

Rete ATM

Introduzione ad ATM	2
ATM UN PO' DI STORIA	2
ATM Asynchronous Transfer Mode	2
<i>STM Synchronous Transfer Mode</i>	3
Applicazioni e benefici	4
ATM networks	5
Introduzione alla terminologia.....	5
L'architettura a strati di ATM	6
ATM Physical Layer	8
ATM Data Link Layer	9
FUNZIONALITÀ.....	9
FORMATO DELLA CELL	9
<i>ATM CELL HEADER</i>	9
Formato NNI.....	9
CELL LOSS PRIORITY	10
CYCLICAL REDUNDANCY CHECKS HEADER.....	10
Formato UNI.....	10
GENERIC FLOW CONTROL	11
SET-UP DI UNA CONNESSIONE.....	11
<i>PROTOCOLLO DI SET-UP PER UNA CONNESSIONE POINT TO POINT</i>	12
ROUTING.....	13
ATM SWITCH.....	14
CONGESTIONE	15
ATM ADAPTATION LAYER	16
FUNZIONALITÀ.....	16
I PROTOCOLLI	17
AAL 1	17
<i>AAL 1 SAR</i>	17
<i>AAL 1 Convergence Layer</i>	18
AAL 3/4.....	18
<i>AAL 3/4 SAR</i>	18
<i>AAL 3/4 Convergence Layer</i>	19
AAL 5	20
<i>AAL 5 SAR</i>	21
<i>AAL 5 Convergence Layer</i>	21

Introduzione ad ATM

ATM UN PO' DI STORIA

A partire dal 1960 è incominciata a livello mondiale l'integrazione della **rete telefonica analogica** con sistemi capaci di gestire sia segnali analogici che digitali. L'introduzione di **fibres ottiche** a bassa perdita di dati ha reso questo mezzo tecnologico il supporto ideale per la costruzione di una *rete di telecomunicazioni ad estensione geografica*. La vera novità è stata però la necessità di gestire *applicazioni multimediali* conglobanti voce, video e dati, il tutto spesso in tempo reale. Il problema che nasceva era quello di unire tipi di traffico con caratteristiche molto diverse. Ad esempio, la voce richiede un'occupazione costante di banda ma in quantità limitata, mentre i dati richiedono spesso molta banda in maniera discontinua. La voce ed il video non ammettono ritardi, ma ammettono perdita di informazioni al contrario del traffico dati.

Inizialmente le compagnie telefoniche pensarono ad un modo per portare lo standard **ISDN** a velocità dell'ordine dei gigabit e consentire così l'integrazione dei nuovi servizi. Ben presto però realizzarono che il modello **ISDN** non si prestava facilmente a tale scopo. Un problema del canale fisso di **ISDN** è quello di non essere mai della dimensione giusta. La larghezza di banda di **64 Kbps** era stata scelta perchè era la quantità richiesta per il trasporto della voce in forma digitale¹. I progressi tecnologici hanno consentito però la drastica riduzione di questa dimensione e di conseguenza la diminuzione della banda necessaria per il traffico vocale. Tuttavia un' applicazione vocale che usa **ISDN** non può beneficiare di questo progresso, in quanto deve comunque allocare l'intero canale da 64 Kbps. Un altro problema è che sebbene sia possibile creare dei canali da 1 gigabit (allocando 16000 canali da 64 Kbps), il loro effettivo utilizzo è difficoltoso. Infatti per spedire una quantità grande di informazioni, bisogna spezzare i dati in più pezzi e spedirli su differenti circuiti (*striping*). Lo **striping** è però molto oneroso in quanto richiede dell'hardware molto complesso che possa risincronizzare i vari pezzi provenienti dai circuiti allocati. Tutto ciò ha portato allo sviluppo di ATM.

ATM ASYNCHRONOUS TRANSFER MODE

ATM è una particolare forma di **cell network** sviluppata prima dal **CCITT** e poi dal **ITU** all'interno del progetto **B-ISDN** (*broadband integrated services digital network*).

Con il termine **cell** si intende un pacchetto di dimensioni fisse, solitamente molto piccole, che nel caso di ATM è di **53 bytes**. Ogni cell consiste di 5 byte di **header** e 48 byte di **payload**.

Il termine **asincrono** che compare nella sigla si riferisce al fatto che le cells di un certo utente non devono essere periodicamente spaziate all'interno del flusso complessivo di cell. In altre parole, agli utenti non sono assegnate posizioni fisse all'interno di un frame come accade per la commutazione a circuito. Per capire meglio la questione vediamo prima il predecessore di ATM.

¹ Assumendo per il segnale vocale una banda lorda di **4 kHz**, la frequenza di *campionamento* (in base al teorema del campionamento) è di 8 kHz, cioè 8000 campioni al secondo; se ad ogni campione si associano, mediante il processo di *quantizzazione*, 8 bit, abbiamo $8000 \cdot 8 = 64000$ bit al secondo.

STM Synchronous Transfer Mode

ATM rappresenta la naturale evoluzione di **STM** (*Synchronous Transfer Mode*), una *rete a commutazione di circuito* usata per il trasferimento dati e voce. In questo caso, una connessione tra due nodi finali deve essere stabilita prima che cominci il trasferimento dei dati e deve essere rilasciata quando la trasmissione finisce. In questo modo i punti finali riservano staticamente ed in anticipo la larghezza di banda richiesta per l'intera durata della connessione.

Il trasferimento di dati in una *STM network* avviene suddividendo la larghezza di banda dell'**STM link** (solitamente T1 e T3 link) in una unità fondamentale di trasmissione chiamata **time-slot** o **bucket**. Questi time-slot sono organizzati in una sorta di treno, che contiene un numero fissato di time-slot, etichettati da 1 a N. Per ogni STM link ci possono essere M differenti treni, etichettati da 1 a M, che si ripetono ogni periodo T di tempo, con i time-slot disposti sempre nel medesimo ordine. I parametri T, N ed M sono stabiliti dal comitato degli standard e variano da Europa ad America.

Ad ogni connessione, tra due **end point** di un certo STM link, viene assegnato un time-slot i ($1 \leq i \leq N$) su di un certo treno j ($1 \leq j \leq M$) e i dati di quella connessione sono sempre trasportati in quel bucket. In questa maniera si possono multiplexare più connessioni su uno stesso link.

Il grosso svantaggio, però, è che quando una connessione non ha traffico, il suo time-slot di output è sprecato (se non ci sono dati da trasmettere il bucket viene spedito vuoto). In questa maniera nessun'altra connessione può usufruire dello spazio vuoto (il time-slot assegnato rimane allocato in modo esclusivo per tutta la vita della connessione) e se il numero di slot inutilizzati per ogni treno è alto, si ha uno spreco significativo della larghezza di banda. In più essendo fissi il numero di treni e di slot, vi potranno essere al massimo $N \cdot M$ connessioni.

ATM rappresenta una soluzione a questi problemi. L'idea di partenza è quella di non identificare una connessione dal numero del time-slot, ma di spedire l'identificatore di connessione nel bucket insieme ai dati. Questo viene fatto usando un pacchetto di dimensioni fisse (cell), contenente uno header in cui viene posto l'identificatore (detto VPI-VCI) di connessione.

La scelta della dimensione della cell incide sulle prestazioni della rete. Infatti una cell piccola garantirà un minor spreco di banda, considerando che mediamente l'ultima cell di un pacchetto viaggerà vuota. Inoltre una dimensione contenuta permetterà la riduzione del tempo di serializzazione (il tempo necessario ad uno switch per trasmettere la cell sul giusto link). Non va dimenticato che uno degli scopi di ATM è quello di permettere un traffico integrato e per far questo è necessario che alcuni tipi di dati (voce, video) non subiscano eccessivi ritardi.

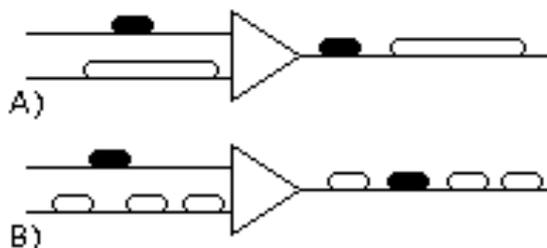


Fig. 1.1. Serializzazione di una cell

Se la dimensione della cell non fosse costante e piccola, potrebbero verificarsi dei casi come quello in figura 1.1.a, dove una cell urgente deve aspettare la trasmissione di un pacchetto più grande prima di poter procedere. Se il pacchetto fosse stato spezzato in più cell la situazione sarebbe stata quella di figura 1.1.b, dove la cell urgente non subisce alcun ritardo.

APPLICAZIONI E BENEFICI

ATM ha diversi benefici :

- Una rete ATM è capace di integrare tutti i tipi di traffico:
 1. voce.
 2. video.
 3. dati.

- Grazie alla sua alta velocità e all' integrazione di tipi di traffico diversi, ATM permetterà lo sviluppo di nuove applicazioni :
 1. Video conferenze.
 2. Video on demand.
 3. Come supporto per reti distribuite più efficienti. Computer distribuiti geograficamente, sia su reti locali che geografiche, possono essere usati per risolvere problemi complessi, la cui soluzione richiedeva l'uso di computer molto costosi. Un programma applicativo, può essere partizionato e distribuito nella rete, per sfruttare le particolari capacità di ogni macchina e l'utilizzo dei circuiti virtuali garantisce un flusso veloce e continuo di dati tra i nodi di calcolo.
 4. Applicazioni grafiche interattive che richiedono velocità dati molto alta e ritardi minimi (visualizzazione di dati scientifici).

- Compatibilità. ATM non è basato su uno specifico mezzo fisico. ATM può essere usato su fibre ottiche, doppino, cavo coassiale.

- ATM è una tecnologia emergente, nata con il proposito di essere uno standard internazionale e non il prodotto di un singolo venditore.

Questi benefici hanno tuttavia un prezzo. Infatti ATM è una tecnologia molto complessa, forse una delle più complicate mai sviluppate nell'industria di rete. Una ATM network richiede un'*infrastruttura di protocolli* molto complessa per gestire il lavoro di *switch* e di *internetworking*.

ATM networks

INTRODUZIONE ALLA TERMINOLOGIA

Una rete ATM è composta da un insieme di ATM switch interconnessi punto a punto tra loro mediante interfacce.

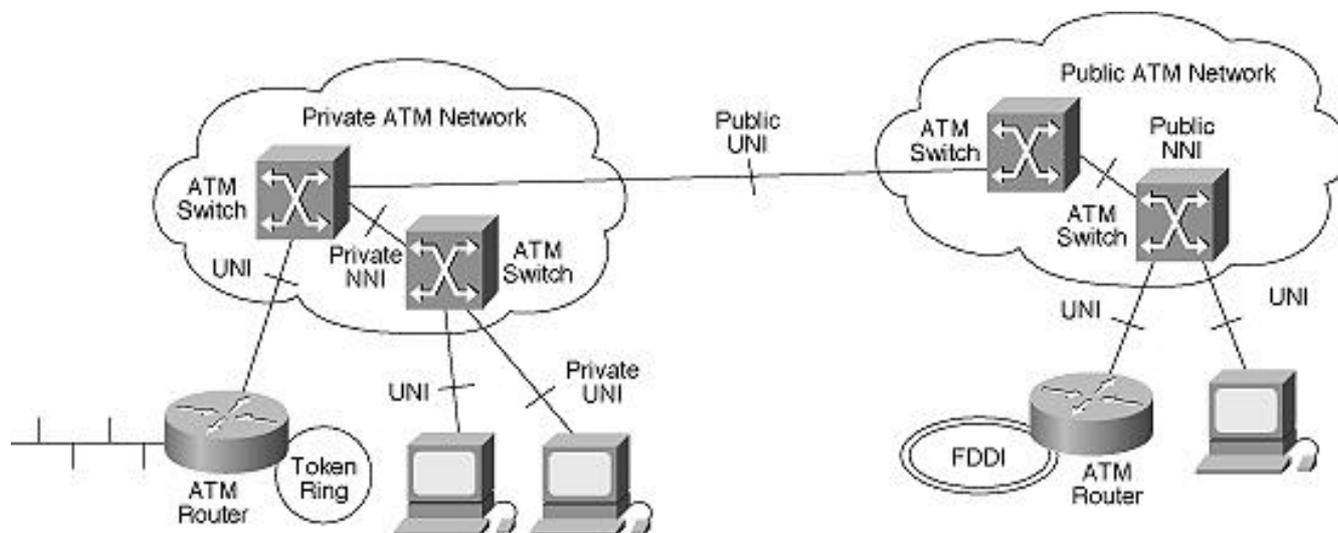


Fig. 2.1. Rete ATM

Le interfacce che uno switch ATM supporta sono di due tipi: la **user-network interface (UNI)** e la **network-network interface (NNI)**.

UNI connette gli **ATM end-system** (*host, routers, ecc.*) ad uno *switch*, mentre NNI è l'interfaccia che connette due switch. Più precisamente è un *link fisico* su cui i due switch si scambiano informazioni secondo il formato NNI. Va fatta questa precisazione perchè l'interfaccia tra uno switch privato ed uno pubblico è di tipo UNI (detta *public UNI*), dal momento che questi switch non si scambiano informazioni secondo il formato NNI.

L'operazione base degli switch è molto semplice: ricevere una cell da un link e determinare la porta su cui ritrasmettere la cell. Vedremo più avanti ed in maggior dettaglio gli switch, il formato UNI e quello NNI.

Diciamo invece fin da ora che *una rete ATM è fondamentalmente orientata alla connessione, ossia prima di poter trasmettere un qualsiasi dato, un circuito virtuale deve essere stabilito nella rete ATM tra sorgente e destinazione*. Questo circuito deve poi essere identificato unicamente in tutta la rete ATM e per farlo si è adottata la seguente strategia distribuita.

Per ogni link, i due *end switch* stabiliscono una coppia di identificatori **VPI** e **VCI** che identificano il **circuito virtuale** in quella tratta.

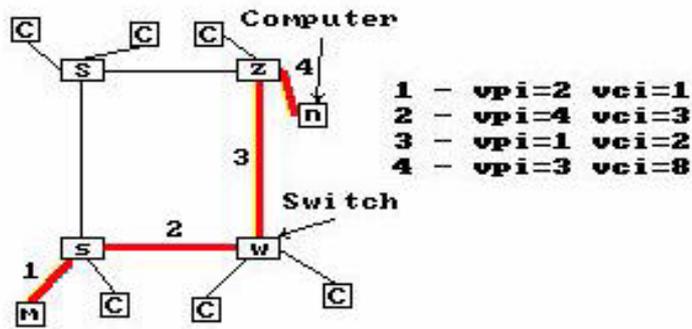


Fig. 2.2. Circuito Virtuale

La figura 2.2 ci mostra un esempio di quanto appena enunciato. Vi sono due end-system (m e n) ed un circuito virtuale che li congiunge. Il circuito passa attraverso quattro diversi link, ed in ognuno di questi la connessione viene identificata da una diversa coppia di VPI-VCI. Poichè per un certo link il valore della coppia è determinato dagli switch, questi saranno sempre in grado di scegliere dei valori distinti per identificare connessioni diverse che passino per quel link.

Il fatto che vi siano due identificatori è strettamente legato ad un'idea gerarchica di ATM. Tutti i circuiti virtuali con lo stesso VPI si dicono appartenere ad un **cammino virtuale** (*virtual path*). In questo modo tutte le cell che competono ad un certo virtual path (cioè quelle con lo stesso VPI) possono essere trasmesse negli switch semplicemente controllando il campo VPI e tralasciando quello VCI, che dovrà essere controllato solo negli switch connessi con un UNI, dove le cell dovranno essere *demultiplexate* sul giusto circuito virtuale. In questo modo si possono multiplexare più connessioni lungo lo stesso link basandosi sulle caratteristiche del loro traffico, contando sul fatto che sia statisticamente improbabile che tutte le connessioni abbiano un picco di trasmissione nello stesso momento. In questo modo la somma complessiva delle richieste massime di banda di ogni connessione può eccedere la larghezza di banda del link.

L'ARCHITETTURA A STRATI DI ATM

ATM ha la sua **gerarchia di protocolli**. Come si vede dalla figura sono stati definiti tre diversi layer.

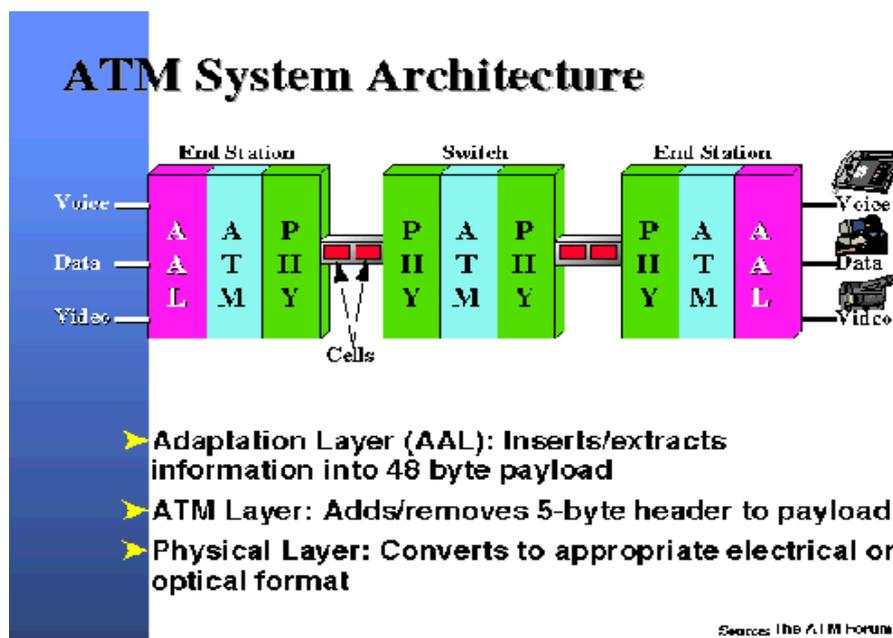


Fig. 2.3. ATM layers

Il primo è il **physical layer** (*strato fisico*), che ha sostanzialmente le stesse funzionalità del *livello 1 di OSI* (definire le caratteristiche elettriche e trasmettere i bit).

Vi è poi l'**ATM data link layer** (detto **ATM layer**), che si occupa sia dell'aggiunta dei 5 byte di *header* al *payload*, che della trasmissione delle cell (compreso il **routing**). Ricopre quindi i *livelli 2 e 3 di OSI*, ma a differenza del *data link OSI* non si occupa però del recupero delle cell perse o danneggiate.

Infine vi è l'**ATM adaptation layer (AAL)**, che divide e riassume i pacchetti nei 48 byte di *payload* che costituiranno le cell ATM, assicurando diverse *classi di servizio*. Qualcosa di simile vi è nel *livello 4 di OSI*. Il servizio fornito dallo strato AAL non è comunque affidabile, dato che non viene effettuata alcuna ritrasmissione e non ci sono degli *acknowledgements (ACK, conferme positive)* per segnalare quello che è stato ricevuto. Un servizio di consegna affidabile deve quindi essere realizzato in uno strato superiore a quelli di ATM, per esempio usando le ATM cells per trasportare il traffico TCP/IP.

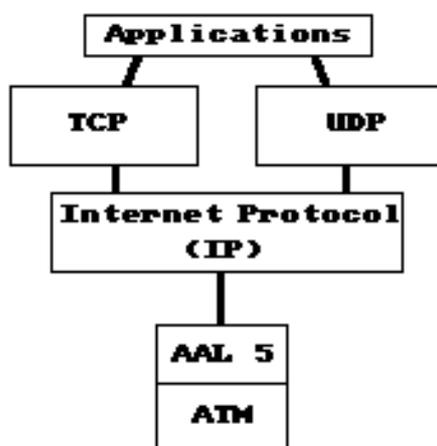


Fig. 2.4. TCP/IP usato con ATM

Nei prossimi capitoli vedremo più in dettaglio i vari strati.

ATM Physical Layer

Lo **strato fisico** (*Physical Layer*) si occupa della trasmissione di una cell tra due nodi della rete attraverso un qualche mezzo fisico. Il PL è diviso in due *sottostrati*: uno dipendente dal mezzo (responsabile della corretta trasmissione e ricezione dei bit nel mezzo fisico) ed un sottostrato chiamato **transmission-convergence** (responsabile del *mapping* delle ATM cell nel mezzo di trasporto usato).

Solitamente per la trasmissione su lunghe distanze vengono usate fibre ottiche e lo standard usato è quello **SONET** (*synchronous optical network*). Il SONET era uno standard esistente negli Stati Uniti, che il **CCITT** ha poi integrato nello standard più generale **SDH** (*synchronous digital hierarchy*).

In SONET l'unità base è un array 9x90 di bytes, chiamato **frame**. Di questi 810 bytes, 27 sono di **overhead** ed i restanti sono di **payload**. Come si vede dalla figura 3.1, l'overhead costituisce le prime tre colonne. Esso viene usato per gestire gli errori, la sincronizzazione e per identificare il tipo di payload.

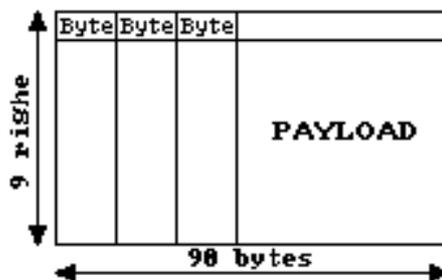


Fig. 3.1. Frame SONET

Un frame SONET è trasmesso ogni 125 μ sec (che è il periodo di campionamento che si ottiene prendendo 8000 campioni al secondo) per un totale di 51.840 Mbps. Questa è considerata la velocità base ed il canale corrispondente è chiamato **OC-1** (*optical carrier*). Il resto della gerarchia è dato dalla figura 3.2.

Level	Line Rate (Mbps)
OC-1	51.84
OC-3	155.52
OC-9	466.56
OC-12	622.08
OC-18	933.12
OC-24	1244.16
OC-36	1866.24
OC-48	2488.32

Fig. 3.2. Gerarchia SONET

Si noti che OC-n indica n canali OC-1 indipendenti, mentre con OC-nc si intende un unico canale ad alta velocità.

Solitamente in ATM vengono usati canali OC-3c (155, 52 Mbps). Alcune volte viene usato il termine **STS-n** per indicare un canale OC-n prima della sua conversione in segnale ottico.

L'ATM cell vengono poi incapsulate in un frame SONET usando uno specifico protocollo.

ATM Data Link Layer

FUNZIONALITÀ

L' **ATM DLL** (*ATM Data Link Layer*) offre agli strati superiori la visione di un servizio di trasferimento di pacchetti di dimensione fissa da 53 bytes (cell) orientato alla connessione. Questo layer è responsabile inoltre delle seguenti attività:

- inizializzazione di una connessione;
- control flow;
- routing delle cell;
- scheduling delle cell negli switch.

Non si occupa del recupero delle cell danneggiate o perse.

È importante notare che le funzioni realizzate dall'ATM DLL sono state ideate per essere eseguite in hardware ad elevata velocità.

FORMATO DELLA CELL

Abbiamo già detto che una ATM cell è composta da **53 bytes**, di cui 5 costituiscono lo **header** e i restanti 48 la parte dati (**payload**, cioè *carico pagante*). La dimensione del campo payload è derivata dal compromesso tra chi voleva una dimensione di 128 bytes, per il trasporto del traffico dati, e chi la voleva di 16 bytes, per gestire il traffico vocale di una rete telefonica.

ATM CELL HEADER

L' **ATM cell header** ha due diversi formati: uno è per le cell che giungono all' UNI da un utente ATM, l'altro per le cell scambiate tra due switch (formato NNI).

Formato NNI

Vediamo per primo il formato NNI, mostrato in figura 4.1.

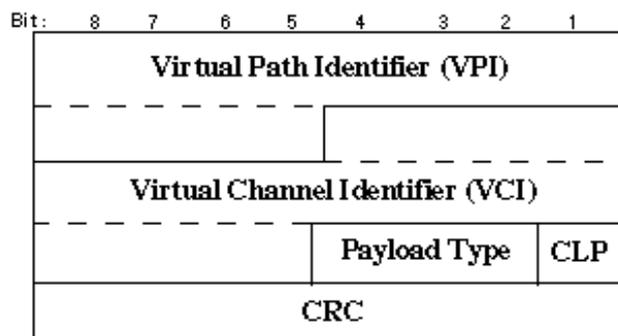


Fig. 4.1. Formato NNI

I primi due campi sono il 12-bit **virtual path identifier (VPI)** e il 16-bit **virtual circuit identifier (VCI)**. Una connessione ATM è unicamente identificata dalla combinazione di questi due identificatori. La loro importanza è legata al fatto che sono indispensabili per il routing delle cell attraverso i vari nodi di switch.

Vi è poi il campo **payload-type** utilizzato per distinguere le cell dati da quelle di controllo.

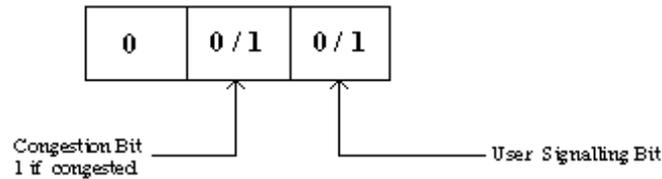


Fig. 4.2. payload-type

Come si vede dalla figura 4.2 esso è composto da **3 bit**:

- se il primo bit è 0 allora la cell è una **user-data cell**: In questo caso il secondo bit è usato per indicare se la cell ha incontrato congestione nella rete ed il terzo bit è usato da uno degli adaptation layer (AAL 5) per segnare la fine di un datagramma.
- se il primo bit è posto ad 1, allora la cell è destinata alla gestione e all'amministrazione della rete.

CELL LOSS PRIORITY

Il campo **CLP** (*Cell Loss Priority*) è costituito da un singolo bit. È usato per indicare se una cell può essere persa durante un periodo di *congestione* della rete. Se uno switch deve togliere dalle sue code di attesa delle cell, le prime ad essere cancellate saranno quelle con il CLP posto ad 1.

Ad esempio i dati vocali possono supportare la perdita di cell senza bisogno di ritrasmissione e quindi un'applicazione può assegnare 1 al CLP del suo traffico vocale.

CYCLICAL REDUNDANCY CHECKS HEADER

Il campo **CRC** (*Cyclical Redundancy Checks Header*) è usato per proteggere lo header dagli errori di trasmissione. Il CRC è quindi calcolato solo sui primi 5 bytes della cell.

È da notare che il CRC va controllato e ricalcolato ad ogni switch della rete ATM, dato che lo header cambia in ogni nodo (il VCI e il VPI vengono sempre variati ed anche il bit di congestione nel PAYLOAD-TYPE può cambiare)

Formato UNI

Vediamo ora il formato **UNI**. Come si vede dalla figura 4.3 l'unica differenza è l'introduzione del campo **GFC** (*Generic Flow Control*) ed il ridimensionamento del campo **VPI** (*Virtual Path Identifier*), ora di solo 8 bit.

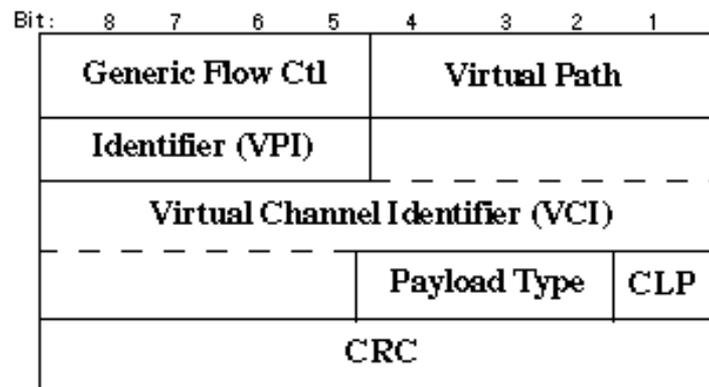


Fig. 4.3. Formato UNI

GENERIC FLOW CONTROL

Il campo GFC attualmente non è ancora utilizzato. Si prevede di adoperarlo per consentire all'UNI di controllare la quantità di traffico in ingresso nella rete ATM. Ha un significato locale, nel senso che il suo valore non è trasportato end-to-end perchè i suoi bits vengono riassegnati al VPI negli ATM switches.

SET-UP DI UNA CONNESSIONE

Un **ATM end-system** che desidera stabilire una connessione attraverso la rete deve inviare delle **cell di segnalazione** attraverso un ben noto canale virtuale (**VPI=0, VCI=5**). Questo canale è riservato al traffico di segnalazione e tutti gli switch sono preconfigurati a ricevere ed elaborare i pacchetti di segnalazione.

In generale all'interno di ogni VPI, tutti i VCI inferiori a 32 sono riservati al traffico di controllo.

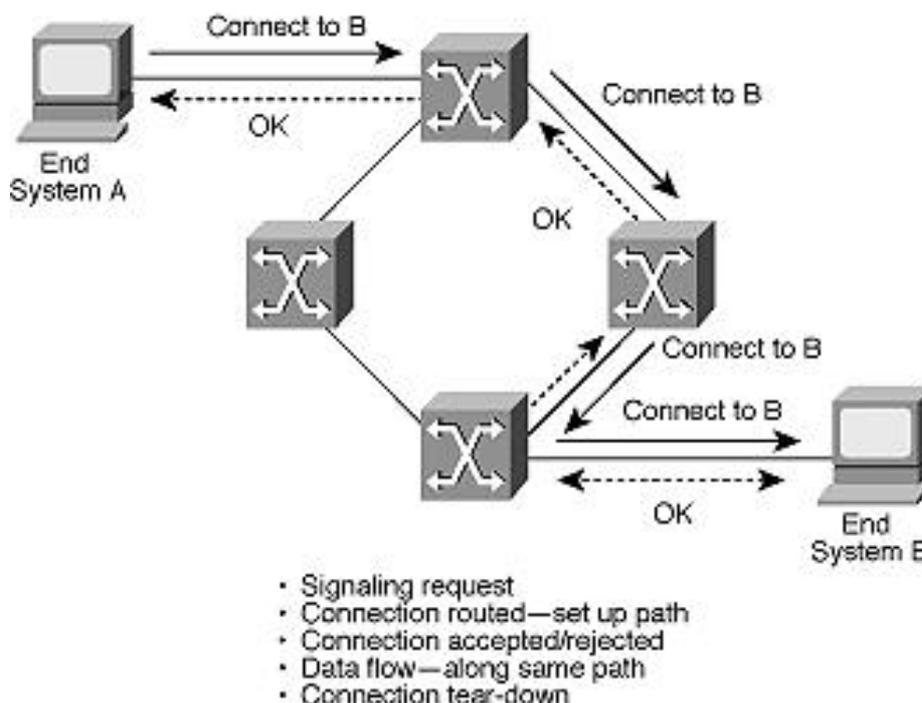


Fig. 4.4. ATM set-up connection

La segnalazione viene quindi trasportata di switch in switch, settando gli identificatori nelle **tabelle di routing**, fino al destinatario, che può o accettare e confermare la connessione o rifiutarla.

Gli identificatori di connessione VPI e VCI sono allocati per un particolare link dal nodo a cui la richiesta è spedita.

Questa connessione viene chiamata **SVC** (*Switched Virtual Connections*) ed è gestita automaticamente dal **protocollo di segnalazione**. Va fatta questa precisazione perchè esiste un secondo tipo di connessione chiamato **PVC** (*Permanent Virtual Connections*), in cui gli switch tra i due end-system vengono inizializzati da un intervento manuale di un operatore.

In una rete ATM sono possibili due tipi di connessioni (sia SVC che PVC):

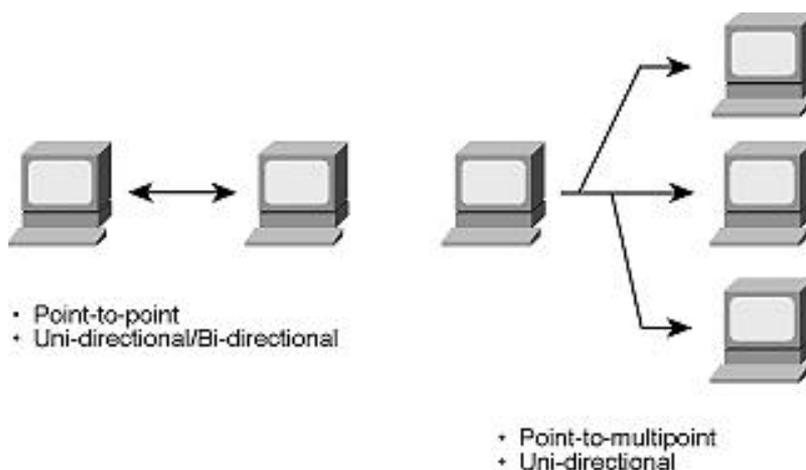


Fig. 4.5. Tipi di connessione

1. connessione **point to point**, che collega due end-system e può essere sia unidirezionale che bidirezionale
2. connessione **point to multipoint**, che collega un end-system (detto root node) a più destinatari (detti leaves) e può essere solo unidirezionale. La replicazione delle cells viene fatta in rete dagli switch.

PROTOCOLLO DI SET-UP PER UNA CONNESSIONE POINT TO POINT

Un esempio di **protocollo di set-up** per una connessione point to point in ATM è il **Q.93B** (specifica *UNI 3.0*), una versione semplificata del protocollo **Q.931** di ISDN.

Come si vede dalla figura 4.6 tre entità sono coinvolte: l'utente chiamante, quello ricevente e la rete.

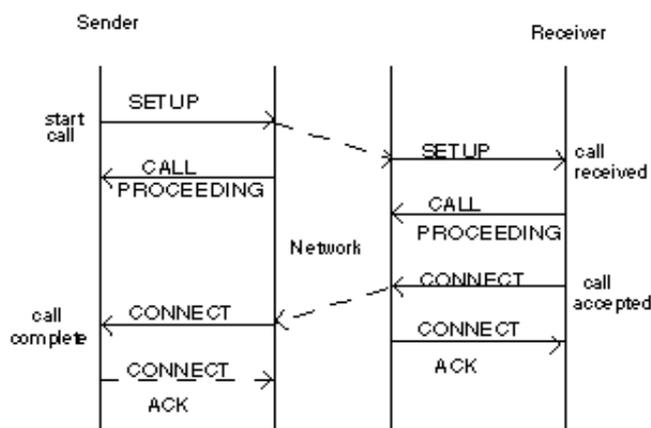


Fig. 4.6. Protocollo di connessione Q93.B

Al fine di stabilire la connessione sono utilizzati 4 messaggi, il cui significato dipende dal contesto di utilizzo. Il chiamante incomincia con un **messaggio di set-up**, contenente informazioni sul chiamante, sul servizio richiesto, sul tipo di AAL da usare e l'indirizzo del ricevente. La rete risponde con un **messaggio di call proceeding**, il cui scopo è quello di dare l'*acknowledgement* (ACK) al SET-UP e fornire il valore del VPI-VCI assegnato al chiamante. La rete notifica quindi il ricevente con un altro messaggio di SET-UP e questi restituisce un messaggio di acknowledgement alla rete mediante una ulteriore CALL PROCEEDING.

Il ricevente, quindi, in base alle informazioni del SET-UP, decide se accettare o meno la chiamata. In caso affermativo invia un **messaggio di connect**, a cui la rete risponde spedendo un *acknowledgement* al ricevente ed un messaggio CONNECT di conferma al chiamante.

Per garantire il set-up della connessione, i messaggi di SET-UP e CONNECT vengono ritrasmessi dopo un certo intervallo di tempo nel caso in cui l'*acknowledgement* non sia stato restituito.

Se avviene un errore nella negoziazione, il ricevente risponde con un messaggio di **release complete**, indicante il motivo del fallimento.

ROUTING

Quando una connessione è stabilita tra due host della rete, un **cammino virtuale** è definito dal nodo di partenza a quello di destinazione. La procedura che instaura la connessione inizializza le **tabelle di routing** interne agli switch. Quando una cell giunge ad uno switch, il suo campo VPI è usato come indice nella tabella di routing, in cui sono contenuti la porta di uscita su cui deve essere posta la cell e il nuovo valore per i campi VPI e VCI.

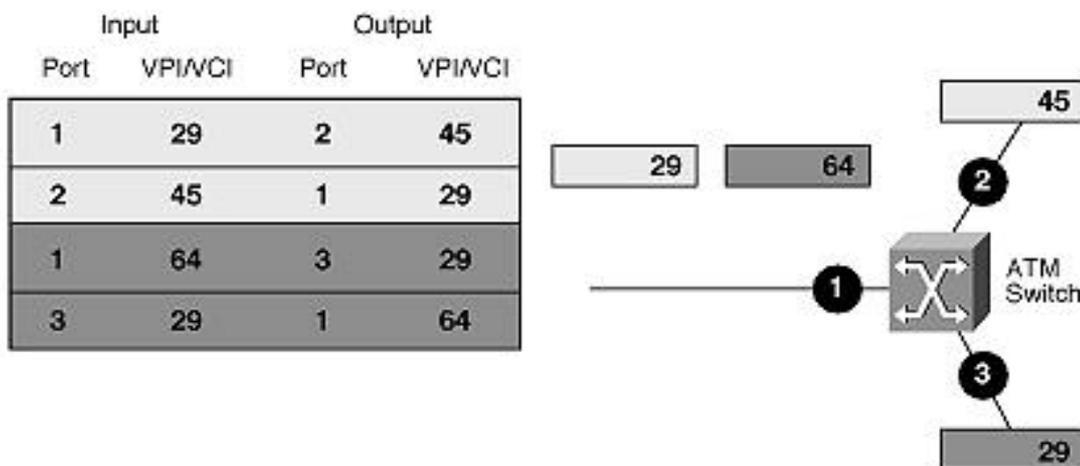


Fig. 4.7. Routing di una cell

La figura 4.7 ci mostra un esempio di funzionamento del routing. La cell con VPI pari a 29 giunge allo switch e dalla tabella ricava che la sua porta di uscita è la 2 e il nuovo VPI è 45. Naturalmente gli switch possono effettuare il routing non solo sul campo VPI ma anche su quello VCI.

ATM SWITCH

Una rete ATM è solitamente composta da fibre ottiche che interconnettono dei nodi di switch. Un esempio è la figura 4.8.

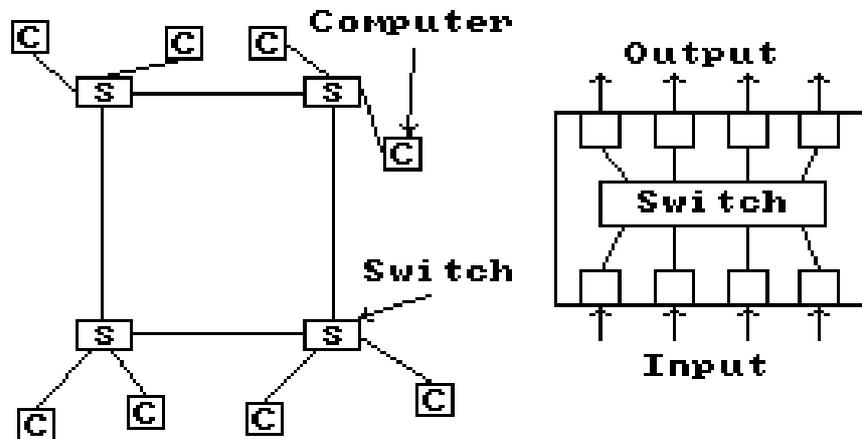


Fig. 4.8. ATM network

Come si vede, ogni switch può avere più porte di ingresso e di uscita (in questo caso quattro). Poiché più cell possono arrivare ad uno switch nello stesso momento, esso deve essere in grado di instradarle in parallelo.

Vi sono due requisiti minimi che deve soddisfare uno switch ATM :

1. deve ridurre al minimo la perdita di cell, anche in situazioni di congestionamento. Poiché una cell è solitamente parte di una struttura dati più grande, la perdita di una cell può causare la perdita di un intero pacchetto. Per questo motivo le perdite devono essere molto contenute, dell'ordine di 1 ogni 10^{12} ;
2. non deve compiere alcun riordino delle cell.

Quando una cell giunge ad uno switch i suoi campi VPI e VCI vengono esaminati e, in base alle informazioni memorizzate nelle tabelle di routing (inizializzate quando la connessione è stata stabilita), viene posta sulla giusta porta di uscita.

Uno switch deve essere in grado di gestire il caso in cui due cell su due differenti porte di ingresso vogliano usare la stessa porta di uscita. In questo caso si rende necessario un **buffer** dove immagazzinare le cell. Sono possibili due strategie:

1. la prima è quella di costruire delle **code di ingresso**. Questa strategia porta in se il rischio di bloccare in coda delle cell destinate a delle porte di output libere (problema dell' *head of line blocking*).
2. La seconda è invece quella di costruire delle **code di uscita**. Questo consente di eliminare l'*head of line blocking* e quindi di migliorare le prestazioni.

Una strategia usata per scegliere quale delle due dovrà partire per prima consiste, a parità di priorità, in una scelta casuale.

CONGESTIONE

All'interno di una rete ATM, le condizioni che portano alla congestione sono estremamente dinamiche e richiedono del hardware molto veloce capace di intervenire e riportare la rete ad una condizione sicura. *Il concetto sui cui si basa la gestione delle congestioni in ATM è quello della prevenzione: controllare velocemente la rete e anticipare la congestione prima di dovervi porre rimedio.*

Ciò può essere fatto controllando le code all'interno degli switches. Infatti un indice di probabile congestione è dato dalla perdita delle cell. Non appena una cell viene scaricata (le code si stanno riempiendo) una **control flow cell** parte per avvisare l'UNI, che sarà quindi in grado di limitare la velocità delle connessioni che stanno causando la congestione, diminuendo il numero di bit per secondo che queste possono trasmettere.

ATM ADAPTATION LAYER

FUNZIONALITÀ

Lo scopo di questo livello è impacchettare efficientemente i vari tipi di dati di più alto livello (come datagrammi, dati audio e video) in serie di cell che possono essere spedite su connessioni ATM e ricostruite nel formato appropriato al punto di arrivo. Date tutte le varie forme in cui possono apparire i dati digitali, quanti protocolli sono necessari per supportare tutti i tipi di dati che possono essere spediti su un network ATM? Sono possibili varie soluzioni:

1. non è richiesto nessun livello di adattamento: semplicemente ogni applicazione ha un proprio tipo di connessione ATM e quindi ha un proprio formato con cui spedire i dati. Il problema è che in questo modo applicazioni differenti che forniscono lo stesso servizio non sono in grado di interagire tra di loro.
2. avere soltanto un livello di adattamento capace di fornire tutti i servizi a tutti gli utenti.
3. sviluppare, all'interno del livello di adattamento, un set di protocolli che siano ottimizzati per i differenti tipi di applicazioni.(soluzione usata per ATM, sviluppata dal CCITT)

Le varie applicazioni sono state quindi divise, a seconda dei servizi di cui necessitano, in quattro classi:

1. con bit-rate costante: queste applicazioni mandano e ricevono dati con una velocità costante di bit. Richiedono che il ritardo dalla sorgente alla destinazione sia controllato e che il servizio sia orientato alla connessione (es.: telefoni e alcuni sistemi video)
2. con bit-rate variabile: queste applicazioni mandano i dati con velocità variabile di bit, ma richiedono ancora il controllo sul ritardo e sono orientati alla connessione. [Classe non standardizzata]
3. orientate alla connessione: queste sono le applicazioni che storicamente hanno usato i servizi network. (es.: X.25)
4. non orientati alla connessione: queste applicazioni sono quelle che usano correntemente i protocolli di datagram networking. (es.: TCP/IP o TP4/CLNP)

Class A	Class B	Class C	Class D	
Required		Not Required		Timing between source and destination
Constant	Variable			Bit Rate
Connection Oriented			Connection less	Connection Mode

Fig. 5.1. Classi di servizi

Il CCITT diede una numerazione ai protocolli che sviluppavano questi tipi di servizi. Così AAL 1 (ATM Adaptation Layer protocol 1) gestiva il servizio con bit-rate costante, AAL 2 gestiva il servizio con bit-rate variabile e così via.

In seguito, poichè i protocolli AAL 3 e AAL 4 sembravano molto simili, essi vennero fusi in un unico protocollo chiamato AAL 3/4. Più tardi però, ci si rese conto che AAL 3/4 non era implementato in maniera adeguata. Quindi, un nuovo protocollo di nome AAL 5, chiamato per la sua semplicità ed efficienza SEAL (Simple and Efficient Adaptation Layer), venne standardizzato e sostituito all'AAL 3/4. Il protocollo AAL 2 è qualcosa di ancora non ben chiaro e quindi non è stato sviluppato nessuno standard.

I PROTOCOLLI

Durante l'iniziale disegno degli AAL, si constatò che era conveniente dividere ogni AAL in due parti ben distinte: un sottolivello chiamato **segmentation and reassembly sublayer (SAR)**, responsabile della divisione in cell e il riassettaggio di esse in una unità più lunga (i pacchetti), e un sottolivello chiamato **convergence sublayer**, responsabile sia della gestione del flusso di dati da e per il sottolivello SAR, che della definizione di un formato unico per i pacchetti che devono essere spediti usando quell'AAL. Va precisato che gli header o i trailer aggiunti dai vari protocolli vengono posti nella parte dati dell'utente, cioè nei 48 byte di payload.

AAL 1

AAL 1 SAR

Come si vede dalla figura vi sono 47 byte dati e 1 byte di overhead (per un totale di 48 byte, ovvero la dimensione del payload della cell).

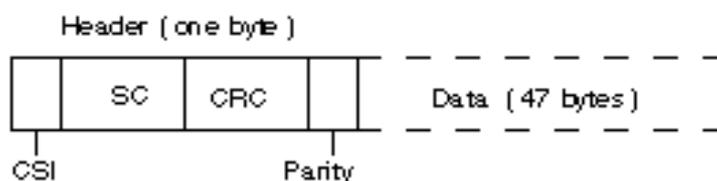


Fig. 5.2. AAL 1 SAR Format

Il byte di overhead viene così diviso : CSI (1 bit), SC (3 bit); CRC (3 bit), Parity (1 bit). Le prime due parti sono chiamate **SN** (*Sequence Number*) perchè usati per mantenere la sequenza, mentre le ultime due sono chiamate **SNP** (*Sequence Number Protection*).

Il bit **CSI** (*Convergence Sublayer Indicator*) serve per indicare i limiti dei blocchi di correzione dell'errore nel sottolivello di convergenza.

I 3 bit del campo **SC** (*Sequence Counter*) rappresentano un contatore (modulo 8) incrementato di uno per ogni cell e serve per verificare perdite o errori nella sequenza delle cell.

Il contenuto (3bit) del campo **CRC** viene calcolato sul SN usando il polinomio x^3+x+1 .

Il **parity bit**, infine, viene calcolato sul SN + CRC.

AAL 1 sarebbe stato migliore se avesse avuto un campo SN più grande e meno controlli di errori, dato che per la struttura propria di ATM gli errori sono rari mentre una serie di perdite di cell dovute al carico di traffico è più probabile.

AAL 1 Convergence Layer

La maggior parte degli **AAL1 Convergence Layer** proposti tenta di emulare ciò che correntemente i circuiti telefonici fanno internamente in un network telefonico, cioè dividere la banda di trasmissione in un numero di circuiti più piccoli a velocità costante di bit.

Un AAL 1 di particolare interesse è invece quello ideato per gestire la trasmissione video.

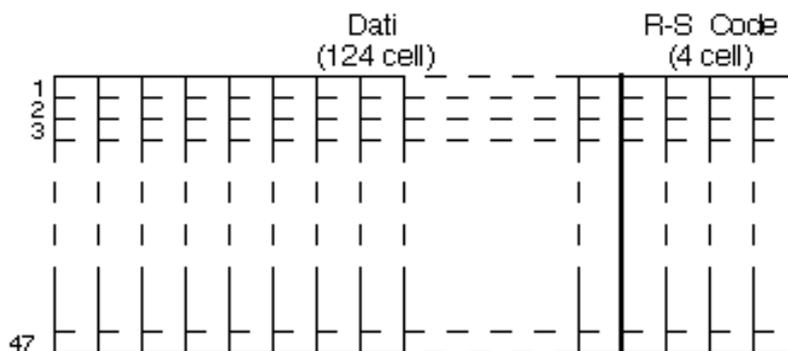


Fig. 5.3. AAL 1 Block for Video

Questo AAL 1 divide i dati video in blocchi di 128 cell, di cui le ultime quattro contengono un **Reed-Solomon code**, capace di correggere fino a 2 byte di errore e di recuperare una perdita massima di 4 byte tra i 124 byte sulla riga. Quindi poichè ci sono 47 righe in un blocco, si possono recuperare fino a 4 cell.

AAL 3/4

AAL 3/4 SAR

Come si vede dalla figura vi sono 44 byte di dati e 4 byte di overhead del SAR (per un totale di 48 byte, ovvero la dimensione del payload della cell).

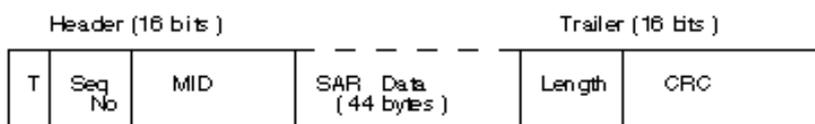


Fig. 5.4. AAL 3/4 SAR Format

I 4 byte di overhead vengono così diviso : T (2 bit), MID (4 bit), SN (10 bit); Length (6 bit), CRC (10 bit). I primi tre sono nella parte anteriore di ogni cell e compongono l'Header (2 byte), mentre gli ultimi due appartengono al Trailer (2 byte).

T: (*Type field*) serve per segnalare se la cell è all'inizio, in mezzo o alla fine del pacchetto o se contiene un intero pacchetto. I suoi valori sono :

- 10 Beginning of Message (BOM)
- 00 Continuation of Message (COM)
- 01 End of Message (EOM).
- 11 Single Segment Message (SSM)

MID: (*Multiplexing Identifier*) permette l'*interleaved multiplexing* di pacchetti diversi sulla stessa connessione ATM, vale a dire che un circuito virtuale viene utilizzato per trasportare pacchetti provenienti da applicazioni distinte. In questo modo si rende necessario dare ad ogni pacchetto un identificatore che lo distingue dagli altri, e questo identificatore dovrà poi essere posto in tutte le cell in cui verrà frammentato (dato che le cell non viaggeranno necessariamente una di seguito all'altra). Se non c'è multiplexing allora il MID è settato a zero. Il problema è che il multiplexing funziona meglio quando il numero di applicazioni è grande abbastanza da permettere un multiplexing statistico, ma solitamente il numero di applicazioni che una singola connessione ATM normalmente serve non è molto grande. Alcuni affermano addirittura che la maggiore potenzialità dell'ATM sia quella di ridurre il multiplexing in modo tale da avere per ogni flusso di dati una propria connessione ATM, rendendo in questo modo il MID ovviamente inconsistente.

SN: (*Sequence Number*), modulo 16, mantiene l'ordine di sequenza delle cell in un pacchetto; ogni pacchetto ha una sequenza propria, perciò SN può partire da un valore arbitrario all'inizio di ogni messaggio.

Length field: conta il numero di byte del campo dati. Esso vale sempre 44, tranne per le cell SSM o EOM; nelle cell EOM può essere settato a 63, indicando in questo modo che il pacchetto è stato abortito o deve essere eliminato)

CRC: capace di correggere un singolo errore di bit nella cell, calcolato sul polinomio $x^{10}+x^9+x^5+x^4+x+1$.

AAL 3/4 Convergence Layer

Definisce il formato per tutti i pacchetti che devono essere spediti usando AAL 3/4.

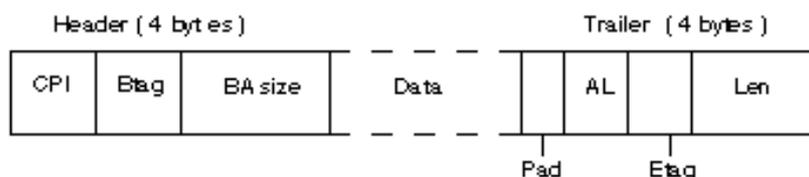


Fig. 5.5. AAL 3/4 Convergence Format

Come si vede dalla figura vengono usati 8 byte così ripartiti: CPI (8 bit), Btag (8 bit), BA size (16 bit); AL (8 bit), Etag (8 bit), Length (16 bit) + pad (3 byte). I primi tre fanno parte dell'header e gli ultimi tre fanno parte del trailer; sia l'header che il trailer devono essere allineati su 32 bit, così il trailer potrebbe essere preceduto da un massimo di 3 byte di padding.

CPI: (*Common Part Indicator*) settato a 0 perchè ancora da definire.

BAsize: (*Buffer Allocation size*) indica il massimo numero di byte che il ricevente deve riservare per conservare il pacchetto del convergence layer. Il problema è che normalmente molte interfacce allocano una dimensione massima per il buffer automaticamente e quindi il BAsize viene ignorato. Inoltre il BAsize è indubbiamente troppo piccolo per pacchetti, ad esempio di 64 KB. Btag e Etag: (Beginning e End tag), modulo 256, sono valori di confronto usati per permettere al ricevente di confermare che l'header e il trailer del convergence layer sono di fatto dello stesso pacchetto.

AL: (*Alignment field*) è un byte di zeri e serve per essere sicuri che il trailer sia di 32 bit.

Length field: è l'attuale numero di byte che sono spediti nel pacchetto (deve essere minore o uguale al BAsize); esso è probabilmente ridondante ma è un buon controllo per verificare se il SAR sta lavorando bene.

I problemi principali del AAL 3/4 riguardano le performance e l'overhead. Il primo è dovuto al fatto che processare una sequenza di piccole cell è sicuramente più costoso che esaminare un intero pacchetto, poichè si ammortizzerebbe meglio il suo costo fisso di overhead. Infatti, AAL 3/4 richiede che ogni cell sia esaminata al proprio arrivo (in particolare, per esaminare il T (segment Type field, nell'header del livello SAR) per controllare se essa è all'inizio, alla fine o in mezzo al pacchetto). Comunque, questo fatto può essere evitato, costruendo un'interfaccia che processi le cell e mandi un interrupt alla CPU solo una volta completato il pacchetto, il che però renderebbe le interfacce inevitabilmente più complesse. Il problema dell'overhead, invece, è molto più grave e significativo del precedente. Infatti, i quattro byte di overhead dell'AAL 3/4 SAR più i cinque byte del ATM header determinano il fatto che il 17 per cento (9 byte su 53 totali) di ogni cell è persa a causa dell'overhead di trasmissione. Inoltre, per pacchetti piccoli, gli otto byte di overhead del AAL 3/4 convergence aggiungono un'ulteriore penalità. Una causa per questo problema si può trovare nell'introduzione di alcuni campi il cui uso poteva essere evitato: il MID nel SAR è indubbiamente un errore; i Btag e Etag riflettono chiaramente un tentativo di nascondere delle pecche nel disegno delle procedure del SAR; infine, l'utilità del BAsize è molto limitata.

AAL 5

Tutte le cell del pacchetto sono composte da 48 byte di dati, tranne l'ultima in cui vi sono 40 byte di dati (+ Pad) e 8 byte di overhead del trailer. Gli obiettivi dei disegnatori di questo protocollo furono:

1. ottenere nella trasmissione, meno overhead di quanto se ne aveva nell'AAL 3/4
2. minimizzare il costo di gestione delle cell
3. se possibile, fare in modo che l'AAL abbia un comportamento il più possibile simile alle interfacce di comunicazione dati già esistenti, come quella di Ethernet o come la FDDI (Fiber Distributed Data Interface).

AAL 5 SAR

Il AAL 5 SAR overhead consiste in un unico bit posto nel header del ATM cell e che viene settato nell'ultima cell di ogni pacchetto.

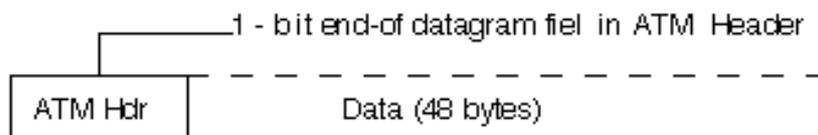


Fig. 5.6. AAL 5 SAR Format

Questo bit è posto nel payload-type field e se la cell è una user-data esso è usato per segnalare se la cell è l'ultima del pacchetto. L'AAL 5 ricevente deve solo mettere in coda le cell fino a quando non riceve una cell con il bit di fine pacchetto selezionato. In seguito controlla il CRC e la lunghezza e quindi il pacchetto viene passato allo strato superiore.

AAL 5 Convergence Layer

L'ultima cell del pacchetto contiene il trailer del AAL 5 convergence layer.

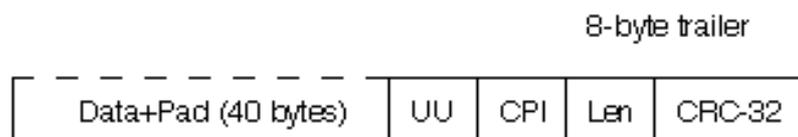


Fig. 5.7. AAL 5 Convergence Format

Il trailer di 8 byte è così ripartito: UU (1 byte), CPI (1 byte); Length (2 byte), CRC (4 byte) + Pad.

Pad: campo composto da una sequenza di zeri, utilizzato per riempire totalmente la parte dati.

UU (*User-to-User Indication*) e **CPI** (*Common Part Indicator*): sono correntemente non usati e settati a zero (addirittura l'AAL 5 in origine non li contemplava).

Length field: è semplicemente il numero di byte di dati nel pacchetto (non includendo il padding tra la fine dei dati e il trailer).

CRC: è un CRC di 32 bit sull'intero pacchetto del convergence layer, includendo il pad e il trailer, è estremamente robusto e può indicare la perdita di cell.

La proposta originale dell'AAL 5 conteneva solamente il bit nell'header e il CRC.

Documento prelevato in rete da **SANDRO PETRIZZELLI** all'indirizzo

http://www.dsi.unive.it/~franz/reti/so/atm/Home_Page_ATM.html