

# 1 Concetto fondamentale: Traffico “bursty” e allocazione dinamica

## 1.1 Il meccanismo di base: la multiplazione statistica

Una caratteristica degli apparati che trasmettono dati è di generare traffico in modo discontinuo. Una stazione che trasmette dati costituisce quindi una sorgente che emette delle “raffiche” (o “burst”) di bit consecutivi, intervallate spesso da lunghi silenzi. Questa caratteristica viene spesso enunciata dicendo che la sorgente è di tipo “bursty”.

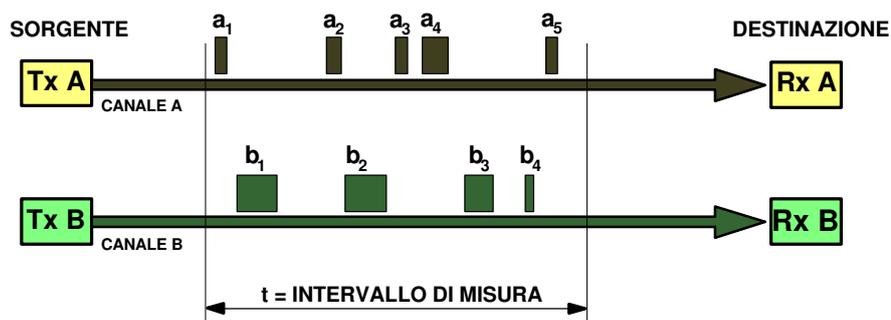


Figura 1 - Traffico di tipo “bursty”

Il rapporto fra la somma delle durate dei bit emessi e la durata totale della trasmissione costituisce il *tasso d’attività* della sorgente. Nell’esempio di Figura 1 una sorgente A invia dati a una destinazione A; il tasso d’attività, misurato nell’intervallo  $t$ , vale:

$$\rho_A = \frac{a_1 + a_2 + a_3 + a_4 + a_5}{t} = 0,29$$

dove  $a_i$  sono le durate dei vari *burst* emessi dalla sorgente. Come conseguenza, il canale trasmissivo che collega sorgente/destinazione risulta utilizzato mediamente al 29%.

Supponiamo ora di avere una seconda coppia B sorgente/destinazione e che la sorgente e la destinazione si trovino rispettivamente vicino alla sorgente A e alla destinazione A. Collegando i due elementi della coppia B mediante un secondo canale, questo risulta utilizzato al 52%. Complessivamente si sono impegnati due canali, ottenendo un’utilizzazione media:

$$\rho = \frac{29+52}{2} = 41,2\%$$

Evidentemente se le due coppie condividessero un unico canale esso si rivelerebbe sufficiente, risultando utilizzato all'83%.

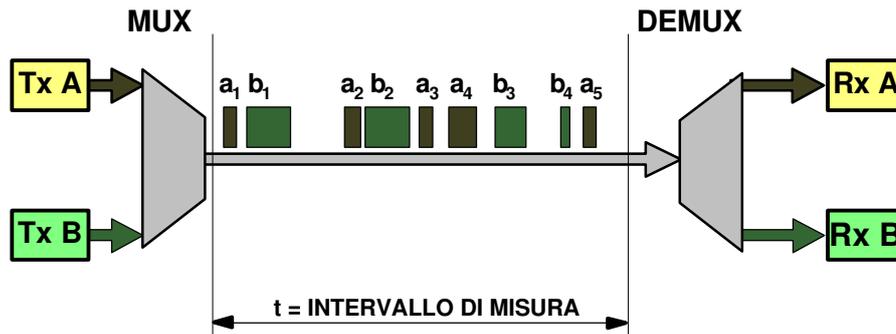


Figura 2 - Principio di funzionamento del moltiplicatore statistico

Il *moltiplicatore statistico*, illustrato in Figura 2, è appunto un apparato che ottiene questo tipo di condivisione concentrando più canali tributari corti (e quindi poco costosi, nell'esempio i due tributari A e B) in un unico *canale aggregato* (fase MUX) e disaggregando i tributari all'altro estremo (fase DEMUX). In un generico istante sul canale aggregato viene trasmesso il *burst* di A oppure di B, a seconda di quale delle due sorgenti (A oppure B) è attiva in quell'istante. Esprimiamo questo fatto dicendo che:

*“il moltiplicatore statistico realizza un’allocazione dinamica del canale aggregato”.*

Il moltiplicatore statistico non deve essere confuso col moltiplicatore TDM (*Time Division Multiplexing*), che serve a ricavare più canali da un unico mezzo fisico trasmissivo e che non realizza alcuna allocazione dinamica.

### 2.1 Il funzionamento “store-and-forward”

La moltiplicazione statistica opera secondo uno schema di funzionamento noto come *Store-and-Forward* (“memorizza e inoltra”) che, come dice il nome, presuppone che gli apparati di moltiplicazione siano dotati di memoria per immagazzinare il traffico in transito. Lo schema di Figura 3 rappresenta alcuni eventi relativi a una coppia MUX/DEMUX.

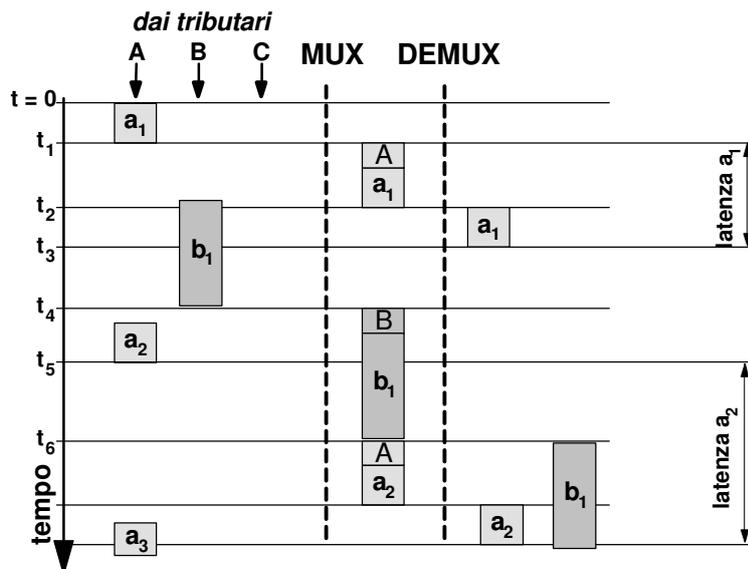


Figura 3 - Principio di funzionamento “Store-and-Forward”

All'istante  $t_1$  il *burst*  $a_1$ , proveniente da A, è ricevuto completamente da MUX. Questi gli antepone un'intestazione che porta l'identificativo di A e incomincia a ritrasmetterlo sul canale composito. All'istante  $t_2$  il *burst* è ricevuto completamente da DEMUX che esamina l'intestazione, da essa vede che il destinatario è A, elimina l'intestazione che non serve più e inizia a ritrasmetterlo sul canale tributario di A. All'istante  $t_3$  il *burst* è giunto a destinazione. Tutti i suoi bit hanno subito un ritardo  $t_3 - t_1$  dovuto alle operazioni di moltiplicazione/demoltiplicazione statistica. Questo ritardo è denominato *latenza*.

All'istante  $t_4$  il *burst*  $b_1$ , proveniente da B, è memorizzato da MUX e riceve lo stesso trattamento di  $a_1$ . All'istante  $t_5$  il *burst*  $a_2$ , proveniente da A, è ricevuto completamente da MUX. Questi gli antepone l'intestazione e tenta di ritrasmetterlo sul canale composito, che però è ancora occupato da  $b_1$ . Allora il *burst* viene accodato nella memoria di MUX e ritrasmesso, non appena il canale si libera, sul canale tributario di A. All'istante  $t_6$  il *burst* è giunto a destinazione. Esso, sebbene lungo quanto  $a_1$ , ha avuto una latenza maggiore.

Riassumendo si osserva che sono state fatte le seguenti ipotesi semplificative:

- Le operazioni di elaborazione richiedono tempi trascurabili
- Si è trascurato il ritardo di propagazione
- La frequenza di cifra (velocità trasmissiva) è la stessa per tutti i canali

In ogni caso valgono le seguenti conclusioni:

- La latenza dipende dalla durata del “*burst*”
- La latenza dipende dal traffico sul canale, quindi:

“*la latenza è una variabile casuale*”

Dal fatto che il processo d'arrivo dei *burst* e di conseguenza la latenza sono fenomeni casuali deriva l'aggettivo "statistico" con cui si caratterizzano gli apparati che funzionano in questo modo.

## 2 Reti di computer e reti per dati

Chiamiamo **reti di computer** il complesso formato dai nodi estremi (computer) e da tutto ciò che li collega. All'interno di questa struttura, chiameremo **sottorete di comunicazione** (o semplicemente *rete*) l'insieme dei nodi intermedi e dei circuiti che li collegano.

Mediante le architetture di marca (come SNA di IBM o DECnet di Digital) vengono costituite reti di computer del tipo cosiddetto "*legacy*" nelle quali la distinzione fra rete e sottorete di comunicazione tende a sparire, dato che entrambe le funzioni sono svolte dagli stessi apparati. Un'alternativa più aperta alla realizzazione di sistemi *multivendor* consiste nell'appoggiarsi alle reti pubbliche per dati (Public Data Network — PDN).

La Figura 4 mostra una rete di computer utilizzando una rete pubblica per dati. La parte di elaborazione viene svolta dai computer, mentre alla rete pubblica (che nel contesto globale è vista come una sottorete) sono affidate tutte le funzioni di comunicazione. Una soluzione del genere può naturalmente essere realizzata anche in ambito privato, nel senso che le tecnologie e gli standard relativi alla "nuvola" pubblica possono essere utilizzati per costituire reti private.

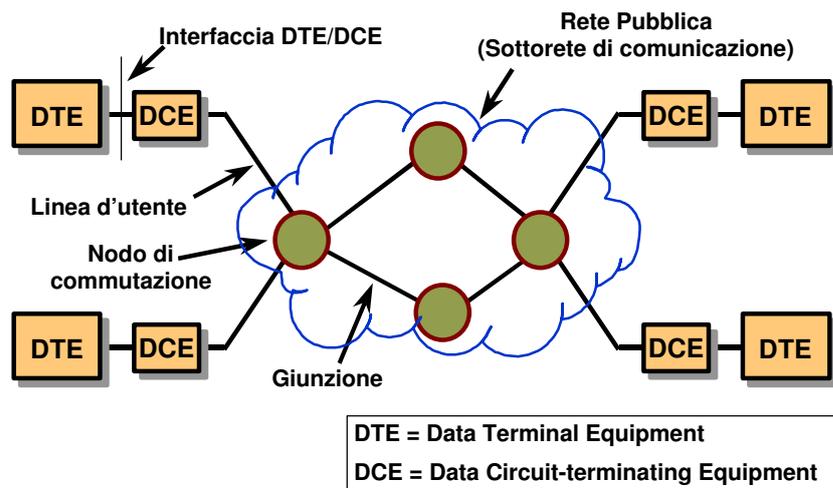


Figura 4 - Architettura di una rete pubblica per dati

La struttura di reti di questo tipo è molto simile a quella di una rete telefonica: anche qui un complesso di nodi di commutazione, collegati fra loro da circuiti di giunzione, fornisce accesso mediante linee d'utente a nodi terminali d'utente. Questi ultimi sono classificati come DTE e sono i computer (o i loro terminali). La linea d'utente è terminata dal DCE (un modem) che presenta verso il DTE un'interfaccia numerica. È a questa interfaccia che la rete pubblica fornisce il servizio all'utente ed è qui che vengono definiti i relativi standard di interazione utente-rete, come ad esempio la segnalazione.

Le differenze maggiori fra la rete telefonica e le reti dati risiedono nelle tecniche di gestione del traffico usate dai nodi di rete e si riflettono nel tipo di servizio fornito all'interfaccia DTE/DCE.

## 2.1 La commutazione di pacchetto

Il concetto *Store-and-Forward* della multiplazione statistica può essere esteso, da una semplice concentrazione fra due siti connessi tramite un circuito, su tutta una rete. Se l'intestazione del *burst* contiene la destinazione finale, ogni nodo attraversato può esaminarla e in base ad essa ritrasmettere il blocco di dati sull'appropriata giunzione, fino all'utente finale. I blocchi così organizzati sono chiamati **pacchetti** e questo modo di smistare il traffico attraverso un'intera rete costituisce la **Commutazione di Pacchetto**.

Attraverso un'interfaccia utente/(rete a pacchetto) possono benissimo transitare, alternati fra loro in modo casuale, pacchetti con differenti destinazioni. Grazie a ciò un computer può mantenere, attraverso un'unica linea fisica d'accesso, più comunicazioni contemporanee con interlocutori diversi. Per fare lo stesso attraverso una rete a circuito occorrerebbero tante linee fisiche quanti sono gli interlocutori.

Le reti a pacchetto sono nate con ARPANET, che era stata progettata per collegare fra loro dei computer. Per questo tipo di impiego la tecnica a pacchetto è perfettamente adeguata. Per il servizio voce i ritardi intrinsecamente maggiori, e soprattutto il fatto che la rete non fornisce alcuna garanzia sulle variazioni di ritardo (*jitter*), fanno sì che il trasporto a pacchetto della voce non costituisca una tecnica accettabile per il servizio telefonico pubblico. Tuttavia essa ha un limitato impiego in servizi (come voce su IP) caratterizzati da qualità bassa e non costante e da prezzi corrispondentemente esigui.

Il funzionamento *Store-and-Forward* permette di ricevere i dati a una certa velocità e ritrasmetterli a una velocità diversa. Questa caratteristica viene sfruttata comunemente, ad esempio il circuito composito fra una coppia di multiplatori statistici ha spesso velocità maggiore dei tributari; similmente, le giunzioni in una rete a pacchetto hanno di solito maggior capacità delle linee d'utente, dovendo smaltire maggior traffico. In definitiva le velocità dei circuiti si dimensionano a traffico, mettendo linee più veloci dove ci sono maggiori esigenze. Di conseguenza possono essere messi in comunicazione due utenti che accedono alla rete a velocità diverse. Questo in una rete a circuito non è possibile.

## 2.2 Reti a Datagramma

La commutazione di pacchetto implica che gli utenti (ossia i DTE) siano capaci di scambiare con la rete, seguendo procedure ben precise, pacchetti opportunamente organizzati. Un DTE avente tali caratteristiche è un DTE a pacchetto. Nell'esempio di Figura 5 due DTE a pacchetto comunicano fra loro.

Il DTE di sinistra consegna alla rete il pacchetto No 1 e poi il pacchetto No 2. Il primo viene instradato attraverso il nodo di transito superiore e quindi consegnato al DTE di destra. Il secondo viene instradato attraverso il transito inferiore. Questo differente instradamento può essere causato da vari motivi, ad esempio dal fatto che all'arrivo del pacchetto No 2 il transito inferiore risulta più scarico ed è quindi preferito. Come conseguenza può succedere, come illustrato, che il pacchetto 2 segue un percorso più veloce e arriva fuori sequenza.

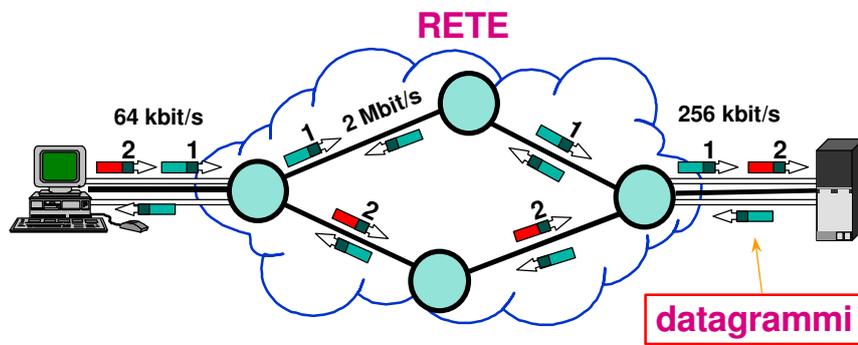


Figura 5 - “Store-and-Forward” in una rete a datagramma

Un funzionamento come quello descritto, in cui ogni pacchetto è considerato dalla rete come un’entità a sé stante privo di correlazioni coi pacchetti che lo precedono o lo seguono, si chiama funzionamento del tipo a *datagramma* e i pacchetti sono chiamati datagrammi.

Ogni datagramma deve recare l’indirizzo completo del destinatario e, oltre a non ricevere garanzie di consegna in sequenza, può andare perso (buffer che traboccano, errori di trasmissione, ..) o arrivare duplicato (timer che scadono e ritrasmissione di una seconda copia). La rete fa del suo meglio per trasferire il traffico correttamente ma non ha alcun meccanismo interno di controllo e di conseguenza non offre alcuna garanzia. Si dice che una rete a datagramma funziona in modo “*best effort*”.

La rete **ARPANET**, che è stata la prima rete a pacchetto del mondo e da cui deriva Internet, funzionava a datagramma.

### 2.3 Circuiti virtuali

Un modo di funzionamento alternativo al datagramma è quello a **circuito virtuale**. Il circuito virtuale è un servizio che consiste nell’emulazione di un circuito fisico.

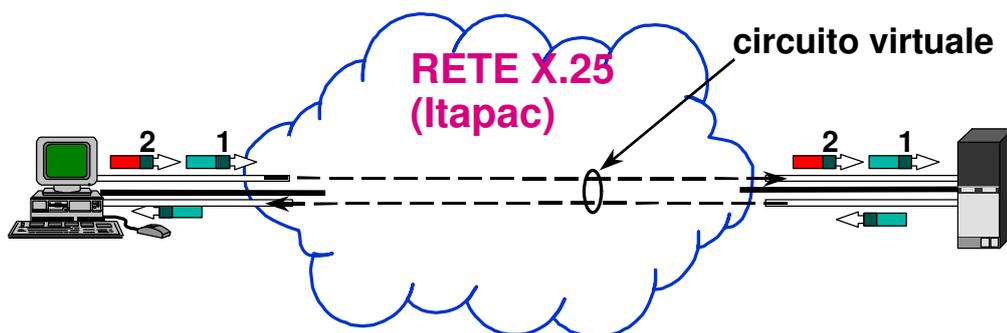


Figura 6 - “Store-and-Forward” in una rete a circuito virtuale

Nell’esempio di Figura 6 il DTE di sinistra consegna alla rete a circuito virtuale il pacchetto No 1 e poi il pacchetto No 2. Dato che — indipendentemente da ciò che avviene nella rete — per i due DTE tutto si svolge (all’incirca) come se fra i due interlocutori esistesse un circuito, i pacchetti non possono sorpassarsi e vengono quindi consegnati in sequenza.

In aggiunta, una rete a circuito virtuale garantisce da perdite (la rete realizza degli algoritmi interni che provvedono a eventuali ritrasmissioni) e da duplicazioni. Esistono inoltre dei meccanismi di controllo di flusso all'interfaccia utente/rete che servono a rallentare il flusso in caso che ci sia pericolo di congestione (una rete a datagramma in caso di congestione scarta semplicemente i pacchetti).

Non c'è bisogno che ogni pacchetto rechi l'indirizzo completo del destinatario, poiché una volta immesso in un circuito virtuale il pacchetto non può arrivare che all'altro estremo del circuito stesso. Al suo posto il pacchetto porta un numero (solitamente di 12 bit) che identifica il circuito virtuale. Come conseguenza di ciò, in una rete a circuito virtuale si ha un *overhead* minore che in una a datagramma, in cui ogni indirizzo occupa uno spazio maggiore (32 bit in ARPANET/Internet).

Come contropartita di quanto visto, occorre per ogni coppia di interlocutori definire un circuito virtuale. Questa definizione può avvenire:

1. su base chiamata, ossia in modo interattivo mediante segnalazione fra l'utente chiamante e la rete. In tal caso si ha il servizio di **Chiamata Virtuale (VC)**.
2. su base contrattuale, ossia facendosi assegnare dal gestore un circuito virtuale fisso fra due punti. In tal caso si ha il servizio di **Circuito Virtuale Permanente (PVC)**.

Questi servizi di rete sono specificati dalla Raccomandazione X.25, che definisce tutto ciò che avviene all'interfaccia fra un DTE a pacchetto e una rete pubblica a pacchetto. In Italia la principale rete pubblica di questo tipo è *Itapac*, gestita da Telecom Italia.

### 3 La Raccomandazione X.25

La Raccomandazione X.25 specifica ciò che accade fra DTE e DCE. Dato che la comunicazione ha luogo solo se il DTE rispetta le regole, l'interfaccia specifica anche i protocolli superiori al fisico che sono interpretati dalla rete. Si dice che la rete **termina** i protocolli di livello 1,2,3.

La Raccomandazione X.25 definisce l'interfaccia fra un DTE e una rete pubblica.

Il DTE e la rete devono scambiarsi dati organizzati a pacchetti. Attraverso l'interfaccia l'utente riceve un servizio di circuito virtuale. Più circuiti virtuali possono coesistere su una stessa interfaccia.

Per poter distinguere fra un circuito virtuale e un altro, i pacchetti scambiati attraverso l'interfaccia devono contenere un identificativo di circuito virtuale.

I protocolli dei livelli 4-7 sono invece terminati dall'altro DTE. In definitiva l'interfaccia equivale a una **pila di protocolli**, tanti quanti sono terminati dalla rete. Chiameremo **interfaccia di telecomunicazione** questa interfaccia, da non confondersi con l'interfaccia del modello OSI che è interna a un sistema e il cui funzionamento è descritto da un insieme di primitive.

A livello 1 l'interfaccia X.25 prevede varie opzioni, essenzialmente interfaccia fisica EIA/TIA-232 (ex RS-232) oppure X.21.

A livello 2 si usa un protocollo della famiglia HDLC definito da ISO, con la procedura LAPB (*Link Access Procedure Balanced*), in pratica con LAPB si indica l'intero protocollo.

A livello 3 si usa un protocollo specifico, chiamato PLP (*Packet Layer Protocol*) che costituisce la parte più significativa dell'interfaccia dato che è qui che vengono usati i vari pacchetti di segnalazione (CALL REQUEST, ecc.) e di trasferimento dati tipici della rete X.25.

All'interno della rete vengono usati protocolli di marca che non sono oggetto di normalizzazione. Spesso (ma non necessariamente) le reti commerciali utilizzano al loro interno protocolli del tipo a circuito virtuale, simili all'X.25.

### 3.1 Gestione delle chiamate

Il diagramma sequenziale di Figura 7 mostra un semplice esempio di Chiamata Virtuale X.25. Le due linee verticali rappresentano le interfacce DTE/DCE dei due utenti che si parlano, fra di esse c'è la rete.

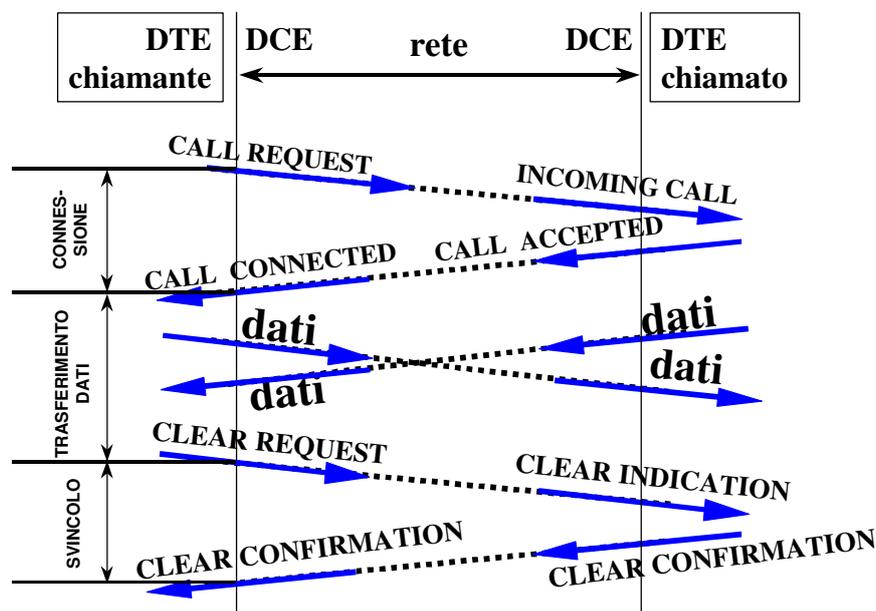


Figura 7 - Chiamata virtuale X.25

Il DTE di sinistra è il chiamante. Per iniziare la chiamata esso consegna alla rete un pacchetto di segnalazione denominato **CALL REQUEST**. Questo pacchetto è l'unico che contiene l'indirizzo completo del DTE chiamato. La rete esamina il pacchetto e avvisa l'altro utente che c'è una chiamata in arrivo, inoltrandogli un pacchetto **INCOMING CALL**. Il chiamato accetta e risponde con un pacchetto **CALL ACCEPTED**. A questo punto la chiamata, che si trovava in fase di **connessione**, passa in fase **trasferimento dati**. In questa fase l'utente può consegnare alla rete pacchetti dati, che vengono recapitati inalterati (tranne alcuni campi nell'intestazione). La rete comunica l'accettazione della chiamata mediante il pacchetto **CALL CONNECTED**. Alla ricezione di questo pacchetto la chiamata, che era in fase di connessione, passa in fase trasferimento dati anche all'interfaccia di sinistra.

Alla fine della chiamata uno dei due utenti (nell'esempio il chiamante) decide di chiudere. Manda pertanto in rete un pacchetto di **CLEAR REQUEST**, la chiamata passa in fase di **svincolo**, avviene uno scambio di pacchetti di segnalazione come illustrato in figura alla fine del quale il circuito virtuale viene eliminato completamente.

Come si vede, una Chiamata Virtuale evolve in modo del tutto analogo a una chiamata telefonica. Un Circuito Virtuale Permanente corrisponde invece a un circuito dedicato (CDN) e si trova sempre in fase trasferimento dati.

### 3.2 I formati dei dati

Ogni quadrato in Figura 8 rappresenta un ottetto. Il *pacchetto* X.25 è contenuto in un *frame*.

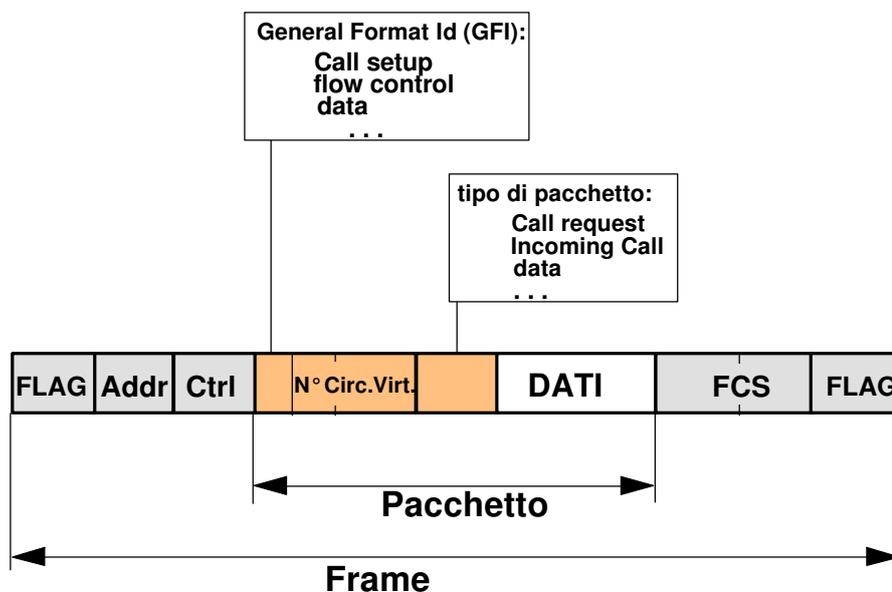


Figura 8 - Formati dei dati in X.25

Il formato del *frame* è quello tipico del protocollo HDLC e comprende: un campo *Address* (residuo di vecchie procedure, in pratica inutilizzato); un campo *Control* che serve a gestire il flusso delle frame fra rete e utente (soprattutto a chiedere ritrasmissioni e ad esercitare il controllo di flusso); un campo *Frame Check Sequence* (FCS) basato su un codice a ridondanza ciclica e che serve al ricevitore per controllare che non ci siano stati errori di trasmissione. Il tutto è compreso fra due FLAG che servono a identificare l'inizio e la fine del *frame*.

Il pacchetto è costituito da un'intestazione, seguita dai dati utili. Il pacchetto può essere di tipo *dati* oppure di tipo *controllo*; in quest'ultimo caso manca il campo DATI.

L'intestazione ha un formato dipendente dal tipo di pacchetto, come minimo occupa tre ottetti.

I primi 4 bit costituiscono il General Format Identifier (GFI) e servono a discriminare fra le classi principali di pacchetto.

I successivi 12 bit sono l'identificativo di circuito virtuale.

Il terzo ottetto identifica il tipo di pacchetto.

### 3.2.1 I vecchi terminali e la funzione PAD

Non tutti gli apparati dati, soprattutto all'epoca in cui vennero introdotte le prime reti pubbliche, sono in grado di gestire la complessità dell'interfaccia X.25.

In particolare i semplici terminali asincroni a carattere non possono collegarsi alla rete, che oltre alle capacità elaborative e di memoria richiede trasmissione sincrona sulla linea d'utente. Per dare servizio a questi terminali occorre convertire il flusso da essi generato in e da pacchetti X.25, per poi inoltrarlo su un circuito virtuale.

Questa funzionalità, definita **PAD** (Packet Assembly/Disassembly) è stata considerata di primaria importanza ed è stata standardizzata in modo da poter essere conglobata dalla rete. Con PAD si indica anche il dispositivo (hardware+software) che svolge la funzione. In pratica le reti pubbliche a pacchetto oltre che ai DTE a pacchetto offrono accesso, su porte speciali, anche a DTE asincroni. La standardizzazione è costituita da tre Raccomandazioni (X.3, X.28, X.29) che specificano rispettivamente:

- X.3**: Caratteristiche del dispositivo PAD di una rete pubblica;
- X.28**: Interfaccia fra DTE a caratteri e PAD;
- X.29**: Procedure di controllo fra PAD e DTE a pacchetto.