UNIVERSITA' DEGLI STUDI DI PAVIA

FACOLTA' DI INGEGNERIA

DIPARTIMENTO DI ELETTRONICA

EFFECT OF TRANSMISSION IMPAIRMENTS ON TCP WINDOW IN A WIRELESS DIFFSERV ENVIRONMENT

Relatore:
Chiar.mo Prof. Lorenzo Favalli

Correlatore:
Ing. Matteo Lanati

Tesi di Laurea
di Maria Pilar
Perez Quilez

Anno Accademico 2006/2007

Indice

Introduzione

Negli ultimi decenni la telefonia mobile e la trasmissione di dati tramite Internet hanno subito uno sviluppo e una diffusione enormi. I sistemi radiomobili hanno raggiunto livelli di utenza paragonabili, e talvolta superiori, a quelli della telefonia fissa. Internet è diventato uno dei mezzi più importanti per raccogliere e distribuire informazioni in tutto il mondo, offrendo numerose applicazioni di genere diverso.

Servizi quali la posta elettronica, lo scambio di file o anche il World Wide Web, che oggi sono utilizzati quotidianamente da milioni di persone in tutto il mondo sul mobile, richiedono un tipo di trasmissione di dati che sia affidabile dal punto di vista dell'integrità dei dati stessi ma che, entro dei limiti ragionevoli, non considera i tempi di trasferimento come un parametro critico per le applicazioni.

I principi base su cui sono costruite le reti a commutazione di pacchetto e le caratteristiche dell'insieme di protocolli implementati su di esse, tra cui i più noti TCP e IP, hanno garantito negli scorsi anni il successo e lo sviluppo di Internet. Ma la possibilità di scambiare enormi archivi di informazioni e di dialogare con il mondo ha portato la rete a decine di milioni di utenti e in consecuenzia la informazioni ha scambiato.

In conseguenza si sta lavorando allo sviluppo di nuovi sistemi, denominati di terza generazione, in grado di conciliare la mobilità dell'utente con la crescente esigenza di comunicazione multimediale. In

altri termini si cerca di far confluire in un'unica realtà la telefonia mobile e i servizi multimediali.

Sfortunatamente il servizio di trasporto offerto da Internet, noto come best effort, non è adatto ad applicazioni come quelle multimediali che necessitano di limitati tempi di transito e di regolarità nella consegna delle informazioni. Tali caratteristiche, ottenute riservando delle risorse a ciascun flusso di dati ed associandolo permanentemente per tutta la durata della comunicazione ad un percorso all'interno della rete, sono proprie delle reti a commutazione di circuito, come la rete telefonica, ma diametralmente opposte a quelle delle reti a pacchetto.

Per fronteggiare questo problema è in corso un processo di aggiornamento degli attuali sistemi radiomobili, quelli di seconda generazione come il GSM (Global System for Mobile Communications), finalizzato ad introdurre nuovi servizi, come l'accesso ad Internet, basati sulla tecnica a commutazione di pacchetto. Il GPRS (General Packet Radio System) ha la capacità di trasferire informazioni a pacchetto via etere tra la stazione radio base e il terminale mobile. i sistemi di terza generazione devono presentare forti elementi di continuità con gli attuali, al fine di garantire una migrazione graduale tra vecchio e nuovo e di riutilizzare quanto fino adesso è stato realizzato.

Il passaggio verso questi sistemi aprirà la strada a servizi a pacchetto con una capacità di comunicazione fino a dieci volte superiore rispetto a quella del GPRS, costituendo quindi il salto verso la comunicazione a larga banda nel mondo wireless.

Delle numerose proposte avanzate negli ultimi anni per la soluzione del problema della qualità del servizio in Internet, l'architettura di rete Differentiated Services è quella che risulta più promettente per le sue caratteristiche di semplice implementazione, elevata scalabilità e possibilità di poter coesistere con le infrastrutture di reti esistenti.

Questo lavoro di tesi ha come obiettivo studiare gli effetti del ritardo di coda e la rispondenza alle aspettative dell'architettura Differentiated

Services per garantire la qualità del servizio in una rete UMTS con core network completamente IP mediante simulazioni in ambiente NS2.

Dopo una breve introduzione alle reti a commutazione di pacchetto e al protocollo TCP/IP trattate nel Capitolo I si descrivono il protocollo UMTS e il protocollo radio RLC. Saranno poi presentati nel Capitolo IV le caratteristiche dell'architettura di rete Differentiated Sevices. Infine, dopo aver visto nel Capitolo V come realizzare un modello teorico di router DiffServ ed introdotto lo scenario scelto per questa caratterizzazione, nel Capitolo VI verranno presentati e discussi i risultati delle simulazioni effettuate con il simulatore di reti NS2.

Capitolo 1 Le reti di calcolatorie e il TCP/IP

1.1 Il modello di riferimento ISO/OSI

Il problema dell'interconnessione dei calcolatori allo scopo di condividere risorse o scambiare dati è nato poco dopo l'informatica stessa. Esso è stato stimolato dal progressivo decentramento delle risorse di calcolo, rispetto ai grossi e costosi macchinari della prima generazione, e dai molti ed evidenti vantaggi offerti dalla comunicazione e condivisione di informazioni, sia per le organizzazioni che per i singoli individui.

Verso la fine degli anni '70 il comitato ***ISO*** ha iniziato lo sviluppo dell'Open System Inter-connection (***OSI***). Questo modello di riferimento si pone l'obiettivo di fornire una struttura comune su cui sviluppare standard di riferimento per l'interconnessione di sistemi informatici, e di fornire un modello con cui confrontare le architetture di rete proprietarie.

Il modello OSI è, quindi, un formalismo astratto che non entra nel dettaglio della definizione di servizi o protocolli specifici di una architettura. Esso definisce una struttura in cui le funzioni logiche necessarie alla comunicazione tra sistemi sono suddivise in sette livelli funzionali (***layer***) ordinati gerarchicamente.

I principi di progetto seguiti durante lo sviluppo del modello furono i seguenti:

- Ogni livello deve avere una funzione ben definita, in modo da minimizzare il numero dei livelli e le funzioni svolte da ciascuno.

- Il passaggio delle informazioni avviene solo tra livelli adiacenti, attraverso opportune interfacce; ogni livello fornisce servizi al livello superiore utilizzando quelli resi disponibili dal livello inferiore.
- La comunicazione tra sistemi diversi avviene tra livelli paritari, attraverso un protocollo relativo a ciascun livello.

Con questa architettura, una volta definite le interfacce tra i livelli ed i servizi messi da loro a disposizione, si rende indipendente la definizione di un protocollo di un determinato livello dai dettagli di implementazione dei servizi messi a disposizione dal livello inferiore.

1.1.1.- I livelli OSI

Ognuno dei sette livelli della struttura gerarchica introdotta dal modello OSI, offre servizi più evoluti nella misura in cui si sale nella gerarchia. La trasmissione dei dati avviene verticalmente per tutti i livelli tranne per quello fisico, ma ciascuno si comporta come se la trasmissione fosse orizzontale, verso lo stesso livello di un host remoto. Lo standard OSI definisce i livelli come segue:

1. ***Livello Fisico***: definisce gli strumenti meccanici (connettori), elettrici (livelli di tensione), funzionali e procedurali per gestire l'operatività del canale fisico.
2. ***Livello Collegamento***: ha lo scopo di realizzare la trasmissione con un certo grado di affidabilità; riceve i pacchetti dal terzo livello e li trasmette sequenzialmente.
3. ***Livello Rete***: gestisce l'instradamento dei messaggi, determinando il percorso attraverso i nodi della rete per mezzo di tabelle di instradamento; fornisce una serie di servizi omogenei al livello trasporto, adattando le diversità dei servizi forniti al livello collegamento.
4. ***Livello Trasporto***: fornisce servizi di trasferimento trasparente dei dati alle entità di livello sessione. Si occupa di frammentare i dati in unità efficientemente gestibili dal livello sottostante, di effettuare la correzione degli errori e gestire le situazioni di congestione della rete. Questo è il primo livello (dal basso) ***end-to-end***, ovvero

che prescinde dalla topologia della rete e dall'esistenza dei sistemi intermedi nel trasferimento dei dati tra sorgente e destinazione.

5. ***Livello Sessione***: è responsabile del dialogo tra due programmi applicativi e del loro scambio di dati; esso consente di dotare le connessioni end-to-end di servizi più avanzati quali la gestione del dialogo, la gestione di token per l'accesso alle risorse condivise e la gestione della sincronizzazione della trasmissione.
6. ***Livello Presentazione***: si occupa di gestire la sintassi e la semantica delle informazioni che verranno trasmesse, codificando i dati secondo uno standard, in modo che calcolatori diversi possano comunicare.
7. ***Livello Applicazione***: interagisce direttamente con l'utente; rappresenta le utilità del sistema operativo e le applicazioni tramite le quali l'utente utilizza la rete.

7	**Applicazione**
6	**Presentazione**
5	**Sessione**
4	**Trasporto**
3	**Rete**
2	**Collegamento**
1	**Fisico**

Figura 1.1: I livelli ISO/OSI.

Il modello OSI ha il pregio di aver fornito per primo una distinzione chiara tra servizi, interfacce e protocolli, ed è dunque molto utile come strumento didattico per spiegare il funzionamento delle reti informatiche. Tuttavia, esso è stato usato pochissimo come guida all'implementazione di protocolli reali per diverse ragioni:

- Quando l'ISO mise a punto il modello i principali produttori di hardware e software stavano già diffondendo prodotti basati sull'architettura TCP/IP.
- Esso risulta molto complesso: la funzionalità di alcuni strati, come il quinto e il sesto, sono poco chiare e alcune funzioni, come il controllo di errore e di flusso, sono ripetute in diversi strati.

- È troppo orientato alle telecomunicazioni tradizionali, come la telefonia, e poco alla struttura hardware e software dei computer e delle reti di calcolatori; inoltre dà poca importanza ai protocolli senza connessione.

1.2 L'architettura di rete TCP/IP

L'architettura di rete TCP/IP si sviluppa secondo una struttura a livelli in analogia al modello ISO/OSI. I livelli sono quattro e sono individuati dai protocolli che li caratterizzano; essi sono riportati di seguito dal più basso (più vicino allo strato fisico) al più alto:

1. L'Host-to-Network
2. L'internet Protocol (IP)
3. I protocolli di trasporto (TCP e UDP)
4. I protocolli applicativi (FTP, Telnet, http, etc.)

Tutti gli standard relativi al TCP/IP e ad Internet vengono raccolti sotto forma di ***RFC*** (Request For Comments) che costituiscono i documenti ufficiali pubblicati dal Deputy Internet Architect ed accessibili in rete.

1.2.1 L'Host-to-Network

Nell'architettura TCP/IP i protocolli dei livelli fisico e di collegamento che consentono l'accesso alla rete di comunicazione non vengono specificati, a differenza di quanto avviene nel modello OSI. Questo perché essa si serve di quelli già disponibili ed aderenti agli standard.

Nell'ambito delle reti locali (LAN) il TCP/IP si appoggia principalmente sui protocolli definiti nello standard IEEE 802 (Ethernet, Token Ring, DQDB, etc.), e, nell'ambito delle reti geografiche (WAN) su PPP, SLIP, X25 ed ATM. La caratteristica del TCP/IP di essere indipendente da come la rete è fisicamente realizzata, raggiunge l'obiettivo di rendere interoperabili sulla rete elementi eterogenei per tecnologie e sistemi operativi, permettendo di utilizzare le infrastrutture di interconnessione già esistenti.

1.2.2 Il protocolo di rete IP

Il protocollo IP (Internet Protocol), con riferimento alla stratificazione del modello ISO/OSI, si colloca al livello tre, strato di rete, ed è un protocollo a datagramma, cioè senza connessione. L'obiettivo di IP, dunque, è quello di fornire allo strato di livello quattro (trasporto) un insieme di primitive omogenee con le quali trasferire le informazioni lungo i nodi della rete, fornendo un'astrazione dalla particolare tecnologia utilizzata dal secondo livello.

Come anticipato, infatti, IP è in grado di interfacciarsi sia con diverse tecnologie dello strato di collegamento (Ethernet, Token Ring, FDDI, etc.), utilizzando i servizi resi disponibili da queste, sia con differenti protocolli del medesimo strato (X25, Frame Relay, ATM, etc.) adattandone le funzionalità ai propri scopi.

Le funzioni svolte da IP sono essenzialmente le seguenti:

- L'astrazione dal particolare strato di collegamento.
- L'eventuale frammentazione e ricomposizione dei messaggi che eccedono le dimensioni consentite dal protocollo che gestisce il link.
- La rivelazione senza correzione degli errori.
- L'instradamento dei pacchetti lungo la rete.

Il passaggio dei dati tra diverse sottoreti, caratterizzate dall'impiego di tecnologie eterogenee nello strato di collegamento, avviene tramite i router. Essi, essendo dotati di porte di connessione molteplici (tante quante sono le sottoreti connesse), provvedono ad instradare i pacchetti da una rete all'altra utilizzando apposite tabelle tenute in memoria.

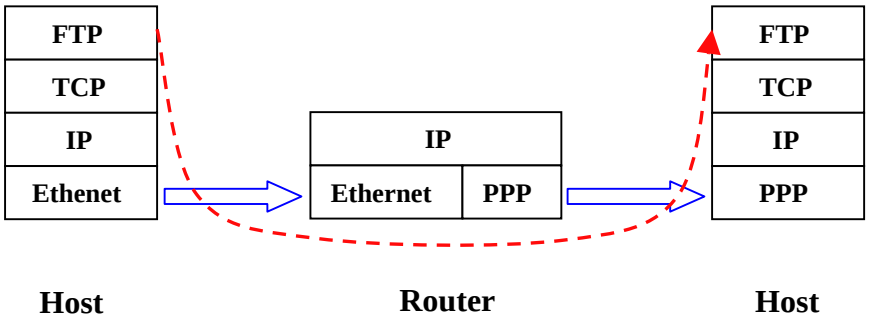


Figura 1.2: Collegamento tra applicazioni appartenenti a reti diverse.

La scelta di un protocollo di rete a commutazione di pacchetto senza connessione è stata motivata dall'esigenza imposta dal Dipartimento della Difesa di garantire efficienza nell'uso e tolleranza ai guasti. Il conseguimento di questi obiettivi fu reso possibile grazie all'impiego della multiplazione statistica dei pacchetti e della ridondanza dei percorsi di instradamento: in caso di caduta di un link, il protocollo IP è, infatti, in grado di calcolare automaticamente, ed in breve tempo, nuovi percorsi per raggiungere la destinazione non più accessibile tramite il link interrotto.

Gli indirizzi IP sono costituiti da stringhe di 32 bit univoche su tutta la rete Internet, solitamente rappresentati per mezzo della notazione "*dotted decimal*": si riportano, separati da punti, ciascuno dei quattro byte che costituiscono l'indirizzo (es. 196.168.3.24). I 32 bit dell'indirizzo IP sono il risultato dell'unione di un indirizzo di rete ed uno di host. L'organizzazione gerarchica originale partizionava lo spazio di indirizzamento in cinque classi distinte per numero di host per rete ed in base alla funzionalità.

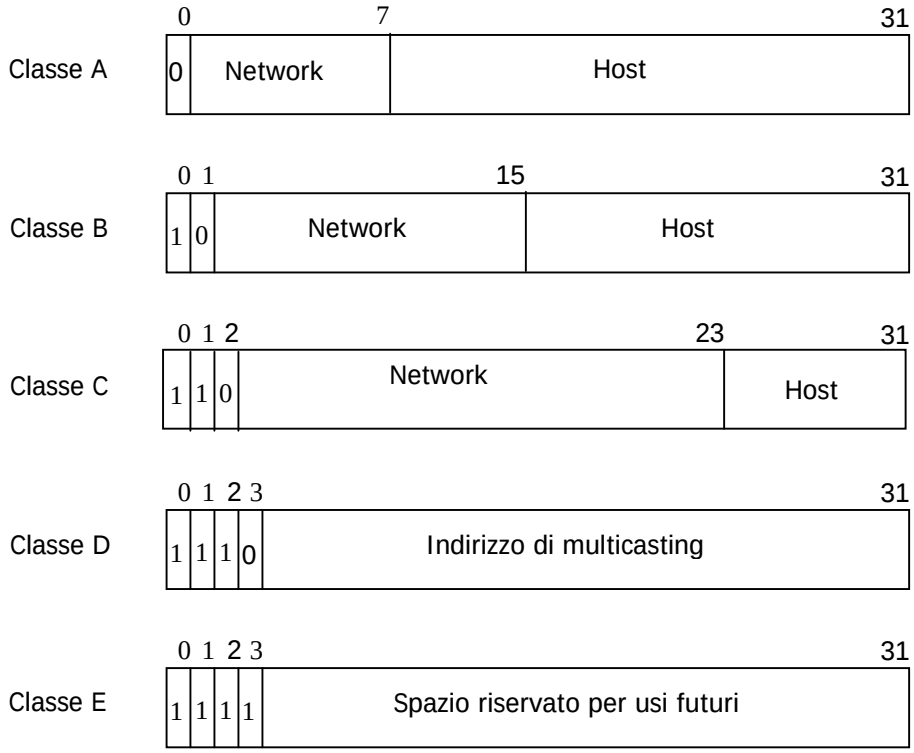


Figure 1.3: Classi di indirizzamento IP.

Successivamente, per sfruttare in modo più efficiente lo spazio di indirizzamento, è stata introdotta l'operazione di ***subnetting***; quest'ultima permette di utilizzare una parte di bit riservati agli host come indirizzo di sottorete.

L'operazione di subnetting utilizza un ulteriore insieme di 32 bit detto ***netmask*** che ha una sequenza di bit consecutivi posti ad 1 in corrispondenza dell'indirizzo di rete e di sottorete, ed i restanti bit a 0 per la parte di indirizzo che distingue l'host. Attraverso un'operazione di AND logico bit a bit degli indirizzi con la netmask è possibile verificare se due nodi appartengono alla stessa sottorete.

Per rendere più semplice agli utenti della rete l'identificazione dei nodi, viene utilizzato un sistema di denominazione simbolico che associa a ciascun indirizzo IP uno o più nomi simbolici. Le tabelle di associazione sono contenute in una base di dati distribuita, gestita da un insieme di nodi organizzati gerarchicamente detti ***DNS*** (Domain Name Server). È necessario sottolineare che gli indirizzi IP sono associati alle interfacce del nodo di rete e non al nodo stesso: dispositivi che possiedono più interfacce, come i router, avranno un indirizzo IP associato a ciascuna di esse. La struttura gerarchica, introdotta dal concetto di subnet nel TCP/IP, porta a distinguere tra due tipi di instradamento: uno interno alla

stessa sottorete ed un altro inter-rete tra sottoreti distinte.

Le tabelle di routing sono gestite al livello di IP e sono composte da elementi contenenti, per ciascun indirizzo remoto di destinazione, l'indirizzo del successivo nodo direttamente connesso; questo, a sua volta, conterrà una tabella con l'indirizzo per il prossimo salto, e così via fino a raggiungere la destinazione. Nelle tabelle di instradamento viene anche tenuta traccia, mediante l'uso di parametri, del costo di un percorso, in modo da poter sempre scegliere il percorso più breve o più veloce.

1.2.3 Il protocolo di servizio ICMP

L'***ICMP*** (***Internet Control Message Protocol***) è trasportato da IP come se fosse un protocollo di livello trasporto, ma è, in realtà, un protocollo di servizio che deve essere gestito da tutti gli elementi della rete. Esso svolge la funzionalità di notificare eventuali anomalie che si verificano durante l'instradamento dei pacchetti IP e di controllare lo stato della rete fornendo uno strumento di gestione della stessa.

Alcuni eventi che causano l'invio di un messaggio ICMP possono essere:

- Il non raggiungimento della destinazione finale da parte di un pacchetto.
- Il verificarsi di una congestione dovuta all'esaurimento del buffer di un router.
- L'esaurimento del ***TTL*** (time to live), ossia il tempo di vita di un pacchetto, che di solito è causa di un loop (ciclo) nel percorso di routing.

I messaggi ICMP, trasportati direttamente da IP, sono costituiti da un header IP in cui viene settato ad 1 un bit del campo "protocol", in modo da identificare che si tratta di un messaggio di servizio.

1.2.4 I protocolli di trasporto TCP e UDP

Il ***TCP*** (***Transmission Control Protocol***) è un protocollo di trasporto ***connection oriented*** (orientato alla connessione). Esso incrementa la qualità del servizio intrinsecamente inaffidabile del protocollo IP offrendo agli strati superiori un servizio di connessione affidabile con riscontro e controllo di flusso.

Il TCP oltre a fornire le funzionalità citate, realizza una multiplazione degli accessi alla rete, aggiungendo ai 32 bit, che individuano la ***NSAP*** (***Network Service Access Point***) di IP, 16 bit di indirizzo di "porta" coincidente con la ***TSAP*** (***Transport Service Access Point***) del modello OSI. Le applicazioni possono così accedere ai servizi dello strato di trasporto collegandosi al SAP identificato dall'insieme <indirizzo IP>:<porta>. Per molte applicazioni che utilizzano il TCP/IP sono riservati alcuni numeri di porta, chiamate ***well-known ports***: ad esempio al Telnet è associata la porta 23, all'FTP la 21, all'Http la 80.

I servizi forniti dal TCP sono:

- La messa a disposizione per gli strati superiori di primitive per l'instaurazione e l'abbattimento delle connessioni.
- La segmentazione del flusso dei dati in pacchetti per la trasmissione e la relativa ricomposizione a destinazione.
- La multiplazione di più connessioni in un unico flusso di pacchetti IP.
- Lo smistamento dei dati ricevuti verso le applicazioni corrispondenti, individuate dal numero di porta (operazione di demultiplexing).
- La garanzia di un servizio affidabile mediante riscontro dei pacchetti e controllo d'errore.
- Il controllo del flusso e della congestione.

Il servizio di riscontro, detto ***acknowledgement***, è basato su un meccanismo a finestra scorrevole, o ***sliding window***. Il nodo che trasmette invia il numero massimo di byte, determinato dalla dimensione della finestra, senza attendere la conferma dell'avvenuta ricezione di ogni singolo pacchetto: la finestra si contrae per ogni byte trasmesso e si espande per ogni byte riscontrato. Così viene massimizzato l'uso del canale

anche quando il tempo di attraversamento del collegamento, detto ***end-to-end delay***, assume valori rilevanti, come nel caso di connessione su reti geografiche (WAN).

Le finestre utilizzate dal TCP sono, in realtà, due: una è basata sulla capacità di ricezione del destinatario e consente il controllo del flusso, in modo da essere sicuri che un mittente veloce non sovraccarichi un ricevente più lento; l'altra, detta ***congestion window***, viene usata per il controllo della congestione.

L'***UDP*** (***User Data Protocol***) è il secondo protocollo di livello trasporto fornito dall'architettura TCP/IP. Esso mette a disposizione un servizio ***connectionless*** (senza connessione) di tipo ***best effort***, senza garanzie di consegna né di rispetto della sequenza. Il protocollo UDP si può considerare come una semplice interfaccia tra lo strato applicativo e il protocollo di rete IP. La funzionalità aggiuntiva è la presenza del numero di porta che consente, anche in questo caso, la multiplazione delle comunicazioni su un singolo nodo. L'UDP, come già specificato, non restituisce riscontri sull'avvenuta ricezione del datagramma né garantisce la sequenzialità di trasmissione in ricezione. Esso, nella sua semplicità, offre il vantaggio di una estrema leggerezza, e quindi velocità, rispetto al più complesso TCP: viene quindi usato in tutte quelle situazioni in cui bisogna gestire comunicazioni fortemente dipendenti dai vincoli temporali. Oltre a ciò, non dovendo gestire la ritrasmissione dei pacchetti perduti, è particolarmente adatto alle applicazioni in cui il dato ricevuto perde di valore se consegnato oltre un limite di tempo, come avviene per i flussi audio e video real-time: in questi casi è preferibile, infatti, scartare qualche pacchetto piuttosto che rallentare l'intera sequenza con il meccanismo di riscontro e ritrasmissione.

Capitolo 2 UMTS (Universal Mobile Telecommunications System) e Delay Link Control

Il termine UMTS (*Universal Mobile Telecommunications System*) fa riferimento all'insieme degli standard emessi secondo le specifiche 3GPP (*3rd Generation Partnership Project*) che sono in continuo aggiornamento.

Il sistema UMTS implementa in Europa le direttive ITU (*International Telecommunication Union*), riconosciute a livello mondiale, per i sistemi di terza generazione, denominate IMT-2000 (*International Mobile Telecommunication 2000*).

Questo sistema è stato progettato per rispondere a nuovi requisiti:

- frequenze di funzionamento uguali in tutto il mondo per permettere una totale interoperabilità;
- compatibilità con gli standard di nuova generazione;
- utilizzo di terminali estremamente piccoli e leggeri;
- integrazione con la rete ISDN e supporto di servizi anche di tipo video e interattivi;

- fornitura di una copertura globale integrando componenti terrestri e satellitari.

Data la varietà dei servizi offerti e la propensione alla copertura continua e totale, è necessaria la definizione di diversi livelli di copertura radio anche sovrapposti nella stessa area geografica e saranno richiesti meccanismi per consentire l'utilizzo di reti di altri operatori, in modo trasparente, sia all'interno della propria nazione sia in reti estere.

Altri requisiti fondamentali sono l'utilizzo più efficiente delle risorse radio attraverso tecniche di accesso innovative e appropriate soluzioni architetturali che assicurino il necessario adattamento dei sistemi mobili ai requisiti dei nuovi servizi emergenti.

2.1 – La banda del sistema UMTS

La WARC (*World Administrative Radio Conference*), responsabile dell'assegnazione delle frequenze radio su scala mondiale, ha assegnato nel 1992 ai sistemi di terza generazione una banda compresa tra 1885-2025 MHz e 2110-2200 MHz.

In Europa la componente terrestre dei sistemi di terza generazione hanno a disposizione 155 MHz, le bande appaiate 1980-2010 MHz e 2170-2200 MHz sono assegnate alla componente satellitare, mentre i primi 15 MHz della banda coincidono con parte della banda attualmente impiegata dal DECT (*Digital Enhanced Cordless Telecommunications*).

La restante porzione di spettro relativa al segmento terrestre è stata suddivisa in parte "appaiata" e non. La prima è costituita da 60+60 MHz da 1920 a 1980 Mhz in uplink e da 2110 a 2170 MHz in downlink, la seconda da 35 MHz da 1900 a 1920 MHz e da 2010 a 2025 Mhz, dove non c'è alcuna distinzione a priori tra le porzioni assegnate alle varie tratte.

Recentemente queste bande sono state rese disponibili per licenze UMTS, in modo da permettere agli operatori il dispiegamento delle reti. La larghezza di banda nominale del singolo canale è di 5 MHz. Le motivazioni che hanno portato a questa scelta sono varie. In primo luogo, le velocità di 144 e 384 Kbit/s sono ottenibili senza sacrificare eccessivamente la capacità del sistema, riuscendo ad ottenere, sotto certe condizioni, anche picchi di 2 Mbit/s. Secondo, la mancanza di disponibilità di spettro richiede l'allocazione di uno spettro ragionevolmente piccolo, specialmente se il sistema è stato

dispiegato in bande già occupate da sistemi di seconda generazione. Terzo, la larghezza di banda di 5 MHz può risolvere meglio il problema dei cammini multipli rispetto a bande più limitate, facendo incrementare le prestazioni del sistema.

2.2 – Acceso radio del sistema UMTS

Le principali caratteristiche dell'interfaccia radio dell'UMTS, definita UTRAN (UMTS *Terrestrial Radio Access Network*), sono riassunte nella seguente tabella.

	UTRA-FDD	UTRA-TDD
Tecniche di accesso	W-CDMA	TD-CDMA
Chip rate	3.84 Mchip/sec	
Canalizzazione	5 MHz	
Durata di trame	10 msec	
Numero di slot per trama	15	
Modulazione	Down-link: QPSK Up-link: B-PSK	QPSK
Ricezione	Coerente	
Velocità di trasmissione dell'informazione	Variabile (ogni 10 sec) Velocità diverse possono essere ottenute variando lo spreading factor, oppure assegnando più codici al segnale da trasmettere oppure affasciando più time slot (somo caso TDD)	

Tabella 2.1: Caratteristiche dell'interfaccia UMTS

Come richiesto dalla normativa ETSI, al fine di facilitare la realizzazione di terminali *dual mode FDD/TDD*, i parametri delle due componenti sono stati armonizzati il più possibile. Di conseguenza, le specifiche per le due componenti risultano essere spesso costituite dagli stessi elementi.

Il livello fisico di UTRAN offre dei servizi ai livelli superiori, trasmettendo sulla portante fisica informazioni generate a partire dal livello due della pila OSI.

In particolare, i canali di trasporto sono i servizi offerti dal livello fisico ai livelli superiori. I canali di trasporto sono definiti in base al tipo di informazione trasferita e al modo di trasferirla sull'interfaccia radio. Essi possono essere raggruppati in due tipologie: canali comuni, dove l'informazione è trasmessa indistintamente a tutti i terminali mobili, e

canali dedicati, dove la comunicazione avviene verso un singolo terminale associando un canale fisico, cioè un codice e una frequenza nella modalità FDD, ed un *time slot* nel caso TDD.

Il numero di canali di trasporto è molto maggiore di quanto specificato per il GSM. Questo è dovuto al fatto che il sistema UMTS non è ottimizzato solamente per servizi vocali, ma deve essere in grado di offrire simultaneamente servizi con caratteristiche di qualità molto diverse.

2.3 – Architettura del sistema UMTS

L'architettura complessiva del sistema UMTS può essere divisa in due segmenti principali: la rete di accesso (*UMTS Terrestrial Radio Access Network- UTRAN*) e l'infrastruttura di commutazione e routing (*Core Network*).

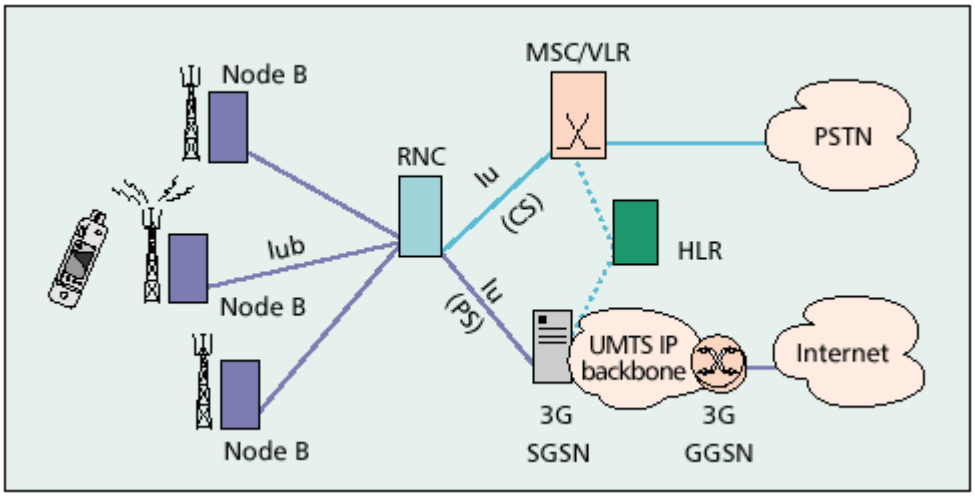


Figura 2.2: Architettura del sistema UMTS

L'architettura proposta da ETSI è strutturata su tre livelli:

- livello fisico: canali fisici, mappati dai canali logici o *stand alone*.
- livello di data link: è costituito dai canali logici o di trasporto.
- livello di rete.

L'UMTS, come gli altri sistemi radiomobili cellulari, è spesso identificato con le sue caratteristiche radio, dimenticando che, oltre alla tratta in etere, vi è una complessa e

vasta rete che costituisce la parte più estesa del trasporto delle informazioni, voce o dati, proveniente dal terminale mobile.

La *Core Network* è la componente di questa rete che mette in comunicazione le varie sezioni della cosiddetta *Rete di Accesso*, la quale raccoglie invece direttamente il traffico proveniente dalle diverse stazioni radio.

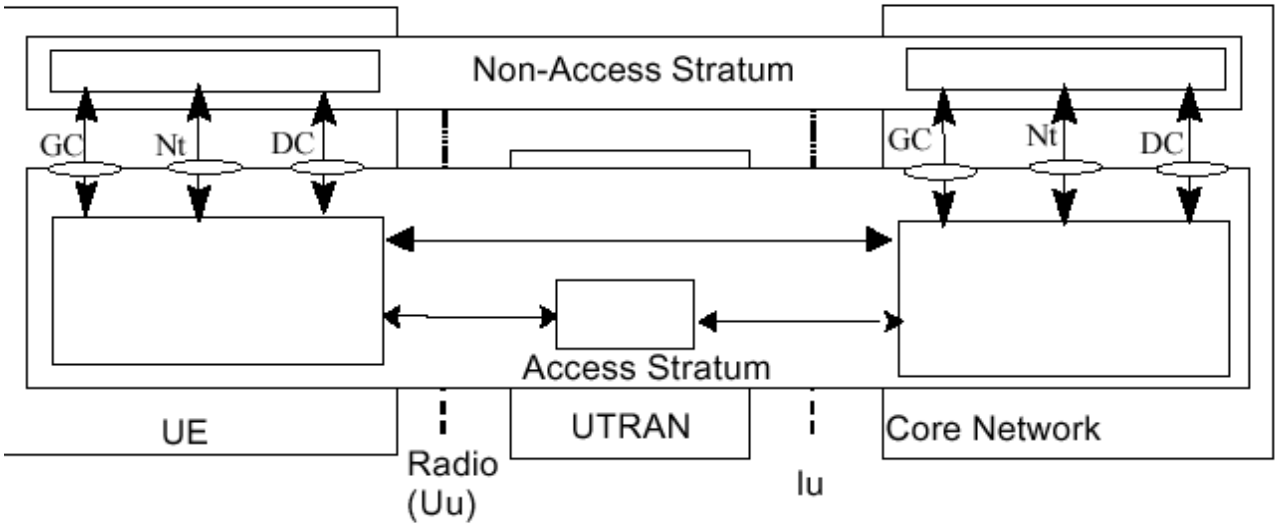


Figura 2.3: Architettura funcionale ad alto livello UMTS

Come detto in precedenza, la rete di accesso del sistema UMTS è denominata UTRAN (*UMTS Terrestrial Radio Access Network*).

L'architettura ad alto livello del sistema UMTS è in funzione delle seguenti unità logiche: UE (*User Equipment*, ossia il terminale mobile), UTRAN e CN. Nell'UMTS si è cercato di introdurre un certo grado di indipendenza dell'interfaccia radio dalle altre parti del sistema. Questa autonomia, studiata negli ambiti della ricerca, è stata parzialmente realizzata mediante la separazione logica tra *Access Stratum* e *Non Access Stratum* del sistema UMTS. Per *Access Stratum* si intende l'insieme dei protocolli e delle funzionalità maggiormente legati all'interfaccia radio. In modo complementare, si indicano quelli indipendenti dall'accesso radio con il termine *Non Access Stratum*.

Funzionalità caratteristiche del *Non Acces Stratum* sono il controllo di chiamata e di sessione, cioè le procedure per l'instaurazione, la modifica e il rilascio delle risorse logiche trasmissive in relazione al servizio richiesto, e il controllo di mobilità, cioè tutte quelle

procedure che permettono all'utente di comunicare, indipendentemente dalla sua posizione e dal fatto che sia in movimento. La funzionalità appena descritta si riferisce alla mobilità tra le diverse aree della rete d'acceso, e come tale è quindi gestita dalla *Core Network*. Al contrario, la mobilità interna a un'area di accesso viene gestita in modo indipendente all'interno dell'*Access Stratum*. Di fatto le funzionalità dell'*Access Stratum* corrispondono all'insieme delle funzionalità implementate nell'UTRAN.

I servizi dati e multimediali sono un requisito fondamentale per lo sviluppo di un sistema di terza generazione quale l'UMTS. Infatti, un'importante caratteristica dell'UTRAN risiede nella possibilità di gestire contemporaneamente servizi diversi. Tale possibilità è tecnicamente supportata da una serie di caratteristiche a livello di protocolli radio:

- Variable bit rate su canali di trasporto dedicati: questa caratteristica è particolarmente utile quando il terminale deve fornire un insieme di servizi differenti. La possibilità di fornire su un canale fisico dedicato bit rate variabili consente un ottimizzato uso delle risorse anche in questo caso.
- Multiplexing di differenti canali logici sullo stesso canale di trasporto dedicato.
- Multiplexing di differenti canali di trasporto dedicati sullo stesso canale fisico.
- Canale comune in uplink il cui utilizzo può offrire un supporto particolarmente efficiente per la fornitura di servizi dati.
- Canale in downlink condiviso da più utenti particolarmente adatto per applicazioni Internet.

2.3.1 Architettura della rete d'accesso

La rete di accesso è delimitata da due interfacce: da un lato l'interfaccia radio denominata Uu delimita l'UTRAN verso il terminale mobile; dall'altro l'interfaccia Iu connette l'UTRAN alla *Core Network*. Quest'ultima in realtà è un'interfaccia che ha doppia valenza, in quanto integra sia l'interfaccia che collega l'UTRAN alla CN a circuito (CS-*Circuit Service*) sia l'interfaccia che collega l'UTRAN alla CN a pacchetto (*PS-Packet Service*).

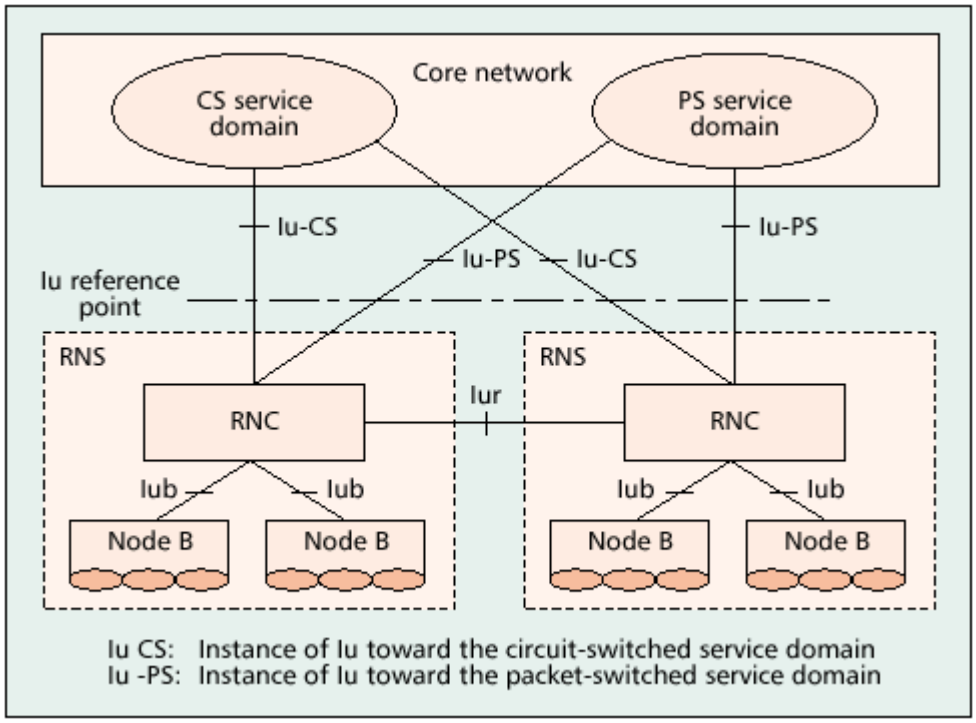


Figura 2.4: Componente e Interfacce dell'UTRAN

L'UTRAN è costituita da un insieme di *Radio Network Subsistem* (RNS) connessi alla CN tramite l'interfaccia I_u. Un RNS è composto da un controllore (*RNC-Radio Network Controller*) e da una o più entità chiamate Node B, connesse all'RNC tramite l'interfaccia I_{ub}. Un Node B sovrintende a un gruppo di celle che possono essere FDD, TDD o miste.

All'interno dell'UTRAN, RNC differenti possono essere collegati tra loro tramite l'interfaccia I_{ur}.

Lo RNC rappresenta il confine tra il mondo radio e il resto della rete, infatti lì si chiudono i protocolli radio che si sono aperti nel terminale per la gestione della tratta in etere. Al di sopra, si trovano i protocolli che permettono l'interconnessione con la *Core Network* e che da esso dipendono. La tratta via etere è limitata al tratto terminale-stazione radio base; in tale ottica i protocolli radio sono definiti tali, non perché i relativi messaggi transitano sulla tratta radio, ma perché relativi alla gestione della tratta radio.

Oltre ad offrire un dimensionamento scalabile dello RNS, questa architettura offre, fra l'altro, una rilevante capacità di gestire la mobilità a livello dell'UTRAN.

Sia il Node B sia lo RNC sono in grado di gestire l'*handover* e la marcodiversità.

L'*handover* è una funzionalità propria dei sistemi radiomobili che consente il mantenimento della connessione radio di un utente che si sposta da una cella all'altra. Per

macrodiversità si intende la capacità di mantenere la connessione in corso fra il terminale mobile e la rete attraverso più di una stazione base. Tale funzionalità è molto importante nei sistemi CDMA.

Handover e macrodiversità possono essere gestiti a livello di Node B, nel caso di celle appartenenti allo stesso Node B, oppure possono essere gestiti a livello di RNC mediante l'impiego dell'interfaccia I_{ub}, nel caso di celle controllate da Node B diversi, ma controllate dallo stesso RNC, o della I_{ur}, nel caso di celle appartenenti a RNS diversi.

Fra RNS diversi l'*handover* può anche essere effettuato tramite la CN, utilizzando l'interfaccia I_u. In questo caso non può esservi macrodiversità perché quest'ultima è realizzata con i protocolli radio che sono limitati all'RNC.

In effetti, le reali motivazioni per l'esistenza dell'interfaccia I_{ur} sono collegate alla gestione della mobilità all'interno dell'UTRAN. Ogni RNS, per quanto di rilevanti dimensioni in termini di territorio e di numero di utenti serviti, non può soddisfare tutte le esigenze della mobilità. La I_{ur} permette da un lato la mobilità continua, con transizioni tra RNS non percepibili dagli utenti, grazie alla macrodiversità, e dall'altro di alleggerire le procedure relative alla CN, limitandone l'intervento in casi in cui tale interfaccia non sia presente.

Un'altra caratteristica peculiare dell'UTRAN è la scelta dei protocolli di trasporto sulle interfacce I_u, I_{ub} e I_{ur}. I protocolli si basano essenzialmente su ATM (*Asynchcronous Trasfer Mode*). La scelta del protocollo ATM è da imputarsi principalmente alla necessità di avere un meccanismo flessibile e potenzialmente adattabile alle diverse combinazioni di traffico multimediale. Quando si vuole una tecnologia di trasporto che emuli i circuiti di trasmissione della voce e che consenta, contemporaneamente, la trasmissione dei dati, la scelta di ATM è obbligatoria.

ATM è una delle tecnologie maggiormente utilizzate nelle reti IP e inoltre, grazie a una serie di protocolli e prestazioni standardizzate per l'indirizzamento e l'instradamento delle chiamate, è in grado di incorporare le funzionalità di *networking* e trasporto delle segnalazione tipica delle reti telefoniche, a differenza degli altri protocolli di trasporto per IP.

ATM è stato progettato per lavorare in ambienti multiprotocollo. Per consentire a questa tecnologia di supportare servizi estremamente diversi, sono stati definiti ATM Adaption Layer (ALL) con caratteristiche molto diverse.

Nel sistema UMTS, per adattare il flusso delle informazioni alle sue caratteristiche

vengono usati: l'ATM Adaption Layer di tipo 2 (ALL2) per il trasporto dei protocolli radio (I_{ub} e I_{ur}) e dei flussi di utente verso il *Circuit Service* (I_u), e IP su AAL5 per i flussi utenti verso il *Packet Service* (I_u).

Si deve tenere conto delle peculiarità delle componenti del traffico che, come la voce, sono caratterizzate da basso *bit rate* e da requisiti di *real time*.

Infatti se si considera la voce compressa usata nei sistemi radiomobili, ipotizzando un bit rate di 8 kbit/s e la durata di trama di 10 ms, si ottiene facilmente che ogni 10 ms è necessario inviare un pacchetto vocale di 80 bit. Nel caso di trasporto su ATM con AAL tradizionali, si verifica un evidente spreco: ATM ha una cella con un payload di 48 byte (384 bit), quindi solo poco più di un quinto della capacità viene sfruttata. ALL2 è in grado di multipare traffico di utenti diversi sullo stesso flusso di celle, sia ponendo pacchetti di utenti diversi nella stessa cella, sia suddividendo un pacchetto di utente su due celle.

2.3.2 Macrodiversità e soft handover

In un sistema radiomobile cellulare con tecnica di accesso CDMA, tutte le celle operano sulla stessa portante, almeno quelle allo stesso livello gerarchico. Questo fa si che il segnale trasmesso da una stazione radio base possa essere ricevuto da chiunque conosca i codici di *spreading* utilizzati e si trovi nella zona geografica in cui i segnali ricevuti siano superiori a delle soglie minime di potenza. E' naturale introdurre in questi tipi di sistema il concetto di *macrodiversità*. Allo scopo di migliorare la qualità della comunicazione, il mobile non si limita a rimanere collegato ad una sola stazione radio base, ma coinvolge nella chiamata tutte le stazioni da cui riceve un segnale di riferimento sufficientemente buono. Viene così definito il concetto di *Active Set* (AS) come l'insieme di stazioni radio base dalle quali il mobile riceve la stessa informazione utile. Il processo di attivazione e rilascio dei *link* paralleli è svolto in funzione della qualità del segnale ricevuto dal mobile lungo i suoi movimenti nella rete.

Quando il terminale mobile opera in macrodiversità, può migliorare la qualità della comunicazione sul downlink combinando trama per trama i segnali di contenuto energetico più elevato. Tali segnali possono provenire da celle appartenenti a Node B e RNC diversi.

Sulla tratta di salita, il segnale trasmesso dal mobile viene decodificato da tutte le

stazioni appartenenti all'*Active Set*. Il segnale vieni ricombinato al livello gerarchico superiore, permettendo così di ottenere un sostanziale miglioramento della qualità.

Da un altro punto di vista, l'impiego in rete della macrodiversità consente un aumento della capacità del sistema. Infatti, la macrodiversità può essere anche utilizzata per mantenere il livello di qualità voluto a fronte di una diminuzione del valore di potenza trasmessa dal mobile. In tal modo il sistema opera a livelli di potenza inferiori, permettendo un conseguente aumento della capacità del sistema.

Trovarsi in uno stato di macrodiversità rende particolarmente facile il processo di *handover*. Per questo, nel sistema UMTS si parla di *soft handover*.

Il processo di *soft handover* è costituito da un insieme di funzioni: l'effettuazione delle misure, la loro interpretazione, la trasmissione dei dati misurati ed interpretati all'entità di rete incaricata di controllare il processo, l'algoritmo di soft handover e l'esecuzione dell'handover stesso.

2.3.3 Gestione delle risorse radio

La gestione delle risorse radio costituisce l'insieme di procedure e algoritmi per un'efficiente gestione dei livelli di *link* radio.

Ci sono sostanziali differenze tra la gestione delle risorse da parte di un sistema basato sul CDMA e uno basato sul TDMA/FDMA, evidenti in una situazione di sovraccarico di una cella in sistemi UMTS e GSM.

In tale situazione, sarebbe opportuno cercare di spostare parte del traffico nelle celle adiacenti.

Può essere realizzato nel GSM semplicemente forzando alcuni utenti ad utilizzare risorse di una stazione radio base differente dalla best server, cioè la stazione il cui segnale è ricevuto dal mobile con potenza maggiore rispetto alle altre stazioni.

Globalmente, il traffico risulterà distribuito più omogeneamente sulle celle del sistema, con ovvi vantaggi in termini di traffico smaltito complessivamente. Per contro, i soli mobili forzati a collegarsi con una stazione alternativa potranno sperimentare una qualità inferiore e potranno interferire maggiormente con quei mobili, e solo con quelli,

che utilizzano le stesse risorse radio, in termini di frequenza e *time slot*, in altre celle del sistema. In particolare, se un utente richiede di instaurare una chiamata in una cella congestionata, il sistema potrà cercare di gestire la richiesta utilizzando le risorse di una cella adiacente.

La stessa operazione è molto più delicata in un sistema CDMA, dove l'obiettivo principale è minimizzare la potenza complessivamente trasmessa dai mobili, ovvero attestare ogni utente alla stazione radio base che permette di utilizzare la minore potenza.

Quest'ultima non è necessariamente la cella il cui segnale è ricevuto dal mobile con maggior potenza. La potenza utilizzata dal mobile è funzione sia dell'attenuazione verso la stazione radio base, sia del livello di carico di quest'ultima. Pertanto è necessario attestare il mobile nella stazione radio base *best choise*, che permette di minimizzare la somma, in unità logaritmiche, dell'attenuazione di tratta e del livello di interferenza.

Forzare un utente ad operare su una cella diversa dalla best choise, significa forzarlo a trasmettere con potenza maggiore e questo peggiorerebbe la qualità di tutti gli utenti. La potenza del mobile forzato su un'altra cella costituirebbe una forte interferenza per la best choise e innalzerebbe ulteriormente il suo livello di interferenza, già elevato a causa del forte carico.

Nel sistema UMTS è presente una precisa politica di *Admission Control*, che prevede di limitare l'accesso di nuove chiamate alla rete anche in presenza di risorse radio disponibili. Tali politiche sono caratteristiche delle tecniche di accesso che, come il CDMA, prevedono una soft capacity, ossia che vedrebbero via peggiorare la qualità con il crescere del numero di chiamate. E' quindi l'operatore che stabilisce quale deve essere il carico massimo e di conseguenza vieta l'accesso ad ulteriori chiamate per mantenere la qualità a livelli accettabili.

2.3.4 Codifica per la correzione degli errori

Per quanto riguarda la codifica di canale, UMTS prevede quattro alternative e la scelta di quale utilizzare dipende dalla qualità richiesta dal servizio e dalle caratteristiche del canale:

- codifica convoluzionale con rapporto 1/3, per BER dell'ordine di 10^{-3}, ed un'eventuale punturatura, con rapporto finale ½, per BER più elevati;

- codifica Reed-Solomon + interleaving + codifica convoluzionale per BER dell'ordine di 10^{-6};
- turbo codici con rapporti di codifica 1/3;
- codifica per servizi particolari con BER diversi;

Per dati a pacchetto che non richiedano la consegna in tempo reale, esiste la possibilità di ritrasmissioni e protocolli ARQ (Automatic Repeat Request).

2.4 Struttura cellulare gerarchica

Un sistema di terza generazione deve essere in grado di supportare una grande varietà di servizi in ambienti radio differenti. Per i differenti requisiti sono necessarie diversi tipi di cella: celle di grandi dimensioni garantiscono una copertura continua, mentre celle piccole sono necessarie per raggiungere un'alta efficienza spettrale un'alta capacità. Inoltre, le diverse modalità cellulari devono poter coesistere all'interno del sistema.

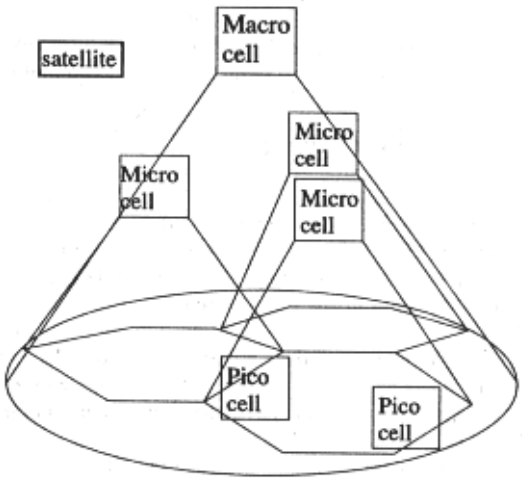


Fig. 2.5 – Struttura cellulare gerarchica

La struttura cellulare gerarchica HCS (*Hierarchical Cell Structure*) descrive un sistema in cui operano sovrapposte due tipi differenti di celle.

Le microcelle hanno tipicamente un raggio di poche centinaia di metri e sono servite da stazioni radio base a bassa potenza, con antenne posizionate a pochi metri dal suolo.

Le macrocelle coprono un area di 1 km di raggio o più. Solitamente sono poste in

aree rurali per fornire una copertura continua di aree coperte da microcelle, in modo da servire gli utenti che si spostano rapidamente.

Le macrocelle coprono un area di 1 km di raggio o più. Solitamente sono poste in aree rurali per fornire una copertura continua di aree coperte da microcelle, in modo da servire gli utenti che si spostano rapidamente.

Le picocelle coprono aree interne agli edifici: una stazione radio a bassa potenza copre un ufficio, un piano o un palazzo.

Le celle satellitari introducono una copertura continua globale, anche in zone dove i sistemi terrestri non possono essere installati.

Il principio da seguire ogni volta sia possibile è quello di dirigere il traffico nella cella più piccola disponibile, in modo da migliorare l'efficienza spettrale del sistema.

Sono stati previsti tre gradi di velocità di trasmissione sostenibili, diversificati in base alla dimensione della cella in cui avviene la comunicazione. I livelli di bit rate massimo ottenibile sono:

- 144 kbit/s nelle macrocelle;
- 384 kbit/s nelle microcelle;
- 2 Mbit/s nelle picocelle.

Queste velocità sono il massimo raggiungibile in un determinato ambiente di servizio, mentre il valore effettivo di trasmissione sarà influenzato anche dalle richieste degli altri utenti.

Si possono individuare due tipologie distinte di *strutture HCS*:

- celle stratificate che condividono le stesse frequenze;
- celle stratificate che utilizzano bande separate.

Nel primo caso, gli utenti che operano su celle di livelli diversi sono separati dagli handover e dagli affievolimenti del segnale. Nel secondo caso, i livelli gerarchici differenti sono divisi nel dominio della frequenza.

Nel caso di macro e microcelle operanti alla stessa frequenza, il fattore di riuso della frequenza è posto a uno e l'isolamento spaziale è utilizzato per dividere i due livelli gerarchici e per controllare l'interferenza inter-stato, mentre l'interferenza intra-stato è

monitorata da un controllo di potenza. In un generico modello di terra piatta, la perdita media di trasmissione va come R-2 fino ad una certa distanza, stabilita in base al ricevitore e all'altezza del trasmettitore, mentre dopo è paragonabile a R-4. Quindi, il segnale proveniente da una stazione radio base di una microcella si attenua più velocemente di quello proveniente da una macrocella.

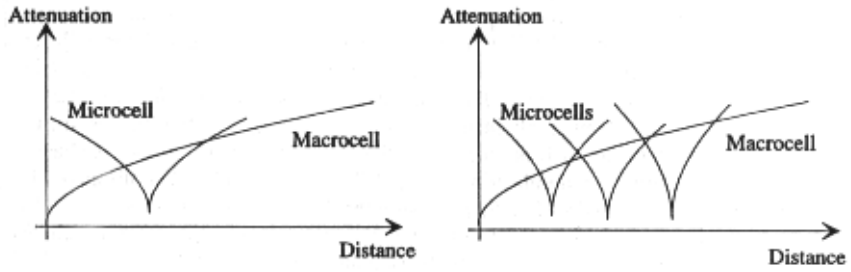


Fig. 2.6 – Separazione spaziale delle celle

Il problema di questa gestione è che il terminale mobile che si trova nella zona di intersezione delle curve di attenuazione dovrebbe essere immediatamente preso in carico dal livello corrispondente. Il soft handover risolve questo problema, tuttavia, se la stazione mobile arriva in una microcella, risulta connessa alla stazione radio base sia della macro che della microcella. Questo accade perché la regione di handover è abbastanza larga per evitare un effetto ping-pong tra le celle, specialmente se il terminale si sta movendo velocemente. L'handover non può essere particolarmente veloce perché il segnale pilota, proveniente dalle diverse stazioni candidate, viene misurato ad intervalli sufficientemente lunghi da filtrare il fading veloce.

Un sistema con i diversi livelli gerarchici a diverse frequenze è più semplice da gestire siccome le varie celle non interferiscono tra di loro. Lo svantaggio è la richiesta di una banda larga, ad esempio 5 Mhz nel sistema UMTS.

2.5 Il livello Data Link

I protocolli radio implementati nei vari nodi di rete del sistema UMTS sono definiti su diversi livelli e, all'interno dei livelli più complessi, esiste un'ulteriore separazione in sottolivelli corrispondente ad una precisa divisione funzionale. Questi protocolli non sono limitati alla parte wireless del sistema, ma si spingono fino al RNC.

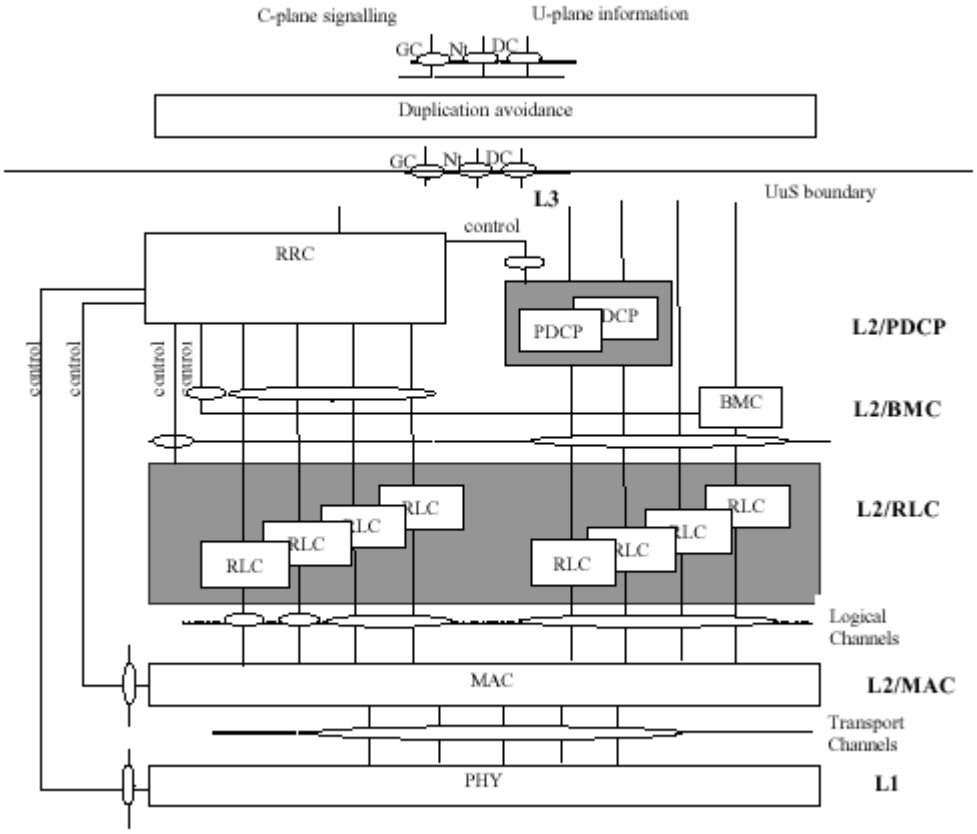


Fig. 2.7 – Architettura dei protocolli radio dell'UTRAN

I protocolli radio dell'UTRAN possono essere divisi in tre livelli:

- Livello Fisico o Livello 1
- Livello Data Link o Livello 2
- Livello Network o Livello 3

Vi è un'ulteriore divisione dei protocolli dell'*Access Stratum* tra piano di controllo e piano utente.

L'informazione che transita attraverso la rete di accesso può essere un'informazione di segnalazione oppure una informazione d'utente, dove la segnalazione ha lo scopo di creare le condizioni per la corretta trasmissione dell'informazione d'utente.

Il livello 3 è fondamentalmente responsabile della gestione della segnalazione

all'interno dell'UTRAN, per questo è collocato completamente nel piano di controllo. Al contrario i primi due livelli sono trasversali ai piani di controllo e d'utente, svolgendo le funzionalità di trasmissione.

Il livello 2 è diviso in alcuni sottolivelli come il MAC oppure il RLC.

2.6 - Livello Medium Access Control

2.6.1 - Servizi forniti ai livelli superiori

I servizi che il MAC fornisce ai livelli superiori sono i seguenti:

- Trasferimento dati: questo servizio fornisce il trasferimento di MAC SDU in modalità unacknowledged tra entità MAC di pari livello, non fornendo però funzionalità di segmentazione e riassemblaggio che dovranno quindi essere svolte dai livelli superiori.
- Riallocazione delle risorse radio e dei parametri MAC: questo servizio esegue, su richiesta del RRC, la riallocazione delle risorse radio e la variazione dei parametri utilizzati all'interno del livello MAC per rendere possibile il cambiamento dei formati di trasporto e del tipo di canale di trasporto.
- Rapporto delle misurazioni: questo servizio realizza locali misurazioni di volume di traffico e di qualità della trasmissione per poi comunicarle al livello RRC.

Il livello MAC fornisce i servizi di trasferimento dati sui canali logici verso il livello RLC. Ogni tipo di canale logico identifica un particolare flusso di informazioni che viene trasferito.

2.6.2 - Funzioni di livello MAC

Le funzioni di livello MAC comprendono:

- Redirezione dei canali logici sui canali di trasporto: i canali logici vengono diretti sui canali di trasporto appropriati.
- Selezione del formato di trasporto: una volta assegnato il Transport Format Combination Set (TFCS) dal RRC, viene selezionato, per ogni canale di trasporto, il

formato di trasporto più opportuno in base alla velocità della sorgente. Il controllo dei formati di trasporto e la possibilità di una loro rapida variazione permettono un utilizzo efficiente dei Transport Channel.

- Gestione delle priorità tra diversi flussi di dati appartenenti ad uno stesso UE: la selezione della particolare combinazione di formati di trasporto all'interno di un TFCS può essere effettuata in base alla priorità del flusso di dati che deve essere inviato sull'opportuno canale di trasporto. In questo modo è possibile assegnare ad un flusso con alta priorità un bit rate più alto ed analogamente un bit rate più basso ad un flusso con una priorità più bassa.
- Gestione della priorità tra UE tramite una schedulazione dinamica: volendo utilizzare le risorse radio in modo efficiente per traffico impulsivo, il MAC realizza la gestione delle priorità su canali comuni e condivisi.
- Identificazione degli UE sui canali di trasporto comuni: quando un UE utilizza un canale comune, vi è la necessità di identificare in banda tale UE. Dato che il livello MAC gestisce l'accesso ai canali di trasporto ed effettua il multiplexing su di essi, l'identificazione degli UE è naturalmente svolta da questo livello.
- Multiplexing e demultiplexing delle PDU dei livelli superiori: il MAC supporta il (de)multiplexing sui canali di trasporto dei transport block e dei transport block set, così da multipare in modo efficiente sullo stesso transport channel i diversi servizi che devono essere forniti.
- Monitoraggio del volume di traffico: questa funzione esegue misurazioni di volume di traffico sui canali logici e le comunica al RRC. In base a tali informazioni, il RRC stabilisce se vi sono modifiche da apportare alle proprietà di trasporto dei canali di traffico.
- Cambiamento del tipo di canale di trasporto: questa funzione esegue la commutazione tra canali di trasporto comuni e dedicati basandosi sulle decisioni effettuate dal RRC.
- Cifratura: questa funzione previene acquisizioni di dati non autorizzate e viene attuata soltanto in modalità di trasmissione RLC trasparente.

2.6.3 – Combinazioni possibili delle modalità di trasmissione

Le tabelle riassumono le possibili combinazioni dei canali nella modalità FDD che possono essere supportate da un UE rispettivamente in *uplink* e in *downlink*. La scelta della modalità di trasmissione, intesa come l'insieme dei canali di trasporto usati sia in uplink che in downlink, dipende da una moltitudine di fattori, quali il tipo di sorgente di traffico (dimensione e velocità di generazione dei pacchetti), la QoS richiesta dal servizio (ritardi di trasferimento e BER) e la disponibilità di risorse.

	COMBINAZIONE CANALI FISICI	COMBINAZIONE CANALI DI TRASPORTO	CAPACITÀ BASE DELLA CONNESSIONE O DIPENDENTE DAL SERVIZIO	COMMENTI
1	PRACH	RACH	Base	Il canale fisico PRACH include il preambolo e il messaggio
2	PRACH	FAUSCH	Dipendente dal servizio	
3	PCPCH (consiste in una parte di controllo e una di dati)	CPCH	Dipendente dal servizio	Il canale fisico PCPCH include il preambolo e il messaggio Il bit rate massimo del canale del canale dalla capacità della stazione radio mobile
4	PCPCH (consiste in una parte di controllo e una di dati)	CPCH	Dipendente dal servizio	Il canale fisico PCPCH include il preambolo e il messaggio Il bit rate massimo del canale del canale dalla capacità della stazione radio mobile
5	DPCCH+DPDCH	Uno o più DCH codificati in un singolo CCTrCH	Dipendente dal servizio	Il numero massimo di canali DCH e il bit rate massimo del canale dipendono dalla capacità della stazione radio mobile
6	DPCCH + diversi DPDCH	Uno o più DCH codificati in un singolo CCTrCH	Dipendente dal servizio	Il numero massimo di canali DCH e il bit rate massimo del canale dipendono dalla capacità della stazione radio mobile

Fig. 2.8 – Modalità di trasmissione uplink

Le caratteristiche dei principali canali di trasporto sono:

- RACH: essendo un canale comune a contesa, l'accesso non è immediato e richiede un certo intervallo di tempo prima di poter effettuare una trasmissione. Per

questo motivo, i servizi che questo canale può supportare non devono risentire di tempi d'attesa elevati; inoltre i pacchetti devono essere di dimensioni limitate e poco frequenti.

- CPCH: le caratteristiche di questo canale sono analoghe a quelle del RACH, con la differenza che il CPCH permette il trasferimento di pacchetti di dimensioni maggiori.

- DCH: essendo un canale di trasporto dedicato ad un particolare UE, non è soggetto a ritardi di accesso come sul RACH e CPCH. Questa caratteristica permette di trasferire pacchetti di dimensioni e frequenza molto variabili con bassi ritardi. La controindicazione all'uso di questo canale risiede nel fatto che l'allocazione di un DCH equivale ad instaurare una connessione a circuito, in quanto viene riservato un codice soltanto per un utente, rendendo quindi posibile il rapido esaurimento delle risorse del sistema. Quindi, è di grande importanza l'utilizzo di adeguate politiche di rilascio del canale, basate per esempio su un timer di inattività, che permettano di sfruttare in modo efficiente l'insieme delle risorse disponibili senza influenzare eccessivamente la QoS che può essere offerta su questo canale.

- DSCH: essendo un canale condiviso, non permette di assicurare un tempo massimo di trasmissione dei dati in downlink, compromettendo così la qualità di ricezione. Questo canale risulta quindi utilizzabile per trasmissioni di pacchetti poco frequenti.

- FACH: anche questo canale è condiviso da più UE e quindi, come per il RACH, possono essere trasmessi soltanto pacchetti di dimensioni limitate; inoltre i tempi di trasferimento possono essere elevati.

- PCH: come il caso precedente, possono essere trasmessi soltanto pacchetti di dimensioni limitate e poco frequenti. Non vi è alcuna garanzia sul ritardo di trasferimento.

	COMBINAZIONE CANALI FISICI	COMBINAZIONE CANALI DI TRASPORTO	CAPACITÀ BASE DELLA CONNESSINE O DIPENDENTE DAL SERVIZIO	COMMENTI
1	PCCPCH	BCH	Base	
2	SCCPCH	FACH + PCH	Base	Il bit rate massimo del canale che può essere supportato dipende dalla capacità del servizio sulla satzione radio

				mobile
3	SCCPCH + AICH	FACH + PCH + RACH in uplink oppure FACH + PCH + CPCH in uplink	Base	Il bit rate massimo del canale che può essere supportato dipende dalla capacità del servizio sulla satzione radio mobile. Questa combinazione del canale fisico facilità la porzione del preambolo del CPCH in uplink.
4	SCCPCH + DPCCH	FACH + PCH + CPCH in uplink	Dipendente dal servizio	Questa combinazione di canale fisico favorisce la porzione di messaggio del CPCH in uplink.
5	Più di un SCCPCH	Più di un FACH + PCH	Dipendente dal servizio	
6	DPCCH + DPDCH	Uno o più DCH codificati in un solo CCTrCH	Dipendente dal servizio	Il numero massimo di canali DCH e il bit rate massimo del canale sono dipendenti dalla capacità della stazione radiomobile.
7	DPCCH + più di un DPDCH	Uno o più DCH codificati in un solo CCTrCH	Dipendente dal servizio	Il numero massimo di canali DCH e il bit rate massimo del canale sono dipendenti dalla capacità della stazione radiomobile.
8	PDSCH + DPCCH + più di un DPDCH	DSCH + uno o più DCH codificati in un solo CCTrCH	Dipendente dal servizio	Il numero massimo di canali DCH e il bit rate massimo del canale sono dipendenti dalla capacità della stazione radiomobile.
9	SCCPCH + DPCCH + più di un DPDCH	FACH + uno o più DCH codificati in un solo CCTrCH	Dipendente dal servizio	Il numero massimo di canali DCH e il bit rate massimo del canale sono dipendenti dalla capacità della stazione radiomobile.
10	SCCPCH + PDSCH + DPCCH + più di un DPDCH	FACH + DSCH + uno o più DCH codificati in un solo CCTrCH	Dipendente dal servizio	Il numero massimo di canali DCH e il bit rate massimo del canale sono dipendenti dalla capacità della stazione radiomobile.
11	Un DPCCH + più di un DPDCH	Più di un DCH codificato in uno o più CCTrCH	Dipendente dal servizio	

Fig. 2.9 – Modalità di trasmissione downlink

2.7 - Livello RLC (Radio Link Control)

2.7.1 - Servizi forniti ai livelli superiori

I principali servizi forniti dal livello RLC sono i seguenti:

- Instaurazione e rilascio di una connessione RLC: viene installata una connessione per ogni radio bearer.
- Trasferimento di dati in modalità trasparente: questo tipo di servizio permette di trasmettere le PDU dei livelli superiori senza aggiungere alcuna informazione di protocollo, limitandosi alla segmentazione e al riassemblaggio.
- Trasferimento dati in modalità unacknowledged: tramite questo tipo di servizio è possibile trasferire le PDU dei livelli superiori senza garanzie sulla consegna all'entità di pari livello. La modalità unacknowledged ha le seguenti caratteristiche:
    - Rilevamento dei dati errati e consegna ai livelli superiori delle sole SDU che non sono state corrotte da errori di trasmissione;
    - Singola consegna di ogni SDU ai livelli superiori utilizzando la funzione di rilevamento dei duplicati;
    - Consegna immediata delle SDU all'entità ricevente del livello superiore.
- Trasferimento dei dati in modalità acnowledged: questo tipo di servizio garantisce la consegna alla pari entità. Nel caso in cui il RLC non fosse in grado di consegnare i dati in modo corretto, viene inviata una notifica all'entità di trasmissione. Per questo servizio vengono supportate entrambe le modalità di consegna in sequenza e fuori sequenza.

  La modalità acknowledged ha le seguenti caratteristiche:
    - Consegna al livello superiore delle sole SDU ricevute correttamente.
    - Singola consegna al livello superiore.
    - Consegna delle SDU nello stesso ordine in cui sono state trasmesse.
    - Consegna fuori sequenza, alternativa alla precedente, delle SDU al livello superiore.

- Impostazione della QoS: il protocollo di trasmissione deve essere configurabile dal livello 3 per fornire diverse qualità del servizio.
- Notifica degli errori irrecuperabili: il RLC notifica al livello superiore che sono presenti degli errori che non possono essere risolti all'interno del livello.

2.7.2 - Funzioni del livello RLC

Le funzioni principali sono:

- Segmentazione e riassemblaggio: questa funzione realizza la segmentazione e il riassemblaggio delle PDU di livello superiore di dimensione variabile in/da RLC *Payload Unit* (PU) di dimensioni minori che possono essere adattate all'attuale insieme di formati di trasporto.
- Concatenazione: se il contenuto di una SDU di livello RLC non rientra in un numero intero di RLC PU, un primo segmento della seguente RLC PU può essere inserito in una nuova RLC PU in concatenazione con l'ultimo segmento della precedente SDU.
- Padding: quando non è possibile applicare la concatenazione e i dati rimanenti che devono essere trasmessi non rientrano in una RLC PDU di una determinata dimensione, nella parte rimanente del campo dati vengono inseriti bit di padding.
- Trasferimento di dati utente: questa funzione è utilizzata per il trasporto dei dati degli utenti dei servizi RLC. Le modalità di trasferimento supportate sono di tipo acknowledged, unacknowledged e trasparente.
- Correzione dell'errore: questa funzione fornisce la correzione dell'errore tramite ritrasmissione (*Selective Repeat, Go Back N, Stop and Wait ARQ*) dei dati ricevuti non correttamente tramite la modalità di trasferimento acknowledge.
- Consegna in sequenza: questa funzione preserva l'ordine con cui le PDU del livello superiore vengono trasferite utilizzando la modalità acknowledged. Se tale funzione non è utilizzata, allora viene fornito il servizio di consegna fuori sequenza.
- Rilevamento dei duplicati: questa funzione rileva la presenza di RLC PDU duplicate e assicura che la PDU del livello superiore venga consegnata solo una volta.
- Controllo di flusso: tramite questa funzione il ricevitore può controllare la velocità

con cui la pari entità RLC in trasmissione invia le informazioni.

- Controllo del numero di sequenza: per la modalità di trasferimento unacknowledged viene controllata l'integrità delle PDU riassemblate e viene fornito un meccanismo per il rilevamento delle RLC SDU corrotte attraverso il controllo del numero di sequenza.
- Rilevamento e recupero degli errori di protocollo: questa funzione permette di continuare le operazioni del protocollo anche in seguito alla presenza di errori.
- Cifratura: previene acquisizioni di dati non autorizzate.

2.7.3 - Modalità trasparente

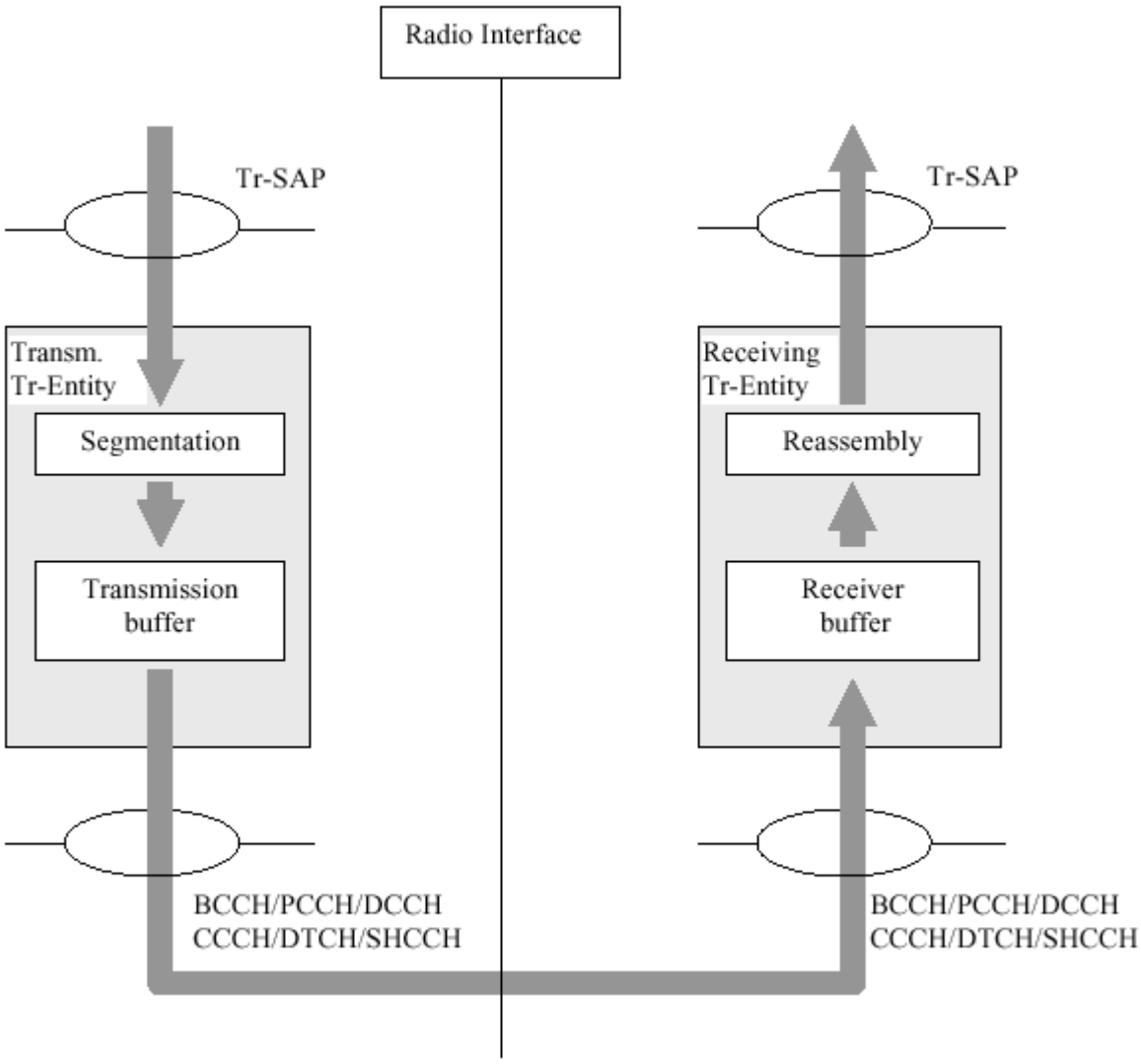


Fig. 2.10– Modalità di trasmissione trasparente

La figura mostra il modello di due entità di pari livello nella modalità trasparente.

L'entità di trasmissione riceve le SDU dai livelli superiori attraverso il Tr-SAP (*Trasparent Service Access Point*) e le segmenta in RLC PDU di dimensioni appropriate senza aggiungere ulteriori informazioni di controllo. Il modo in cui realizzare la segmentazione viene deciso una volta stabilito il servizio. Il livello RLC consegna al MAC le RLC PDU attraverso i canali logici BCCH, PCCH, SHCCH, SCCH, e DTCH a seconda che il livello superiore sia collocato nel piano di controllo o in quello utente. Il canale logico CCCH usa la modalità trasparente soltanto in uplink. All'entità ricevente vengono passate dal MAC le PDU attraverso i canali logici. Il livello RLC riassembla queste ultime in RLC SDU e le consegna al livello superiore attraverso il Tr-SAP.

2.7.4 - Modalità unacknowleged

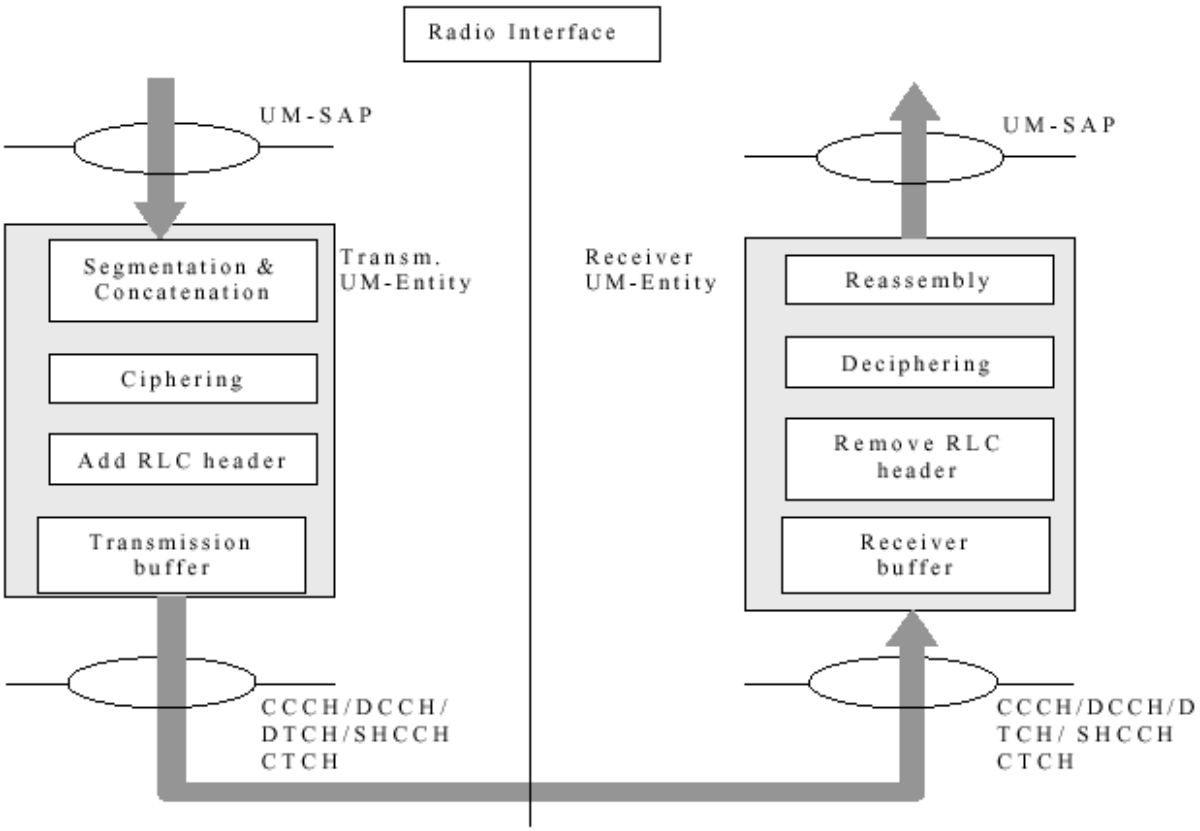


Fig. 2.11 – Modalità di trasmissione unacknowledged

Nella modalità unacknowledged, l'entità trasmittente riceve le SDU dal livello superiore e le segmenta in PDU di dimensioni appropriate con la possibilità di concatenare più SDU.

Il livello RLC aggiunge un'intestazione e la PDU risultante viene inserita nell'apposito buffer di trasmissione. Le PDU vengono poi trasferite al sottolivello MAC attraverso i canali logici DCCH, SHCCH, CTCH o DTCH e tramite il CCCH solo in downlink. L'entità in ricezione riceve le PDU dal MAC attraverso i canali logici. Il livello RLC elimina le intestazioni dalle PDU e le riassembla in RLC SDU per passarle in seguito al livello superiore.

2.7.5 - Modalità acknowledged

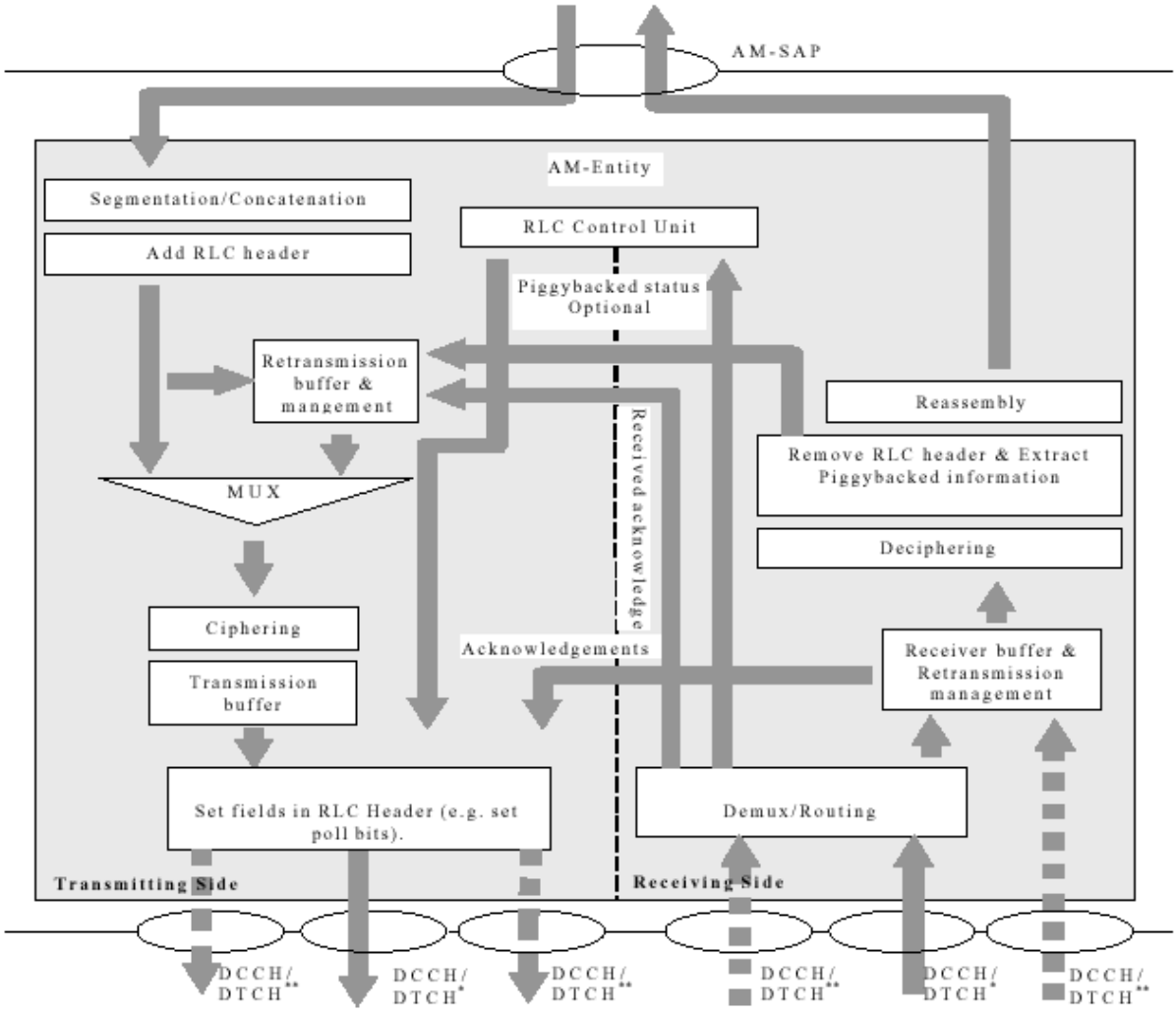


Fig. 2.12 – Modalità di trasmissione acknowledged

Il modello rappresenta due entità di pari livello in modalità acknowledged.

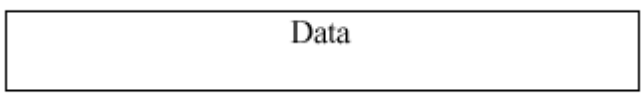
L'entità di trasmissione riceve le SDU dal livello superiore e le segmenta o concatena in PU di lunghezza prefissata, che corrispondono alle PDU. Tale lunghezza è un

valore deciso durante la fase di bearer setup e può essere modificata soltanto dal livello RRC attraverso una riconfigurazione del bearer. Se più SDU vengono concatenate in un'unica PU, appropriate indicatori di lunghezza (Lengh Indicator) vengono inseriti nell'intestazione della PDU. In seguito le PDU vengono nel buffer di trasmissione ed in quello di ritrasmissione. L'entità denominata MUX stabilisce quando e quali PDU consegnare al livello MAC per poter ad esempio gestire la trasmissione di PDU di controllo o di dati utente. Le PDU sono consegnate tramite una funzione che completa l'intestazione e rimpiazza il padding con informazioni di stato inviate in piggybacking. Questo permette di aumentare l'efficienza della trasmissione e rende possibile uno scambio di messaggi più veloce tra le entità di pari livello. Le informazioni di controllo contenute nella piggybacked STATUS PDU di dimensioni variabili (a sua volta contenuta nella AMD-PDU) e inviate in piggybacking, non vengono memorizzate in nessun buffer di ritrasmissione. L'entità precedente accetta le PDU che vengono passate dal livello MAC attraverso i canali logici ed estrae da esse le eventuali informazioni di stato inviate in piggybacking per poi memorizzare le PU nel buffer di ricezione fino a quando l'intera SDU non è stata ricevuta. Il buffer di ricezione può chiedere le ritrasmissioni delle PU tramite l'invio di un NACK alla pari entità di trasmissione. In seguito alla rimozione delle intestazioni ed alla operazione di riassemblaggio, l'intera SDU viene consegnata al livello superiore. All'entità ricevente giungono anche i riscontri di avvenuta ricezione della pari entità e quindi è necessario che siano passati al buffer di ritrasmissione.

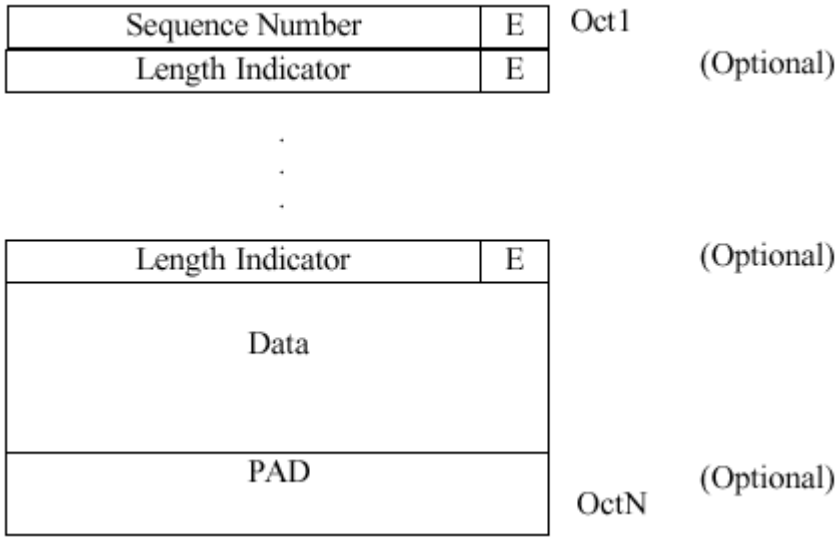
2.7.6 - PDU di livello RLC

Le RLC PDU possono essere divise in due sottosistemi costituiti dalle PDU che trasportano dati di utente e quelle che trasportano informazioni di controllo. Le RLC Data PDU sono:

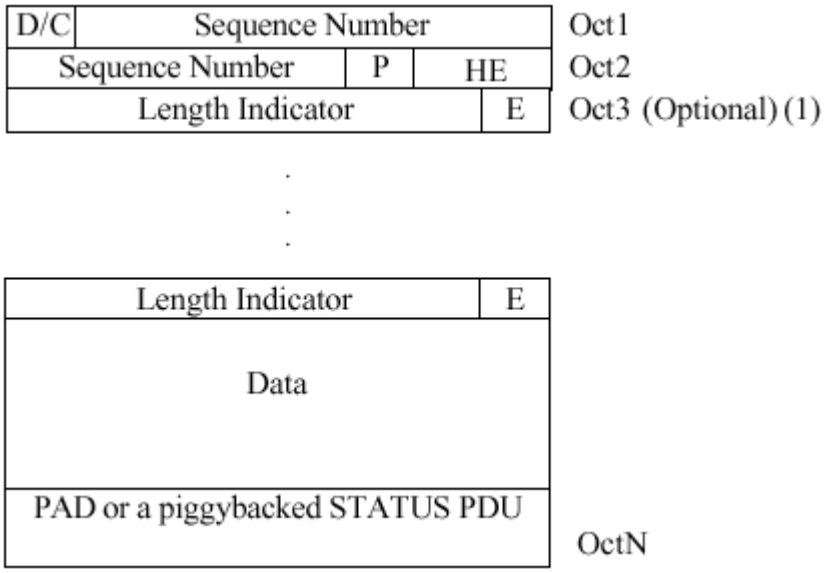
- TrD PDU (*Transparent mode data PDU*): serve per il trasporto dei dati in modalità trasparente. Il livello RLC non aggiunge alcuna intestazione.



- MD PDU (*Unacknowledged mode data PDU*): è utilizzata per trasferire RLC SDU in modalità acknowledged tramite una numerazione sequenziale.

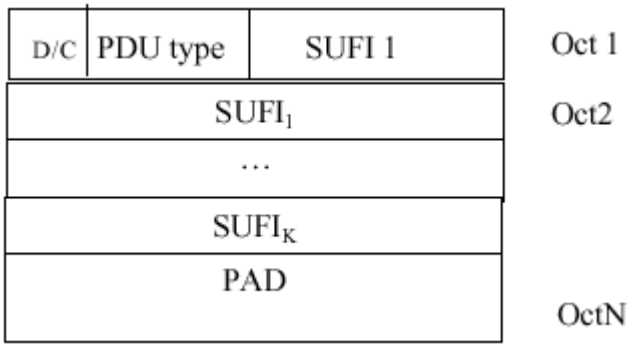


- AMD PDU (*Aknowledged mode data PDU*): è utilizzata in modalità acknowledged e trasporta dati utente o informazioni di stato in piggyback.

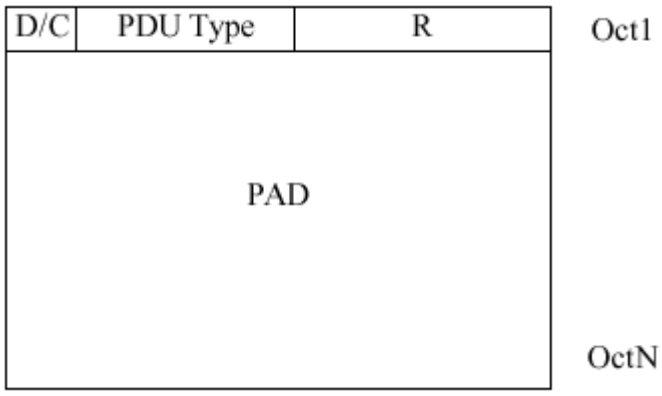


Le control PDU sono:

- Status PDU e piggybacked STATUS PDU: sono utilizzate in modalità AM dall'entità ricevente per comunicare alla controparte la dimensione consentita della finestra di trasmissione o la mancata ricezione di alcune PU.



- RESET PDU: in modalità acknowledged, viene utilizzata per inizializzare tutti gli stati, le variabili e i timer di protocollo dell'entità di pari livello per raggiungere la sincronizzazione.



- RESET ACK PDU: costituisce un ACK per la RESET PDU.

I campi principali di una RLC PDU sono i seguenti:

- D\C field: occupa 1 bit e indica il tipo della AM PDU, ovvero se è di controllo o di dati.
- PDU type: occupa tre bit e indica il tipo di control PDU utilizzato.
- Sequence Number (SN): indica il numero di sequenza della PU. Occupa 12 bit nel caso di AMD PDU o 7 bit nel caso di UMD PDU.
- Polling bit (P): occupa 1 bit ed è utilizzata dall'entità RLC per richiedere l'invio di una STATUS PDU.
- Exstension bit (E): occupa 1 bit e indica se l'ottetto successivo è da interpretare come *length indicator*.
- Lenght Indicator (LI): può occupare 7 o 15 bit e indica la fine di una SDU all'interno della PU; inoltre determina il numero di ottetti dall'ultimo LI fino al termine della

SDU in esame.

2.7.7- Variabili di stato

Il livello RLC si serve di variabili di stato per la gestione delle trasmissioni. Le PU sono numerate in modo sequenziale da 1 a *n*-1, dove *n* è pari a 2^{12} per la modalità acknowledged ed a 2^7 per la modalità unacknowleged.

Le principali variabili di stato utilizzate dal trasmettitore sono:

- VT(S) (*Send State Variable*): è il numero di sequenza della prossima PDU da trasmettere per la prima volta; viene aggiornata dopo la trasmissione di una PDU che non include PU trasmesse precedentemente. Questa variabile viene posta inizialmente a 0.
- VT(A) (*Acknowledged State Variable*): è il numero di sequenza della prossima PDU per la quale si attende un ACK e rappresenta l'elemento nella parte inferiore della finestra dei messaggi di acknowledgement accettati come validi. Il valore iniziale è 0.
- VT(DAT): questa variabile conta il numero di volte che una PU è stata trasmessa. Vi è una di queste variabili in ogni PU e viene incrementata ogni volta che la PU è trasmessa. Il valore iniziale è 0.
- VT(MS) (*Maximum Send State Variable*): è il numero di sequenza della prima PU non ritenuta valida dal ricevitore della pari entità. Questo valore rappresenta l'elemento della parte superiore della finestra di trasmissione. Il trasmettitore non invia nuove PU se VT(S)>VT(MS). Questa variabile è aggiornata sulla base di informazioni ricevute tramite le STATUS PDU.

Le principali variabili di stato utilizzate dal ricevitore sono:

- VR(R) (R*eceive State Variable*): è il numero di sequenza della prossima PU che il ricevitore si aspetta di ricevere. Il valore iniziale è 0.
- VR(H) (*Highest Expected State Variable*): è il numero più alto di PU atteso. Questa variabile di stato è aggiornata quando una nuova PU viene ricevuta.

- VR(MR) (*Maximum Acceptable Receive State Variable*): è il numero di sequenza della prima PU non accettata dal ricevitore., ovvero VR(MR)=VR(R)+Rx_Window_Size. Il ricevitore rifiuta le PU con SN>VR(MR).

2.7.8 - Il flusso dati attraverso il livello 2

I flussi delle informazioni attraverso il livelli 2 sono caratterizzati dalle modalità di trasferimento applicate dal livello RLC e dal MAC. Nel caso in cui non sia richiesta l'aggiunta di un'intestazione MAC, la trasmissione attraverso questo livello è definita trasparente.

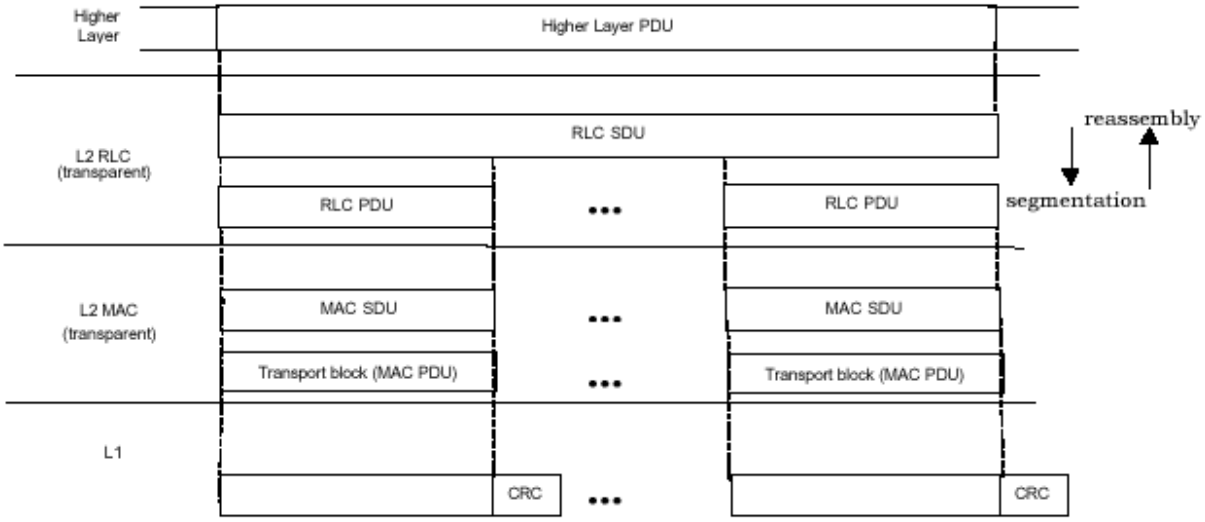


Fig. 2.13 – Flusso dati con RLC e MAC trasparenti

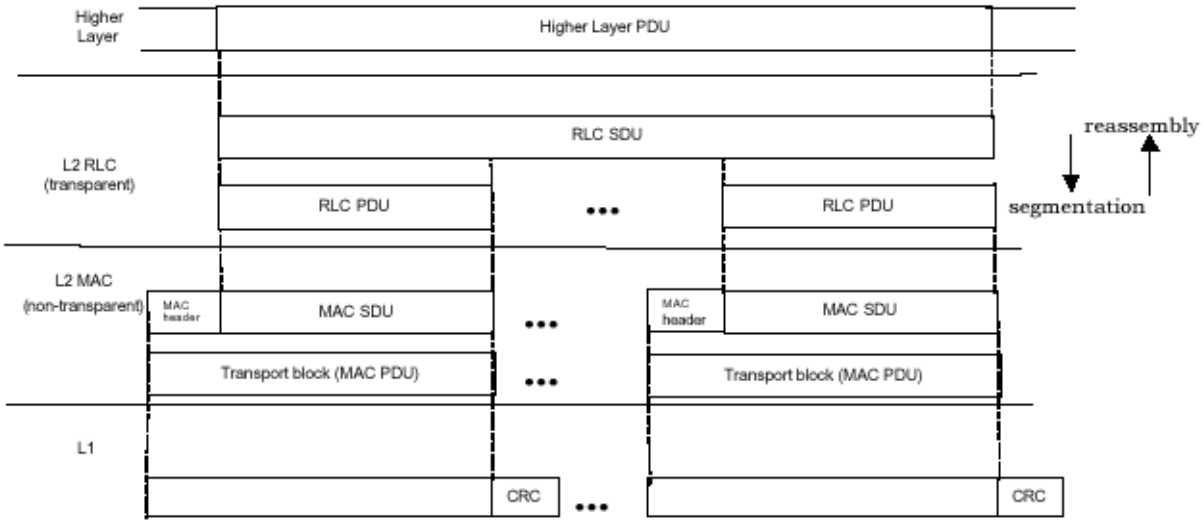


Fig. 2.14 – Flusso dati con RLC trasparente e MAC non trasparente

Le modalità RLC unacknowledged e acknowledged richiedono comunque

un'intestazione RLC. Nel primo caso viene scambiato un solo tipo di PDU dati tra le pari entità RLC, mentre nel secondo caso è possibile lo scambio sia di PDU dati che di controllo. Il termine trasmissione trasparente è utilizzato per caratterizzare il caso in cui un protocollo, MAC o RLC, non richieda alcuna informazione di controllo da inserire nell'intestazione. Tuttavia, in certi casi, alcune funzioni del protocollo possono comunque essere applicate, come ad esempio la segmentazione e il riassemblaggio per il livello RLC.

Nelle immagini si sono accorpate le modalità RLC acknowledged e unacknowledged nella modalità di trasmissione non trasparente.

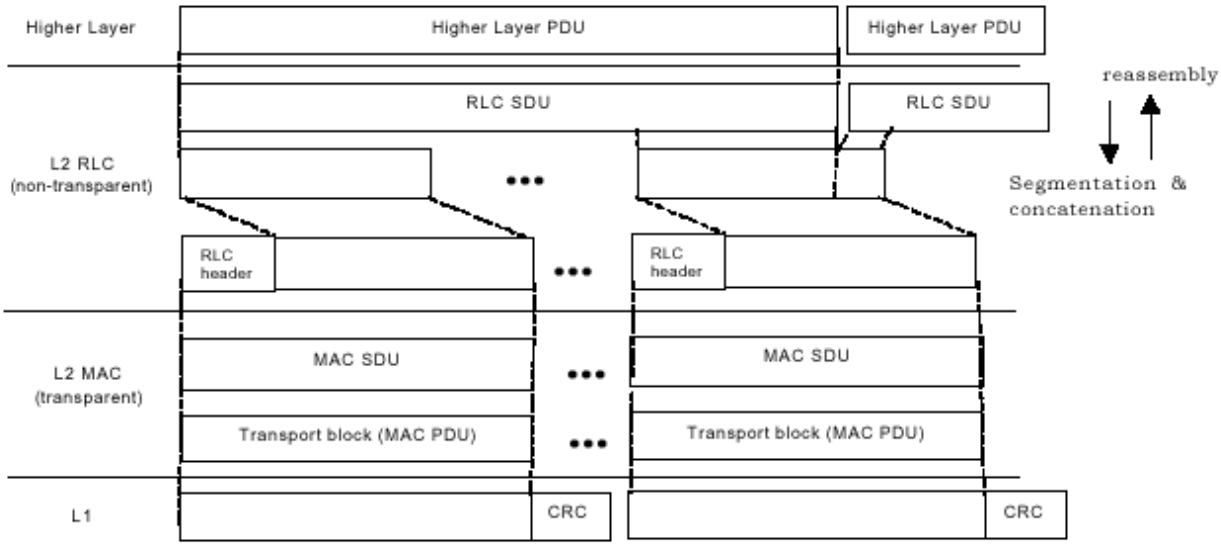


Fig. 2.15– Flusso dati con RLC non trasparente e MAC trasparente

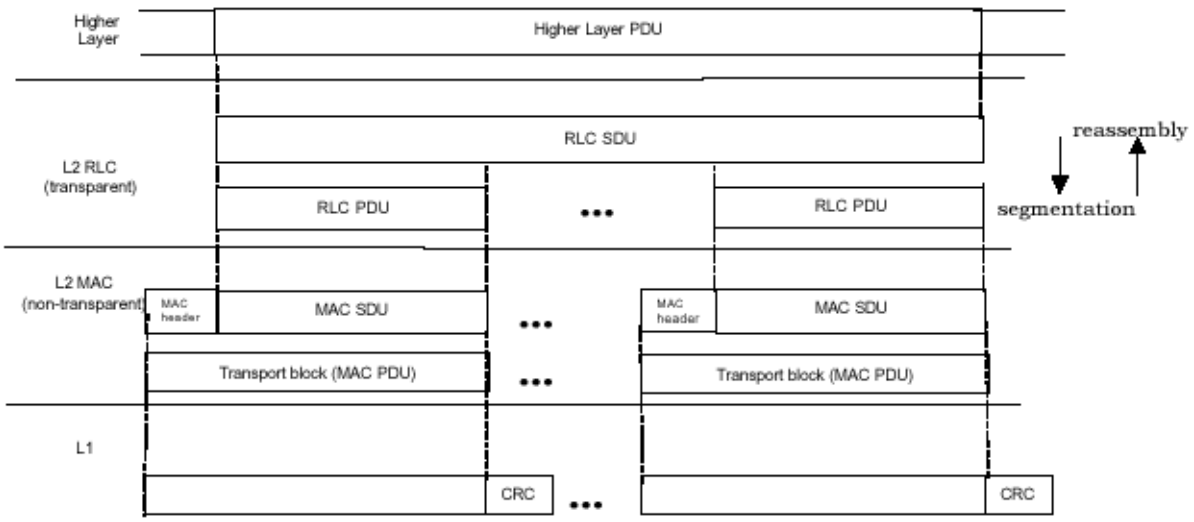


Fig. 2.16 – Flusso dati con RLC e MAC non trasparenti

2.8 ARQ (Automatic Repeat Request).

I protocolli Radio Link Control (RLC) sono utilizati di solito per la cosegna affidabile di SDU in sequenza. Per la correzione di errori sul canale radio si usa la ritrasmissione dei blocchi errati ricorrendo allo schema Selective Repear Request (SR-ARQ).

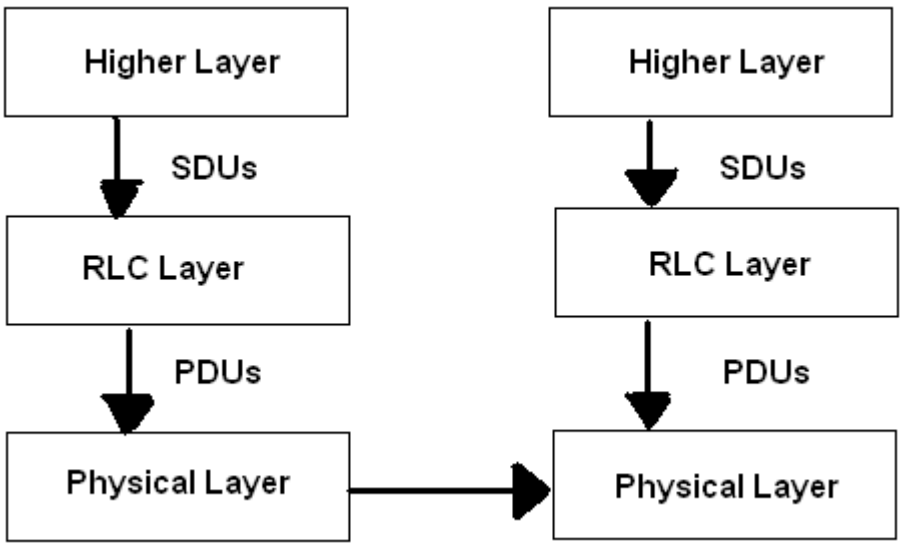


Fig. 2.17 – Struttura RLC in comunicazioni wireless

Le SDU sono divise in blocchi, chiamati PDU (Payload Data Unit), da trasmettere sul canale radio e protetti con un CRC (Cyclic Redundancy Check). Il ricevitore utilizza il CRC per controllare se la PDU è corretta: in questo caso si manda un riscontro (ACK, acknowledgment) al mittente. Se la PDU è errata si invia un riscontro negativo (NACK), in modo che il trasmettitore spedisca nuovamente la PDU. I vantaggio del SR-ARQ è che si ritrasmette solo i blocchi errati all'interno di una finestra di trasmissione. Per semplificare, le PDU errate occupano le stesse posizioni all'interno delle finestre successive.

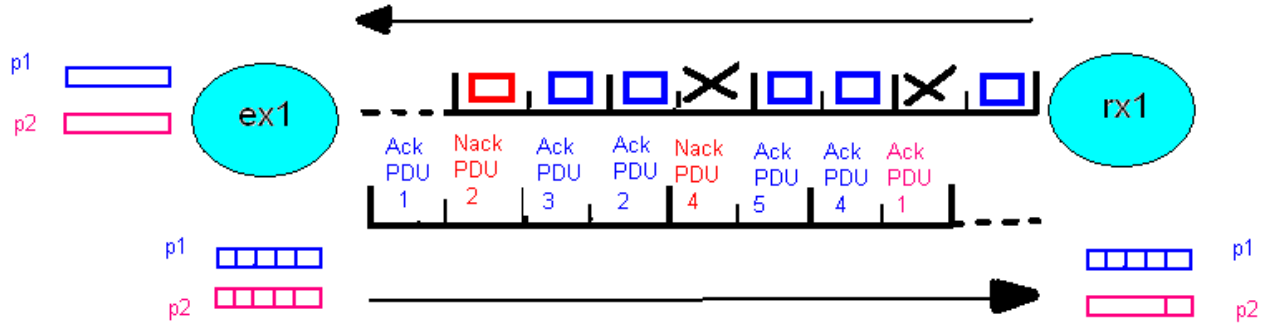


Fig. 2.18 – Schema da funzionamiento da SR-ARQ

Capitolo 3 L’architettura Diffserv

L’incredibile aumento di dimensioni di Internet ha messo in luce le debolezze intrinseche di IP e del servizio best effort che esso consente.

Il problema più grosso non è l’aumento del volume del traffico ma è piuttosto il cambiamento della natura di tale traffico: oggi Internet è usata da applicazioni che erano impensabili quando il protocollo è stato concepito, ciascuna delle quali genera un tipo diverso di traffico con diversi requisiti di prestazioni. Ad esempio, molte di queste nuove applicazioni, come il multimedia, richiedono risorse sostanziali in termini di banda, altre, quali la telefonia su Internet, hanno requisiti stringenti in termini di ritardo di consegna.

L’IETF nel 1997 ha studiato un diverso approccio al problema della QoS per definire un’architettura che tenesse conto dei seguenti vincoli:

- essere sufficientemente elastica da adattarsi ad un’ampia varietà di servizi, anche di tipologie non previste al momento della stesura.
- garantire un’elevata scalabilità.
- operare con le applicazioni esistenti senza modifiche all’interfaccia software tra esse e la rete.
- non dipendere da segnalazioni tra l’applicazione e tutti i nodi coinvolti nel cammino dei dati.

- evitare di mantenere informazioni di stato riguardo ai singoli microflussi, o alle singole sorgenti, nei nodi interni alla rete, limitandosi, invece, allo stato di un numero ragionevole di aggregati.
- basarsi su un numero ristretto di politiche di trattamento dei pacchetti, dotate di implementazioni semplici ed efficienti per non penalizzare i nodi che trattano grosse quantità di traffico.
- poter coesistere con reti che non offrono il controllo della QoS, in modo da poter essere adottata gradualmente.

Il risultato di quattro anni di lavoro dell'apposito *Working Group* è l'architettura a servizi differenziati *Differentiated Services*, o brevemente *DiffServ.*

3.1 Panoramica sull'architettura

L'architettura a servizi differenziati viene presentata come divisa in due parti distinte. I membri del DiffServ Working Group hanno operato questa distinzione ispirandosi al progetto originale di Internet, in cui si scelse di separare le funzionalità di *forwarding* e quelle di *routing*. L'operazione di forwarding, infatti, deve essere effettuata separatamente per ogni pacchetto ed è estremamente semplice e veloce, poiché è costituita dalla scelta di una riga nella tabella di routing in base al contenuto dell'header del pacchetto in esame. Il routing è, invece, la gestione della tabella ed è un'operazione delicata e laboriosa, anche se viene eseguita ad intervalli di tempo di molto superiori al tempo di transito di un pacchetto.

Allo stesso modo, l'architettura DiffServ comprende una componente ben definita che opera a livello dei singoli pacchetti, e un'altra, molto più complessa, relativa alle politiche di allocazione delle risorse, cui spetta il compito di configurare i parametri usati dalla prima.

La prima componente, che prende il nome di "comportamento al nodo" o *Per-Hop Behavior* (*PHB*), realizza il trattamento differenziato che ciascun pacchetto riceve in un nodo, definito in termini di disciplina di servizio della coda, buffer allocati, priorità rispetto alle altre code, etc. Nel documento l'attenzione viene focalizzata sulla semantica generale

dei PHB piuttosto che sui dettagli implementativi, presumendo che le caratteristiche "esterne" evolvano più lentamente delle implementazioni.

I PHB, insieme con le funzionalità necessarie per assegnare ad essi i pacchetti (operazione di *classifying*), possono essere implementati nelle reti esistenti con poche modifiche. In aggiunta, possono rendersi necessari dei meccanismi per il monitoraggio del traffico (operazione di *metering*) e per una sua eventuale manipolazione (operazioni di *policing*, *shaping* e *dropping*), in modo da renderlo conforme alle condizioni necessarie per poter usufruire di tale trattamento.

Queste funzionalità, insieme ai PHB, costituiscono i blocchi con i quali è possibile costruire i servizi differenziati, siano essi end-to-end o interni ad un dominio.

Viceversa, le metodologie appartenenti alla seconda componente, che permettono di effettuare la scelta dei pacchetti da destinare a trattamenti particolari e di stabilire che tipo di regole adottare per l'utilizzo delle risorse, sono ancora poco approfondite. Nonostante ciò, è possibile ottenere un'efficace differenziazione dei servizi di rete per mezzo di politiche semplici e di configurazioni statiche.

3.1.1 Il ruolo di core ed edge router

Per illustrare l'approccio DiffServ occorre innanzi tutto introdurre il concetto di "dominio a servizi differenziati", chiamato *Differentiated Services domain* o *DS-domain*. Questo è costituito da un insieme di nodi contigui in cui le politiche di fornitura dei servizi differenziati e la loro gestione sono amministrate in modo coordinato ed indipendente. Un dominio a servizi differenziati, spesso indicati in letteratura come nuvole o *clouds*, può comprendere al suo interno una o più reti sotto la stessa amministrazione, come è il caso di una Intranet o di un ISP.

All'interno di un dominio, l'architettura DiffServ distingue tra nodi di frontiera, detti *edge routers*, e nodi interni ad esso, detti *core routers*. I nodi di frontiera costituiscono l'interfaccia di collegamento con i clienti, ai quali si intende fornire differenziazione dei servizi, o con altre reti gestite da altre entità, siano esse DiffServ compatibili o meno. I nodi interni, invece, collegano solamente altri nodi interni o di frontiera, ma tutti appartenenti allo stesso dominio.

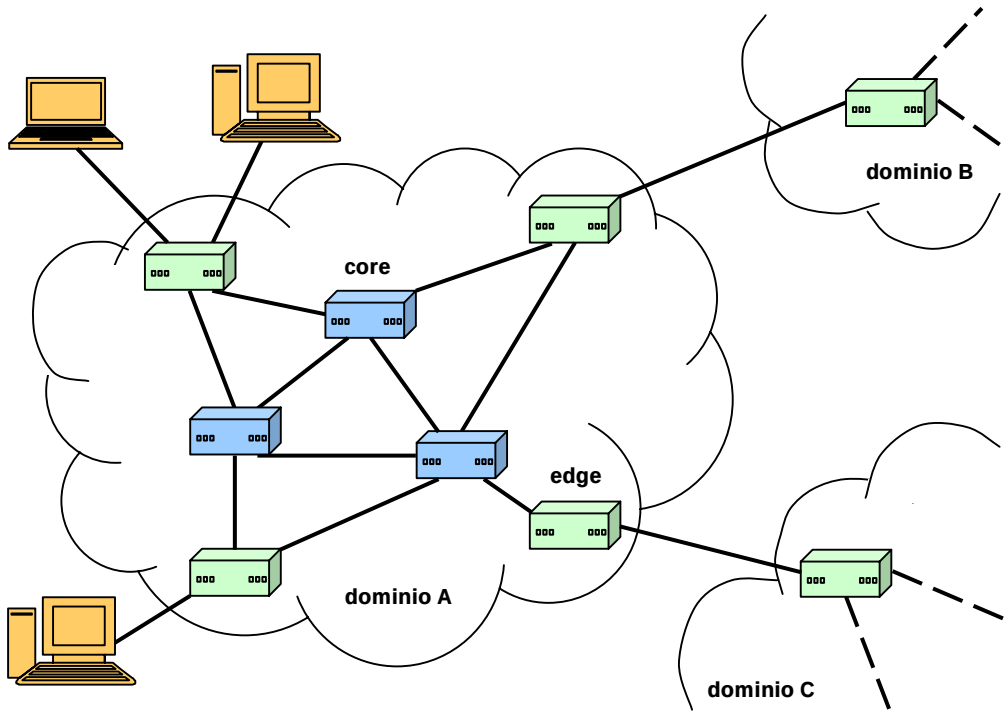


Figura 4.1: Esempio di un dominio DiffServ.

Il requisito di scalabilità e di alta velocità nel trattamento dei singoli pacchetti è ottenuto a scapito della granularità nella gestione dei flussi. Anziché identificare ciascun flusso in ogni nodo, nell'architettura DiffServ tale distinzione viene effettua soltanto nei nodi di frontiera: i pacchetti entranti in un dominio a servizi differenziati vengono, infatti, distinti come appartenenti ad un numero limitato di classi di servizio. La classe di appartenenza è indicata nel pacchetto stesso sotto forma di un opportuno codice, detto *Differentiated Services Codepoint* o *DSCP*. Il codice viene inserito nell'intestazione del pacchetto in un campo predisposto per questo scopo, chiamato *Differentiated Services Field*, dal primo nodo DiffServ compatibile incontrato dal pacchetto durante il percorso, sia esso la sorgente stessa o il primo router di frontiera del dominio. A valle del nodo che ha effettuato la marcatura, l'insieme dei pacchetti appartenenti ad una stessa classe, provenienti in generale da più sorgenti diverse, viene trattato come un'entità unica chiamata *Behavior Aggregate* o *BA*, definito come "una collezione di pacchetti con lo stesso DSCP che attraversano un link in una determinata direzione". In contrapposizione all'aggregato, si parla di *microflusso* come "un singolo flusso di pacchetti, da applicazione ad applicazione, identificato da indirizzi di sorgente, destinazione e, quando possibile, dai numeri di porta". In generale un BA è quindi costituito dall'insieme di più microflussi.

Ad ogni classe di servizio è associato un diverso trattamento del corrispondente aggregato, ossia un PHB: i nodi interni possono, quindi, limitarsi ad esaminare il contenuto del DSCP dei pacchetti in arrivo e a fornire a ciascuno il PHB corrispondente. La quantità di informazioni di stato da mantenere nei nodi interni è, dunque, proporzionale al numero delle classi, che negli intenti dell'IETF deve rimanere molto limitato, anziché al numero dei microflussi entranti nel dominio.

La scelta e l'assegnazione dei PHB sono operazioni estremamente rapide e quindi adatte a router che trattano grossi volumi di traffico, ma non sono sufficienti a costruire dei servizi *end-to-end*. Occorre infatti, come già accennato, almeno verificare che le sorgenti o i domini "a monte" non richiedano servizi di tipologie non previste o che non si è in g rado di offrire; questa verifica deve essere effettuata a livello di microflussi. L'architettura DiffServ, invece, delega tutte le operazioni necessarie a questo scopo alla frontiera del dominio, nella quale, verosimilmente, ciascun nodo tratta un numero limitato di microflussi. Ed inoltre, nel caso di un eventuale cambiamento futuro delle politiche di offerta di QoS, sarà sufficiente riconfigurare solo alcuni componenti della frontiera, senza modificare i meccanismi presenti nei nodi interni.

3.1.2 Classificazione e condizionamento del traffico

Ogni dominio DiffServ si accorda con le entità direttamente connesse alla sua rete, siano esse clienti o altri domini confinanti, stipulando un accordo detto *Service Level Agreement* o *SLA*. Questo contiene al suo interno delle specifiche di servizio relative ai BA, chiamate *Service Level Specifications* o *SLS*, che descrivono il servizio che il dominio fornisce da un punto di vista esterno, cioè per mezzo di parametri come *throughput, ritardo, jitter, probabilità di perdita* dei pacchetti. Bisogna anche notare che, nell'architettura DiffServ, i servizi si intendono sempre in una sola direzione: in generale, ad ogni frontiera si stabiliranno coppie di SLS per le due direzioni di traffico.

Dalle specifiche di servizio contenute negli SLS si ricavano le specifiche sul condizionamento del traffico, indicate in letteratura come *Traffic Conditioning Specifications* o *TCS*, che definiscono a livello di pacchetto il trattamento che gli aggregati ricevono all'ingresso della frontiera del dominio. Questo accordo sul condizionamento si traduce in modo quantitativo sotto forma della configurazione di un blocco funciónale

presente in tutti i nodi del dominio, detto *Traffic Conditioner Block* o *TCB*, che ha lo scopo di manipolare i flussi per renderli conformi alle specifiche date.

La prima operazione da effettuare sul traffico, per poter attuare una differenziazione dei servizi, consiste nel decidere quale servizio sarà offerto a ciascun pacchetto per mezzo di una classificazione. Nell'architettura DiffServ questa operazione è svolta da un blocco funzionale, chiamato *Traffic Classifier*, che seleziona i pacchetti in ingresso per mezzo di filtri applicati a vari campi dell'intestazione, come indirizzo di sorgente e destinazione, protocollo, numero di porta, DS-Field, o altre proprietà come l'interfaccia di arrivo. Sono definiti due tipi di classificatori:

- *BA Classifier:* seleziona i pacchetti solo in base al valore del DSCP; è pensato essenzialmente per i nodi interni di un dominio, dove i pacchetti devono soltanto essere instradati verso il blocco funzionale che realizza il PHB associato.
- *Multi-Field Classifier:* è una generalizzazione del BA Classifer adatto alla frontiera di un dominio; i flussi in uscita vengono diretti verso altrettanti blocchi di condizionamento del traffico che verificano la loro conformità con gli SLS.

3.2 I Per-Hop Behavior

I *Per-Hop Behavior* o *PHB* vengono definiti come "una descrizione del trattamento di forwarding da applicare ad un aggregato, come appare dall'esterno". Si possono definire gruppi di PHB, dove con gruppo si intende un insieme di due o più PHB che possono soltanto essere definiti ed implementati insieme per via di un vincolo comune, come può essere una disciplina di gestione di una coda, o un algoritmo di scheduling. In questo caso nella definizione deve essere chiaramente indicato se e sotto quali condizioni è possibile marcare un pacchetto con un PHB o un altro dello stesso gruppo.

Lo spazio dei 64 possibili codepoint è suddiviso in tre gruppi, chiamati ***DSCP pools***, che vengono illustrati nella seguente figura:

Gruppo	Codepoint	Assegnazione
1	**xxxxx0**	**Azioni standard (EF, AF, Default, Class Selector Codepoint)**
2	**xxxx11**	**Sperimentale/Locale**
3	**xxxx01**	**Sperimentale/Locale/Standard Futuri**

Figura 4.4: DSCP pools.

Nel gruppo 1 sono raccolti i codepoint che possono avere significato anche tra un dominio e un altro, e otto di questi sono riservati per indicare i Class Selector PHB, come descritto nel paragrafo precedente. Il gruppo 2 comprende 16 codepoint riservati per uso sperimentale o interno ad un dominio, come anche i 16 appartenenti al gruppo 3, i quali dovranno essere usati per le future definizioni quando lo spazio di indirizzamento del secondo gruppo sarà esaurito.

Ogni dominio può implementare, indipendentemente dalle scelte di altri domini, tutti i PHB che ritiene utili per il tipo di servizio offerto, siano essi standardizzati oppure no. È opportuno sottolineare che, benché il campo DSCP sia limitato a 64 valori, non esiste alcun limite al numero di PHB che sarà possibile realizzare: lo scenario più probabile vede un numero limitato di PHB di maggiore utilità e diffusione, più o meno universalmente mappati sui codepoint del gruppo 1, insieme a molti altri di uso locale che ciascun dominio assegnerà ai codepoint appartenenti ai gruppi 2 e 3. Infatti, il DiffServ Working Group, pur incentivando queste consuetudini, rifiuta ogni standardizzazione nella mappatura PHB-codepoint continuando a parlare solo di valori raccomandati, allo scopo di garantire la massima flessibilità.

3.2.1 Il default PHB

In tutti i nodi DiffServ compatibili deve essere disponibile un PHB di default e deve coincidere con il normale trattamento best effort che oggi caratterizza le reti IP. In assenza di accordi con i clienti o con i domini adiacenti, si assume che tutti i pacchetti in transito appartengono a questo aggregato, come anche quelli marcati con un codepoint

non riconosciuto alla frontiera di attraversamento del dominio.

Per implementare questo aggregato bisogna scegliere una disciplina di scheduling che invii i pacchetti di questo tipo ogni volta che i link di uscita non siano impegnati da pacchetti appartenenti ad altri PHB; diventa, però, necessario prevedere dei meccanismi di controllo per far sì che all'aggregato sia garantita una certa porzione di banda minima. In questo modo si garantisce alle sorgenti non DiffServ compatibili la possibilità di continuare ad usare la rete come al solito, se pur con risorse limitate.

Per il default PHB viene raccomandato il codepoint "000000", anche se un dominio può mappare il comportamento di default su un qualunque insieme di codepoint, con il vincolo che ai pacchetti marcati con DSCP "000000" deve essere assegnato il PHB di default. Così viene assicurata la completa compatibilità con le applicazioni tradizionali, le quali inviano i pacchetti IP con precedence uguale a "000" e TOS pari a "000". Bisogna anche notare che questo codepoint risulta essere quello con valore più basso tra i Class Selector Codepoint. I pacchetti indicati come di default possono poi essere promossi ad altri PHB per mezzo di una operazione di remarking, in modo da avere la possibilità di fornire servizi differenziati ad applicazioni non DiffServ compatibili.

3.2.2 L'Expedited Forwarding PHB group

L'*EF-PHB* (*Expedited Forwarding Per-Hop Behavior*) rappresenta il blocco costitutivo fondamentale per la costruzione di servizi end-to-end di tipo *premium*, che sono caratterizzati da basso ritardo, basso jitter, banda garantita e bassa probabilità di perdita dei pacchetti.

Un servizio di tipo *premium* offre garanzie di tipo deterministico: cliente e fornitore negoziano una SLA in cui è indicato un tasso massimo che il cliente si impegna a non superare e, fintanto che l'intensità del traffico offerto si mantiene inferiore a questo tasso, il fornitore assicura che la banda garantita sarà disponibile e che ritardo, jitter e probabilità di perdita saranno inferiori ad altrettanti limiti prefissati.

Le principali cause di ritardo in una rete a pacchetto sono il ritardo fisso di propagazione e il ritardo dovuto alle code negli switches e router che il traffico incontra nell'attraversamento della rete. Poiché il ritardo di propagazione è una proprietà fissata dalla topologia, per garantire ad un aggregato basso ritardo, jitter e probabilità di perdita

occorre che questo incontri lungo il suo percorso soltanto code di dimensioni molto ridotte ed unicamente quando è necessario, in modo da minimizzare il ritardo di coda accumulato. L'obiettivo dell'EF PHB è, dunque, quello di fornire un PHB nel quale i pacchetti marcati in modo appropriato incontrano solitamente code molto ridotte se non assenti del tutto; in questo caso si è anche sicuri che lo scarto dei pacchetti sia ridotto al minimo. Il codepoint raccomandato per questo servizio è pari a "101110".

L'EF PHB può essere implementato in un nodo ricorrendo ad alcuni degli algoritmi di scheduling più noti. Ad esempio, un gruppo di code servite da uno scheduler a priorità stretta può fornire il comportamento richiesto, qualora il traffico EF sia assegnato alla coda con priorità più alta o, comunque, ad una coda tale che nessun'altra possa ritardarne l'emissione di un pacchetto per un tempo superiore al transito di una ***MTU*** (maximum transfer unit) al bitrate configurato. Un'altra possibilità è quella di utilizzare uno scheduler di tipo Weighted Round Robin nel quale la porzione di banda assegnata alla coda EF sia proprio uguale al tasso configurato.

I servizi di tipo EF sono adatti alle applicazioni real-time come videoconferenza, voice-over-IP, video live broadcast, che oggi sono considerate trainanti per lo sviluppo futuro di Internet. Tali applicazioni, originariamente supportate da reti a circuito, sono quelle che meno si adattano al trattamento dei pacchetti di tipo best effort offerto dalle reti IP, il quale non consente di prevedere né tanto meno di manipolare in alcun modo parametri importanti come il ritardo medio e il jitter a cui andranno incontro i pacchetti, anche in presenza di link molto sovradimensionati.

3.2.3 L'Assured Forwarding PHB group

Nel paragrafo precedente si è visto come il PHB Expedited Forwarding può essere usato per fornire agli utenti garanzie deterministiche sul livello di servizio offerto al prezzo, oltre che di un elevato costo monetario, dell'impossibilità di inviare traffico ad una intensità anche di poco superiore a quella stabilita nell'SLS. In molti casi, tuttavia, è preferibile un servizio che offra garanzie di tipo statistico anziché deterministico, consentendo di inviare traffico oltre i limiti definiti dal profilo contrattuale, in modo da sfruttare eventuali situazioni di basso carico della rete. Servizi di questo genere sono desiderabili sia dagli utenti, che acquistano una QoS minima ma senza vincoli su quella massima, sia dai

fornitori, che possono in questo modo sfruttare la banda lasciata libera nei periodi di basso carico della rete.

L'*AF PHB Group* (*Assured Forwarding*) comprende alcuni PHB in grado di fornire ai domini DiffServ il supporto per implementare dei servizi caratterizzati da diversi livelli di garanzia di recapito dei pacchetti. Nel documento vengono definite quattro classi di servizio, chiamate classi AF, alle quali ogni nodo DiffServ compatibile fa corrispondere quattro diversi livelli di garanzia in termini di banda e buffer allocati.

L'implementazione dell'AF PHB non è necessaria, come nel caso dell'EF PHB, affinché un nodo possa dirsi DiffServ compatibile. Tuttavia, nel caso in cui sia presente in un dominio, i nodi che lo compongono devono rispettare i seguenti vincoli:

- ogni implementazione del gruppo dovrebbe prevedere tutte e quattro le classi di servizio e trattare separatamente i pacchetti appartenenti a ciascuna classe.
- in ogni nodo deve essere riservato ad ogni classe un quantitativo minimo configurabile di risorse, in termini di buffer e banda sui link di uscita; inoltre, l'algoritmo di scheduling deve garantire a ciascuna classe il tasso minimo di servizio in ogni situazione, sia di breve che di lungo periodo.
- le classi possono essere configurate per ricevere un tasso di servizio superiore a quello minimo qualora la banda necessaria sia disponibile.

La caratteristica di consentire contemporaneamente sia un livello minimo di servizio, sia la possibilità di sfruttare la banda in eccesso, viene realizzata introducendo in ogni classe AF tre diversi livelli di priorità di scarto dei pacchetti. Questi livelli corrispondono ad altrettanti valori del DSCP, indicati con la dicitura ***AFxy***, dove x corrisponde alla classe di servizio (da 1 a 4) ed y alla priorità di scarto (da 1 a 3). La priorità di scarto fornisce una misura dell'importanza del pacchetto all'interno della classe, senza ricorrere ad espliciti messaggi di segnalazione, e deve essere trattata dai nodi come segue:

- all'interno della stessa classe AF, un nodo non deve inoltrare i pacchetti marcati con probabilità di scarto p rispetto a quelli marcati con probabilità q se $p < q$: nel caso si verifichi una congestione lo scarto dei pacchetti deve procedere in ordine di

priorità, partendo da quelli a priorità più alta.

- un nodo deve accettare tutte e tre le priorità di scarto e, all'interno di ogni classe, queste devono portare ad almeno due diversi livelli di probabilità di perdita: in questo caso i codepoint AFx2 e AFx3 devono corrispondere allo stesso livello (il più alto).
- l'implementazione di questo PHB non deve mai causare il riordino dei pacchetti di un microflusso appartenenti alla stessa classe AF, indipendentemente dalla priorità di scarto.

L'Assured Forwarding, quindi, è in grado di tollerare le congestioni nel breve periodo, in modo da consentire alle classi di sfruttare tutta la banda disponibile, accettando i burst di traffico e reagendo in maniera graduale ad esse. Naturalmente ciò comporta l'impossibilità di garantire parametri istantanei, come un ritardo inferiore ad una soglia, ma consente di soddisfare dei vincoli statistici, come il throughput medio nel lungo periodo. Questo comportamento viene realizzato implementando le seguenti specifiche:

- le congestioni di lungo termine all'interno di ciascuna classe devono essere minimizzate, consentendo soltanto quelle di breve termine dovute ai picchi di traffico; è necessario, perciò, un algoritmo di gestione attiva delle code come può essere il RED.
- la risposta alle congestioni di lungo termine deve consistere nello scarto dei pacchetti, mentre quelle a breve devono essere gestite mediante accodamento; è dunque necessaria una funzione di filtraggio che calcoli un indice medio di congestione nel lungo periodo in base alla dimensione istantanea delle code: lo scarto deve basarsi sul livello di congestione medio e non su quello istantaneo.
- l'algoritmo di scarto deve essere insensibile alle caratteristiche di breve termine dei microflussi che compongono ciascuna classe: flussi con diversi tipologie di burst, ma con lo stesso bitrate medio, devono essere sottoposti alla stessa probabilità di perdita; bisogna, cioè, introdurre degli elementi aleatori nella funzione di scarto.
- l'algoritmo di scarto deve trattare allo stesso modo tutti i pacchetti di una stessa classe e del medesimo livello di priorità, in modo che, all'interno di essa, i flussi che

occupano una frazione maggiore di banda siano penalizzati proporzionalmente di più rispetto a quelli che ne occupano meno.

- poiché il tasso di scarto viene usato come segnale di feedback per le applicazioni, senza il bisogno di segnalazione esplicita, la transizione da un basso ad un alto tasso di scarto deve essere graduale, in modo da permettere al sistema di raggiungere un punto di lavoro stabile.

Un servizio end-to-end a banda garantita può, in base a quanto detto, essere costruito destinando i pacchetti ad una determinata classe AF indicata nell'SLS stipulato tra cliente e fornitore; esso deve avere un profilo di servizio espresso nel TCS mediante, ad esempio, i parametri di un Token Bucket, che servirà a configurare all'interno del dominio del fornitore le risorse da destinare alla classe. Alla frontiera, il traffico emesso dal cliente sarà inviato ad un meter configurato con il profilo contrattato ed i pacchetti eccedenti, anziché essere scartati, saranno marcati con una probabilità di scarto maggiore, rinviando la decisione sulla loro sorte ai nodi interni. Qualora la rete sia sufficientemente scarica i nodi interni non avranno necessità di scartare pacchetti, e tutti potranno essere recapitati sfruttando così al meglio la banda disponibile. Nel caso in cui, invece, la classe sia congestionata, i nodi interni potranno scartare i pacchetti basandosi soltanto sul livello di priorità riportato nel DSCP. Al contrario, i clienti che emettono traffico entro il proprio profilo non sono penalizzati, poiché le congestioni nel lungo periodo possono essere causate soltanto da pacchetti che altri hanno inviato oltre le specifiche e pertanto marcati con priorità di scarto media o alta.

Nella figura seguente sono riportati i dodici codepoint raccomandati per l'*Assured Forwarding PHB group*:

Probabilità di scarto	**Classe 1**	**Classe 2**	**Classe 3**	**Classe 4**
Bassa	**001010**	**010010**	**011010**	**100010**
Media	**001100**	**010100**	**011100**	**100100**
Alta	**001110**	**010110**	**011110**	**100110**

Figura 4.5: Codepoint raccomandati per l'AF-PHB group.

Questi codepoint fanno parte dello spazio destinato alle azioni standard e non interferiscono con il codepoint destinato al best effort, né con quello raccomandato per l'EF-PHB e né con i Class Selector PHB.

3.3 I Per-Domain Behavior

Una volta standardizzate le *politiche di forwarding* da applicare in ogni nodo della rete, ovvero i *PHB, il Working Group* ha focalizzato l'attenzione sui problemi connessi con il tentativo di garantire la qualità del servizio da estremo ad estremo di un dominio a servizi differenziati.

Nell'architettura DiffServ i pacchetti vengono classificati alle frontiere del dominio, raggruppati in base al trattamento che ricevono nei nodi e resi conformi a determinate regole, tramite operazioni di condizionamento del traffico, che riflettono accordi amministrativi tra cliente e fornitore. La scalabilità di questo approccio è garantita solamente se le definizioni dei PHB danno luogo ad una qualche forma di invarianza delle proprietà con l'aggregazione del traffico: un PHB è, infatti, lo stesso in tutti i nodi, ma l'insieme dei pacchetti che ciascun nodo tratta secondo quel PHB è in generale diverso. Un PHB, dunque, dovrebbe essere definito in modo che le caratteristiche su cui si basa non dipendano dal volume di traffico dell'aggregato ad esso destinato che entra in un nodo, né dal particolare percorso che i pacchetti che compongono l'aggregato seguono all'interno del dominio.

La possibilità di specificare con l'esattezza come un particolare aggregato di traffico, individuato solo dal codepoint, viene modificato all'attraversamento di un dominio è un aspetto cruciale per la costruzione della QoS. Solo creando e quantificando dei comportamenti, che rimangano invariati quando gli aggregati confluiscono o si dividono, sarà possibile estendere le garanzie di QoS a tutta la rete.

Il problema della scalabilità ad un livello superiore dei comportamenti descritti dai PHB a livello di singolo nodo, viene affrontato dall'IETF con la definizione del *Per-Domain Behavior* o *PDB*, che formalizza la descrizione delle garanzie di QoS fornite ad un aggregato nell'attraversamento di un singolo dominio.

Le proprietà di un PDB costruito su un particolare PHB sono scalabili se non risentono delle modifiche, intese come aggregazioni e suddivisioni, che il traffico subisce

nell'attraversare il dominio. Se, al contrario, esistono dei limiti alla validità delle caratteristiche di un PDB, ciò si traduce necessariamente in vincoli sulla topologia o sulle dimensioni del dominio che lo utilizzi. Naturalmente, sono desiderabili dei PDB invarianti rispetto alle dimensioni del dominio: essi possono essere dotati di una semplice relazione con le dimensioni, o tali che le proprietà possano essere garantite applicando più volte le regole di condizionamento del traffico (ad esempio inserendo delle frontiere fittizie all'interno di un unico dominio).

Non bisogna confondere il PDB con la descrizione del servizio contenuta in un SLS: il primo comprende, infatti, dettagli tecnici, come i PHB necessari e le configurazioni delle frontiere, che non compaiono nel secondo. Il PDB deve essere visto, dunque, come un blocco per costruire un servizio normalmente invisibile agli utenti che sottoscrivono l'SLS, se non attraverso i suoi effetti. Uno stesso servizio potrebbe essere implementato, ad esempio, per mezzo di PDB diversi, anche non standardizzati e sviluppati appositamente per un particolare dominio.

3.3.1 Il Best Effort PDB

Il ***Best Effort Per-Domain Behavior*** o ***BE-PDB*** è utilizzato per spedire il normale traffico Internet attraverso una rete DiffServ, conservando, per quanto possibile, le aspettative di consegna dei pacchetti che caratterizzano i domini non DiffServ compatibili.

La definizione non contiene regole per condizionare tassi, burst, ritardi e perdita dei pacchetti, a parte quelle implicite date dalla limitata capacità dei link di ingresso. La frontiera del dominio dovrà assicurarsi che i pacchetti destinati a questo PDB siano marcati con il PHB di default. L'unica limitazione consiste nel non permettere che l'aggregato subisca starvation, ovvero rimanga completamente privo di servizio, quando le risorse sono disponibili; inoltre gli elementi di rete dovrebbero essere configurati per permettere all'aggregato di utilizzare eventuali risorse disponibili.

Sebbene alcune reti possano decidere di limitare il ritardo e la perdita dei pacchetti appartenenti a questo PDB, per motivi inerenti alle rispettive politiche interne di gestione, tali attributi non sono parte integrante della definizione.

Capitolo 4 L'ambiente di simulazione

4.1 Il modulo DiffServ

Il modulo DiffServ è stato sviluppato dalla Nortel Networks ed è stato aggiunto ad NS2 il 2 Novembre 2000. Le funzionalità DiffServ sono racchiuse in cinque moduli che sono delle sottoclassi della classe Queue. Poiché le istanze di questa classe sono sviluppate nell'elemento link, essa modellizza l'interfaccia di uscita di un router: è possibile, dunque, simulare le funzionalità DiffServ solo nelle interfacce di output. I cinque moduli, che a loro volta sono delle classi, sono:

- la classe dsREDQueue che modellizza le funzionalità di base di un router DiffServ, come le code multiple con le quali implementare le diverse classi di servizio; questa è la classe genitore di tutte le altre, e contiene tutti i parametri che sono comuni alle altre.
- la classe redQueue con la quale gestire i meccanismi di controllo della congestione di ogni singola coda.
- la classe CoreQueue che modellizza le funzionalità di un core router.
- la classe EdgeQueue che modellizza le funzionalità di un edge router.
- la classe PolicyClassifier, che è sottoclasse della precedente, con la quale si gestiscono le funzionalità di metering e marking dei flussi.

La struttura delle code fornita dalla classe dsREDQueue consiste in quattro code fisiche, ognuna delle quali contiene tre code RED “virtuali” che si riferiscono ai diversi livelli di precedenza. Ogni coda fisica corrisponde ad una classe di traffico e ogni combinazione di una coda e di un livello di precedenza è associata con un codepoint: i pacchetti vengono accodati in base al codepoint con il quale sono stati marcati.

La classe *dsREDQueue* contiene anche una struttura dati chiamata ***PHB Table***, che contiene un vettore con tre campi:

- Codepoint
- Class (Physical Queue)
- Precedence (Virtual Queue)

La *PHB Table* permette ad edge e core router di mappare i codepoint con le varie code e i vari livelli di precedenza.

Per esempio vediamo come se fa la configurazione dei parametri DS RED da un router edge, o di frontiera, a uno core, o interno:

```
$dsredq set numQueues 1
$dsredq setNumPrec 3
```

Con questo codice si crea una coda física con tre code virtuale nel router, come nella seguente figura:

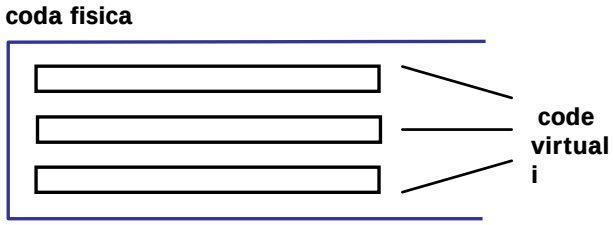


Figura 4.1: Esempio di una coda fisica RED con tre code virtuali.

```
$dsredq setMREDMode WRED 0
$dsredq configQ 0 1 10 20 0.10
```

Con questi comandi scegliamo l'algoritmo di scarto selettivo dei pacchetti nella modalità Weighted RED per ogni coda fisica e i parametri RED per la coda virtuale. In questo esempio la coda virtuale 1 appartenente alla coda fisica 0 scarta un pacchetto con una probabilità del 10% se il contenuto della coda stessa varia tra un valore minimo di 10 pacchetti e uno massimo di 20 pacchetti. Per le code DropTail sono richiesti soltanto i primi tre parametri del comando poiché queste non hanno la nozione di livello di precedenza. Un altro parametro importante per la configurazione dell'algoritmo di scarto selettivo è la dimensione media dei pacchetti, da specificare in byte, che può essere settato con il seguente comando:

```
$dsredq meanPktSize 1500
```

La classe dsREDQueue mette anche a disposizione dei comandi per la stampa della PHB Table e delle statistiche relative alle classi di servizio simulate:

```
$dsredq printPHBTable
$dsredq printStats
```

Con il seguente codice configuriamo un edge router:

```
$dsredq addPolicyEntry [$n1 id] [$n2 id] TokenBucket 10 1000000 10000
$dsredq addPolicerEntry TokenBucket 10 11
```

Con il comando addPolicyEntry viene aggiunta una riga alla Policy Table, che è un vettore dove sono memorizzate le informazioni necessarie a caratterizzare il traffico, come il nodo sorgente e quello di destinazione del flusso, il tipo di meter e i parametri con cui viene misurato, il codepoint iniziale assegnato e altre informazioni di stato; nell'esempio riportato al flusso con sorgente nel nodo n1 e destinazione nel nodo n2 viene assegnato il codepoint 10 e viene misurato con un Token Bucket con CIR (Committed Information Rate, specificato in bit per secondo) pari a 1000000 e un CBS (Committed Burst Size, specificato in byte) pari a 10000. Token Bucket, che usa un CIR e un CBS e due livelli di priorità.

Con il comando addPolicerEntry, invece, viene aggiunta una riga nell'array Policer Table che contiene la mappa dei codepoint iniziali e di quelli da assegnare ai pacchetti quando eccedono i profili stabiliti dai rispettivi meter: nell'esempio riportato i pacchetti con cedepoint 10 che eccedono il profilo del Token Bucket vengono rimarcati con il codepoint 11.

4.2 Lo scenario di simulazione

In questo paragrafo viene descritto lo scenario utilizzato per verificare, mediante simulazione in ambiente NS2, gli efetti del ritardo di coda e la rispondenza alle aspettative dell'architettura Differentiated Services per garantire la qualtità del servizio in una rete UMTS con core network completamente IP. Servizi con requisiti di qualità diversi verranno fatti coesistere in una medesima rete e concorrere per l'assegnazione di risorse scarse, come la banda di un link condiviso.

In particolare si è voluto mettere in evidenza come possa essere garantito un servizio con requisiti di qualità molto stringenti, ad esempio una videoconferenza a discapito di applicazioni insensibili al ritardo, ad esempio la lettura della posta elettronica. In questo lavoro di caratterizzazione si tenterà di garantire le esigenze dei servizi con requisiti più stringenti utilizzando le funzionalità di forwarding e di condizionamento messe a disposizione dal modulo DiffServ implementato nella versione 2.29 di NS2, descritto nel paragrafo precedente. L'effettiva utilità dell'architettura DiffServ ai fini della QoS verrà verificata valutando il *througput* della rete simulata e confrontandolo con quello ottenuto ricorrendo unicamente all'architettura best effort tradizionale.

4.2.1 Topologia della rete simulata

La topologia della rete utilizzata per questo scenario di simulazione è mostrata nella figura 1 ed è composta da diciassette nodi, di cui tre nodi router e dieci nodi host. Nella figura è messo in evidenza il dominio DiffServ che è composto da due nodi di frontiera (*edge router*), contrassegnati con i numeri 1 e 2 , e dal nodo interno contrassegnato con il numero 0, che è il core router del dominio.

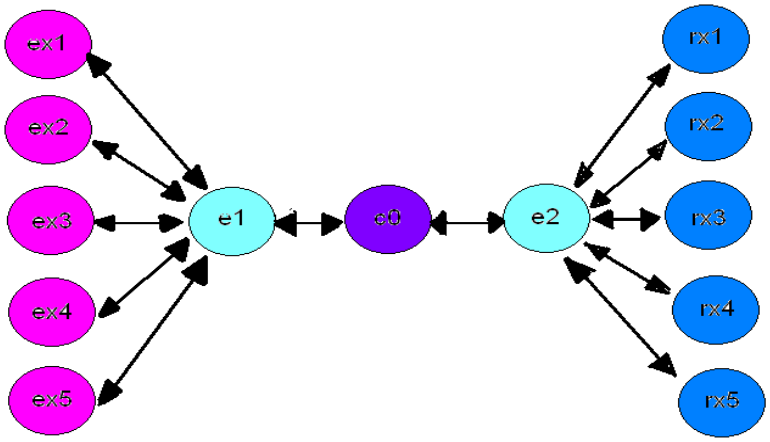


Figura 4.2: Schema della rete

In questa topologia è possibile ritrovare:

- la core network UMTS, coincidente con il dominio DiffServ (nodi 0, 1 e 2);
- la rete d'accesso radio UMTS, costituita dai nodi 8, 9, 10, 11 e 12 collegati al router di frontiera 2;
- le sorgenti, situate nella rete Internet, dunque esterne alla core network, rappresentate dai nodi 3, 4, 5, 6, 7 e collegati al router di frontiera 1.

I nodi router sono connessi tra di loro con dei collegamenti simplex-link e tutti i nodi host, sono connessi tra di loro con dei collegamenti duplex-link, in modo che gli acknowledgement delle sorgenti che usano il TCP come agente di trasporto abbiano un percorso di ritorno.

4.2.2 Le sorgenti di traffico

Nelle figure 5.2 viene mostrato il percorso dei flussi di traffico presenti nella rete. In base al percorso dei flussi e a come sono stati dimensionati i collegamenti tra i router, è chiaro che il *bottle-neck* (collo di bottiglia) dell'intera rete è il link tra il router 1 e il core 0. Saranno, quindi, le code di questi due router a introdurre i ritardi e a causare la perdita di pacchetti in condizioni di sovraccarico della rete.

Analizziamo adesso più in dettaglio le caratteristiche delle sorgenti e dei flussi di

traffico scelti per questa simulazione.

Tra i nodi 3 e 8 abbiamo un traffico di tipo Premium da *PHB Expedited Forwarding* in ambiente Diffserv, queste traffico potrebbe essere ben rappresentato da una teleconferenza. Nella simulazione il flusso di teleconferenza è generato da un'applicazione CBR (*Constant Bit Rate*) che produce pacchetti della grandezza di un frame, i quali usano l'agente di trasporto *RTP* che implementa l'omonimo protocollo di trasporto per applicazioni real time.

Oltre alla classe EF usata per la teleconferenza ed a quella BE (*Best Effort*), che deve comunque esserci per garantire la compatibilità all'indietro, nella simulazione sono presenti tre classi di tipo *Assured Forwarding*, contrassegnate come AF1, AF2 ed AF3, ciascuna associata ad un diverso tipo di traffico. Con un gruppo di classi di questo tipo può essere implementato un modello di servizio, conosciuto come *Olympic service*, che consiste in tre classi a livello di priorità decrescente indicate come gold, silver e bronze.

Alla classe AF1 gold (colore rosso) è associato un flusso di dati che supponiamo essere uno streaming multimediale. I pacchetti di questo flusso, ciascuno di 1000 byte, vengono generati da una sorgente CBR ed usano come protocollo di trasporto l'UDP, generato dall'omonimo agente di trasporto del simulatore.

Alla classe AF2 silver (colore verde) è stato associato il servizio per il trasferimento di file di grosse dimensioni. I pacchetti vengono generarti dall'applicazione FTP messa a disposizione dal simulatore, che può essere configurata soltanto come "on" o "off", ed usano come agente di trasporto il FullTCP, che implementa la versione Reno del TCP, in modo da avere una garanzia sulla effettiva consegna dei dati. Come dimensione dei pacchetti è stata scelta quella di default del simulatore, che consiste in pacchetti di 576 byte di cui 536 di dati e 40 di header TCP.

Alla classe AF3 bronze (colore giallo) è stato associato il servizio di web server. Il flusso dati prodotto da un servizio di questo tipo è stato simulato mediante il generatore di traffico Pareto, che usa una distribuzione paretiana, ed è trasportato anch'esso da un agente FullTCP. Come parametri di configurazione dell'applicazione sono stati scelti 150 ms per il burst_time_ (tempo medio di "on" della sorgente), 50 ms per l'idle_time_ (tempo medio di "off" della sorgente) e 1,5 per lo shape_ ("forma" della distribuzione). Network Simulator mette a disposizione due generatori di traffico non costante, uno con distribuzione di tipo esponenziale e l'altro con distribuzione paretiana. Nelle simulazioni effettuate è stato scelto il secondo tipo perché è stato dimostrato che la distribuzione

paretiana approssima meglio di quella esponenziale il traffico presente in Internet.

Alla classe BE, infine, è stato associato il traffico di web browsing e tutto il traffico generato degli acknowledgement delle connessioni TCP. Questo traffico è generato con un'applicazione Pareto con parametri 50 ms per il burst_time_, 5 ms per l'idle_time_ e 1,5 per lo shape_, e viene trasportato da un agente FullTCP.

4.2.3 Configurazione dei router

Per la configurazione dei router, bisogna naturalmente distinguere i due casi di modello di servizio, best effort e DiffServ, che sono stati implementati e messi a confronto nelle simulazioni effettuate.

Il caso best effort è piuttosto semplice poiché tutti router sono stati configurati con code FIFO e algoritmo di scarto selettivo Drop Tail: raggiunto il limite della coda, che è stato lasciato al valore di default di 50 pacchetti, tutti i pacchetti in arrivo vengono scartati. In base alla topologia e alla configurazione dei link scelti, si avranno pacchetti scartati soltanto nei router 1 e 0. Da notare che Network Simulator permette di settare il limite delle code soltanto in funzione del numero dei pacchetti e non, più realisticamente, in funzione del numero di byte del buffer.

La scelta dei parametri di condizionamento del traffico e di configurazione dei router nel caso DiffServ è molto più complessa per l'alto numero di variabili in gioco e per l'assenza in letteratura di specifiche per la loro configurazione ottimale.

Per quanto riguarda il condizionamento del traffico si è scelto di misurare tutti i flussi con dei Token Bucket in modo da distinguere, marcando i pacchetti con DSCP diversi, due livelli di precedenza per i pacchetti considerati "*in profile*" o "*out of profile*" di ogni flusso. Il frammento di codice riportato di seguito, come esempio, mostra la configurazione dei parametri di condizionamento per il flusso della teleconferenza.

```
$queue_e1_c0 addPolicyEntry [$ex1 id] [$rx1 id]  TokenBucket 36 $rrx_1 1536
$queue_e1_c0 addPolicerEntry TokenBucket 36 38
```

```
$queue_e1_c0 addPHBEntry 36 0 0
$queue_e1_c0 addPHBEntry 38 0 1
```

I pacchetti del flusso generato tra i nodi 3 (ex1) e 8 (rx1) sono marcati con DSCP 36 e sono misurati con un Token Bucket con CIR, pari alla velocità nominale del flusso, e CBS di un pacchetto e mezzo; i pacchetti considerati in profile vengono assegnati alla coda fisica 0 - coda virtuale 0, mentre i pacchetti out of profile vengono rimarcati con DSCP 38 e sono assegnati alla coda fisica 0 - coda virtuale 1.

Il numero di code fisiche presenti nei router è stato scelto in base al numero di flussi gestiti da ognuno di essi, mentre il numero di code virtuali, e quindi di livelli di precedenza, per ogni coda fisica è fissato per tutte le code. Per lo scarto selettivo dei pacchetti è stato scelto WRED per tutte le code, a titolo di esempio si riporta la configurazione per il buffer che gestisce il flusso di teleconferenza.

```
$queue_e1_c0 setMREDMode WRED 0
$queue_e1_c0 configQ 0 0 5 10 0.1
$queue_e1_c0 configQ 0 1 0 0 1.0
```

La coda virtuale 0 dei pacchetti in profile ha una soglia minima di 5 pacchetti e quella massima pari a 10, con una probabilità di scarto di 0,1 poiché questa coda dovrebbe rimanere sempre quasi vuota per garantire basso ritardo e basso jitter; la coda virtuale 1 dei pacchetti fuori profilo, invece, ha le due soglie coincidenti pari a 0 pacchetti con una probabilità di scarto di 1,0, in modo da scartare automaticamente tutti i pacchetti marcati con DSCP 38 e, quindi, considerati non conformi alle specifiche.

La scelta dell'algoritmo di scheduling per le code è stata fatta seguendo le indicazioni trovate in letteratura. Lo schedulatore prioritario (`$queue_e1_c0 setSchedularMode PRI` nel file di configurazione) è quello più indicato per garantire al servizio premium un basso ritardo di propagazione e un jitter contenuto. Per evitare il blocco dei flussi con priorità più bassa è stata fatta un'assegnazione statica della banda residua, indicando per ogni coda il rate assegnato grazie al comando `addQueueRate`.

4.2.4 Le uscite verso l'interfaccia radio UMTS

Analizziamo adesso più in dettaglio le caratteristiche delle uscite verso ~~internet~~ l'interfaccia radio UTRAN e del suo comportamento quando riceve un paccheto.

L'interfaccia radio è simulata come un collegamento tra due nodi con accodamento DropTail, aggiungendo, però, un semplice modello d'errore markoviano per simulare la rumorosità del canale. Inoltre il ritardo sulla linea ha due componenti. La prima

componente tiene conto dei ritardi di propagazione e di elaborazione de è costante, mentre la seconda tiene conto dei ritardi di trasmissione introdotti dal meccanismo di ritrasmissione (ARQ, *Automatic Repeat reQuest*) a livello RLC (*Radio Link Control*).

Come si vede dal codice seguente la configurazione del collegamento tra il router di frontiera, e2, e il nodo mobile, rx1, non è diversa dai casi precedenti, a parte la capacità di canale ridotta a 64 kbit/s. La variabile delayRLC_ contiene la componente fissa del ritardo. La componente di ritardo variabile è introdotta con il comando dynalink associato al collegamento tra i nodi e2 e rx1, come si vede sulla seconda riga. In questo modo si specifica che su quel particolare collegamento bisogna tenere conto degli errori introdotti dal canale e del tempo necessario alla ritrasmissione.

```
$ns duplex-link $rx1 $e2 64kb $delayRLC_ DropTail
$ns dynalink $e2 $rx1 $cap_
```

Le modifiche effettuate non riguardano solo l'interfaccia TCL del simulatore, ma si spingono fino al codice C++. Il comando TCL dynalink fa riferimento a una nuova classe, Dynalink, aggiunta al compilatore C++ per essere usata dal metodo *recv(Packet* p, Handler* h)* della classe *Delay*. Se il paccheto ricevuto dal *router di frontiera* 2 ha come destinatario un nodo con un collegamento dinamico le interfacce radio verso i nodi 8, 9, 10 ,11 e 12 si divide l'informazione in diverse parti dette PDU (*Payload Data Unit*) da 40 bytes. L'invio delle PDU è regolato da una finestra (pari al numero di PDU che si possono trasmettere in un *time slot*) e utilizza dei riscontri o *acknowledgement*, sia positivi che negativi, per comunicare alla sorgente il risultato della trasmissione. Se qualche PDU è persa nella trasmissione, oppure, non si riceve il relativo riscontro, il router reinvia questa PDU persa nel prossimo *time slot*, occupando la posizione precedente. Si procede in questo modo fino a che tutte le PDU del pacchetto sono ricevute correttamente dal destinatario. Se un pacchetto non ocupa tutte le posizioni di un *time slot* per completare la sua trasmissione, le posizione vuote che mancanno sono occupate con le PDU delle seguente pacchetto da trasmitere. Nella seguente figura si espone il funzionamento:

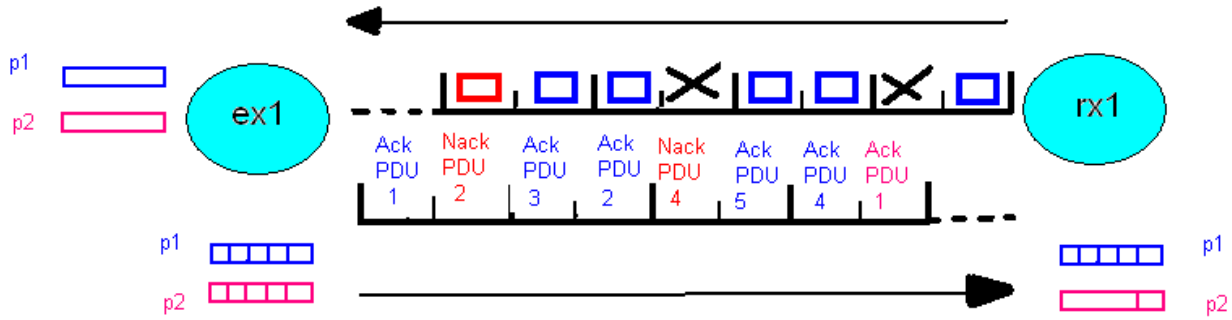


Figura 4.3: Funcionamiento dal core router

Per sapere valore dalla fiestra basta dividere la capacitá del canale per la dimensione della PDU (40 bytes), così sappiamo quante PDU possono essere trasmessa nel stesso *time slot*.

Come già accennato prima, è necessario introdurre un modello di Markov per simulare un canale reale in cui esiste la possibilità ~~de~~ che un pacchetto siea perso nella trasmissione. Per il nostro scopo sono sufficienti due stati, i cui valori di probabilità di trovarsi nello stato Good, in quello Bad oppure che la trasmissione sia errata in ciascuno di questi stati sono introdotti con i seguenti comandi:

```
$ns probState $e2 $rx1 $probState_
$ns probDelayGood $e2 $rx1 $probDelayGood_
$ns probDelayBad $e2 $rx1 $probDelayBad_
```

Per fare le simulazione dall seguente capitolo usiamo li valori da probabilità dal articolo [19].

Capitolo 5 Analisi Risultati

5.1 Introduzione all'analisi dei risultati

Prima di analizzare i risultati è necessario specificare il formato dell'uscita fornita dal simulatore. Il risultato della simulazione di NS2 è un file, che prende il nome di *trace file o file di traccia*, il quale riporta in modo schematico tutti gli eventi accaduti durante la simulazione. Il formato è riportato nella tabella seguente

Columna	1	2	3	4	5	6	7	8	9	10	11	12
Inform.	Action	Time	n1	n2	Ptype	Dim	Flag	Fid	Src	Dst	Sn	uid
Esempio	+	0.125	6	0	Tcp	40	-----	5	6.0	11.0	0	0

Tabella 1: Disposizione del trace file

Brevemente il significato di ogni singola colonna è:

1 Azione: indica l'ingresso o l'uscita di un pacchetto da una coda;

2 tempo: istante di tempo in cui accade l'evento specificato nella colonna precedente;

3 e 4, n1 e n2: sono i due nodi che identificano il link su cui transita il pacchetto in quel momento;

5 Ptype: tipo di pacchetto;

6 Dim: dimensione del pacchetto, in byte;

7 Flag: bit di segnalazione utilizzati da TCP

8 Fid: identificatore di flusso, eventualmente assegnato per identificare più comodamente un flusso

9 Src: nodo sorgente

10 Dst: nodo destinazione

11 Sn: numero progressivo assegnato al pacchetto dal protocollo utilzzato per il trasporto

12 uid: identificativo unico assegnato ad ogni pacchetto, diverso dal precedente.

La possibilità di avere informazioni così dettagliate su ogni flusso rende semplice l'elaborazione successiva. E' stato necessario preparare un semplice programma in C per riorganizzare le informazioni precedenti ed estrarre il throughput di ogni flusso, usando la seguente formula:

```
throughput = (dim*8)/(time- timePrima)
```

Dove al numeratore si trova la dimensione in bit del pacchetto (*dim*, infatti, è la dimensione in byte, fornita dalla colonna numero 6) mentre al denominatore c'è il tempo di interarrivo (time è l'istante in cui si calcola il throughput, *timePrima* è l'istante di arrivo del pacchetto precedente*)*

Il monitoraggio di questi parametri è stato effettuato per due scenari con diverse capacità di canale:

Collegamento tra		
Sorgente e Router (Kb)	Router e Router (Kb)	Router e interfaccia radio (Kb)
70	200	64
160	500	144

Tabella 2: Capatità tra le diversi link ne gli scenari da simulazione

Come si può notare la capacità sul link finale, tra il router e il nodo mobile, è quella di un canale UMTS, mentre la sorgente ha una capacità superiore a quella nominale del canale, in modo da avere sempre dei pacchetti da trasmettere nel buffer. La capacità della *core network UMTS* (il link centrale) è minore rispetto alle richieste del canale: si vuole simulare il caso in cui le altre risorse sono già state assegnate a delle comunicazioni non esaminate, ma che hanno la funzione di traffico di disturbo. Il *collo di bottiglia* è stato messo nel dominio DiffServ, quando si effettua la decisione su come dividere le risorse tra le connessioni.

Facciamo due simulazioni: una con lo *scenario DiffServ* e un'altra senza differenziare i servizi.

Nell scenario con DiffServ la tabella dei *Per Hop Behaviour* la seguente:

Ex1 -> Rx1	Premium Service
Ex2 -> Rx2	AF1 (Gold)
Ex3 -> Rx3	AF2 (Silver)
Ex4 -> Rx4	AF3 (Bronze)
Ex5 -> Rx5	Best Effort

Tabella 3: PHB

5.2 Capacità del canale radio a 64 Kb/s

- Con *Diffserv*:

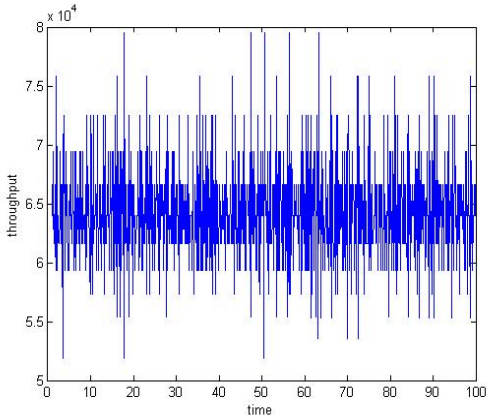


Fig. 1: Collegamento tra e2-rx1 (P. S.)

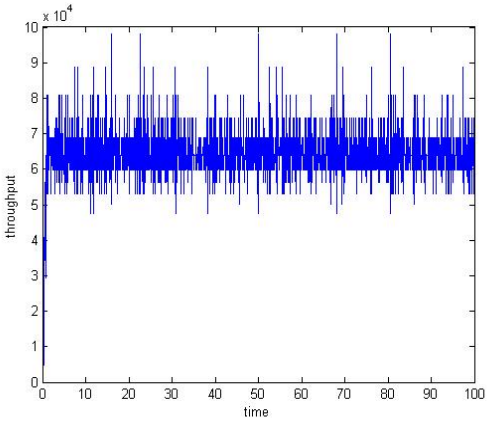


Fig. 2: Collegamento tra e2-rx3 (AF2)

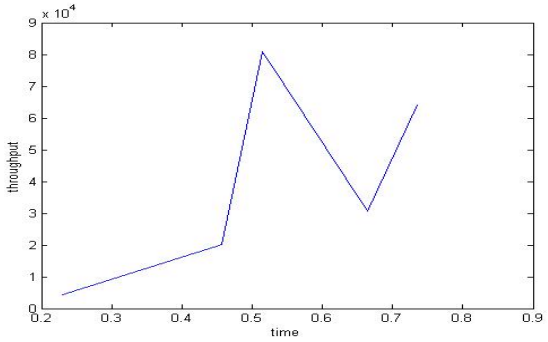


Fig. 3: Collegamento tra e2-rx5 (Best Effort)

- Senza *Diffserv*:

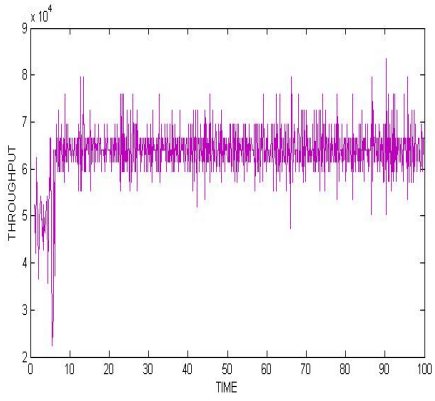


Fig. 4: Collegamento tra e2-rx1

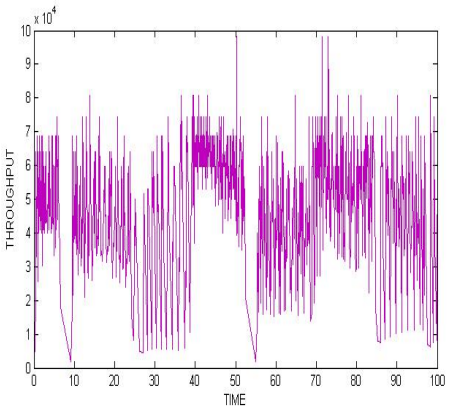


Fig. 5:Collegamento tra e2-rx3

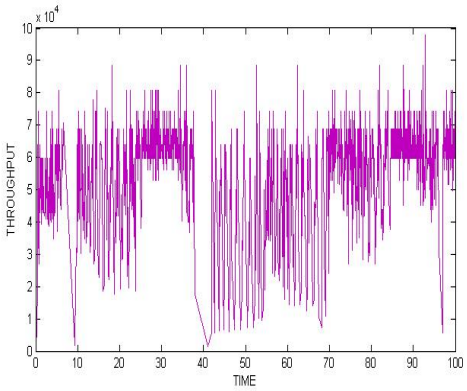


Fig. 6:Collegamento tra e2-rx5

Possiamo vedere che il throughput per il collegamento Premium Service ottiene un significativo miglioramento rispetto agli altri due flussi, portando ad un utilizzo efficiente e continuato del canale radio. Anche il flusso classificato come AF2 ottiene dei miglioramenti, esattamente come ci si poteva aspettare applicando una differenziazione dei servizi. Il guadagno ottenuto dai servizi a cui viene data priorità si ottiene a spese dei servizi best effort, che invece vanno incontro ad una riduzione del *throughput*. In questo ultimo caso la situzione più favorevole è quella che non prevede l'utilizzo di DiffServ perchè tutta la banda a disposizione è divisa in modo uguale tra i collegamenti sul router d'uscita.

5.3 Capacità del canale radio a 144 Kb/s

- Con *Diffserv*:

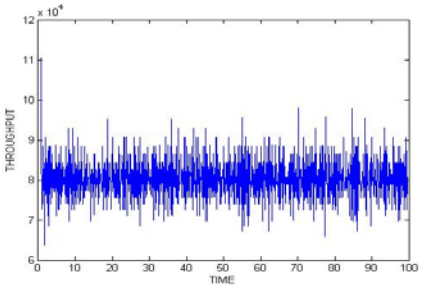


Fig.7: Collegamento tra e2-rx1 (P. S.)

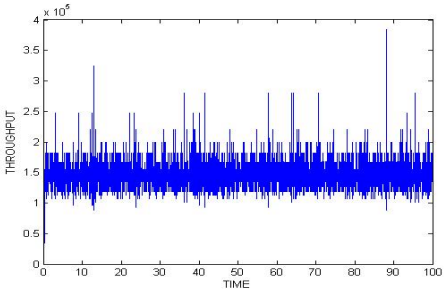


Fig.8: Collegamento tra e2-rx3 (AF2)

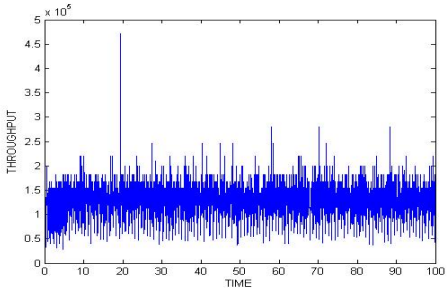


Fig.9: Collegamento tra e2-rx5 (Best Effort)

- Senza *Diffserv*:

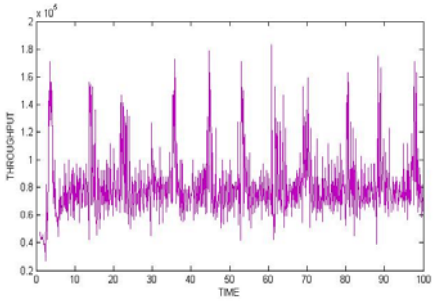


Fig.10: Collegamento tra e2-rx1

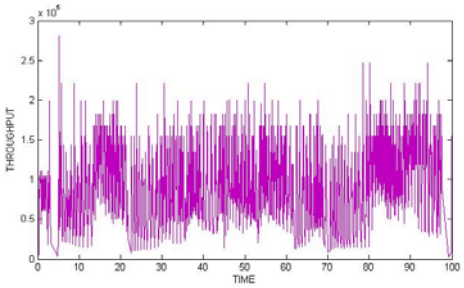


Fig.11:Collegamento tra e2-rx3

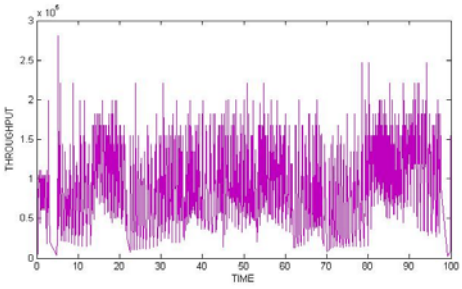


Fig.12:Collegamento tra e2-rx5

Le simulazioni effettuate per un canale radio a 64 kbit/s sono state riproposte per un canale a 144 kbit/s, scalando opportunamente le sorgenti, come visto nella tabella 2. I risultati, come prevedibile sono gli stessi del caso precedente.

5.4 Link con diversa capacità

A queste punto non rimane che valutare la variazione del traffico in ingresso al dominio DiffServ. Si sceglie di usare quattro sorgenti a 140 kb/s e 4 canali radio a 144 kb/s. Il dominio DiffServ prevede due sole classi di servizio, Best Effort e Premium Service, e ha una capacità di 350 kb/s, dunque le risorse sono scarse per garantire tutti i flussi.

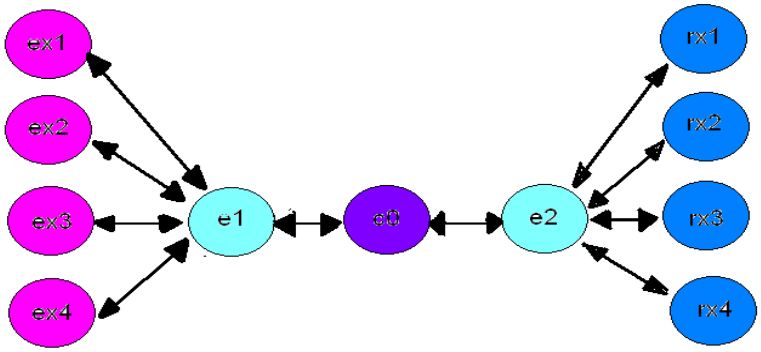


Fig.13: Schema della rete

I flussi sono assegnati alle varie classi secondo lo schema seguente:

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
RX1	Premium Service	Best Effort	Best Effort	Best Effort
RX2	Premium Service	Best Effort	Best Effort	Premium Service
RX3	Premium Service	Premium Service	Premium Service	Premium Service
RX4	Premium Service	Best Effort	Premium Service	Premium Service

- *Scenario 1*:

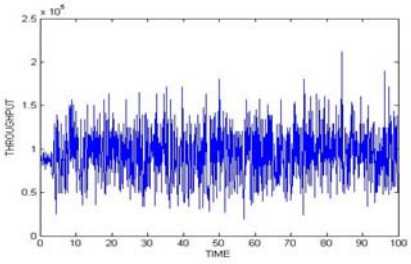


Fig.14: Collegamento tra e2-rx1

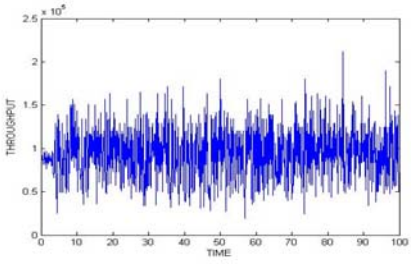


Fig.15: Collegamento tra e2-rx3

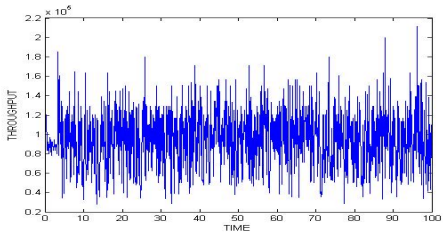


Fig.16: Collegamento tra e2-rx5

- *Scenario 2*:

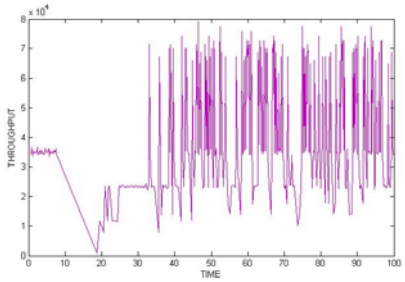


Fig.17: Collegamento tra e2-rx1

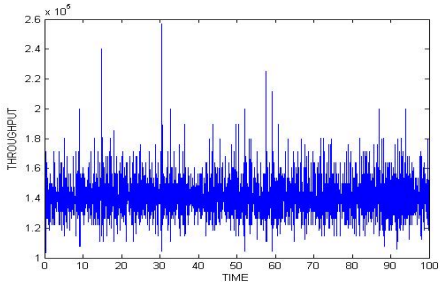


Fig.18: Collegamento tra e2-rx3

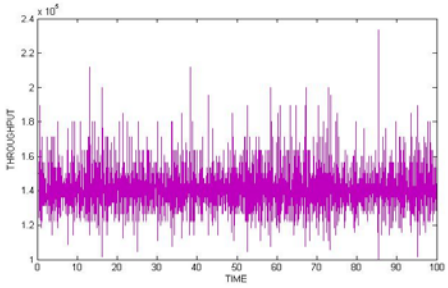


Fig.19: Collegamento tra e2-rx5

- *Scenario 3*:

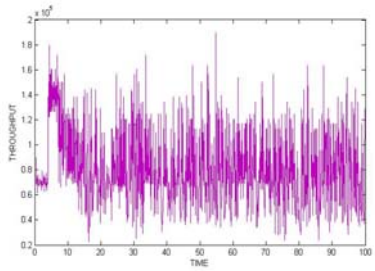


Fig.20: Collegamento tra e2-rx1

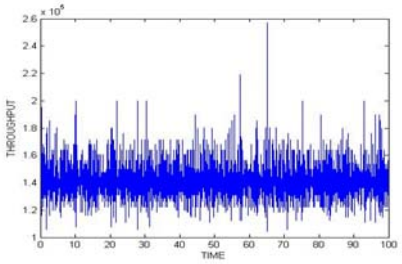


Fig.21: Collegamento tra e2-rx3

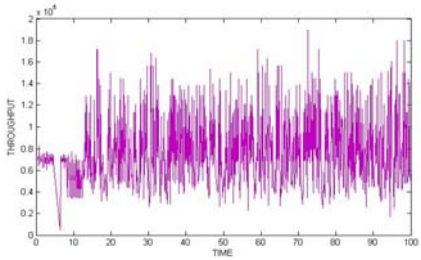


Fig.21: Collegamento tra e2-rx5

- *Scenario 4*:

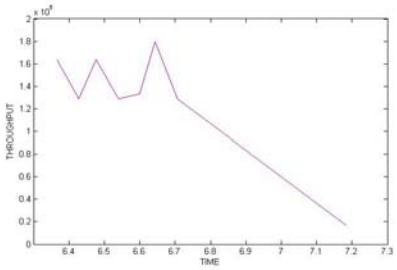


Fig.22: Collegamento tra e2-rx1

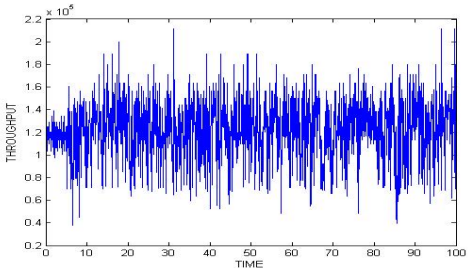


Fig.23: Collegamento tra e2-rx3

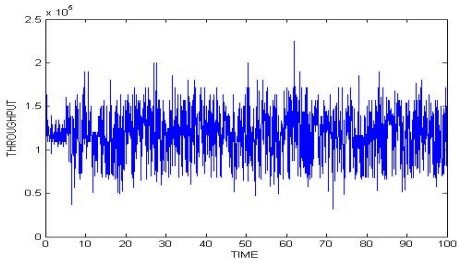


Fig.24: Collegamento tra e2-rx5

Come si può vedere si spazia dal caso in cui si cerca di servire tutti i flussi con la massima priorità al caso in cui si da priorità ad un solo flusso.
Se c'è un solo traffico classificato come Premium Service (scenario 2), le considerazioni sono analoghe a quelle viste nel paragrafo precedente.

Se si cerca di servire tutti i flussi con priorità massima (scenario 1) la differenziazione scompare e si ottiene lo stesso effetto di una rete best effort, cioè la condivisione delle risorse in parti uguali.

Nei casi intermedi (scenari 3 e 4) si nota la differenziazione dei servizi. Maggiore è il peso del traffico best effort rispetto al traffico totale, migliori sono le prestazioni del flusso Premium Service dato che quest'ultimo può utilizzare le risorse tolte al traffico con priorità più bassa.

In conclusione il DiffServ è in grado di introdurre priorità, ma la capacità di fornire garanzie dipende molto da come è composto il traffico in ingresso.

5.5 Diversi probabilità di errore nella trasmissione

Nell scenario da 144 kb proviamo a sappere che cosa sucede quando la probabilità di paccheto svagliato nell stato markoviano Bad é modificata. Dunque studiamo le seguire probabilità:

- PB (*Probabilità Errore nell State Bad*): 0.2

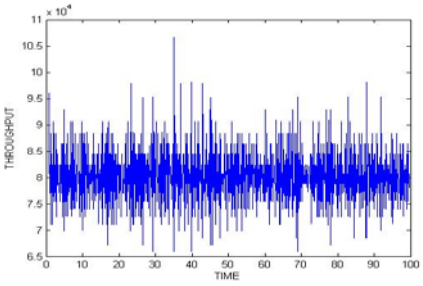


Fig.25: Collegamento tra e2-rx1 (P. S.)

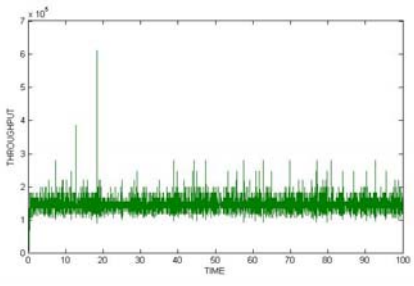


Fig. 26: Collegamento tra e2-rx3 (AF2)

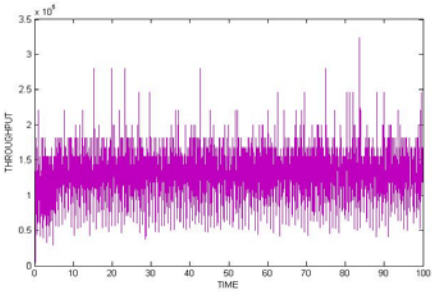


Fig.27: Collegamento tra e2-rx5 (BE)

- PB (*Probabilità Errore nell State Bad*): 0.4

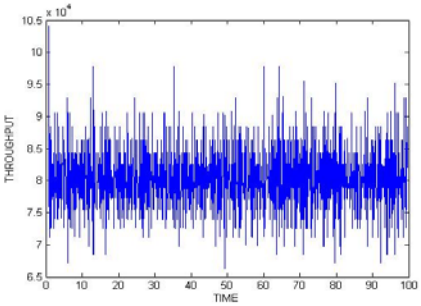


Fig.28: Collegamento tra e2-rx1 (P. S.)

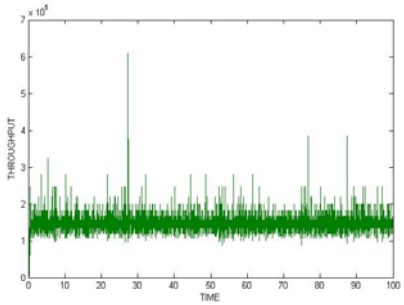


Fig. 29: Collegamento tra e2-rx3 (AF2)

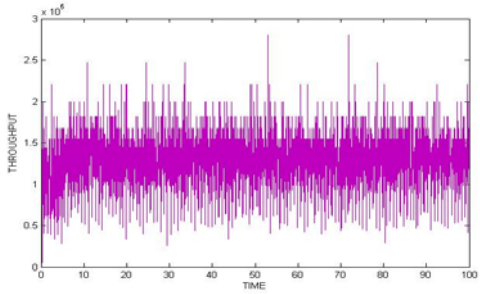


Fig.30: Collegamento tra e2-rx5 (BE)

- PB (*Probabilità Errore nell State Bad*): 0.6

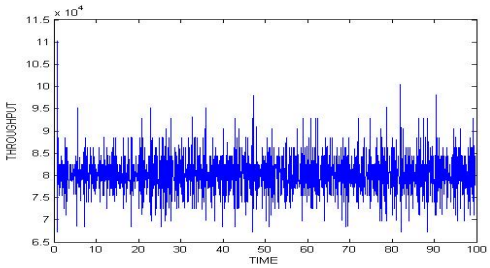


Fig.31: Collegamento tra e2-rx1 (P.S.)

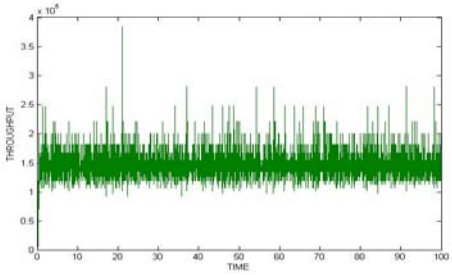


Fig.32: Collegamento tra e2-rx3 (AF2)

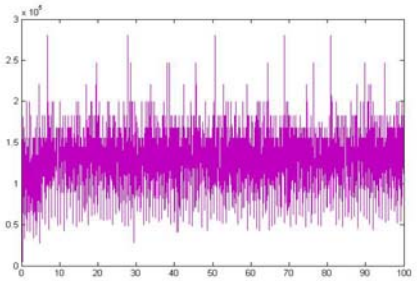


Fig.33: Collegamento tra e2-rx5 (BE)

- PB (*Probabilità Errore nell State Bad*): 0.8

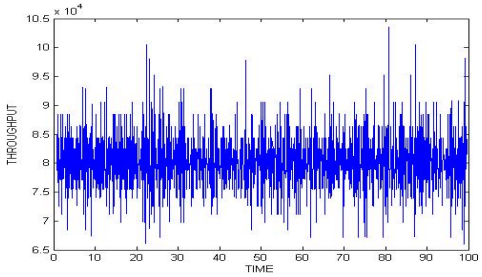


Fig.34: Collegamento tra e2-rx1 (P. S.)

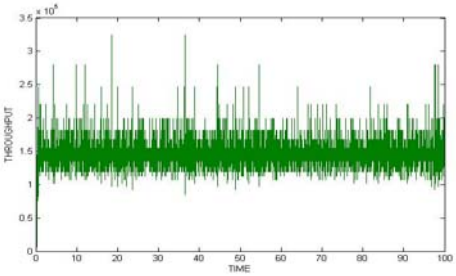


Fig.35: Collegamento tra e2-rx3 (AF2)

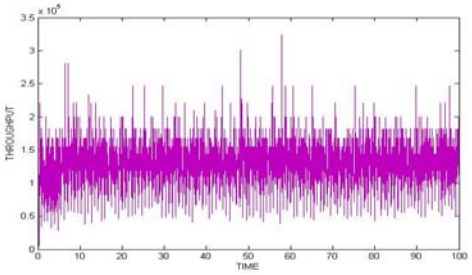


Fig.36: Collegamento tra e2-rx5 (BE)

Variando la probabilità di errore sul canale DiffServ continua ad effettuare una differenziazione dei servizi, ma in generale le prestazioni peggiorano. Infatti con un canale più rumoroso servono un maggior numero di ritrasmissioni che incidono pesantemente sul ritardo complessivo.

Conclusioni

Le simulazioni svolte durante questo lavoro di tesi hanno mostrato come l'architettura di rete Differentiated Services e in grado di realizzare un'efficace differenziazione dei servizi rispetto agli effetti del ritardo di coda.

Confrontando i risultati ottenuti per lo scenario di servizio best effort con quelli relativi allo scenario DiffServ, si può vedere come il throughput per il collegamento a cui viene data maggiore priorità è migliore rispetto a quello ottenuto nel scenario Senza DiffServ, a spese, però, dei servizi best effort. In questo ultimo caso la situzione più favorevole è quella che non prevede l'utilizzo di DiffServ perchè tutta la banda a disposizione è divisa in modo uguale tra i collegamenti sul router d'uscita.

Si può vedere anche che, quando tutti i flussi hanno la stessa priorità, il DiffServ ottiene lo stesso effetto di una rete best effort, cioè la condivisione delle risorse in parti uguali. Infatti, maggiore è il peso del traffico best effort rispetto al traffico totale, migliori sono le prestazioni del flusso Premium Service dato che quest'ultimo può utilizzare le risorse tolte al traffico con priorità più bassa. Dunque si può concludere che il DiffServ è in grado di introdurre priorità, ma la capacità di fornire garanzie dipende molto da come è composto il traffico in ingresso.

Variando la probabilità di errore sul canale DiffServ continua ad effettuare una differenziazione dei servizi, ma in generale le prestazioni peggiorano. Infatti con un canale più rumoroso servono un maggior

numero di ritrasmissioni che incidono pesantemente sul ritardo complessivo.

Per finire, si mostra che quanto più rumoroso è un canale serve un maggior numero di ritrasmissioni che incidono pesantemente sul ritardo complessivo.

Bibliografia

[1] A. S. Tanenbaum: “Reti di computer”, edizioni italiana a cura di Paolo Ciancarini, UTET Libreria, 1997.

[2] B. E. Carpenter e K. Nichols: “Differentiated Services in the Internet”, Procedings of the IEEE, volume 90, n° 9, Settembre 2002

[3] J. Postel: “Internet Protocol”, Internet RFC 791, IETF, Settembre 1981.

[4] P. Almquist: “Type of Service in the Internet Protocol Suite”, Internet RFC 1349, IETF, Luglio 1992.

[5] R. Braden, D. Clark e S. Shenker: “Integrated Service in the Internet Architecture: an overview”, Internet RFC 1633, IETF, Giugno 1994.

[7] Y. Bernet, P. Ford, R. Yavatkar, F. Baker (Cisco), M. Speer, R. Braden, B. Davie, F. Baker (UCLA), J. Wroclawski, E. Felstaine: “A Framework for Integrated Services Operation over Diffserv Networks”, Internet RFC 2998, Novembre 2000.

[9] R. Atkinson: “Security Architecture fornthe Internet Protocol”, Internet RFC 1825, IETF, Agosto 1995.

[10] S. Bradner e A. Mankin: “The Recommendetion for the IP Next Generation Protocol”, Internet RFC 1752, IETF, Gennaio 1995.

[11] K. Nichols, V. Jacobson e L. Zhang: “A Two-Bit Differentiated Service Architecture for the Internet”, Internet RFC 2638, IETF, Luglio 1999.

[12] S. Floyd, V. Jacobson: “Random Early Detection Gateways for Congestion Avoidance”, IEEE/ACM Transactions on Networking, Agosto 1993.

[13] Cisco IOS Quality of Service Solutions Configuration Guide: manuale on line disponibile all’indirizzo: http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/

[14] De Vriendt J., Laine P., Lerouge C., Xiaofeng X, ”Mobile network evolution: a revolution on the move”, IEEE Communications Magazine , Volume: 40 Issue: 4 , Apr 2002.

[15] Mihavska A., Wijting C., Prasad R., Ponnekanti S., Awad Y., Nakamura M., “A novel flexible technology for intelligent base station architecture support for 4G systems”, Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on , Volume: 2 , 2002.

[16] Mihavska A., Wijting C., Prasad R., Ponnekanti S., Awad Y., Nakamura M., “A novel flexible technology for intelligent base station architecture support for 4G systems”, Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on , Volume: 2 , 2002.

[17] Haardt M., Klein A., Koehn R., Oestreich S., Purat M., Sommer V., Ulrich T.; “The TD-CDMA based UTRA TDD mode”, Selected Areas in Communications, IEEE Journal on, Volume:18, Issue: 8, Aug 2000, Page(s): 1375 -1385

[18] Prasad R., Ojanpera T., “An overview of CDMA evolution towards wideband CDMA”, IEEE Communication Surveys, Fourth Quarter 1998, Vol. 1 No. 1.

[19] Francesco Vacirca, Andrea De Vendictis, Alfredo Todini, Andrea Baiochi: “On the effects of ARQ mechanisms on TCP performance in wireless environments”.

[20] Wei Luo, Krishna Balachandran, Sanjiv Nanda, Kirk K. Chang: “Delay Analysis of Selecive-Repeat ARQ With Applications to Link Adaptation in Wireless Packet Data Systems”.