Tema 5

Protocolo PPP *(Point-to-Point Protocol)*

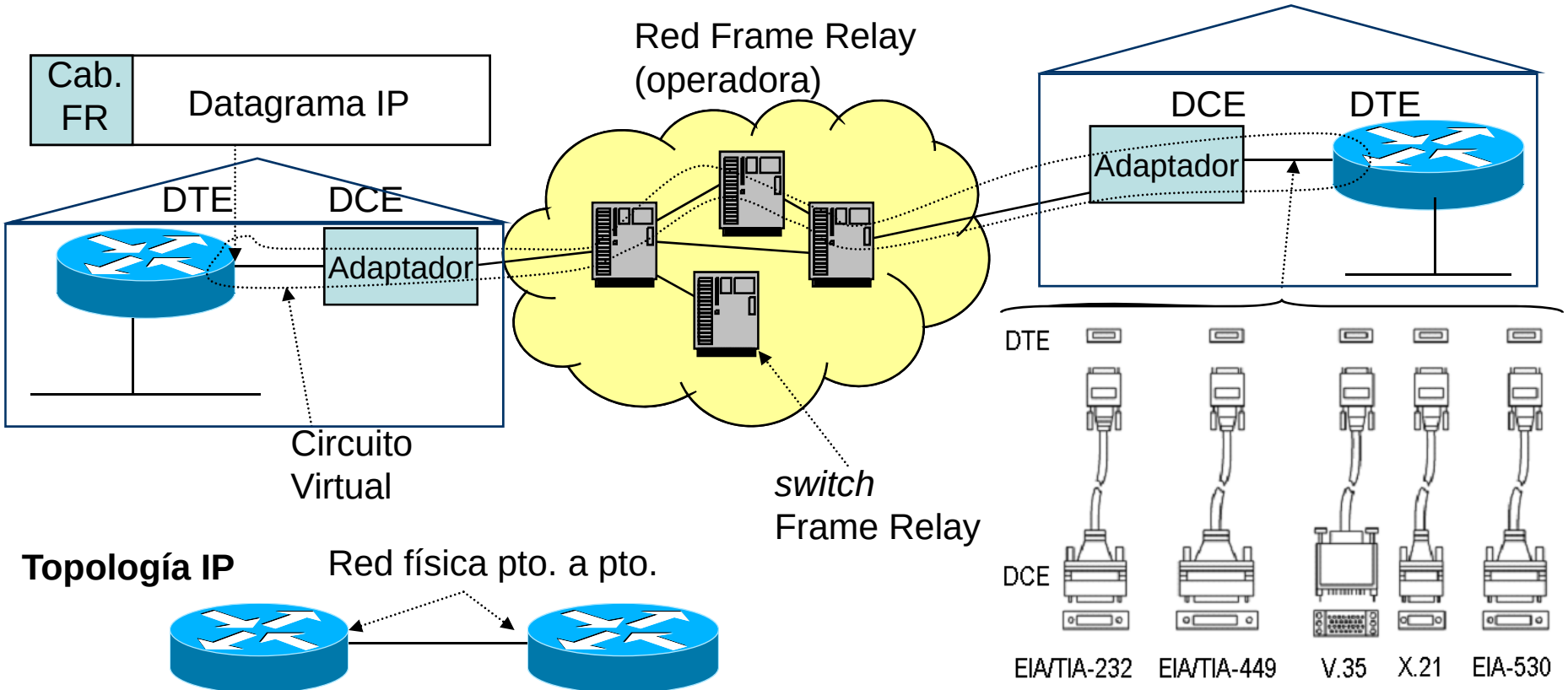
Índice

Protocolos de nivel de enlace para redes punto a punto (I)

- Desde el punto de vista IP: una red física punto a punto interconecta dos (y sólo dos) dispositivos IP.
- Este tipo de redes físicas son las más habituales para conectar dispositivos IP separados largas distancias (WAN, *Wide Area Networks*). Ejemplos:
  - Enlace con módem analógico a través de la Red Telefónica Básica (RTB).
  - Enlace con módem RDSI (Red Digital de Servicios Integrados).
  - Circuito Virtual ATM.
  - Circuito Virtual *Frame Relay*.
  - Línea dedicada de ancho de banda fijo.
  - ...

Protocolos de nivel de enlace para redes punto a punto (II)

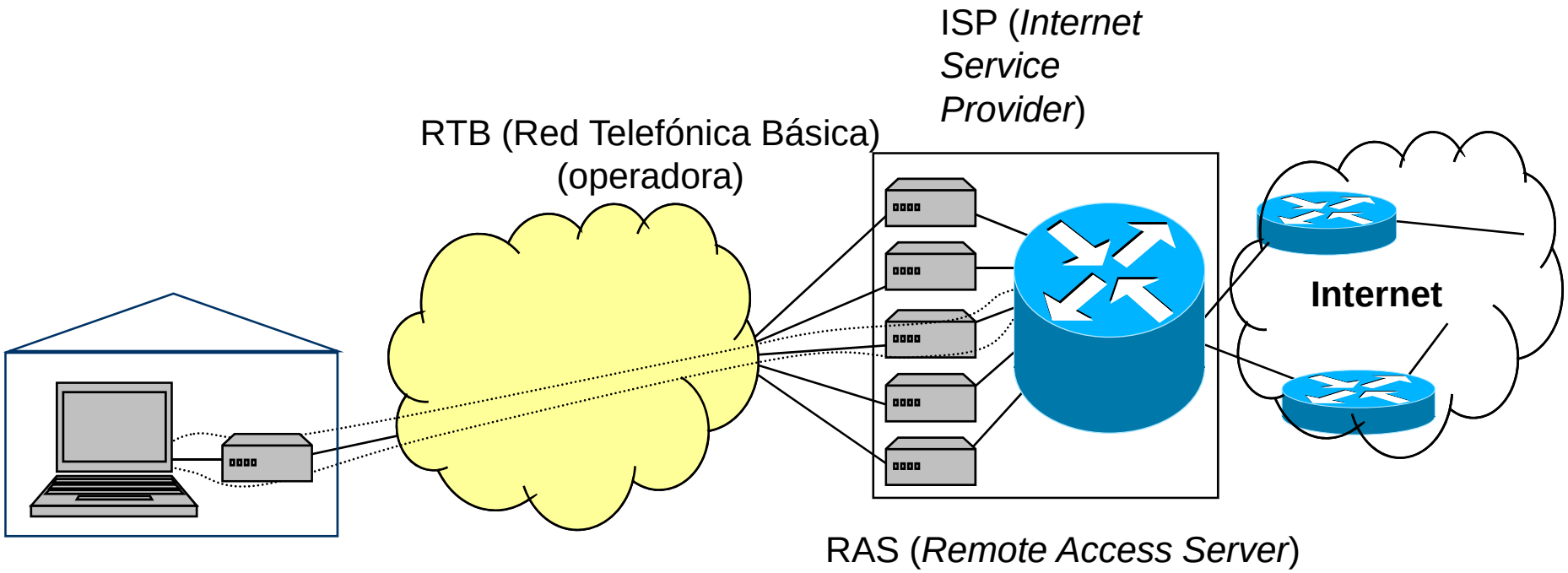
Ejemplo: Interconexión de *routers* mediante CV *Frame Relay*



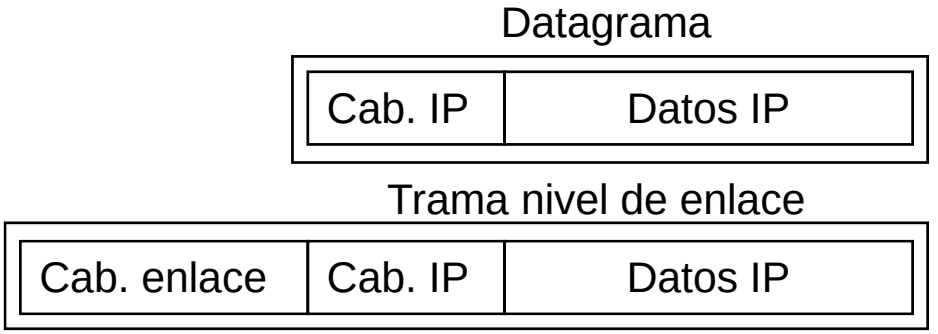
Topología IP

Protocolos de nivel de enlace para redes punto a punto (III)

Ejemplo: Interconexión con módem analógico con el Proveedor de Servicios de Internet (ISP, *Internet Service Provider*)



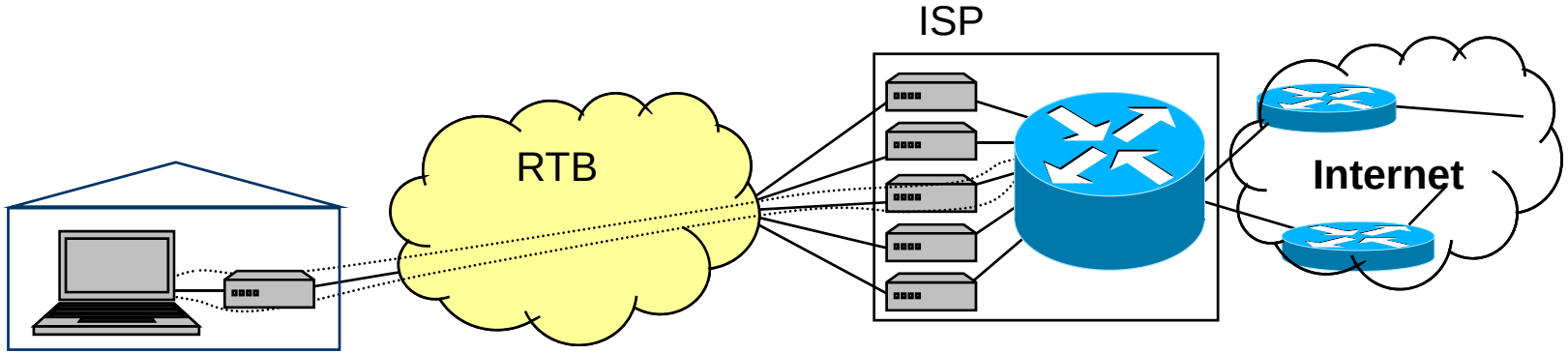
Protocolos de nivel de enlace para redes punto a punto (IV)



...100111010... (entramado red física punto a punto)

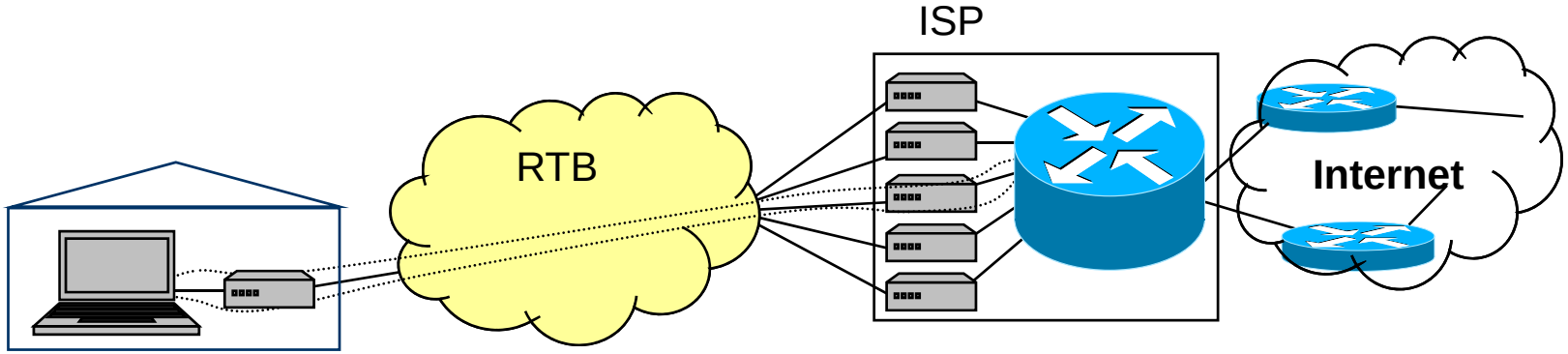
- ¿Cómo transmitir datagramas IP sobre enlaces punto a punto?. Hay enlaces de muy distintas características en cuestiones de capa física (entramado, etc.).
- Existe la necesidad de crear un protocolo de nivel de enlace, que esconda al nivel de red la problemática asociada a las distintas capas físicas.

Protocolos de nivel de enlace para redes punto a punto (V)



- Servicios deseables para un protocolo de nivel de enlace:
  - Entramado de datagramas: Implica, p.e., marcar inicio y fin de datagrama, para poder distinguir entre datagramas consecutivos.
  - Detección/corrección de errores.
  - Control de flujo.
  - Demultiplexado/multiplexado de protocolos de red. Poder usar un enlace para enviar simultáneamente tráfico de varios protocolos de nivel de red (p.e. IP e IPX).
  - Control de la calidad del enlace. Determinar y monitorizar la calidad del enlace (basado habitualmente en la medición de los errores de transmisión).

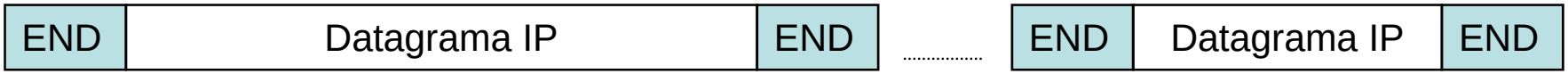
Protocolos de nivel de enlace para redes punto a punto (VI)



- Resolver problema de *transparencia de tráfico.* Determinados enlaces no permiten la transmisión de ciertos caracteres o secuencias de bits reservadas (p.e. caracteres XON/XOFF en módems analógicos). Es necesario implementar un mecanismo de transparencia para poder transmitir tráfico que incluya estos patrones reservados.
- Autenticación. Ofrecer la posibilidad de autenticar la identidad de los extremos de los enlaces.
- Asignación dinámica de direcciones de red. P.e. que un extremo del enlace pida que se le asigne una dirección IP dinámicamente.
- ... (otras funcionalidades, asociadas principalmente a las necesidades de los ISPs).

SLIP (I)

- El primer protocolo de nivel de enlace empleado masivamente en la década de los 80 fue SLIP (*Serial Line Internet Protocol,* RFC 1055).
- Se trata de un protocolo extremadamente sencillo:
  - Sirve únicamente para transmitir datagramas IP (no de otros protocolos de nivel de red) entre los dos extremos del enlace punto a punto.
  - Añade un carácter END (0xC0) para marcar el inicio y el final de cada datagramas.



  - Si el datagrama incluye el carácter END, se emplea un mecanismo de transparencia mediante el carácter ESC (0xDB).
  - No realiza detección ni corrección de errores.
- Sin embargo, no llegó a estar completamente estandarizado, y existían distintas versiones de SLIP.

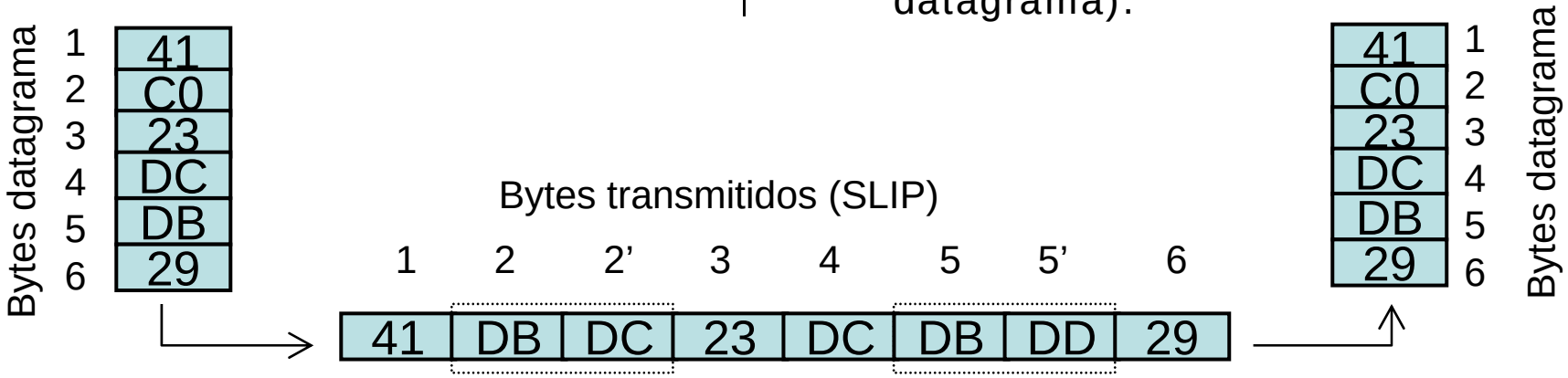
SLIP (II)

Mecanismo de transparencia (transmisión)

1. Transmitir carácter END (marca inicio datagrama).
2. Para cada carácter *c* del datagrama:
   - Si *c*=END (0xC0) => transmitir caracteres {ESC,0xDC}.
   - Si *c*=ESC (0xDB) => transmitir caracteres {ESC,0xDD}.
   - Sino, transmitir el carácter
3. Transmitir carácter END (marca fin de datagrama).

Mecanismo de transparencia (recepción)

1. Esperar recepción carácter END.
2. Para cada carácter *c* recibido:
   - Si *c*=ESC (0xDB) => recibir siguiente carácter *cc*.
     - Si cc=0xDC => añadir END a datagrama.
     - Si cc=0xDD => añadir ESC a datagrama.
     - Sino, error.
   - Si *c*=END => fin de datagrama.
   - Sino, añadir *c* a datagrama.
3. Recibir carácter END (fin datagrama).



Point-to-Point Protocol (I)

- SLIP: ampliamente utilizado en la conexión usuario-ISP de acceso a Internet, en la década de los 80.
- Sin embargo, ofrece muy pocas funcionalidades, que los ISP debían resolver con aplicaciones independientes (p.e. autenticación de los usuarios, con vistas a la tarificación, asignación dinámica de direcciones IP...).
- Parecía clara la necesidad de un nuevo protocolo de nivel de enlace más completo, que sea estandarizado (incluido en los S.O. comerciales etc.).
- El nuevo protocolo desarrollado es el protocolo PPP (*Point-to-Point Protocol*).

Point-to-Point Protocol (II)

- Descripción servicio ofrecido por PPP:
  - Servicio orientado a conexión (conexión PPP).
  - Pueda operar sobre cualquier tipo de enlace punto a punto, con el único requisito de ser *full-duplex*.
  - Entramado de paquetes (separación entre datagramas consecutivos).
  - Transparencia de datos. En caso de enlaces que empleen caracteres reservados, aplicar el necesario mecanismo de transparencia.
  - Multiplexación/demultiplexación. Múltiples protocolos de nivel de red pueden ser empleados transportados en la misma conexión PPP.

Point-to-Point Protocol (III)

- Detección de errores (no corrección).
- Control de calidad de la conexión.
- Autenticación de los extremos de la conexión.
- Compresión de datos.
- Negociación de la dirección de nivel de red.
- Posibilidad de configurar los parámetros de la conexión PPP.
- Extensible: diseñado de tal manera que puedan incorporarse nuevas funcionalidades en futuras versiones.

Point-to-Point Protocol (IV)

- Funcionalidades no ofrecidas por PPP:
  - Corrección de errores, o posibles retransmisiones en caso de detección de error.
  - Control de flujo.
  - Control del orden en la entrega de tramas (si el enlace punto a punto desordenase las tramas, PPP no las ordenaría).
- Todas estas funcionalidades son legadas a capas superiores.

Point-to-Point Protocol (V)

Formato de trama

1 byte	1 byte	1 byte	1 ó 2 bytes	nº entero bytes	2 ó 4 bytes	1 byte
Flag 01111110	Dirección 1111111	Control 00000011	Protocolo	Datos	CRC	Flag 01111110

- PPP usa una estructura de trama similar a la del protocolo de nivel de enlace HDLC (*High-level Data Link Control*).
- Flag (01111110 = 0x7E): Empleado para marcar inicio y fin de trama.
- Dirección (11111111 = 0xFF). Campo heredado de HDLC. En PPP tiene siempre el valor 0xFF (indica que todas las estaciones deben aceptar la trama).
- Control (00000011 = 0x03). Campo heredado de HDLC. En PPP tiene siempre el valor 0x03 (indica “trama no numerada”).

Point-to-Point Protocol (VI)

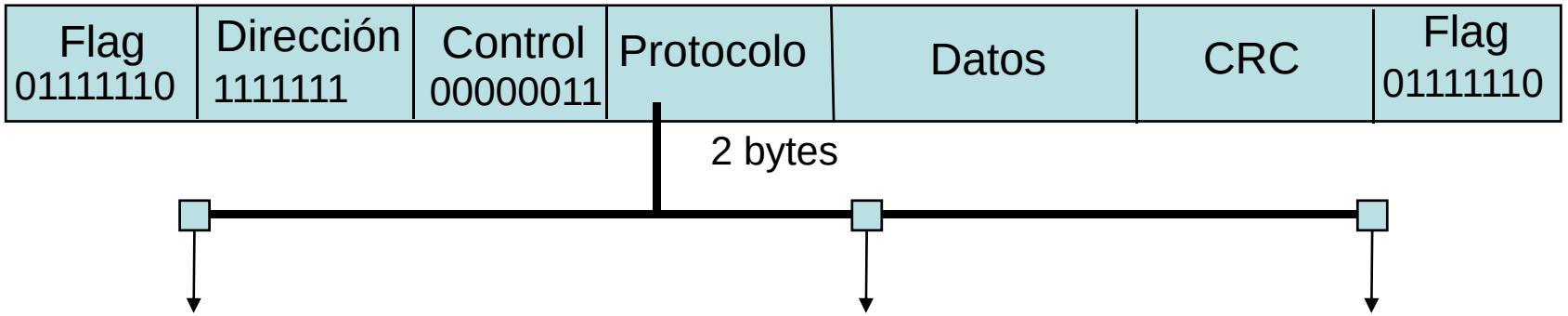
Formato de trama

1 byte	1 byte	1 byte	1 ó 2 bytes	nº entero bytes	2 ó 4 bytes	1 byte
Flag 01111110	Dirección 1111111	Control 00000011	Protocolo	Datos	CRC	Flag 01111110

- Protocolo.
  - Indica tipo de protocolo que se transporta en el campo de datos. P.e. LCP, IPCP, IP, PAP, CHAP...
  - Usualmente ocupa 2 bytes, aunque es posible configurar el envío comprimido de este campo en 1 byte.
- Datos.
  - Tamaño variable, con un máximo por defecto de 1500 bytes, aunque es un parámetro a negociar durante el establecimiento de conexión.
- CRC (detección de errores).
  - Usualmente de 2 bytes.
  - Calculado sobre el campo de dirección, control, protocolo, datos. Es calculado antes de aplicar el mecanismo de transparencia.

Point-to-Point Protocol (VII)

Formato de trama



Protocolos de tipo *control de enlace.*
Comienzan por 0xC0
C021h – Link Control Prot. (LCP)
C023h – PAP
C025h – Link Quality Report
C223h – CHAP

Asociados a 1º parte establecimiento de conexión PPP

Protocolos de tipo *control de red* (NCP, Network Control Protocols). Comienzan por 0x80.
0x8021 – IP Control Protocol (IPCP)
0x8029 – Appletalk Control Protocol
0x802b – Novell IPX Control Protocol
0x803d – Multi-Link Control Protocol
0x80fd – Compression Control Protocol

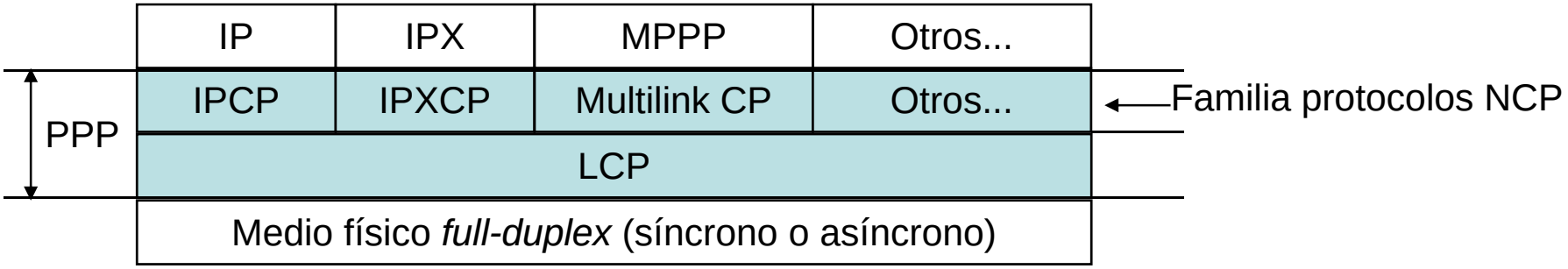
Asociados a 2º parte establecimiento de conexión PPP

Protocolos de nivel de red.
Comienzan por 0x00:
0x0021 – IP
0x0029 – Appletalk
0x002b – Novell IPX
0x003d – Multilink

Empleados durante fase de envío de datos

Point-to-Point Protocol (VIII)

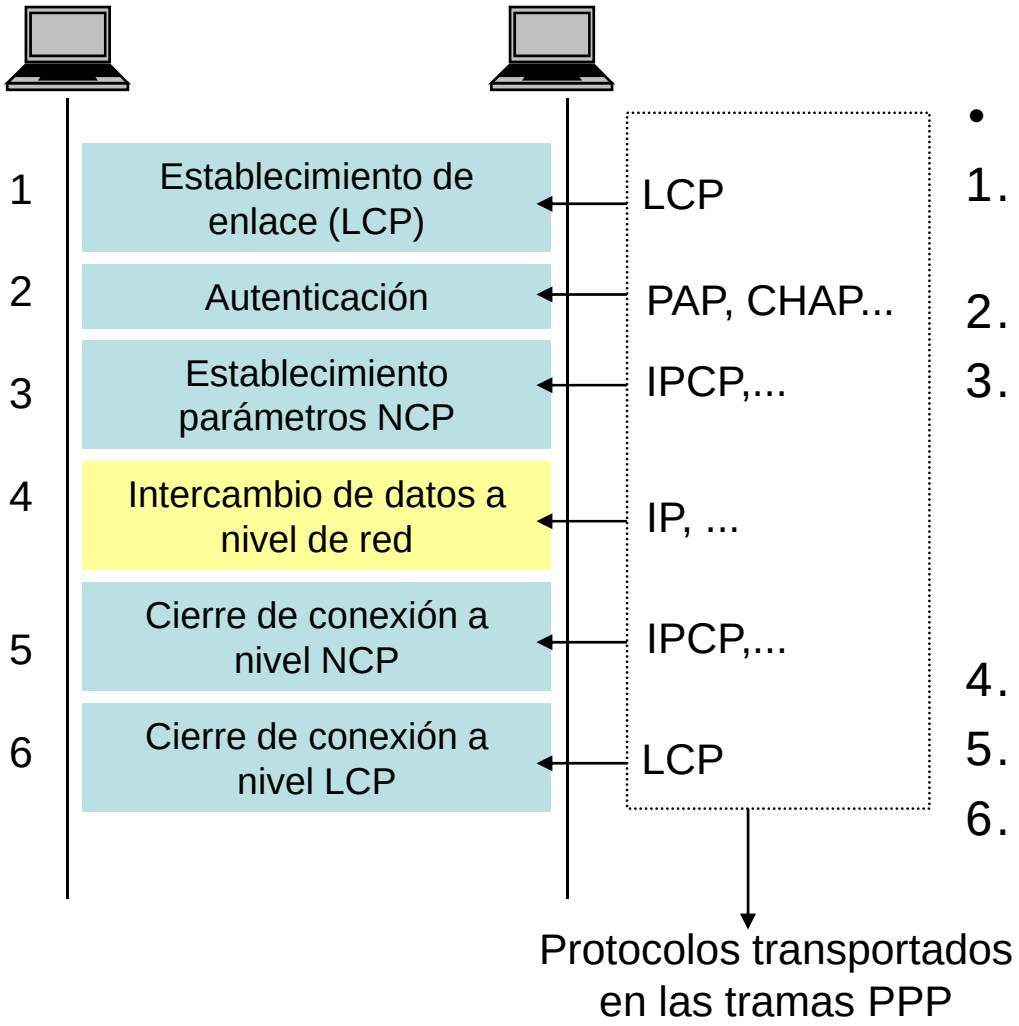
Componentes



- El protocolo PPP distribuye parte de sus funciones en otros protocolos. Componentes de PPP:
  - Protocolo LCP (*Link Control Protocol*) de control de enlace, encargado de la configuración de parámetros asociados al enlace físico, parámetros como el protocolo de autenticación a emplear, o protocolo de control de calidad del enlace a emplear.
  - Protocolos de autenticación (p.e PAP, CHAP).
  - Protocolos de control de calidad del enlace (p.e. LQR).
  - Un protocolo de la familia NCP (*Network Control Protocol*), para cada protocolo de nivel de red que utilice el enlace. Se encarga de la configuración de parámetros asociados a cada protocolo de nivel de red, e independientes de la capa física (p.e. posible asignación dinámica de dirección IP).

Point-to-Point Protocol (IX)

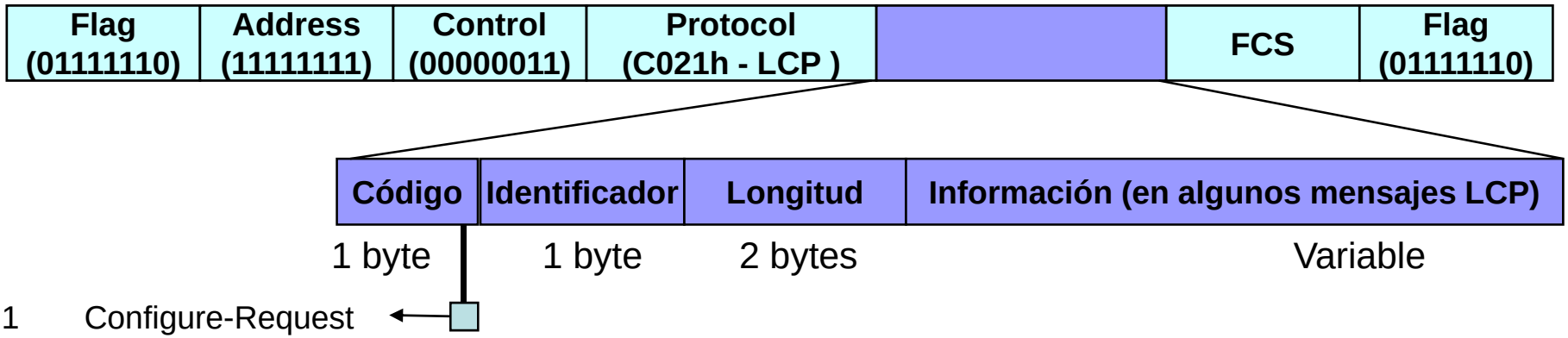
Fases de una conexión PPP



- Fases de la conexión:

1. Configuración parámetros de enlace.
2. Autenticación (opcional).
3. Configuración de parámetros de nivel de red. Tantos procesos en paralelo como protocolos de nivel de red utilizarán el enlace.
4. Intercambio de datagramas.
5. Cierre de conexiones NCP.
6. Cierre de conexión LCP.

LCP (*Link Control Protocol*) (I)



1	Configure-Request
2	Configure-Ack
3	Configure-Nak
4	Configure-Reject
5	Terminate-Request
6	Terminate-Ack
7	Code-Reject
8	Protocol-Reject
9	Echo-Request
10	Echo-Reply
11	Discard-Request
12	RESERVED

- **Código**: Tipo de mensaje LCP.
- **Identificador**: Utilizado para asociar mensajes LCP de consulta (REQUEST) y mensajes LCP de respuesta (ACK, NACK, REJECT).
- **Longitud**: Tamaño del mensaje LCP.
- **Información**: Datos auxiliares, necesarios en algunos mensajes LCP.

LCP (*Link Control Protocol*) (II)

Cód.	Tipo de mensaje	Descripción
1	CONFIGURE-REQUEST	Propuesta de lista de opciones de nivel de enlace
2	CONFIGURE-ACK	Acepta todas las opciones propuestas
3	CONFIGURE-NAK	Anuncia que alguna de las opciones no es aceptada
4	CONFIGURE-REJECT	Anuncia que alguna de las opciones no es reconocida
5	TERMINATE-REQUEST	Solicita el cierre de conexión
6	TERMINATE-ACK	Acepta el cierre de conexión
7	CODE-REJECT	Anuncia recepción de mensaje LCP de código desconocido
8	PROTOCOL-REJECT	Protocolo desconocido
9	ECHO-REQUEST	Tipo de mensaje HELLO para comprobar la actividad del otro extremo
10	ECHO-REPLY	Respuesta el ECHO-REQUEST
11	DISCARD-REQUEST	Solicitud de descartar un mensaje LCP

- Mensajes 1...4: Empleados durante el establecimiento del enlace.
- Mensajes 5 y 6: Empleados durante el cierre del enlace.
- Mensajes 7...11: Empleados en el mantenimiento y prueba del enlace.

LCP (*Link Control Protocol*) (III)

Fase de establecimiento de conexión

- Fase de establecimiento: Ambos extremos A y B configuran los parámetros de enlace simultáneamente.
- Extremo A:
  1. Envío de mensaje CONFIGURE-REQUEST con las opciones deseadas. Para las opciones no incluidas, se asumen los valores por defecto. Campo IDENTIFIER con valores distintos (consecutivos) para cada CONF-REQ.
  2. B responde a la consulta:
     - Si B acepta todas las opciones => envía CONFIGURE-ACK, con mismo identificador.
     - Si B no acepta alguna de las opciones =>envía CONFIGURE-NAK con mismo identificador. Se incluyen las opciones NO aceptadas.
     - Si B no entiende alguna de las opciones => envía CONFIGURE-REJECT.
  3. Ante un CONFIGURE-NAK, se genera un nuevo mensaje CONFIGURE-REQUEST (incrementando el identificador), variando las opciones no aceptadas. Si A no puede aceptar el variar esas opciones, se termina el establecimiento del enlace, que no puede producirse.
- Extremo B: proceso en paralelo, similar y simétrico.

LCP (*Link Control Protocol*) (IV)

Fase de establecimiento de conexión

- *Maximum Receive Unit* (MRU): En el mensaje CONFIGURE-REQUEST se especifica el tamaño máximo del campo de datos que se está dispuesto a recibir. Valor por defecto: 1500 bytes.
- Protocolo de Autenticación: En el mensaje CONFIGURE-REQUEST, se especifica el protocolo de autenticación con el que se quiere que el otro extremo se autentique (el otro extremo debe por tanto enviar su *login* y *password*). Opciones disponibles:
  - Autenticación mediante protocolo PAP.
  - Autenticación mediante protocolo CHAP.
  - No es necesaria autenticación (opción por defecto).

LCP (*Link Control Protocol*) (V)

Fase de establecimiento de conexión

- Protocolo de control de la calidad del enlace: El extremo que envía el mensaje CONFIGURE-REQUEST, indica que espera recibir información de calidad del enlace desde el otro extremo, siguiendo el protocolo indicado.
  - Opciones disponibles: *Link Quality Report* (LQR).
  - Ninguno (opción por defecto).
- Otras opciones (ver RFC 1570, “PPP LCP Extensions”):
  - Compresión del campo protocolo (de 2 bytes a 1). Opción desactivada por defecto.
  - Compresión de los campos Dirección y Control.
  - Utilización de CRC de 32 bits (opción por defecto: CRC 16 bits).
  - ...

LCP (*Link Control Protocol*) (VI)

Fase de mantenimiento de la conexión

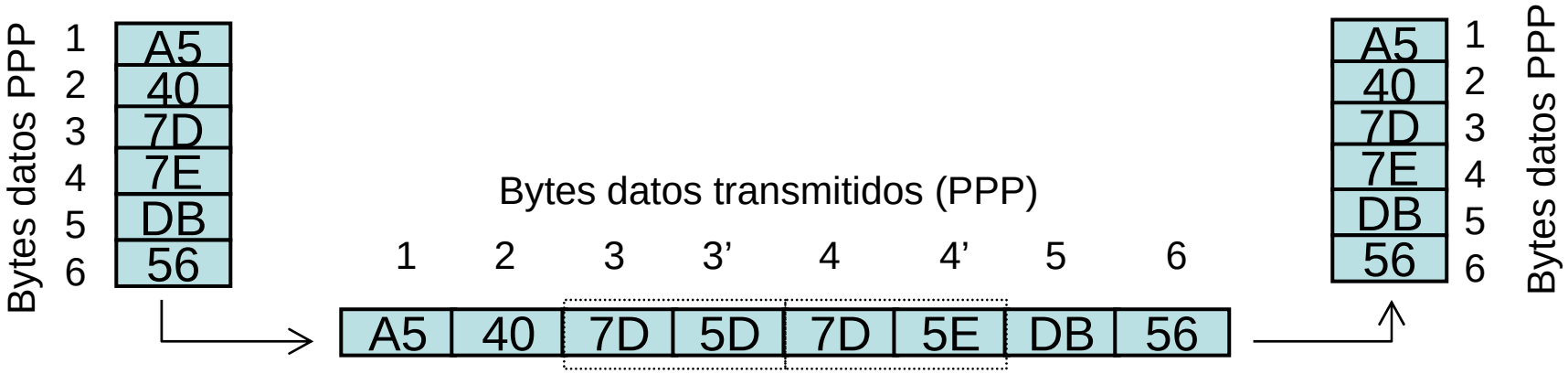
7	CODE-REJECT	Anuncia recepción de mensaje LCP de código desconocido
8	PROTOCOL-REJECT	Protocolo desconocido
9	ECHO-REQUEST	Tipo de mensaje HELLO para comprobar la actividad del otro extremo
10	ECHO-REPLY	Respuesta el ECHO-REQUEST
11	DISCARD-REQUEST	Solicitud de descartar un mensaje LCP

- Mensajes CODE-REJECT, PROTOCOL-REJECT: Enviados por un extremo ante la recepción de una trama PPP de formato desconocido.
- ECHO-REQUEST / ECHO-REPLY: Un extremo que recibe un ECHO-REQUEST, debe responder mediante un ECHO-REPLY. Usados por algunos protocolos de prueba de la calidad del enlace, y como comprobación de actividad.
- DISCARD-REQUEST: Usados en determinadas situaciones para la prueba de un enlace. El receptor de este mensaje, debe simplemente descartarlo.

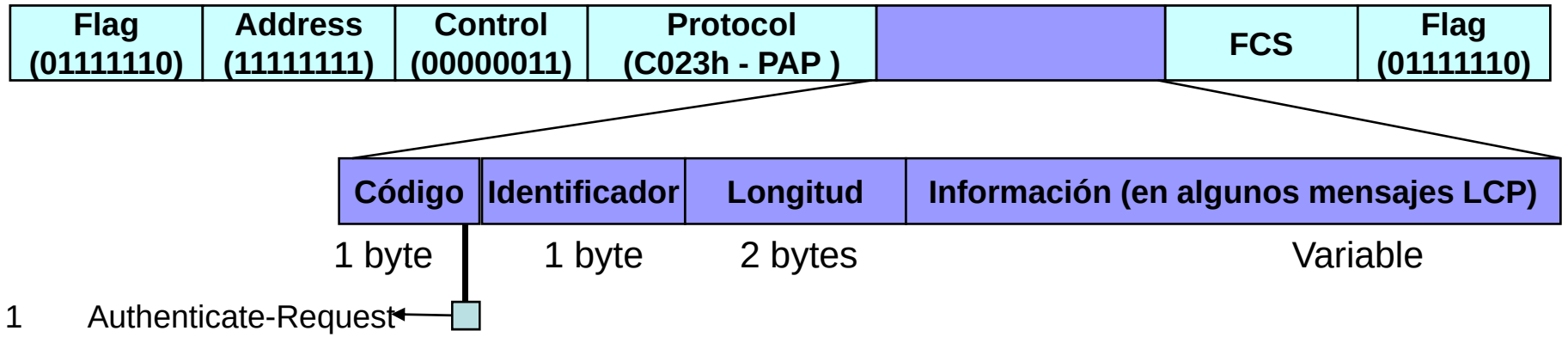
LCP (*Link Control Protocol*) (VII)

Mecanismo de transparencia PPP

- PPP emplea un mecanismo de transparencia a nivel de byte, con carácter de escape ESC = 0x7D.
- Transmisión de carácter *c* del campo de datos:
  - Si *c*==0x7E (flag) => enviar {0x7D,XOR(*c*,0x20)}.
  - Si *c*==0x7D (ESC) => enviar {0x7D,XOR(*c*,0x20)}
  - Sino: enviar *c*.



PAP (*Password Authentication Protocol*) (I)



- **Código**: Tipo de mensaje PAP.
- **Identificador**: Utilizado para asociar mensajes PAP de envío de login,password (AUTH-REQUEST) y mensajes PAP de respuesta (ACK, NACK).
- **Longitud**: Tamaño del mensaje PAP.
- **Información**: Datos auxiliares.

PAP (*Password Authentication Protocol*) (II)

usuario: pedro
clave: aqui

Servidor de acceso (BD claves)
usuario: pedro, clave: aqui
usuario: alicia, clave: xxyc

Negociación LCP [C-R: Autentic = PAP]

A-REQ (ID=1) {login=pedro, password=aquino}

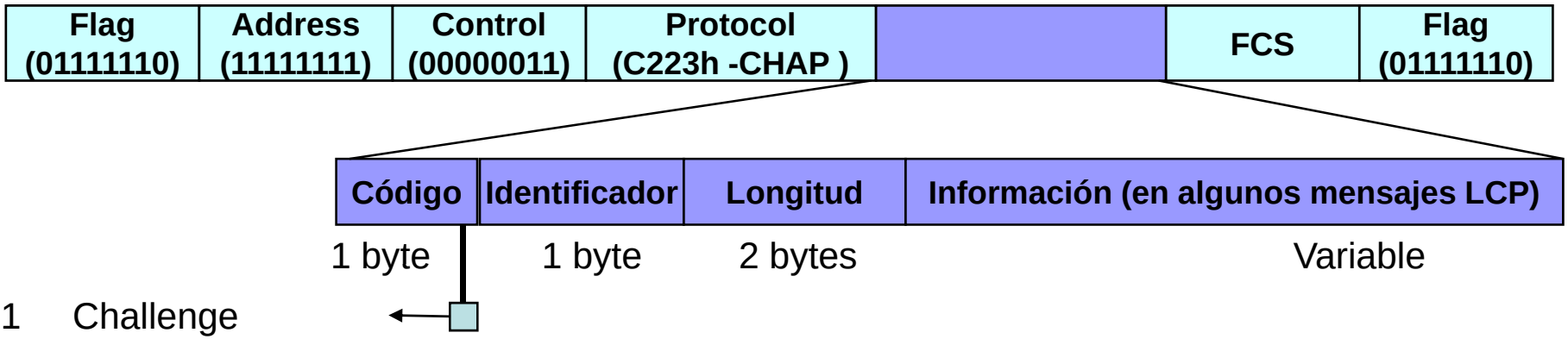
A-NAK (ID=1) {mensaje de texto (opcional)}

A-REQ (ID=2) {login=pedro, password=aqui}

A-ACK (ID=2) {mensaje de texto (opcional)}

- Los mensajes AUTH-REQUEST envían en texto claro la clave del usuario.
- El otro extremo comprueba el par {login,password} enviado, y acepta (A-ACK), a rechaza (A-NAK) la autenticación.
- El establecimiento del enlace no progresa si no se supera la autenticación.

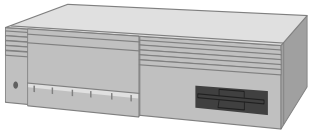
CHAP (*Challenge-Handshake Authentication Protocol*) (I)



1 Challenge
2 Response
3 Success
4 Failure

- **Código**: Tipo de mensaje CHAP.
- **Identificador**: Utilizado para asociar mensajes CHAP de solicitud y respuesta.
- **Longitud**: Tamaño del mensaje PAP.
- **Información**: Datos auxiliares.

CHAP (*Challenge-Handshake Authentication Protocol*) (II)



usuario: pedro
clave: aqui

Servidor de acceso (BD claves)
usuario: pedro, clave: aqui
usuario: alicia, clave: xxyc

- El usuario no envía su clave en texto claro por la red.
- Se utiliza la clave, mediante una función *hash,* para crear una nueva cadena que se envía al servidor.

Negociación LCP [C-R: Autentic = CHAP]

El servidor genera una cadena aleatoria

Challenge (ID=1) {c=cadena de caract. aleatoria}

El usuario devuelve la función *hash* de la cadena {ID,c,password}

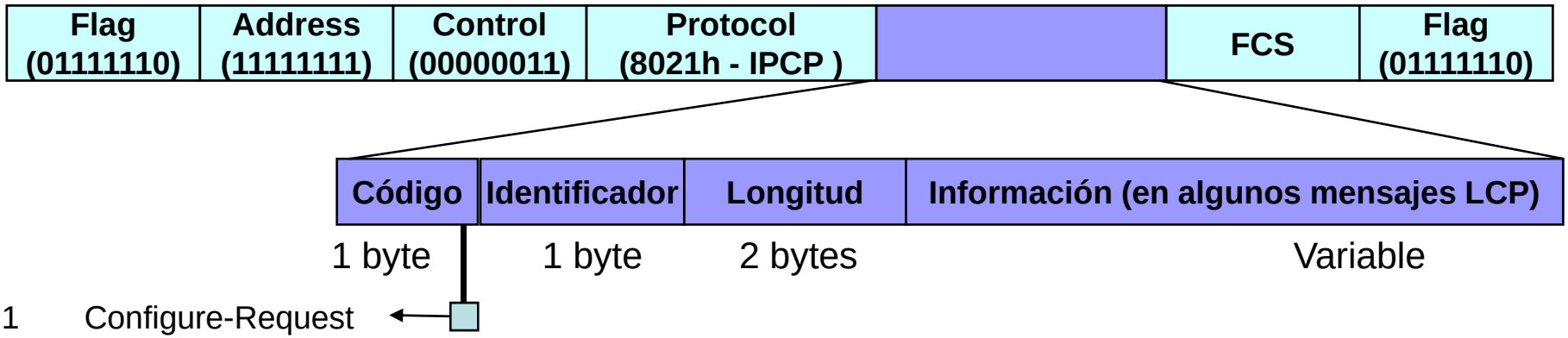
Response (ID=1) {*hash*(ID,c,"aqui")}

Success (ID=1) {mensaje de texto (opcional)}

El servidor comprueba el resultado enviado, con el mismo cálculo realizado con la clave de usuario almacenada

Nota: Opcionalmente, el proceso de autenticación puede repetirse en otros momentos tras el establecimiento de la conexión.

IPCP (*IP Control Protocol*) (I)



- La fase de establecimiento NCP no comienza hasta que no finaliza con éxito el establecimiento LCP, la autenticación y la medición de calidad del enlace, si las hubiera.
- Si el enlace se emplea para el intercambio de datagramas IP, el protocolo NCP que gobierna la configuración de parámetros es IPCP.
- Campos *Código*, *Identificador*, *Longitud* e *Información*, tienen similares objetivos al de los mensajes LCP.

IPCP (*IP Control Protocol*) (II)

Cód.	Tipo de mensaje	Descripción
1	CONFIGURE-REQUEST	Propuesta de lista de opciones IPCP
2	CONFIGURE-ACK	Acepta todas las opciones propuestas
3	CONFIGURE-NAK	Anuncia que alguna de las opciones no es aceptada
4	CONFIGURE-REJECT	Anuncia que alguna de las opciones no es reconocida
5	TERMINATE-REQUEST	Solicita el cierre de conexión a nivel IPCP
6	TERMINATE-ACK	Acepta el cierre de conexión a nivel IPCP
7	CODE-REJECT	Anuncia recepción de mensaje IPCP de código desconocido

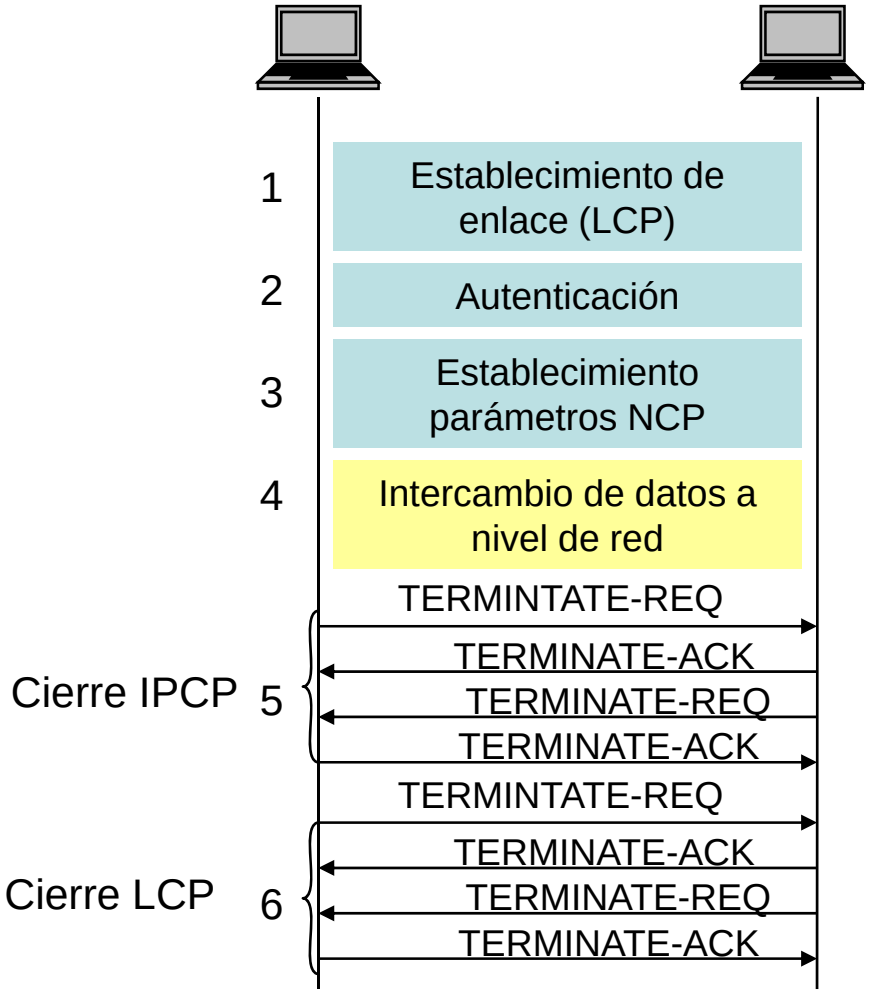
- Fase de establecimiento: Ambos extremos A y B configuran los parámetros de enlace simultáneamente, mediante mensajes CONFIGURE-REQUEST/ACK/NAK/REJECT, de manera similar a como se realiza la negociación a nivel LCP.
- No puede existir intercambio de datagramas hasta que no se cierre con éxito (C-ACK) la fase de negociación IPCP.
- Las diferencias LCP-IPCP existen en las opciones que se configuran.

IPCP (*IP Control Protocol*) (III)

Opciones configurables

- *IP-Compression-Protocol:* Permite negociar la utilización de compresión de datagramas IP. P.e. compresión mediante protocolo Van Jacobson. La opción por defecto es “no compresión”.
- *IP-Address*: Permite negociar la dirección IP que se va a utilizar en el extremo local (extremo A, que envía el CONFIGURE-REQUEST).
  1. El extremo A puede anunciar la dirección IP que desea tener.
     - El extremo A envía esta información en un mensaje C-REQUEST, con la dirección IP local (A). El extremo B puede aceptar (C-ACK) o rechazar esta petición. Dentro del mensaje C-NAK, B puede sugerir una dirección que aceptaría.
  2. El extremo A puede pedir al otro extremo que le asigne una dirección IP.
     - El extremo A envía esta información en un mensaje C-REQUEST, con la dirección IP 0.0.0.0. El extremo B debe rechazar la opción (C-NAK), incluyendo una dirección IP en la respuesta.
  3. Opción por defecto: no se asigna ninguna dirección IP durante el establecimiento IPCP.
- La negociación IPCP se produce simultánea e independientemente en ambos sentidos (al igual que LCP).
- Otras opciones: puede consultarse una lista actualizada en la RFC 1060.

Cierre de conexión PPP



- El cierre de conexión se realiza mediante los mensajes TERMINATE-REQUEST y TERMINATE-ACK.
- En un cierre ordenado:
  - Cada protocolo NCP realiza un cierre ordenado.
  - A continuación, existe un cierre ordenado LCP.

Bibliografía

- James **Carlson,** "PPP Design, Implementation, and Debugging", 2nd Edition, James Carlson, Addison Wesley, 2000. **Capítulos 1, 2, 3, 4, y 5.**
- http://www.workingcode.com/ppp/reference.html
- *Request For Comments* (RFC):
  - (1994) RFC 1661: The Point-to-Point Protocol (PPP).
  - (1994) RFC 1570: PPP LCP Extensions.
  - (1992) RFC 1332: The PPP Internet Protocol Control Protocol (IPCP).
  - (1992) RFC 1334: PPP Authentication Protocols.
  - (1996) RFC 1994: PPP Challenge-Handshake Authentication Protocol (CHAP).
  - (1996) RFC 1989: PPP Link Quality Monitoring.