

Livello Data Link



I compiti del livello

Il livello fisico lascia irrisolti problemi legati agli errori e disturbi occasionali e al ritardo di propagazione del segnale, problemi che devono essere risolti per il corretto funzionamento della comunicazione in rete.

Il compito principale del livello è **garantire una linea di trasmissione priva di errori non segnalati.**

Cosa si deve fare?

Per svolgere il proprio compito le entità di livello data link devono:

- raggruppare i bit in frame e determinare come i bit del livello fisico sono raggruppati in frame (framing);
- gestire gli errori di trasmissione;
- regolare il flusso della trasmissione dalla sorgente al destinatario.

Come? In trasmissione...

- spezza il flusso di bit che arriva dal livello di rete in una serie di frame;
- calcola un'apposita funzione (checksum) per ciascun frame;
- inserisce il checksum nel frame;
- consegna il frame al livello fisico, il quale lo spedisce come sequenza di bit

... in ricezione

- riceve una sequenza di bit dal livello fisico;
- ricostruisce da essa un frame dopo l'altro;
- per ciascun frame ricalcola il checksum;
- se esso è uguale a quello contenuto nel frame, il frame viene accettato, altrimenti il frame viene considerato errato e quindi scartato;

Servizi offerti al livello di rete

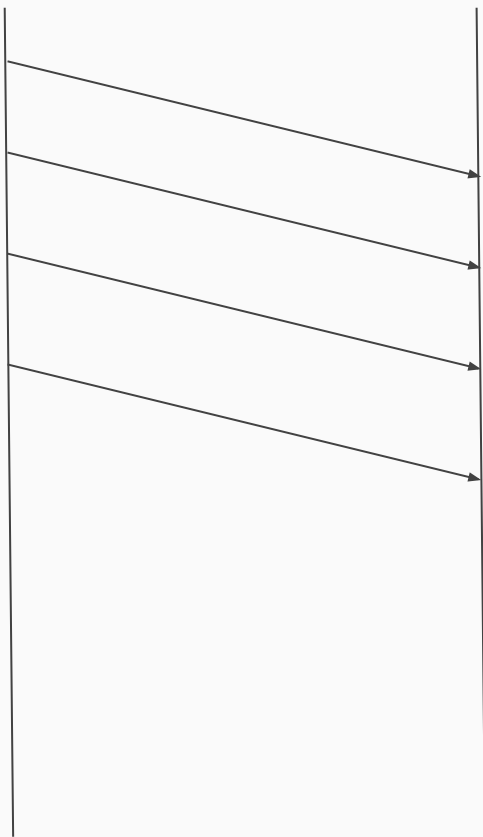
Senza connessione e senza riscontro:

- ★ non viene stabilita alcuna connessione tra mittente e destinatario;
- ★ il mittente non attende alcun riscontro circa l'arrivo e la correttezza dei frame trasmessi;
- ★ i frame persi non vengono recuperati a questo livello, sono i livelli superiori che si occupano della ricostruzione del messaggio.

Questo tipo di servizio è appropriato per canali con tasso d'errore molto basso oppure in sistemi real-time, in cui non c'è il tempo necessario all'attesa dei riscontri.

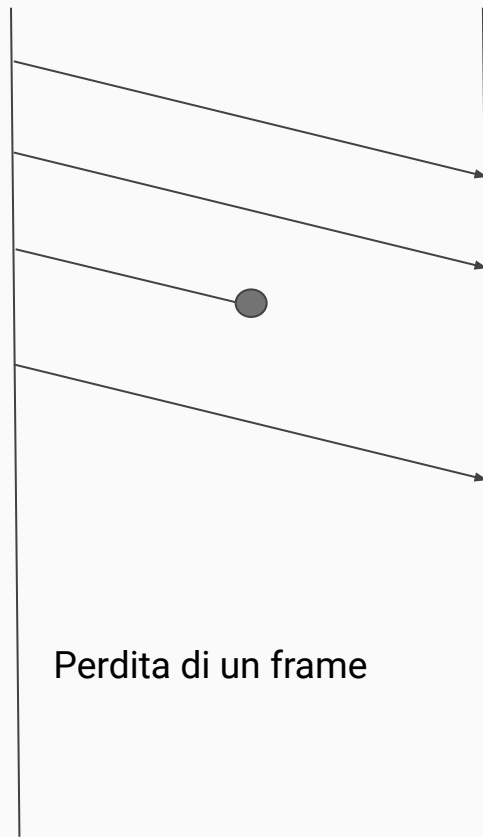
Mittente

Destinatario



Mittente

Destinatario



Perdita di un frame

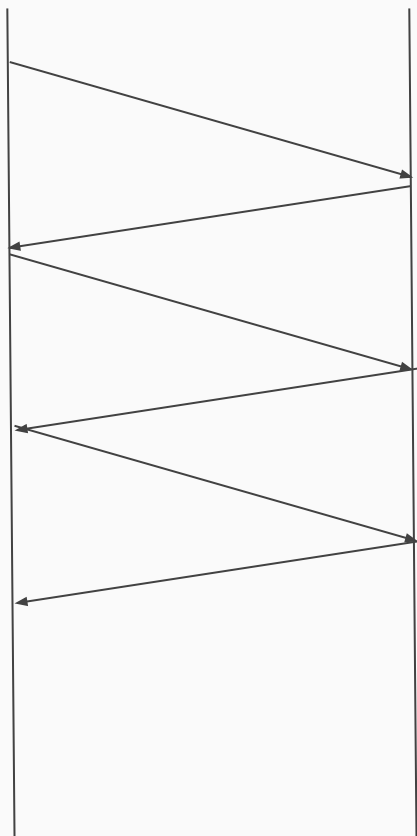
Servizi offerti al livello di rete

Senza connessione e con riscontro:

- ❖ non viene stabilita la connessione tra i due partner, ma i frame ricevuti dal destinatario vengono confermati (o in senso positivo o in senso negativo);
- ❖ se entro un certo intervallo di tempo la conferma non arriva, il mittente può rispeditore il frame;
- ❖ la perdita di un riscontro positivo può causare la ricezione di più copie dello stesso frame.

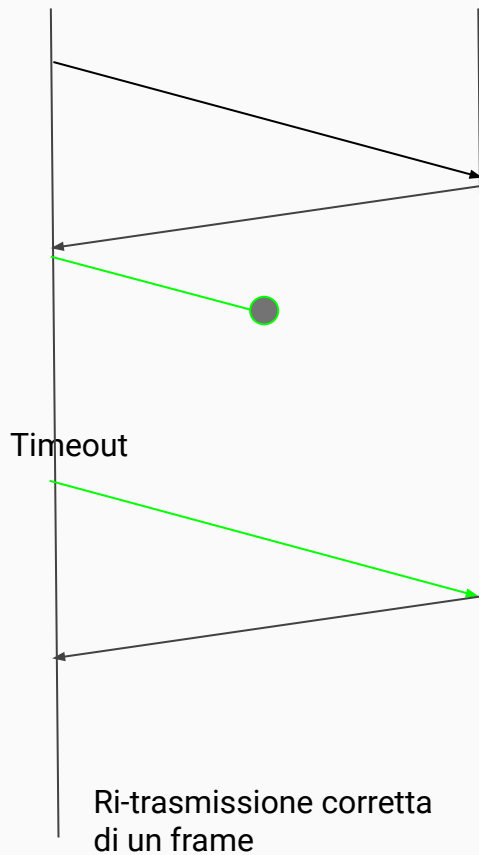
Mittente

Destinatario



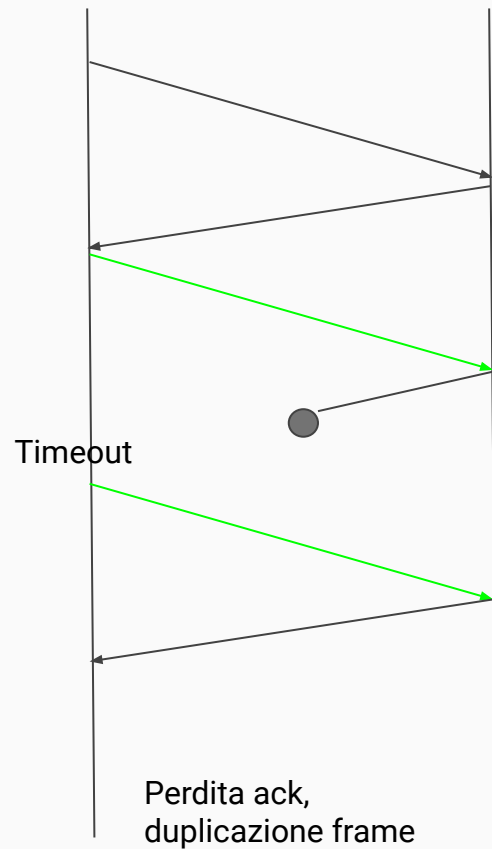
Mittente

Destinatario



Mittente

Destinatario



Servizi offerti al livello di rete

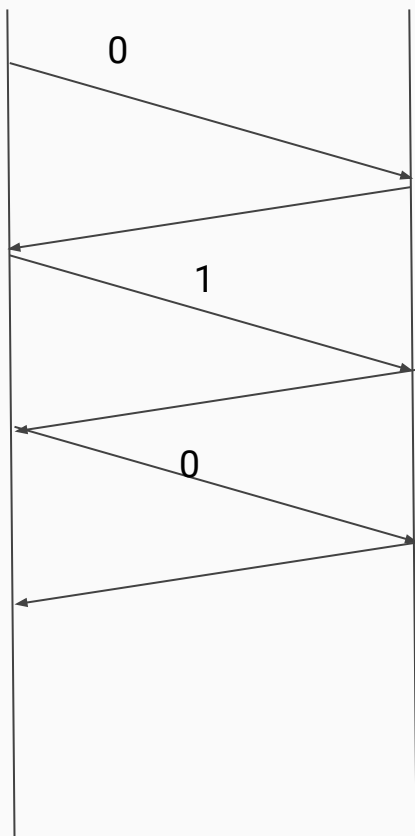
Con connessione e con riscontro:

- ❖ si apre la connessione tra trasmettitore e ricevitore;
- ❖ i frame vengono numerati, il ricevitore emette un riscontro per ciascuno;
- ❖ la connessione viene chiusa.

Con questo tipo di servizio ogni frame viene ricevuto una e una sola volta e i frame arrivano al destinatario nello stesso ordine di trasmissione, fornendo così al livello di rete un flusso di bit affidabile.

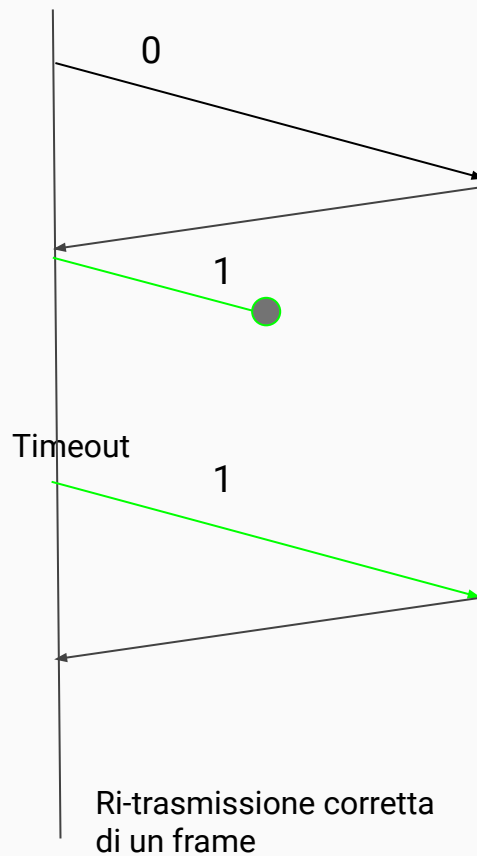
Mittente

Destinatario



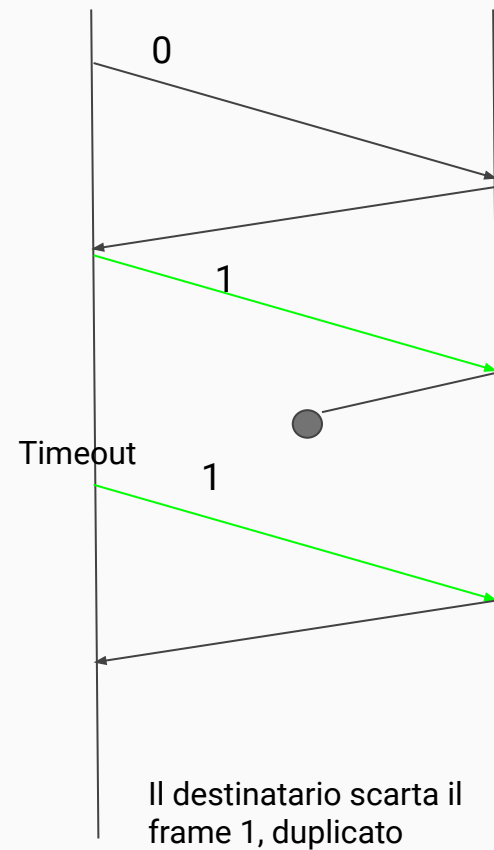
Mittente

Destinatario



Mittente

Destinatario



Delimitazione dei frame con sequenza di caratteri

- ❖ ogni frame inizia e finisce con una particolare la sequenza di caratteri ASCII, di solito DLE STX e DLE ETX;
- ❖ un carattere DLE contenuto nei dati da trasmettere viene duplicato;
- ❖ in ricezione la coppia DLE DLE viene interpretata come un singolo DLE nel campo dati.

Delimitazione dei frame con sequenza di bit

- ❖ ogni frame inizia e finisce con la sequenza di bit 1000 0001 (flag byte);
- ❖ in trasmissione, dopo 5 bit 0 consecutivi del campo payload si inserisce un bit 1;
- ❖ in ricezione, se dopo 5 bit 0 consecutivi c'è un bit 1, questo viene rimosso.

Esempio

La sequenza di bit da trasmettere è:

110000000001100000111

Il mittente aggiunge un bit 1 ogni 5 bit 0 consecutivi:

1100000**1**00001100000**1**111

Esempio

Il destinatario riceve la sequenza:

11000001000011000001111

rimuove il bit 1 che segue una sequenza di 5 bit 0 consecutivi:

11 00000 1 000011 00000 1111

ottenendo di nuovo:

110000000001100000111

Errori di trasmissione

Gli errori sono più frequenti nelle trasmissioni telefoniche e in quelle wireless.

In genere sono dovuti a:

- ❖ rumore di fondo,
- ❖ disturbi (ad es. fulmini) improvvisi,
- ❖ interferenze (ad es. motori elettrici).

Si gestiscono inserendo nel frame dei bit ridondanti ricavati dal payload (checksum).

Codici a correzione d'errore

Si include nel fame abbastanza informazione aggiuntiva in modo da poter ricostruire il messaggio originario.

Non sono utilizzati perché gli errori non si distribuiscono uniformemente, ma si presentano “a raffica”.

Un esempio è il codice di Hamming.

Codici a rilevazione d'errore

Si include nel frame una quantità limitata di informazione aggiuntiva in modo da accorgersi che c'è stato un errore.

L'errore può essere solo rilevato ma non corretto, utilizzando servizi affidabili il problema si risolve con la ritrasmissione del frame errato.

Un esempio è il CRC.

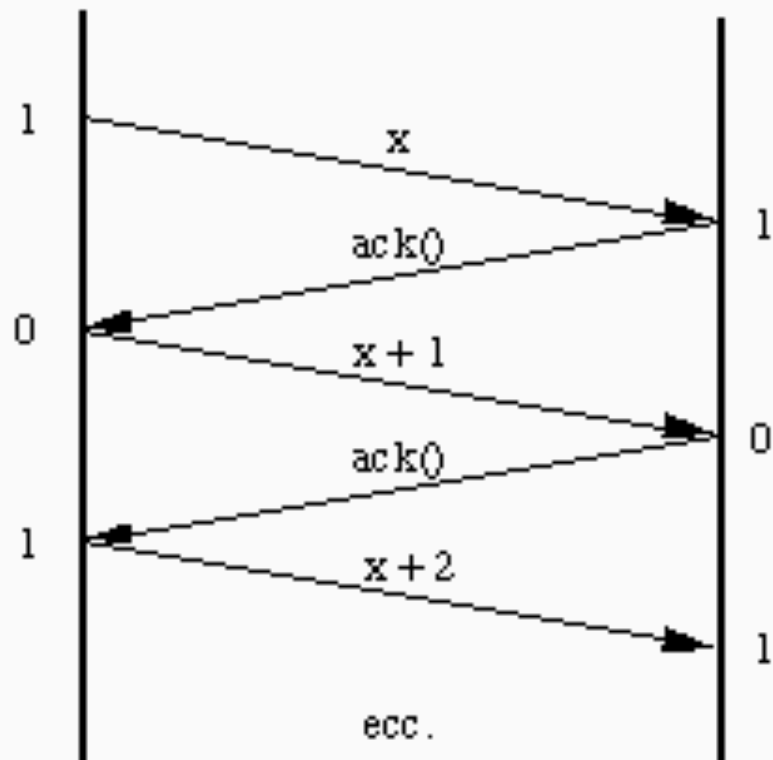
Struttura generale di un frame

FLAG	KIND Intestazione del frame	SEQ	ACK	PAYLOAD	CRC	FLAG
------	--------------------------------	-----	-----	---------	-----	------

- kind: specifica se il frame conterrà dati o è di controllo
- seq: è il numero progressivo del frame
- ack: contiene informazioni relative al riscontro dei frame ricevuti
- payload: contiene le informazioni del frame, può ad esempio corrispondere a un pacchetto completo del livello di rete.

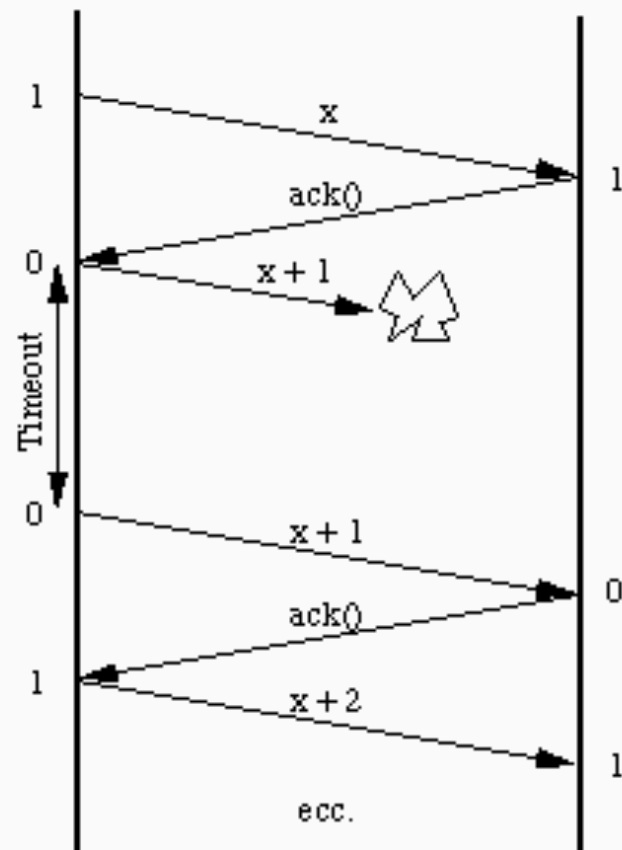
Mittente

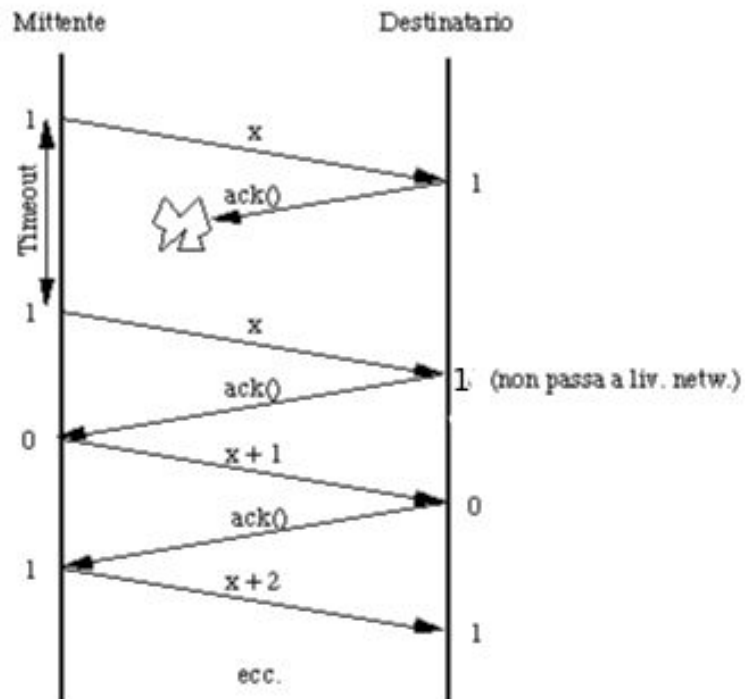
Destinatario



Mittente

Destinatario





Protocollo simplex stop & wait per canale rumoroso (mittente)

- 0) $n_seq = 1$;
- 1) $n_seq = 1 - n_seq$;
- 2) attende un pacchetto dal livello network;
- 3) costruisce frame dati e copia n_seq in $[seq]$;
- 4) passa il frame dati al livello fisico;
- 5) resetta il timer;
- 6) attende un evento:
 - * timer scaduto: torna a 4)
 - * arriva frame di ack (controllo) non valido o nack: torna a 4)
 - * arriva frame di ack (controllo) valido: torna ad 1)

Protocollo simplex stop & wait per canale rumoroso (destinatario)

0) $n_exp = 0$;

1) attende evento;

* arriva frame dati valido da livello fisico:

2) se ($[seq] == n_exp$)

2.1) estrae pacchetto

2.2) lo consegna al livello network

2.3) $n_exp = 1 - n_exp$

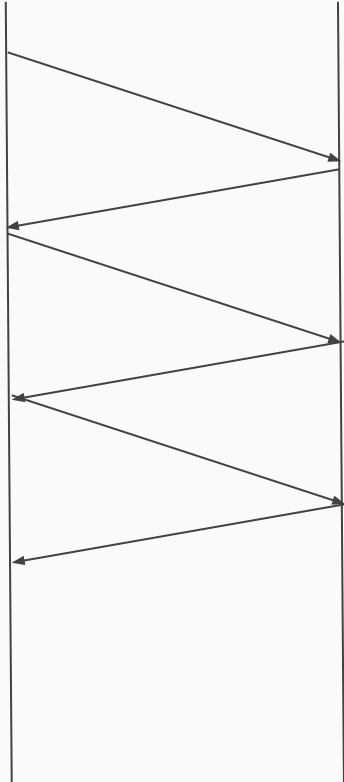
3) invia frame di ack (vuoto)

4) torna ad 1)

* arriva frame non valido: invia nack e torna ad 1)

Efficienza protocolli stop&wait

Mittente Destinatario



Prima di trasmettere un nuovo frame si deve aspettare l'ack del precedente.

Questo comporta un utilizzo inefficiente del canale di comunicazione, che dipende da molti fattori:

- ritardo di propagazione;
- velocità di trasmissione;
- lunghezza della trama

Efficienza protocolli stop&wait: esempio

Lunghezza trama 1000 bit
Velocità 10 Mps
Ritardo di propagazione 20 μs

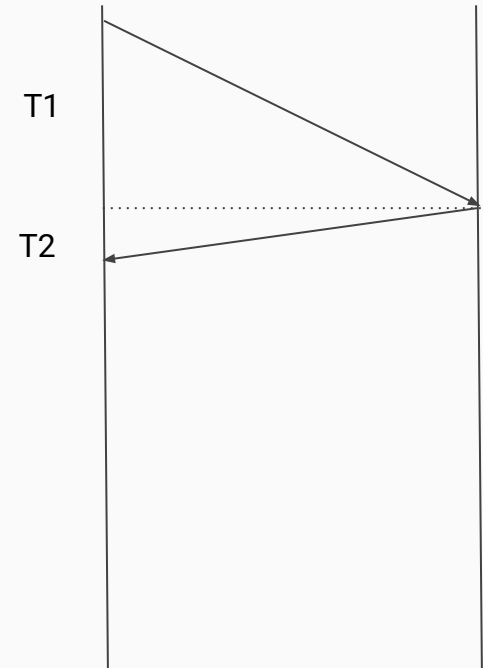
Durata trama = $1000 \text{ bit} / 10 \cdot 10^6 \text{ bit s}^{-1} = 100 \mu\text{s}$

$T1 = 120 \mu\text{s}$ $T2 = 20 \mu\text{s}$

Per trasmettere 1000 bit servono quindi 140 μs , la velocità effettiva

$V_e \approx 7,14 \text{ Mps}$

Efficienza $7,14/10 = 71\%$



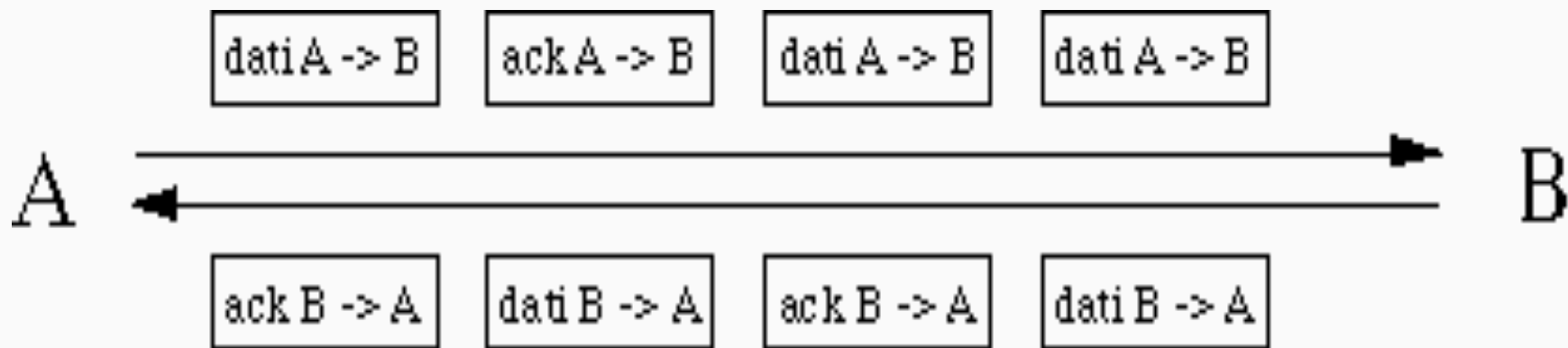
Piggybacking

Negli esempi che abbiamo visto i dati viaggiano solo in una direzione e i riscontri nell'altra, ma normalmente la comunicazione è bidirezionale.

Utilizzando il protocollo appena visto in una comunicazione sia fra A e B, si avrà che:

- ❖ nella direzione da A a B viaggiano i frame dati inviati da A a B e i frame di ack inviati da A a B (in risposta ai frame dati inviati da B ad A);
- ❖ nella direzione da B a A viaggiano i frame dati inviati da B a A e i frame di ack inviati da B a A (in risposta ai frame dati inviati da A ad B);

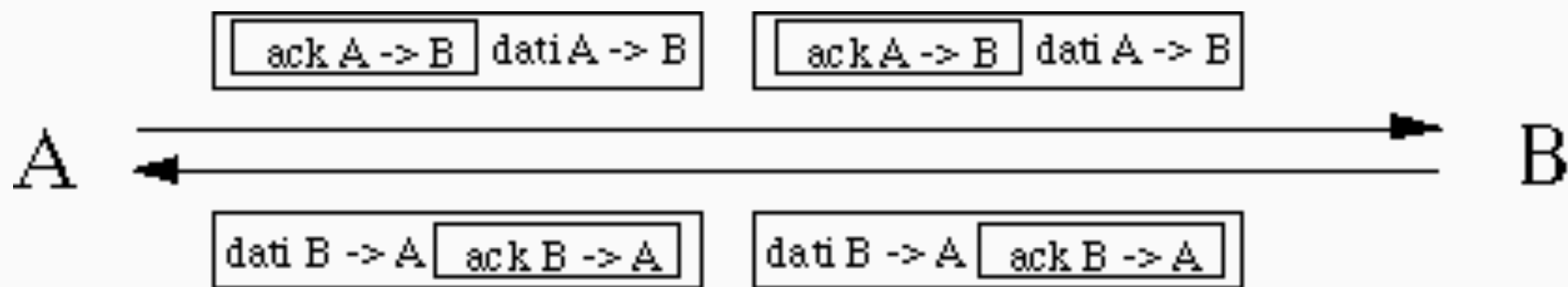
Il campo kind che serve a distinguere fra i due tipi di frame, dati e di ack.



Piggybacking

La soluzione appena proposta però non è efficiente, una soluzione migliore consiste nell'inserire le informazioni di ack nei frame dati.

Questa tecnica si chiama piggybacking (letteralmente, portare a spalle), e migliora l'efficienza perché le informazioni di ack non richiedono la costruzione di un apposito frame (e quindi il tempo necessario alla creazione ed al riempimento della struttura, al calcolo del checksum, ecc.) né la sua trasmissione (e quindi l'uso di banda).



Efficienza protocolli stop&wait con piggybacking: esempio

Lunghezza trama 1000 bit
Velocità 10 Mps
Ritardo di propagazione 20 μs

Durata trama = $1000 \text{ bit} / 10 \cdot 10^6 \text{ bit s}^{-1} = 100 \mu\text{s}$

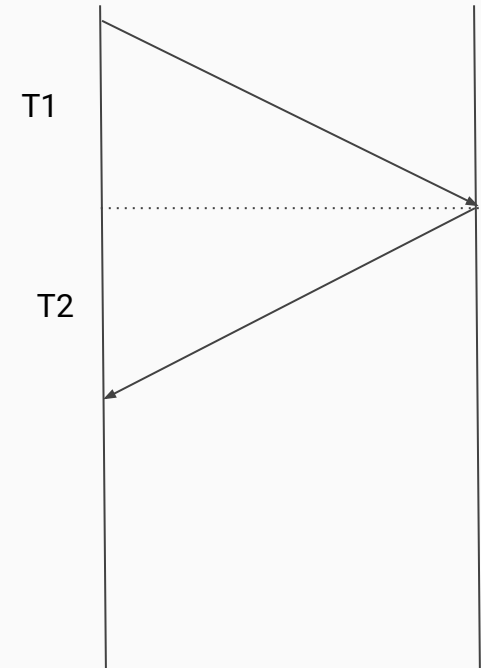
In questo caso anche T2 trasporta 1000 bit di dati, quindi

T1 = 120 μs T2 = 120 μs

Per trasmettere 2000 bit servono quindi 240 μs , la velocità effettiva

$V_e \approx 8,33 \text{ Mps}$

Efficienza $8,33/10 = 83\%$



Sliding windows

Usando il piggybacking, per quanto tempo si può aspettare un frame su cui trasportare un ack che è pronto e deve essere inviato?

Non troppo, perché se l'ack non arriva in tempo il mittente ritrasmetterà il frame anche se ciò non è necessario. Dunque si stabilisce un limite al tempo di attesa di un frame sul quale trasportare l'ack, trascorso tale tempo si crea un frame apposito nel quale si trasmette solo l'ack.

Sliding windows

Diminuendo il numero di ack da trasmettere, si velocizzano le operazioni di trasmissione, per cui alcuni protocolli hanno introdotto le finestre scorrevoli (sliding-windows).

I protocolli a finestra scorrevole stabiliscono un certo numero (larghezza della finestra) di frame dati che si possono inviare consecutivamente senza attendere l'ack di riscontro.

Sliding windows

I protocolli a finestra scorrevole sono full-duplex (per i dati), sfruttano il piggybacking e sono più robusti di quelli stop&wait.

Ogni frame inviato ha un numero di sequenza, da 0 a $2^n - 1$ (il campo seq è costituito da n bit).

Ogni stazione mantiene due finestre, una di trasmissione e una di ricezione.

Sliding windows: finestra di trasmissione

Immaginando i frame che vengono trasmessi durante una connessione, la finestra di trasmissione è “posizionata” su 2^n-1 frame, che possono essere frame da spedire o spediti ma non ancora confermati.

I frame dentro la finestra devono essere mantenuti in memoria per la possibile ritrasmissione; se il buffer è pieno, il livello data link deve costringere il livello rete a sospendere la consegna di pacchetti.

Quando arriva la conferma di ricezione di un frame, la finestra si “sposta” avanti di uno, e sarà possibile ricevere un nuovo frame dal livello di rete.

Sliding windows: finestra di ricezione

I frame che arrivano vengono memorizzati in dei buffer in attesa di essere verificati ed inviati al livello superiore (rete).

Normalmente la finestra di ricezione è grande 1 (un solo frame può essere memorizzato temporaneamente), oppure ha la stessa dimensione della finestra di trasmissione (possono essere tenuti in memoria fino a $2^n - 1$ frame), a seconda del protocollo utilizzato per la ritrasmissione.

Sliding windows: finestra di ricezione

I frame che arrivano vengono memorizzati in dei buffer in attesa di essere verificati ed inviati al livello superiore (rete).

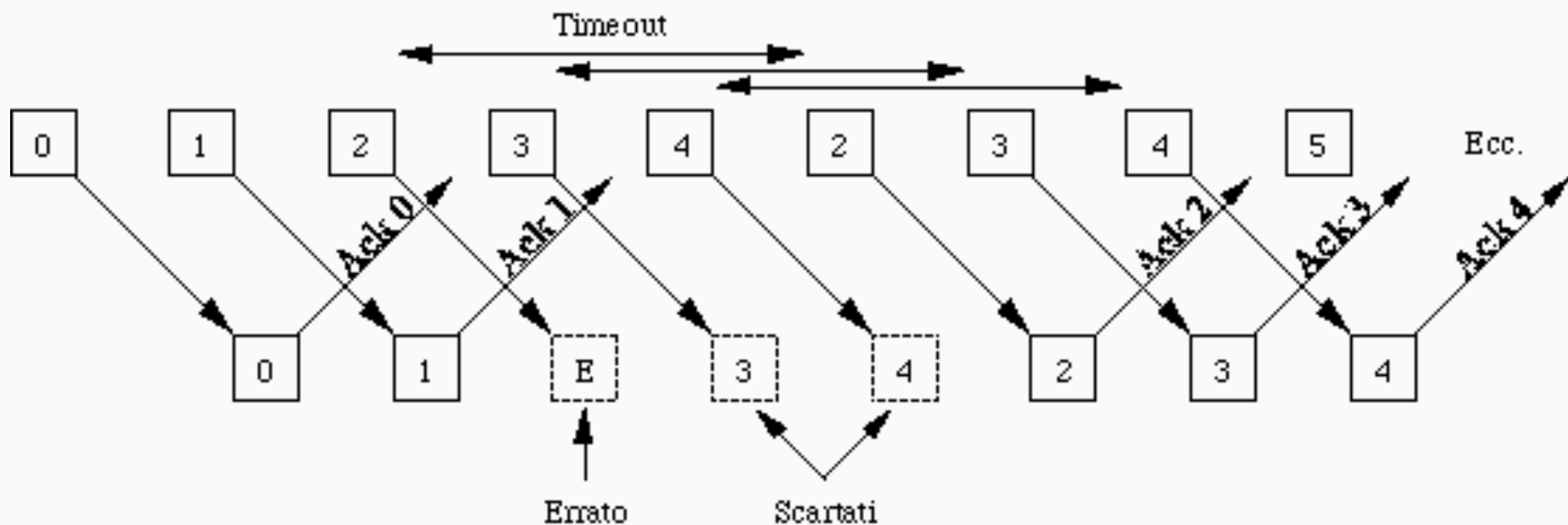
La finestra di ricezione può avere una dimensione diversa rispetto alla finestra di trasmissione.

Se arriva un frame il cui indice è fuori dalla finestra il frame viene scartato (e non si invia il relativo ack), se arriva un frame il cui indice è entro la finestra il frame viene accettato, viene spedito il relativo ack e la finestra viene spostata in avanti.

Sliding windows: go back n

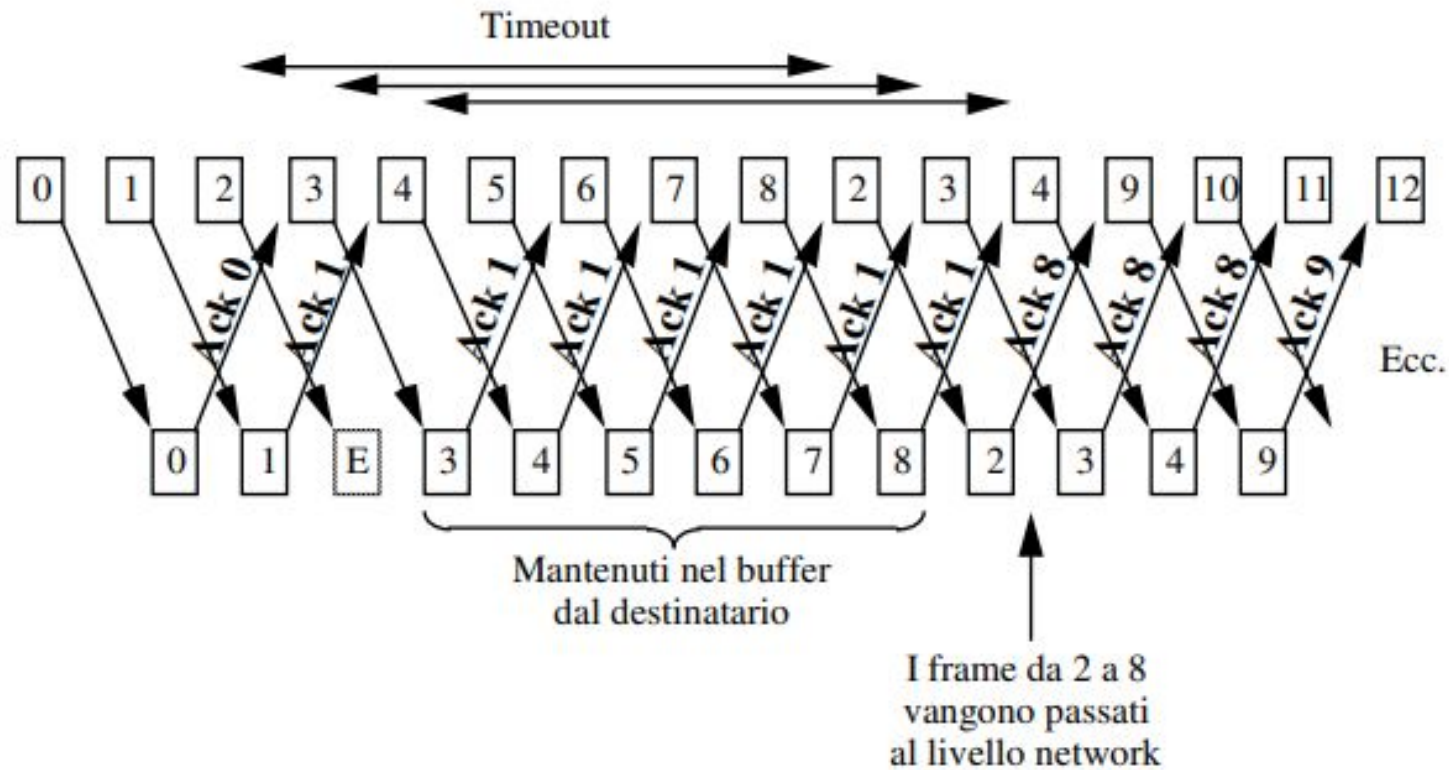
Se finestra di ricezione è di un solo buffer, il destinatario accetta i frame solo nell'ordine giusto.

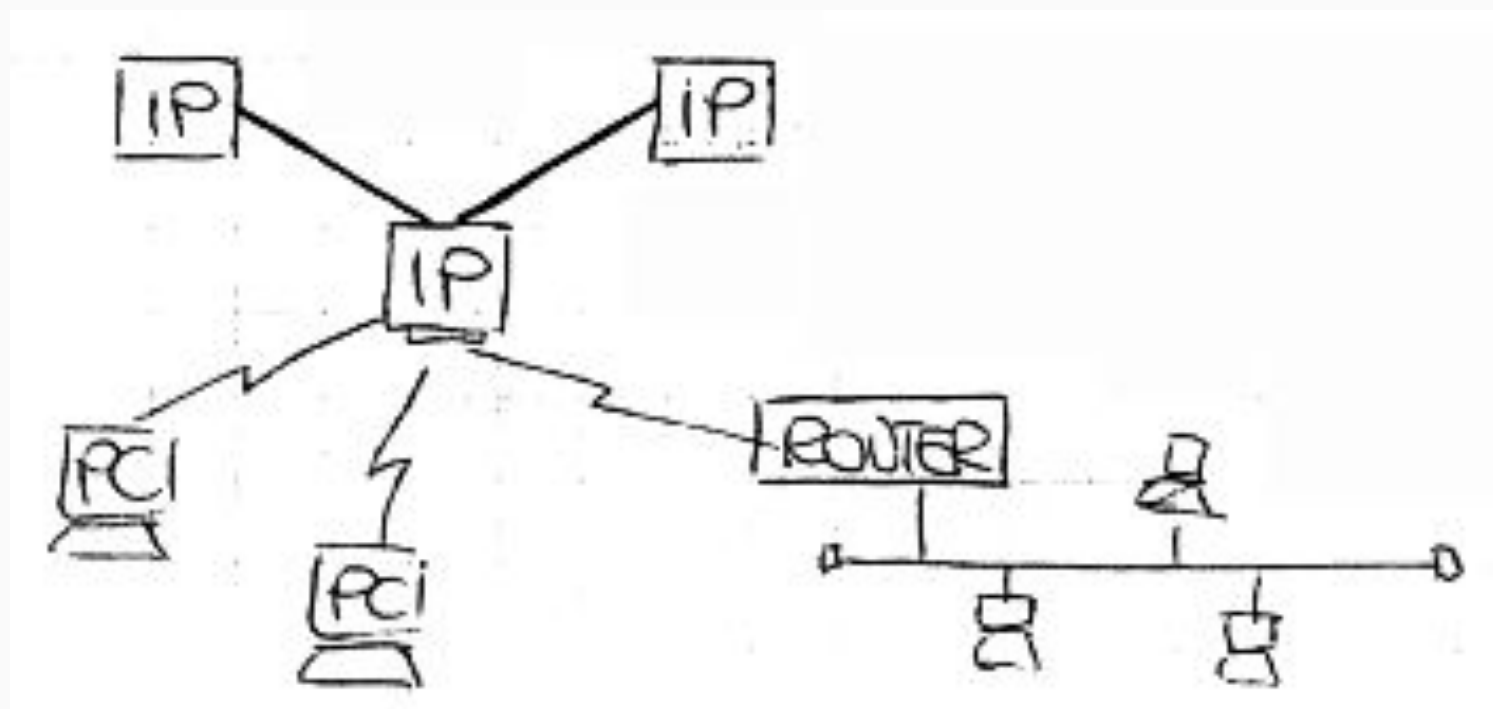
Se arriva un frame contenente errori (o fuori sequenza), si richiede la ritrasmissione di quel frame (o di quello mancante) e di tutti i frame successivi che nel frattempo potrebbero arrivare.



Sliding windows: selective repeat

Se la dimensione della finestra di ricezione è 2^n-1 (come quella di trasmissione), nel caso di mancato arrivo di un frame o di arrivo di un frame errato, se ne richiede selettivamente la trasmissione, ma si accettano e memorizzano frame successivi che potrebbero arrivare (al massimo, 2^n-2).





Protocolli SLIP (Serial Line Internet Protocol)

È il primo protocollo data link utilizzato in Internet.

È un protocollo senza connessione e senza riscontro, i pacchetti IP vengono trasmessi sulla linea con uno speciale byte indicatore (0xC0) alla fine per la suddivisione.

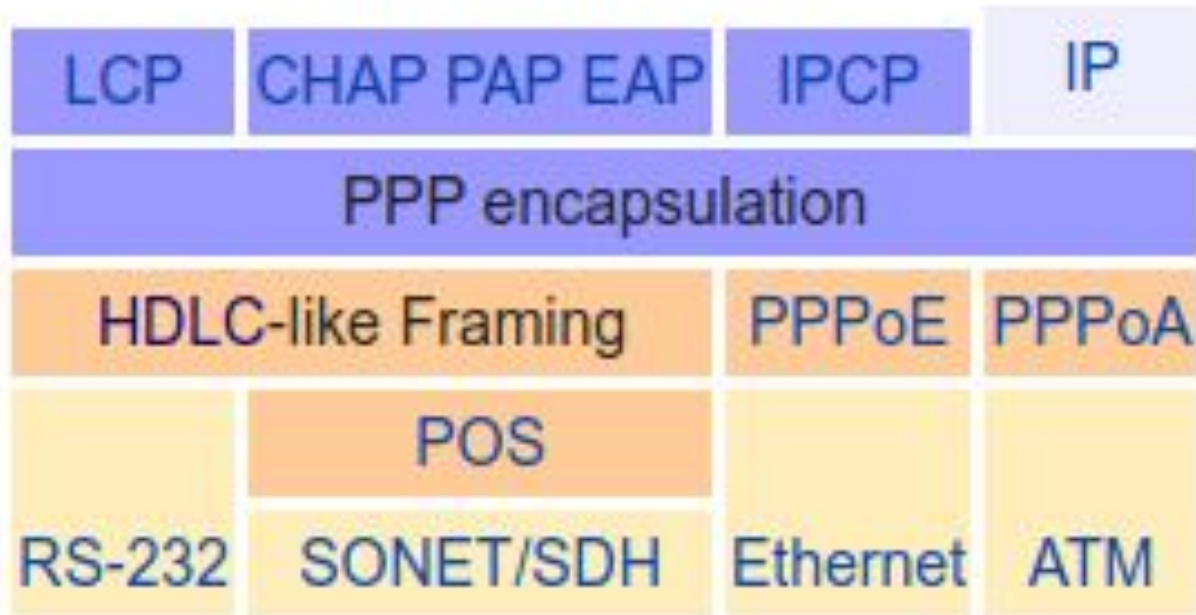
Se tale byte si trova all'interno del pacchetto si utilizza una tecnica a riempimento di carattere, sostituendolo con la coppia 0xBD, 0xDC.

Richiede una porta seriale ad 8 bit ed ormai è utilizzato solo per i microcontrollori, dato il suo overhead basso.

PPP (Point to Point Protocol)

- Descritto dalla RFC 1661 della IETF;
- orientato alla connessione e affidabile, prevede la gestione dell'errore e l'autenticazione delle parti;
- utilizza altri protocolli, tra cui NCP (Network Control Protocol), LCP (Link Control Protocol), CHAP (Challenge-Handshake Authentication Protocol)

PPP architecture



PPP (Point to Point Protocol)

LCP è il protocollo di controllo del collegamento per agganciare le linee, testarle, negoziare opzioni e ricederle educatamente quando non sono più necessarie.

NCP è il protocollo di controllo di rete, consente di negoziare le opzioni del livello di rete; ne esiste uno per ogni livello di rete supportato (IPCP per l'IP). Si utilizza per negoziare alcune caratteristiche di livello tre, tra cui ottenere dal provider la configurazione IP del router (o della stazione).

PPP (Point to Point Protocol)

- Il router si collega al router del provider via modem;
- il modem del router provider risponde e stabilisce una connessione fisica;
- i router si scambiano una serie di frame LCP per selezionare i parametri PPP da utilizzare;
- il router si autentica con CHAP;
- vengono scambiati una serie di frame NCP per configurare il livello di rete, ottenendo ad esempio un indirizzo IP da utilizzare per il collegamento;

PPP (Point to Point Protocol)

Ora il router è un nodo di Internet e può inviare e ricevere pacchetti IP.

In genere quando la rete non viene utilizzata, la connessione termina.

NCP viene utilizzato per sganciare la connessione del livello di rete e liberare l'indirizzo IP e LCP sgancia la la connessione del livello di linea.

PPPoE (Point to Point Protocol over Ethernet)

PPPoE è un protocollo di tunneling che permette di incapsulare il livello IP su una connessione tra due porte Ethernet pur mantenendo le caratteristiche di un collegamento PPP, per questo motivo è utilizzato per "comporre il numero" virtualmente di un altro computer dotato di connessione Ethernet e stabilire una connessione punto-punto con esso, su questa connessione vengono poi trasportati i pacchetti IP in base alle caratteristiche del PPP.

È protocollo normalmente utilizzato per il collegamento del router su linea DSL.