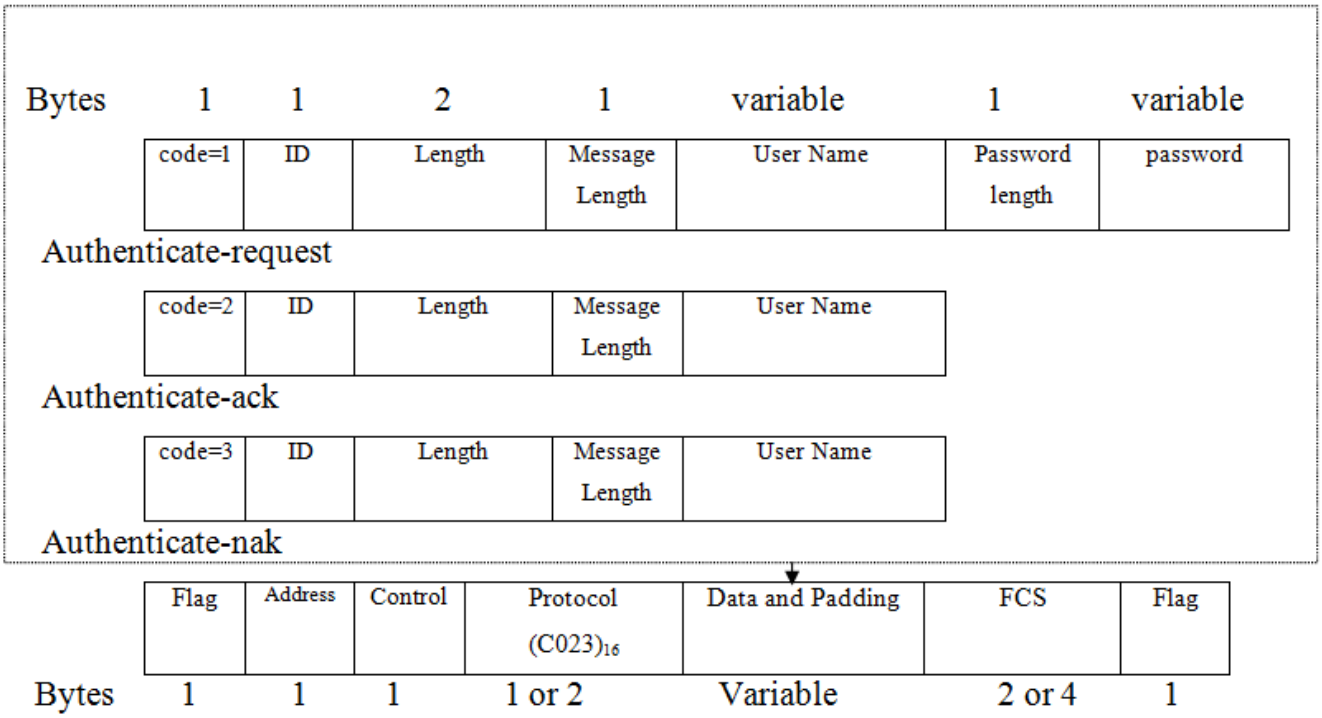
1- The user who wants to access a system sends authentication identification (usually the user name) and password.
2- The system checks the validity of the identification and password and either accepts or denies connection.

The following figure shows the idea of PAP.



PAP Packets:

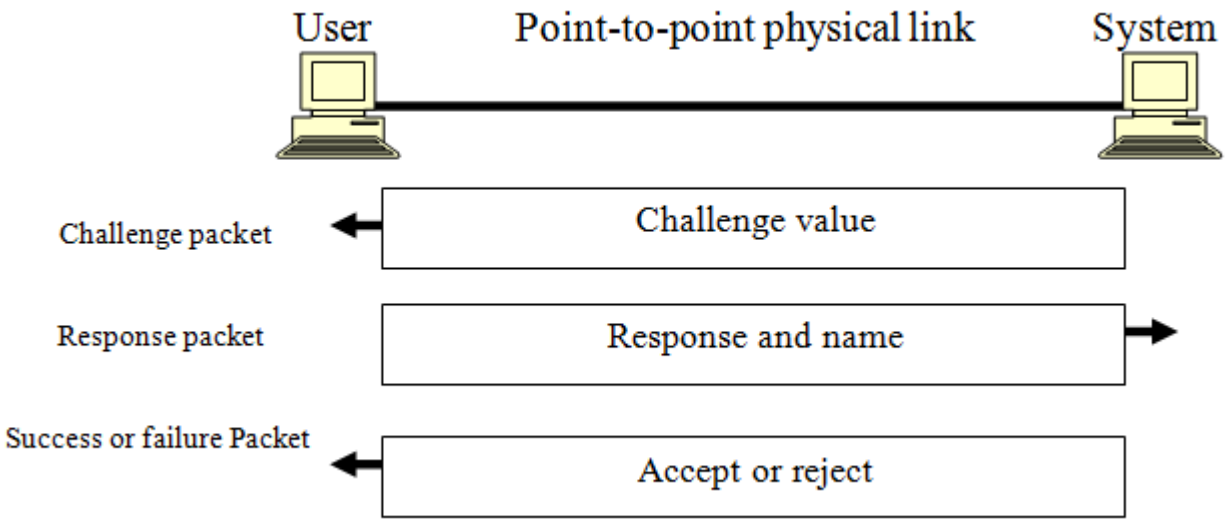
PAP packets are encapsulated in a PPP frame. What distinguishes a PAP packet from other packets is the value of the protocol field, $(C023)_{16.}$ There are three PAP packets: authenticate-request, authenticate-ack, and authenticate-nak. The first packet is used by the user to send the user name and password. The second is used by the system to allow access. The third is used by the system to deny access. The following figure shows the format of the three packets.



CHAP:

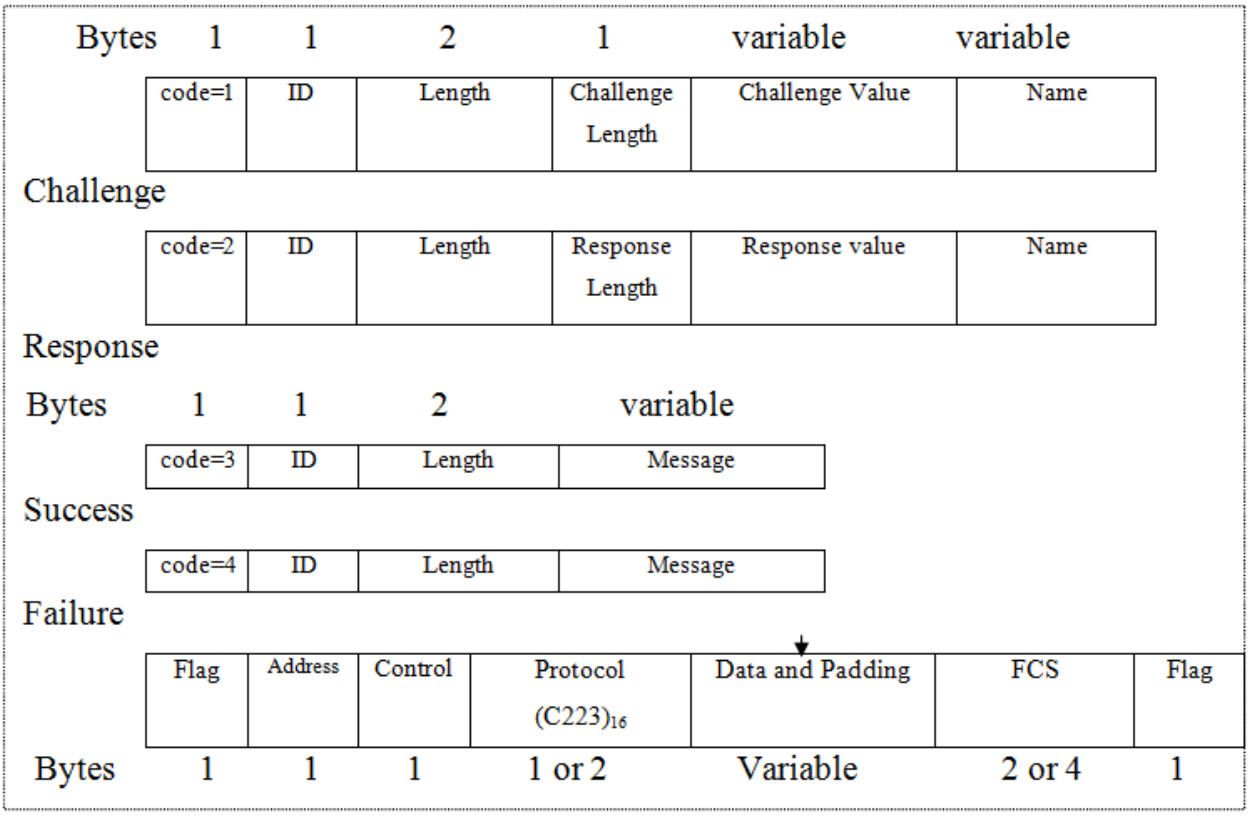
The challenge Handshake Authentication Protocol (CHAP) is a three way hand shaking authentication protocol that provides more security than PAP. In this method, the password is kept secret; it is never sent on line.

1- The system sends to the user a challenge packet containing a challenge value, usually few bytes.
2- The user applies a predefined function that takes the challenge value and the user's own password and creates a result. The user sends the result in the response packet to the system.
3- The system does the same. It applies the same function to the password of the user (known to the system) and the challenge value to create a result. If the result created is the same as the result sent in the response packet, access is granted; otherwise, it is denied.
4- **CHAP** is more secure than PAP, especially if the system continuously changes the challenge value. Even if the intruder learns the challenge value and the result, the password is still secret. The following figure shows the idea:



CHAP Packets:

CHAP packets are encapsulated in the PPP frame. What distinguishes a CHAP packet from other packets is the value of the protocol field, $(C223)_{16}$. There are four CHAP packets: challenge, response, success, and failure. The first packet is used by the system to send challenge value. The second is used by the user to return the result of the calculation. The third is used by the system to allow access to the system. The fourth is used by the system to deny access to the system. The following figure shows the format of the four packets.

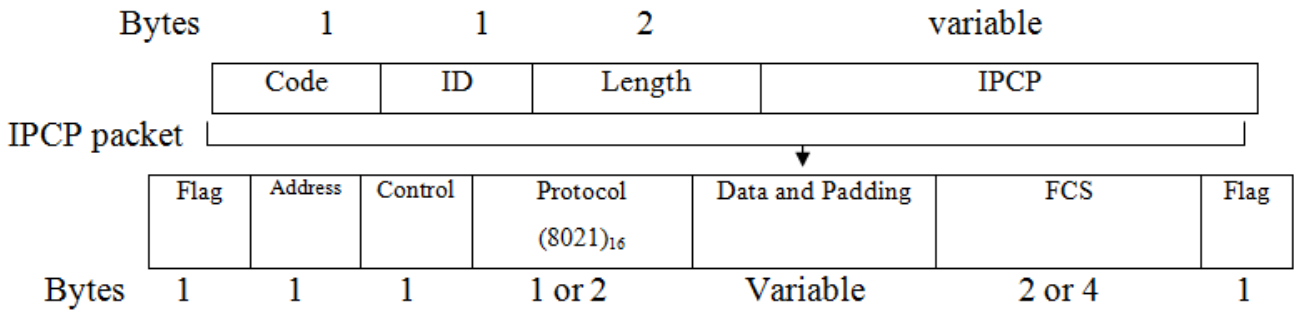


Network Control Protocol (NCP):

After the link has been established and authentication (if any) has been successful, the connection goes to the networking state. In this state, PPP uses another protocol called Network Control Protocol (NCP). NCP is a set of control protocols to allow the encapsulation of data coming from network layer protocols (such as IP) in the PPP frame.

IPCP:

The set of packets that establish and terminate a network layer connection for IP packet is called Internetwork Protocol Control Protocol (IPCP). The format of an IPCP packet is shown in following figure. Note that the value of the protocol field is $(8021)_{16}$.



Seven packets are defined for the IPCP protocol, distinguished by their code values as shown in table below:

Code	IPCP packet
01	Configure-request
02	Configure-ack
03	Configure-nak
04	Configure-reject
05	Terminate-request
06	Terminate-ack
07	Code-reject

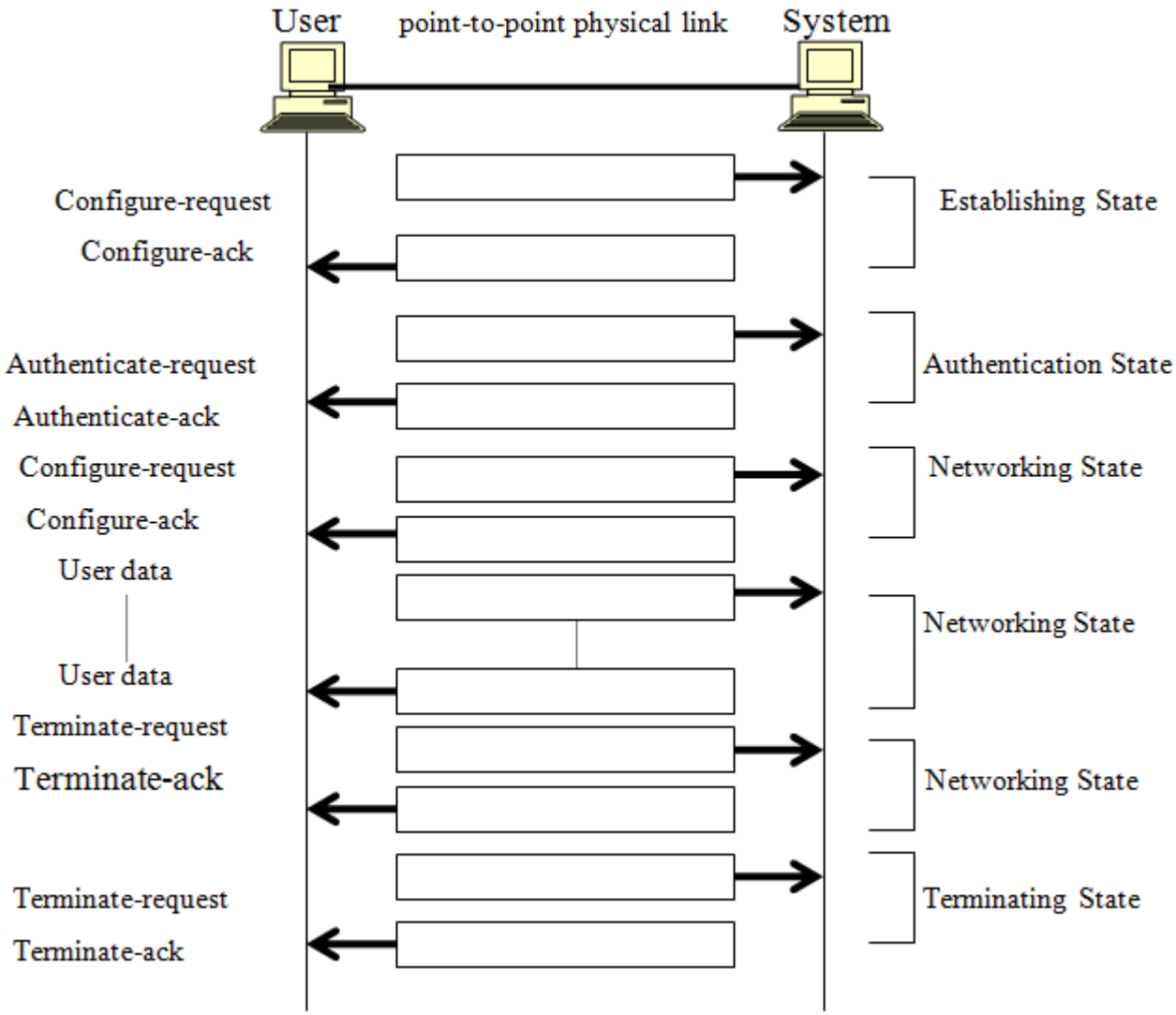
A party uses the configure-request packet to negotiate options with the other party and to set the IP addresses, and so on.

After configuration, the link is ready to carry IP protocol data in the payload field of a PPP frame. This time, the value of the protocol field is $(0021)_{16}$ to show that the IP data packet, not the IPCP packet, is being carried across the link.

After IP has sent all of its packets the IPCP can take control and use the terminate-request and terminate-ack to end the network connection.

Example:

An example of the states through which a PPP connection goes to deliver some network layer packets is shown in figure below:


User
point-to-point physical link
System
Configure-request
Configure-ack
Establishing State
Authenticate-request
Authenticate-ack
Authentication State
Configure-request
Configure-ack
Networking State
User data
User data
Networking State
Terminate-request
Terminate-ack
Networking State
Terminate-request
Terminate-ack
Terminating State