Network Infrastructures A.A. 2009-2010

Point-to-Point Protocol (PPP)

Name of authors:

Ianiri Giuseppe

Mangione Daniele

Masi Giacomo

I. Slip: il predecessore di PPP

Il **Serial Line Internet Protocol** (**SLIP**), è un protocollo di rete, ormai superato, di Livello datalink, usato, in passato, per stabilire connessioni dirette tra due nodi, specie su connessioni seriali asincroni via modem.

Incapsula in uno specifico frame gli IP datagram che vengono trasmessi sequenzialmente sulla linea seriale. Ogni frame è formato aggiungendo il carattere Ascii END (0xc0) in testa ad ogni IP datagram.

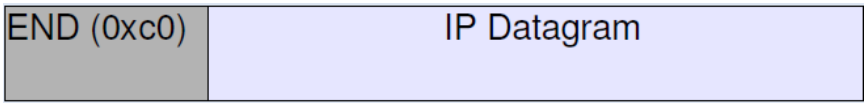

FIGURA 1: FRAME SLIP

Le caratteristiche principali del protocollo slip sono:

- Può essere utilizzato solo per incapsulare IP datagram.
- Non prevede l'utilizzo di checksum per il controllo d'integrità.
- E' necessario conoscere a priori l'indirizzo IP del destinatario (no ARP).
- I frame vengono spediti immediatamente senza alcun protocollo di negoziazione di parametri.
- Semplice e di facile implementazione.

Attualmente il protocollo Slip è utilizzato da BlueCore Serial Protocol per la comunicazione fra moduli diversi e un computer utilizzando un collegamento wireless Bluetooth.
Oggi il protocollo Slip è stato sostituito dal protocollo PPP in quanto le sue caratteristiche erano troppo limitate per gestire l'incremento delle connessioni ad Internet.

II. Introduzione al PPP

Il protocollo PPP (Point to Point Protocol) è un protocollo di rete di Livello datalink che consente, analogamente a Slip, la trasmissione di pacchetti IP su linea seriale ma in modo più flessibile per facilitare il collegamento tra apparati di rete eterogenei.
I collegamenti che utilizza PPP sono Full-Duplex e si assume che la consegna dei pacchetti avvenga in maniera ordinata.
Viene utilizzato con l'ADSL per la sua capacità di connettere direttamente un utente al Central Office del suo ISP, inoltre fornisce autenticazione (Protocolli PAP e CHAP), riconoscimento di errori, configurazione automatica delle interfacce di rete e supporta l'indirizzamento IP dinamico (DHCP – Dynamic Host Configuration Protocol).
Il protocollo PPP fornisce un metodo standard per il trasporto di datagrammi multi-protocollo su un collegamento punto-punto.
PPP è costituito da tre componenti fondamentali.

1. Un metodo per incapsulare pacchetti muti-protocollo;
2. Un Link Control Protocol (LCP) per stabilire, configurare e testare la connessione data-link;
3. Una famiglia di Network Control Protocols (NCPs) per stabilire e configurare differenti network layer protocols;

Esistono due tipi di incapsulamento per i frame PPP: **PPPoE** (PPP over Ethernet) , **PPPoA** (PPP over ATM).

Incapsulamento

L'incapsulamento PPP permette il multiplexing di differenti network-layer protocol simultaneamente sullo stesso link. L'incapsulamento PPP è stato progettato per essere compatibile con i supporti hardware usati più comunemente.

Solo 8 byte addizionali sono necessari per formare l'incapsulamento.

Per supportare implementazioni ad alta velocità, l'incapsulamento di default usa solo campi semplici , di cui uno solo dei quali ha bisogno di essere esaminato per il de multiplexing.

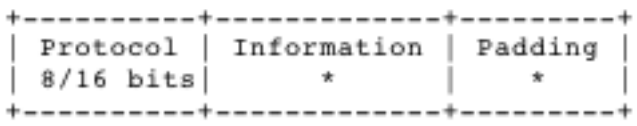

Figura 25: PPP encapsulation

PROTOCOL

È uno o due byte e il suo valore identifica il datagramma incapsulato nel campo Information del pacchetto.

Un esempio dei valori più importanti:

c021:Link Control Protocol

c023: Password Authentication

c025: Protocol Link Quality Report

c223: Challenge Handshake Authentication Protocol

INFORMATION

Il campo Information contiene il datagramma del protocollo specificato nel campo Protocol.

La lunghezza massima per il campo Information, incluso Padding, ma non il campo Protocol, è determinato dalla Maximum Receive Unit (MRU), che di default è 1500 Byte.

PADDING

Il campo Information può essere esteso con un numero arbitrario di Byte per raggiungere la dimensione definita da MRU. È responsabilità di ogni protocollo distinguere fra i byte di informazione reale da quelli di padding.

Network Control Protocols

NCP è una famiglia di protocolli che gestiscono le specifiche richieste dei rispettivi protocolli di livello network al fine di risolvere alcuni problemi, quali ad esempio la gestione ed assegnazione degli indirizzi IP.

Alcuni esempi di protocolli NCP sono IPCP (per IP), IPv6CP (per IPv6) i quali usano lo stesso formato di pacchetto di LCP

Fasi di PPP

Nel processo di configurazione, gestione e terminazione del collegamento punto-punto, il collegamento PPP passa attrverso diverse fasi esplicitate dal seguente diagramma degli stati:

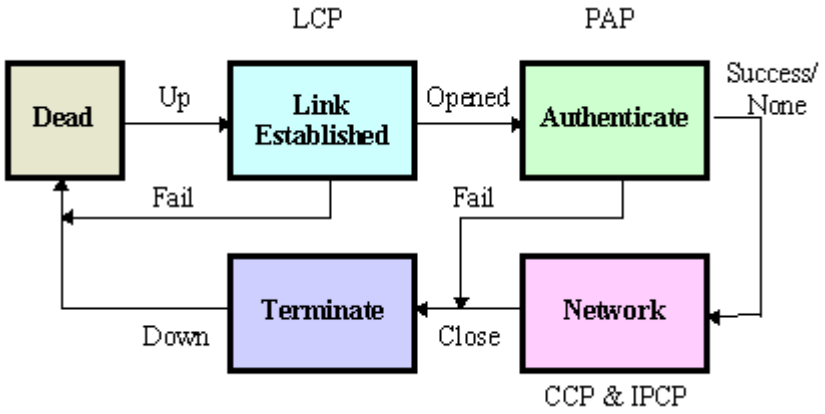

Figura 16: diagram of PPP phases

Link Dead

Il collegamento necessariamente inizia e finisce con questa fase. Quando un evento esterno indica che il livello fisico è pronto per essere usato PPP procede alla fase Link Establishment.

Link Establishment Phase

Il Link Control Protocol (LCP) è usato per stabilire una connessione tramite lo scambio di Configuration packet. Questo scambio è completo, e quindi la connessione è stabilita, quando un Configure-Ack packet (descritto in seguito) è stato sia inviato che ricevuto.

Solo opzioni di configurazione che sono indipendenti dal particolare network layer protocol sono configurati da LCP, mentre quelle dipendenti dal particolare network layer protocol sono gestiti da Network Control Protocols (NCPs) durante la fase Network-Layer Protocol.

La ricezione di un LCP Configure-Request causa il ritorno alla fase Link Establishment dalla fase Network-Layer Protocol o dalla fase Authentication.

Authentication Phase

Su alcuni collegamenti può essere desiderabile richiedere ai peer di autenticarsi prima di consentire lo scambio di pacchetti di protocollo network-layer. Di default, l'autenticazione non è obbligatoria. (trattata in seguito)

Network-Layer Protocol Phase

Una volta finita la fase precedente, ogni network-layer protocol (ad esempio IP) deve essere configurato separatamente dall'appropriato Network Control Protocol (NCP) e PPP può trasportare i pacchetti di questo protocollo. Durante questa fase, il traffico consiste di ogni possibile combinazione di pacchetti LCP, NCP, network-layer protocol.

Link Termination Phase

PPP può terminare il collegamento in ogni istante. Questo può avvenire a causa della perdita della portante, fallimento dell'autenticazione, fallimento dei livelli di qualità richiesti al collegamento, chiusura ecc.

LCP è usato per chiudere la connessione attraverso uno scambio di pacchetti di terminazione. Quando il collegamento è in chiusura, PPP informa i protocolli network-layer affinché prendano azioni appropriate.

Alcune implementazioni dopo lo scambio di pacchetti di terminazione può segnalare al livello fisico di disconnettersi al fine di imporre la terminazione del collegamento.

Formato pacchetti LCP

Ci sono tre classi di pacchetti LCP:

1. Link Configuration packets usati per stabilire e configurare un colegamento(Configure-Request, Configure-Ack, Configure-Nak and Configure-Reject).
2. Link Termination packets usato per terminare il colegamento (Terminate- Request and Terminate-Ack).
3. Link Maintenance packets usati per la gestione (Code-Reject, Protocol-Reject, Echo-Request, Echo-Reply, and Discard-Request).

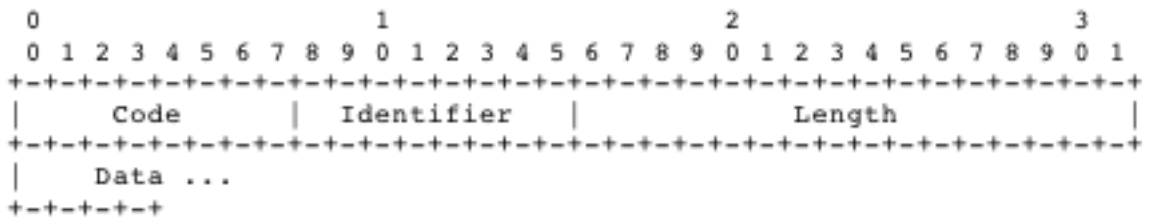

FIGUra 17:Pacchetto LCP

I valori che può assumere il campo Code identificano i tipi di pacchetti

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

Configure-Request

Per aprire una connessione occorre trasmettere un pacchetto Configure-Request che contiene la lista di zero o più Configuration Options che si desidera negoziare. Tutte le opzioni di configurazione sono sempre negoziati simultaneamente.

Configure-Ack
Se ogni Configuration Options ricevuta in un Configure-Request è ammissibile e tutti i valori sono accettabili, si deve trasmettere un pacchetto Configure-Ack contenente la lista delle opzioni accettate.

Configure-Nak
Se ogni Configuration Options ricevuta in un Configure-Request è ammissibile ma alcuni valori sono non accettabili, si deve trasmettere un pacchetto Configure-Nak contenente la lista delle opzioni non accettate.

Configure-Reject
Se qualche Configuration Options ricevuta in un Configure-Request è non ammissibile o i valori sono non accettabili, si deve trasmettere un pacchetto Configure-Reject contenente la lista delle opzioni rigettate.

Terminate-Request & Terminate-Ack
LCP include Terminate-Request e Terminate-Ack per fornire un meccanismo di chiusura della connessione. Se si vuole chiudere una connessione bisogna trasmettere u pacchetto Terminate-Request ripetendo l'invio finché non si riceve un pacchetto Terminate-Ack o il livello sottostante indica che è andato giù o si è ripetuto un numero elevato di volte per cui si può assumere con ragionevolezza che il peer è giù.
Alla ricezione di un Terminate-Request un Terminate-Ack deve essere inviato.

Code-Reject
La ricezione di un pacchetto LCP con un valore per il campo Code sconosciuto sta ad indicare che il peer sta operando con una differente versione del protocollo occorre dunque segnalarlo inoltrandogli un pacchetto Code-Reject.

Protocol-Reject
La ricezione di un pacchetto PPP conil valore del campo Protocol sconosciuto sta ad indicare che il peer sta tentando di usare un protocollo che non è supportato, occorre dunque segnalarglielo inoltrando un pacchetto Protocol-Reject che conterrà il protocollo non supportato.

Echo-Request & Echo-Reply
LCP include Echo-Request e Echo-Reply al fine di fornire un meccanismo di loopback di livello Data Link da utilizzare su entrambe le direzioni di un collegamento al fine di eseguire test su qualità, performance e numerose altre funzioni.
Pacchetti Echo-Request ed Echo-Reply possono essere usati solo nello stato Opened di LCP, alla ricezione di Echo-Request si deve rispondere con Echo-Reply.
Sono questi pacchetti che trasmettono il Magic-Number utilizzato per determinare quali collegamenti sono nella condizione looped-back.

Discard-Request
Il pacchetto Discard-Request può essere inviato soltanto nello stato Opened di LCP e ha lo scopo di fornire uno strumento per eseguire test su qualità, performance lato locale di un collegamento verso remoto.
Alla ricezione di un pacchetto Discard-Request bisogna solo eliminarlo senza alcuna segnalazione.

LCP Configuration Options

LCP Configuration Options consente di modificare le caratteristiche di default di un link punto-punto.

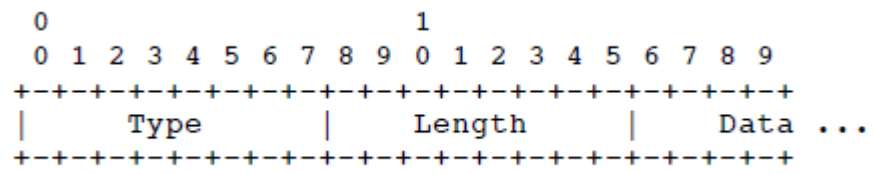

Figura 18: Formato delle Configuration Options

TYPE: indica il tipo di Configuration Option

0 RESERVED
1 Maximum-Receive-Unit
3 Authentication-Protocol
4 Quality-Protocol
5 Magic-Number
7 Protocol-Field-Compression
8 Address-and-Control-Field-Compression

LENGHT: indica la lunghezza di tutta la Configuration Option.
DATA: contiene informazioni specifiche della configuration option.

Maximum-Receive-Unit (MRU)
Questa Configuration Option può essere inviata per informare i peer che si possono ricevere pacchetti più grandi o per richiedere che i peer inviino pacchetti più piccoli. Il valore di default è 1500 byte.
Il campo Data relativo a questa Configuration Option specifica il numero massimo di byte consentiti per i campi information e padding.

Authentication-Protocol
Questa Configuration Option fornisce un metodo per decidere l'utilizzo di uno specifico protocollo di autenticazione.

Quality-Protocol
Questa Configuration Option fornisce un metodo per stabilire l'utilizzo di un protocollo per il monitoraggio della qualità del link. Ovvero che sia in grado di determinare quando e quanto spesso i dati sul link vengono scartati.

Magic-Number
Questa Configuration Option fornisce un metodo per rilevare la presenza di loop-back sui link o altre anomalie sempre a livello Data Link.
Il valore da assegnare al Magic-Number deve essere scelto nel modo più casuale possibile.
Quando viene ricevuta una Configure-Request con una Configuration Option di tipo Magic-Number, tale Magic-Number viene confrontato con quello dell'ultima Configure-Request inviata al peer. Se i due Magic-Number differiscono allora non è presente un loop-back sul link e dovrebbe essere inviato un ack relativo al Magic-Number. Se invece sono uguali allora è possibile che ci sia un loop-back sul link, e che l'attuale Configure-Request sia effettivamente l'ultima inviata.

Protocol-Field-Compression
Questa Configuration Option fornisce un metodo per stabilire la compressione del campo Protocol.

Address-and-Control-Field-Compression
Questa Configuration Option fornisce un metodo per stabilire la compressione dei campi Data Link Layer Address e Control. Tali campi non devono essere compressi quando si invia qualsiasi pacchetto LCP.
Quando questi campi sono compressi il campo FCS (Frame Control Sequence) di livello Data Link è calcolato sul frame compresso.

III. ATM-Asynchronous Transfer Mode

ATM è un protocollo di rete che incapsula il traffico in celle a lunghezza fissa. Una cella ha una dimensione di 53 byte, di cui 48 di payload (corpo) e 5 di header.
ATM utilizza una tecnica di commutazione a circuito virtuale: prima di inviare i dati si invia un pacchetto di *handshake* per configurare la connessione. Man mano che questo pacchetto attraversa gli switch, questi calcolano l'instradamento, attribuiscono un identificatore ai pacchetti di questa connessione, e riservano risorse per la connessione. A questo punto tutti i successivi pacchetti della connessione seguiranno lo stesso percorso.
Le celle successive verranno identificate sulla base di un'etichetta. Quando una cella raggiunge uno switch, questo dovrà consultare una tabella indicizzata da porta in ingresso ed etichetta, ricavando la porta di uscita e la nuova etichetta da assegnare alla cella. Questa architettura molto semplice facilita l'instradamento in hardware, permettendo di realizzare switch ad alta velocità.
L'etichetta è composta di due valori presenti nell'header di ciascuna cella, VPI e VCI:

- Il VPI (*Virtual Path Identifier*) identifica il *path* virtuale su cui il circuito virtuale è stato attivato.
- Il VCI (*Virtual Channel Identifier*) identifica il canale virtuale su cui il circuito virtuale è stato attivato.

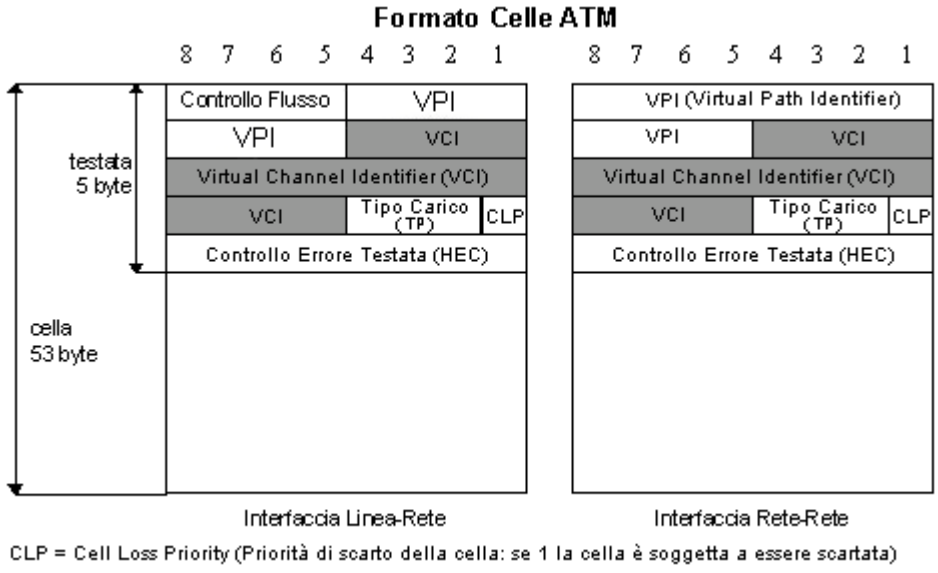

FIGURA 3: CELLE ATM

Gerarchicamente si ha che un circuito virtuale viene stabilito tramite il collegamento di più connessioni virtuali VC. Il VP è un canale virtuale gerarchicamente superiore al VC ed infatti un VP può contenere fino a 2^16 VC. Appositi dispositivi hardware, ad esempio switch ATM, sono in grado di gestire VP (con tutti i VC in essi contenuti) o anche direttamente i singoli VC.
Visto che tutti i pacchetti seguono la stessa strada, è garantita la consegna in ordine, ma non che tutti i pacchetti siano consegnati, perché sono sempre possibili code sugli switch e conseguenti perdite di pacchetti.
La velocità va da 2 a 622 Mbps, e anche oltre. È questa la velocità adatta alla tv ad alta definizione.
ATM, inoltre, consente di segmentare la banda sui diversi canali virtuali per i diversi tipi di servizi di trasmissione appunto tramite l'uso dei VCC (VPI:VCI)
Per supportare vari tipi di traffico su ATM (qualità di servizio), sono stati definiti una varietà di modelli di servizio, che si adattano al traffico telefonico (CBR: banda costante, forti garanzie su banda e ritardo) o a quello IP (VBR: banda variabile, nessuna garanzia).
Sono definiti anche molteplici strati di adattamento ("Adaptation Layer"), per permettere il trasporto su ATM di vari tipi di dati.

- **AAL – ATM Adaptation Layer**
  - "Adatta" i livelli superiori (IP o applicazioni ATM native) al livello ATM sottostante
  - Presente solo negli end system, non negli switch
  - Un segmento AAL (header, dati e trailer) viene frammentato in molteplici celle ATM

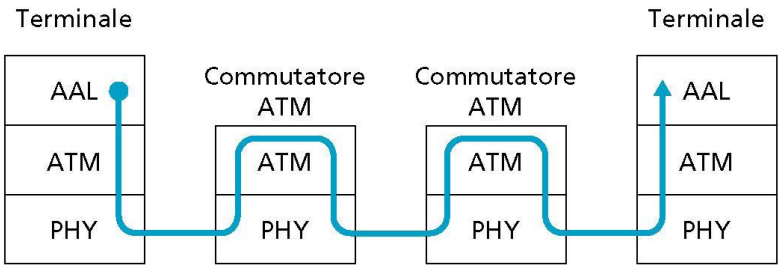

FIGURA 4:ARCHITETTURA PROTOCCOLLARE AAL

Differenti versioni del livello AAL, in funzione della
classe di servizio ATM:
AAL1: per i servizi CBR *(Constant Bit Rate)*:
Es: emulazione di circuito
AAL2: per i servizi VBR *(Variable Bit Rate)*:
Es: video MPEG
AAL5: per la trasmissione dati

- Usato solitamente per trasportare datagrammi IP
- Basso overhead
- E' suddiviso in 2 sottosrati:
  - CS (Convergence Sublayer):
    - Impiego di un codice CRC
    - Il payload è reso multiplo di 48 byte grazie ad un campo di riempimento (PAD)
  - SAR (Segmentation And Reassembly sublayer):
    - Celle dati AAL5 di grandi dimensioni vengono frammentate in celle ATM di 48 byte ciascuna

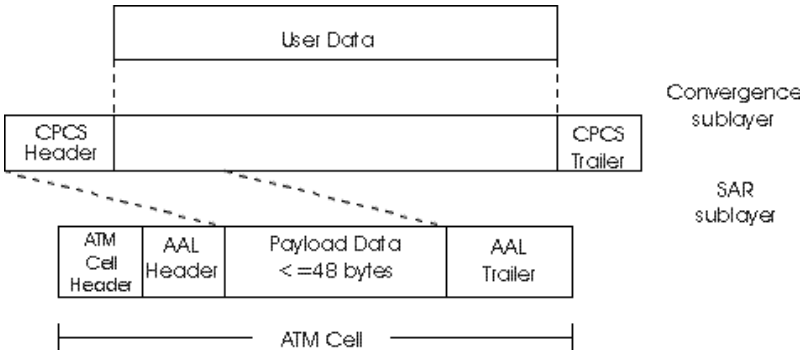

FIGURA 5: PACCHETTO AAL5

Per concludere, i principali vantaggi che ATM offre sono:

- Flessibilità - ATM può essere facilmente ampliata per servizi futuri
- Efficienza nell'utilizzo delle risorse a sua disposizione
- E' una rete universale ma semplice
- Riduzione dei costi di funzionamento, amministrazione e manutenzione
- Riduzione dei costi di trasporto (grazie alla multiplazione statistica)
- Disponibilità dinamica della banda

IV.PPPoA - PPP over ATM

PPPoA (Point-to-Point to Protocol **over ATM**) è un protocollo per il trasporto di frame PPP all'interno di celle ATM AAL5.
ATM AAL5 è stato progettato per fornire connessioni virtuali tra stazioni terminali della stessa rete; queste connessioni offrono un servizio di consegna dei pacchetti con rilevamento d'errore ma senza correzione.

Lo strato PPP tratta il sottostante strato ATM AAL5 come un link bit-synchronous punto-punto. In questo contesto il collegamento PPP corrisponde ad una connessione virtuale ATM AAL5. La connessione virtuale deve essere full-duplex, punto-punto, e può essere o dedicata (permanente) o commutata (impostata su richiesta).
Esistono due meccanismi per identificare il PDU (Protocol Data Unit):

- Virtual Circuit based multiplexing
- Logical Link Control (LLC) encapsulation

Nella prima tecnica il tipo di protocollo del payload è implicitamente accettato dagli end-point per ogni circuito virtuale che usa procedure di provisioning o control plane. Quando si usa invece la tecnica di incapsulamento LLC il tipo di protocollo è esplicitamente identificato tramite un header LLC.

Virtual Circuit Multiplexed PPP Over AAL5
Per trasportare un payload PPP su AAL5 si deve supportare un virtual circuit multiplexed PPP payload.

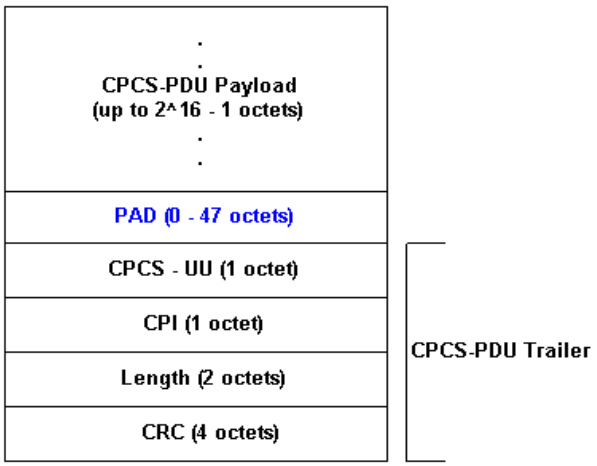

FIGURA 5: VIRTUAL CIRCUIT MULTIPLEXED PPP PAYLOAD.

- Il campo Payload contiene le informazioni utente.
- Il campo PAD adatta i CPCS-PDU per inserirli esattamente nelle celle dell'ATM.
- Il campo CPCS-UU non ha funzione in questo contesto.
- Il campo CPI (Common Part Indicator) allinea il CPCS-PDU a 64 bits.
- Il campo Lenght indica la lunghezza, in ottetti, del campo Payload.
- Il campo CRC protegge l'intero CPCS-PDU.

LLC Encapsulated PPP Over AAL5

Per trasportare un payload PPP su AAL5 si deve supportare un LLC encapsulated PPP payload su PVC.

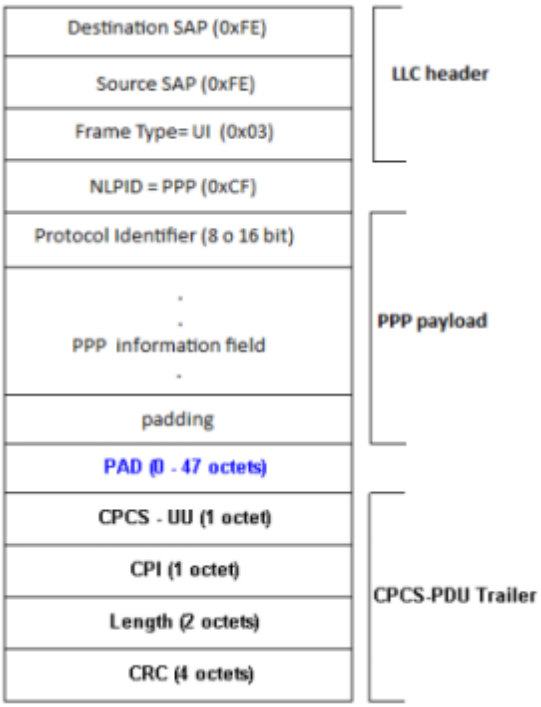

FIGURA 6: LLC ENCAPSULATED PPP PAYLOAD

- Destination SAP: Service Access Point di destinazione.
- Source SAP: Service Access Point di sorgente
- Frame Type: Frame d'informazione senza numerazione.
- NLPID: Network Layer Protocol Identifier.
- Protocol Identifier: Campo d'identificazione del protocollo di livello rete.
- PPP information field: Campo delle informazioni.

Gli end-point possono inviare altri protocolli incapsulati in LLC oltre a PPP attraverso la stessa connessione virtuale; ad eccezione però dei protocolli che hanno un NCP attivo al'interno della sessione PPP.

Quando viene stabilita una connessione a circuito virtuale commutato AAL5, deve essere specificato il meccanismo d'identificazione che si desidera utilizzare (VC-multiplexed PPP o LLC-encapsulated PPP o entrambi).

Quando una virtual connection si interrompe intervengono procedure di detection e recovery che possono ripristinare la connessione effettuando le seguenti transizioni di stato:

VC-multiplexed PPP → LLC encapsulated PPP

LLC encapsulated PPP → VC-multiplexed PPP

Generalmente le reti ATM sono basate su circuiti virtuali e la sicurezza è implicita nei servizi amministrativi del provider. La probabilità di una falla causata dal mancato routing di celle ATM è da considerare trascurabile.

Per le applicazioni che richiedono una sicurezza maggiore è consigliato l'utilizzo di header di autenticazione, payload criptati e/o servizi di sicurezza aggiuntivi a livello ATM.

V. PPPoE - PPP over Ethernet

PPP over Ethertnet consente di connettere una rete di host con un Access Concentrator remoto.

Per fornire una connessione punto-punto su Ethernet, ogni sessione PPP deve conoscere l'indirizzo Ethernet del peer remoto e in più deve stabilire un identificatore univoco di sessione.

PPPoE ha due fasi distinte: Discovery stage e PPP Session stage.

Quando un host vuole iniziare una sessione PPPoE, deve prima effettuare una fase di Discovery per identificare l'indirizzo MAC Ethernet remoto e stabilire un PPPoE SESSION_ID.

Mentre PPP definisce una relazione peer-to-peer, la fase di Discovery stabilisce una relazione client-server in cui un host (il client) trova un Access Concentrator (il server). A seconda della topologia della rete ci può essere più di un Access Concentrator con cui comunicare, quindi la fase di Discovery consente all'host di trovare tutti gli Access Concentrator e di selezionarne uno. Quando questa fase è conclusa con successo, sia l'host che l'Access Concentrator selezionato potranno effettuare la connessione point-to-point su Ethernet. Quando la sessione PPP è stabilita, l'host e l'Access Concentrator devono allocare le risorse per un'interfaccia PPP virtuale.

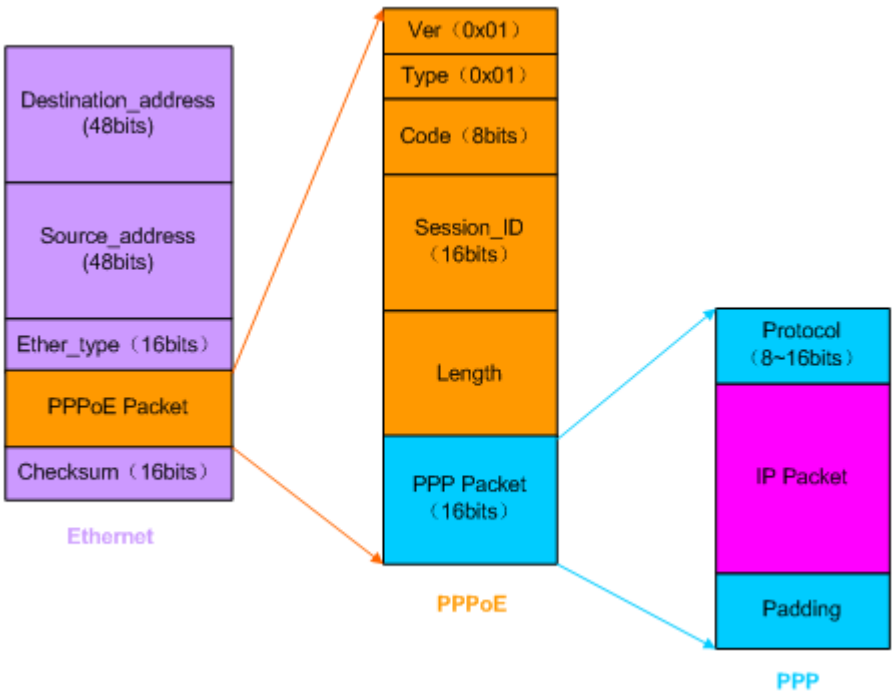

FIGURA 7: PACCHETTI ETHERNET/PPPOE

Pacchetto Ethernet:

- **DESTINATION_ADDR**: contiene i un indirizzo Ethernet unicast di destinazione oppure l'indirizzo Ethernet Broadcast (0xffffffff).
  Per il traffico della sessione PPP questo campo deve contenere l'indirizzo unicast del peer remoto che viene determinato nella fase di Discovery.

- **SOURCE_ADDR**: deve contenere l'indirizzo MAC Ethernet del dispositivo di origine.

- **ETHER_TYPE**: contiene il valore 0x8863 per la fase di Discovery oppure il valore 0x8864 per la fase di PPP Session.

Pacchetto PPPoE:

- **VER** (4 bit): deve essere settato a 0x1 per questa versione di PPPoE.

- **TYPE** (4 bit): deve essere settato a 0x1 per questa versione di PPPoE.

- **CODE** (8 bit): definito in seguito.

- **SESSION_ID** (16 bit): dati un SOURCE_ADDR ed un DESTINATION_ADDR identifica univocamente una sessione.

- **LENGHT** (16 bit): indica la lunghezza del payload PPPoE. Non include la lunghezza delle header Ethernet o PPPoE.

Discovery Stage

1. L'host invia in broadcast un Initiation Packet
2. Uno o più Access Concentrator inviano un Offer Packet
3. L'host invia in unicast un Session Request Packet
4. L'Access Concentrator selezionato invia un Confirmation Packet

Quando l'host riceve il Confirmation Packet può procedere alla fase di PPP Session.
Il payload PPPoE contiene zero o più TAG. I TAG sono costrutti TLV (type-lenght-value) definiti nel seguente modo:

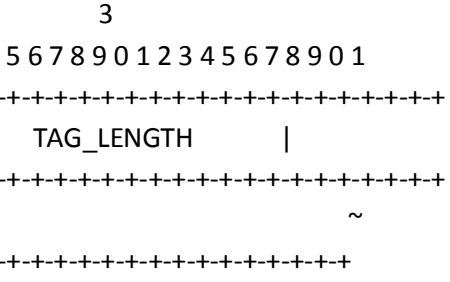

FIGURA 8: PAYLOAD PPPoE

TAG_TYPE (16 bit): contiene un valore che identifica il tipo di TAG.
TAG_LENGTH (16 bit): indica la lunghezza in byte del TAG_VALUE.
TAG_VALUE: ha una lunghezza variabile e può contenere diversi valori in base al tipo di TAG.

PPPoE Active Discovery Initiation (PADI) packet

L'host invia un PADI packet con il DESTINATION_ADDR settato all'indirizzo di broadcast. Il campo CODE è settato al valore 0x09 e il campo SESSION_ID deve essere settato a 0x0000.

Il PADI packet deve contenere esattamente un solo TAG di tipo Service-Name, che indica il servizio richiesto dall'host e un numero qualsiasi di altri tipi di TAG. Un PADI packet (compresa l'header PPPoE) non può superare i 1484 byte.

PPPoE Active Discovery Offer (PADO) packet

Quando l'Access Concentrator riceve un PADI, risponde inviando un PADO packet. Il campo DESTINATION_ADDR contiene l'indirizzo unicast dell'host che ha inviato il PADI packet.

Il campo CODE è settato al valore 0x07 e il campo SESSION_ID a 0x0000.

Il PADO packet deve contenere un AC-Name TAG che contiene il nome dell'Access Concentrator, un Service-Name TAG identico a quello del PADI packet ricevuto e altri Service-Name TAG che indicano i servizi che l'Access Concentrator offre.

Se l'Access Concentrator non può servire il PADI ricevuto, non deve rispondere con un PADO.

PPPoE Active Discovery Request (PADR) packet

Dal momento che il PADI viene inviato in broadcast, l'host può ricevere più di un PADO. L'host guarda quindi i PADO packet ricevuti e ne sceglie uno in base all'AC-Name oppure ai servizi offerti. L'host può allora inviare un PADR packet all'Access Concentrator che ha scelto. Il campo DESTINATON_ADDR è settato con l'indirizzo Ethernet dell'A.C. che ha inviato il PADO. Il campo CODE è settato a 0x19 e SESSION_ID a 0x0000.

Il PADR packet deve contenere esattamente un TAG di tipo Service-Name che indica il servizio richiesto dall'host e un numero qualsiasi di altri tipi di TAG.

PPPoE Active Discovery Session-confirmation (PADS) packet

Quando un Access Concentrator riceve un PADR packet, è pronto ad iniziare una sessione PPP. Esso genera un SESSION_ID univoco per la sessione PPPoE e risponde all'host con un PADS packet. Il campo DESTINATION_ADDR contiene l'indirizzo Ethernet dell'host che ha inviato il PADR packet. Il campo CODE è settato a 0x65 e SESSION_ID deve essere settato al valore univoco generato per tale sessione PPPoE.

Il PADS packet contiene esattamente un TAG di tipo Service-Name, che indica il servizio in base al quale l'Access Concentrator ha accettato la sessione PPPoE, e un numero qualsiasi di altri tipi di TAG.

Se l'Access Concentrator non può fornire il servizio indicato nel PADR, allora deve rispondere con un PADS contenente un TAG di tipo Service-Name-Error. In questo caso SESSION_ID deve essere settato a 0x0000.

PPPoE Active Discovery Terminate (PADT) packet

Questo pacchetto può essere inviato per indicare che una sessione PPPoE è stata terminata. Può essere inviato sia dall'host che dall Access Concentrator. Il campo DESTINATION_ADDR contiene l'indirizzo unicast Ethernet, il campo CODE contiene il valore 0xa7 e il SESSION_ID deve indicare quale sessione è stata terminata. Nessun TAG è richiesto. Quando un PADT viene ricevuto non è più consentito inviare traffico PPP utilizzando tale sessione. Neanche i normali pacchetti di terminazione PPP devono essere inviati dopo aver inviato o ricevuto un PADT.

Se l'host intende cominciare una nuova sessione PPP, deve ripartire dalla fase di Discovery.

PPP Session Stage

Quando una sessione PPPoE inizia, i dati PPP sono inviati come in ogni altro incapsulamento PPP. Tutti i pacchetti Ethernet sono unicast. Il campo ETHER_TYPE è settato a 0x8864 e il PPPoE CODE a 0x00. Il SESSION_ID deve mantenere il valore assegnato nella fase di Discovery per tutta la durata della sessione PPPoE.

Altre considerazioni

Quando un host non riceve un PADO packet entro una specifico intervallo di tempo, dovrebbe reinviare il suo PADI packet e raddoppiare il periodo di attesa. Questo meccanismo viene ripetuto per un numero di volte desiderate.

Se l'host sta aspettando di ricevere un PADS packet, viene utilizzato un meccanismo simile con l'host che reinvia il PADR. Dopo un numero specificato di tentativi, l'host dovrebbe reinviare un PADI packet.

VI. PPP Authentication

Al fine di stabilire comunicazioni su link punto-punto, ogni capo del link PPP deve prima inviare pacchetti LCP per configurare il link dati. Dopo che il link è stato stabilito, PPP prevede una fase di autenticazione opzionale prima di procedere alla fase Network-Layer Protocol. Di default l'autenticazione è quindi non obbligatoria.

NB Di seguito ci riferiremo ad Autenticatore come l'entità del link punto-punto che richiede l'autenticazione e stabilisce il protocollo da utilizzare da parte dell'altra estremità che chiameremo peer che deve farsi autenticare.

Password Authentication Protocol (PAP)

PAP fornisce un metodo semplice per il peer per stabilire la sua identità tramite un 2-way handshake solo dopo la fase di instaurazione del link.

Una coppia (Id , Password) è ripetutamente inviata dal peer all'autenticatore finché l'autenticazione non è confermata (tramite Ack) o la connessione è terminata.

PAP non è un metodo di autenticazione robusto; le Password sono inviate sul canale in chiaro e non vi è protezione da attacchi a ripetizione o tentativi ripetuti. È il peer ad avere il controllo sulla frequenza e la sincronizzazione dei tentativi.

Esattamente un solo pacchetto PAP è incapsulato nel campo informazione di un frame PPP Data Link Layer.

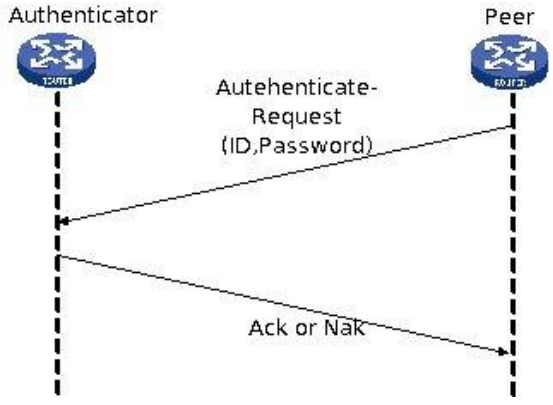

FIGURA 9: PAP AUTHENTICATION

Authenticate Request

Questo pacchetto è usato per iniziare il protocollo PAP.

Il peer deve inoltrare un pacchetto PAP con campo Code settato a 1 (Authenticate-Request) durante la fase di autenticazione, ripetendo l'invio finché non riceve un pacchetto di risposta valida o un contatore di reinoltro (opzionale) spira.

L'autenticatore alla ricezione di un pacchetto Authenticate-Request deve ritornare un pacchetto appropriato di risposta.

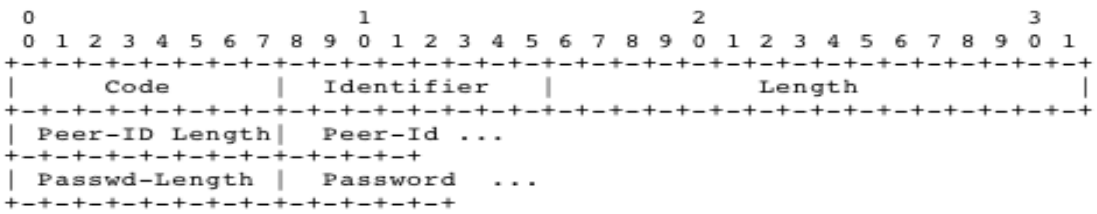

FIGURA 10: AUTHENTICATE-REQUEST PACKET FORMAT

CODE : 1 per Authenticate-Request.

IDENTIFIER: Ha lo stesso significato che in precedenza e deve cambiare ogni volta che un pacchetto Authenticate-Request è emesso.

PEER-ID-LENGTH: È un byte ed indica la lunghezza del campo Peer-ID.

PEER-ID: È zero o più byte e dindica il nome del peer che deve essere autenticato.

PASSWD-LENGTH: È un byte ed indica la lunghezza del campo Password.

PASSWORD: È zero o più byte ed indica la password che deve essere usata per l'autenticazione.

Authenticate-Ack and Authenticate-Nak

Se la coppia (Peer-Id , Password) ricevuta è ammissibile e accettabile l'autenticatore deve trasmettere un pacchetto PAP con campo Code settato a 2 (Authenticate-Ack) in caso contrario deve inviare un pacchetto PAP con campo Code settato a 3 (Authenticate-Nak) ed intraprendere azioni per terminare il collegamento.

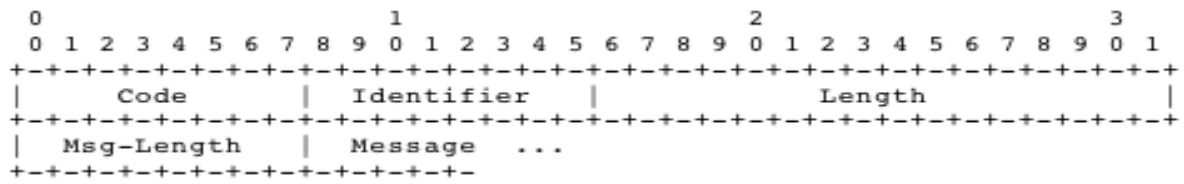

FIGURA 11: AUTHENTICATE-ACK AUTHENTICATE-NAK PACKET FORMAT

CODE: 2 per Authenticate-Ack
3 per Authenticate-Nak

IDENTIFIER: Il valore deve essere copiato dal campo Identifier dell'Authenticate- Request che causa questa risposta.

MSG-LENGTH: È un byte ed indica la lunghezza del campo Message.

MESSAGE: È zero o più byte ed il suo contenuto dipende dall'implementazione, tenta d'essere comprensibile all'uomo e non deve influire sulle operazioni del protocollo.

N.B. Siccome gli Authenticate-Ack possono essere persi, l'autenticatore deve permettere pacchetti Authenticate-Request ripetuti dopo il completamento della fase di autenticazione. Il protocollo deve inoltrare lo stesso valore di Code ritornato quando si è completata la fase di autenticazione.

Challenge-Handshake Authentication Protocol (CHAP)

CHAP è usato per verificare periodicamente l'identità del peer usando un 3-way handshake dopo l'instaurazione del collegamento iniziale.

1. L'autenticatore invia un messaggio "challenge" al peer .
2. Il peer risponde con un valore calcolato usando una funzione hash non invertibile.
3. L'autenticatore controlla la risposta con calcoli interni sul valore hash atteso e se il valore coincide l'autenticazione è confermata altrimenti la connessione deve essere terminata.

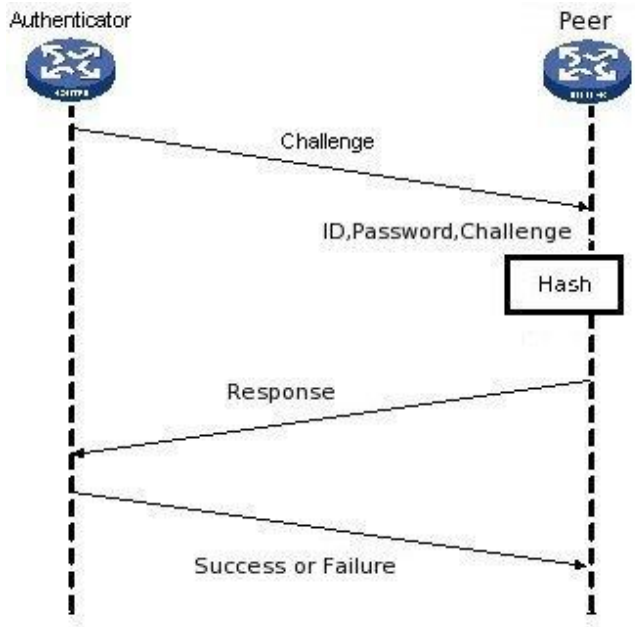

FIGURA 12: CHAP AUTHENTICATION

CHAP fornisce protezione contro attacchi a ripetizione tramite l'utilizzo di un identificativo incrementale e di un valore di challenge variabile. È l'autenticatore ad avere il controllo sulla frequenza e la sincronizzazione dei chellenges.

Questo metodo di autenticazione si basa su un "segreto" conosciuto solo dal peer e dall'autenticatore che non è inviato sul canale.

L'algoritmo CHAP richiade che il segreto sia lungo almeno 1 byte, anche se è preferibile che abbia almeno la lunghezza del valore restituito dalla funzione hash scelta al fine di fornire protezione contro attacchi a ricerca esaustiva.

Il valore di "challenge" deve invece rispettare le importanti proprietà di unicità e impredicibilità.

Tuttavia protocolli come CHAP sono incapaci di proteggere contro attacchi realtime di intrusi attivi.

Challenge and Response

Il pacchetto Challenge è usato per iniziare il protocollo CHAP, l'autenticatore invia un pacchetto CHAP con campo Code settato a 1 (Challenge) e ripete l'invio finché un pacchetto Response valido è ricevuto oppure un contatore di ritrasmissioni (opzionale) spira.

Un pacchetto Challenge può essere trasmesso in ogni momento durante la fase Network-Layer Protocol al fine di garantire che la connessione non sia stata alterata.

Quando il peer riceve un pacchetto Challenge deve rispondere con un pacchetto CHAP con campo Code settato a 2 (Response).
Alla ricezione di una Response l'autenticatore compare il valore di risposta ricevuto con quello calcolato e quindi atteso e se coincidono invia un pacchetto Success altrimenti un pacchetto Failure.

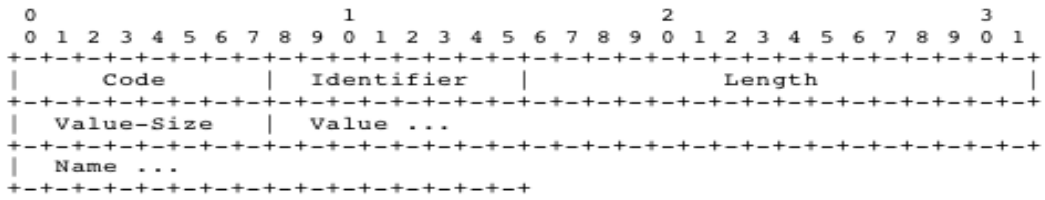

FIGURA 13: CHALLENGE AND RESPONSE PACKET FORMAT

CODE: 1 per Challenge
2 per Response.

IDENTIFIER: Il suo valore deve essere cambiato ogni volta che un Callenge è inviato.
Il valore Identifier per il pacchetto Response deve essere copiato dal campo Identifier del pacchetto Challenge che ha causato la risposta.

VALUE-SIZE: È un byte ed indica la lunghezza del campo Value.

VALUE:È uno o più byte;
Il valore per Challenge deve essere cambiato ogni volta che un Challenge è spedito e la sua lunghezza dipende dai metodi usati per generarlo ed è indipendente dalla funzione hash scelta.
Il valore per Response è l'hash non invertibile calcolato su uno stream di byte costituito dall'Identifier seguito dal "segreto" seguito dal Value di Challenge. La lunghezza dipende dall'algoritmo di hash usato.

NAME: È uno o più byte rappresentanti l'identificazione del sistema che trasmette i pacchetti. Non ci sono limitazioni sul contenuto di questo campo la cui dimensione è determinata dal campo Length.

Success and Failure
Se il valore ricevuto in una Response è uguale al valore atteso, allora l'autenticatore deve trasmettere un pacchetto CHAP con campo Code settato a 3 (Success) altrimenti deve trasmettere un pacchetto CHAP con campo Code settato a 4 (Failure) e deve intraprendere azioni per terminare il collegamento.

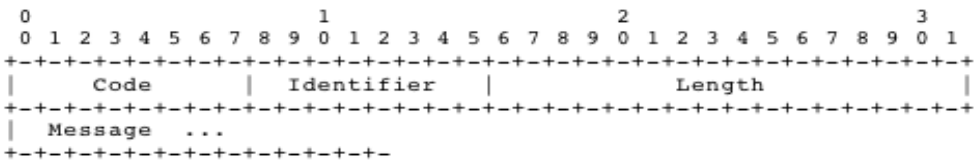

FIGURA 14: SUCCESS AND FAILURE PACKET FORMAT

CODE: 3 per Success
4 per Failure

IDENTIFIER:
Deve essere ricopiato dal campo Identifier del pacchetto Response che ha causato questa replica.

MESSAGE:

È zero o più byte e il contenuto dipende dall'implementazione, tenta d'essere comprensibile all'uomo e non deve influire sulle operazioni del protocollo.

Extensible Authentication Protocol(EAP)

EAP rappresenta un framework per supportare differenti medoti di autenticazione.

1. L'autenticatore invia un messaggio "request" al peer .
2. Il peer risponde con un pacchetto "response" con valori in relazione alla request ricevuta.
3. L'autenticatore termina la fase di autenticazione con un pacchetto Success o Failure.

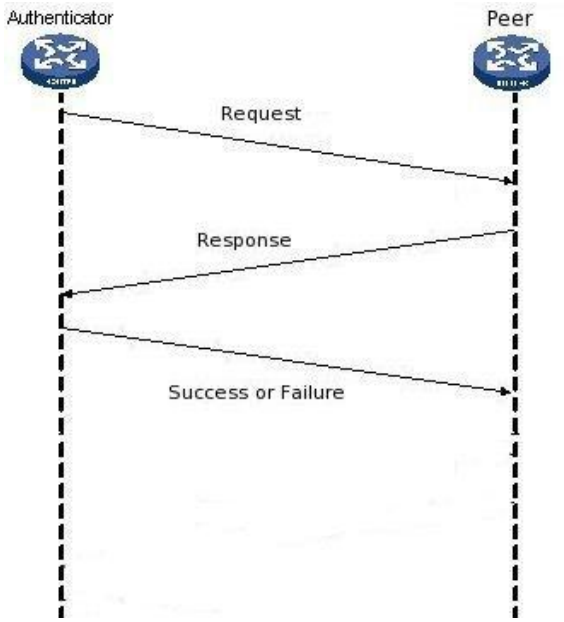

FIGURA 15 EAP AUTHENTICATION

EAP è un protocollo generale per l'autenticazione PPP che supporta meccanismi di autenticazione multipli.
EAP non non seleziona uno specifico meccanismo di autenticazione nella fase LCP, ma pospone questa scelta fino alla fase di autenticazione. Questo permette all'autenticatore di richiedere maggiori informazioni prima di determinare uno specifico meccanismo.
Questo permette eventualmente l'utilizzo di un server esterno che implementa realmente il meccanismo di autenticazione, mentre l'autenticatore PPP effettua solo scambio di pacchetti.

Request and Response

L'autenticatore invia un pacchetto EAP con campo Code settato a 1 (Request) e ripete l'invio finché un pacchetto Response valido è ricevuto oppure un contatore di ritrasmissioni (opzionale) spira.
Quando il peer riceve un pacchetto Request deve rispondere con un pacchetto EAP con campo Code settato a 2 (Response).
Il contenuto dei campi dipende dal tipo di Request ricevuta e una Response può essere solo inviata in risposta ad una richiesta e mai come ritrasmissione legata ad un timer

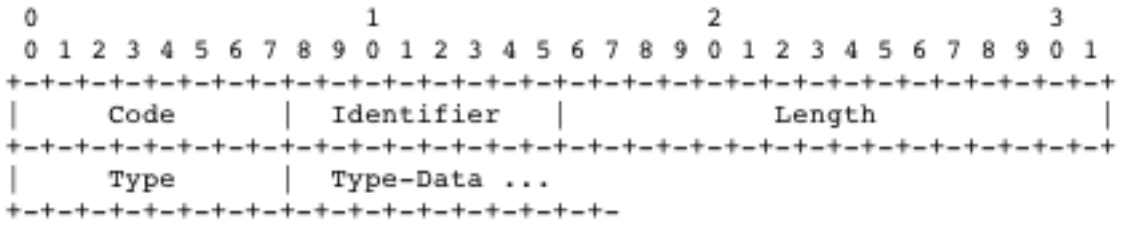

FIGURA 16 REQUEST AND RESPONSE PACKET FORMAT

CODE: 1 per Request
2 per Response.

IDENTIFIER: Il suo valore deve essere cambiato ogni volta che una nuova Request è inviato.
Il valore Identifier per il pacchetto Response deve essere copiato dal campo Identifier del pacchetto Request che ha causato la risposta.

LENGTH: È un byte ed indica la lunghezza del pacchetto intero

TYPE: È un byte;
Il valore per Request dipende dal tipo di richiesta che si effettua:

1 Identity	Usato per richiedere l'identità di un peer
2 Notification	Usato per inviare un messaggio al peer che deve essere mostrato all'utente
4 MD5-Challenge	E' analogo al protocollo PPP CHAP con MD5 come hash predefinito
5 One-Time Password	Password usata per una singola sessione di autenticazione
6 Generic Token Card	Richiede input di valori della token card da parte dell'utente

Il valore per Response è lo stesso valore della Request oppure un valore Nak (valore=3) per indicare che tale valore non è accettabile dal peer

TYPE-DATA: Il suo valore varia in base al valore del campo Type di Request o Response.

Success and Failure
L'autenticatore deve trasmettere un pacchetto EAP con campo Code settato a 3 (Success) per segnalare che l'autenticazione ha avuto successo, altrimenti deve trasmettere un pacchetto EAP con campo Code settato a 4 (Failure).

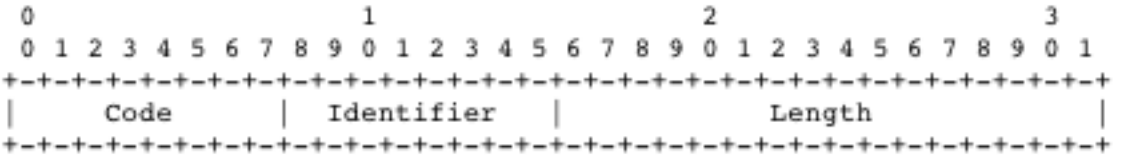

FIGURA 17 SUCCESS AND FAILURE PACKET FORMAT

CODE: 3 per Success
4 per Failure

IDENTIFIER: Deve essere ricopiato dal campo Identifier del pacchetto Response che ha causato questa replica.

LENGTH: 4 (il pacchetto ha lunghezza fissa)

VII. Reference

I. Wikipedia: Serial Line Internet Protocol.

II. RFC 1661 - The Point-to-Point Protocol (PPP), Sito web:"www.openskill.info", "Access Network: xDSL family" (Prof. Cuomo).

III. Wikipedia: Asynchronous Transfer Mode, RFC 2761 - Terminology for ATM

IV. RFC 2364 - PPP Over AAL5

V. RFC 2516 - A Method for Transmitting PPP Over Ethernet (PPPoE)

VI. RFC 1334 - PPP Authentication Protocols, RFC 1994 - PPP Challenge Handshake Authentication Protocol (CHAP), RFC 2284 - PPP Extensible Authentication Protocol