

Capitolo 5

Il simulatore UMTS – Il livello RRC

In questo capitolo, si propone una descrizione dettagliata della struttura architetturale e protocollare del simulatore realizzato. In particolare ne viene descritto il livello RRC e le procedure che consentono a questi di interagire con gli altri strati protocollari. Particolare attenzione è infine rivolta all'implementazione dell'albero dei codici OVSF, agli algoritmi di *Radio Resource Management* ed alle politiche di *Admission Control*.

5.1 Architettura di rete nel simulatore

Si è deciso di modellare l'architettura di rete del simulatore analizzando in dettaglio il dominio d'utente e l'UTRAN, ovvero quella parte dell'*Infrastructure domain* che si occupa dell'accesso radio. La soluzione proposta è riportata in figura 5.1 e riproduce la struttura descritta negli standard.

La versione attuale del simulatore definisce un unico *Radio Network Subsystem* (RNS) all'interno dell'UTRAN; questo implica la presenza di un solo *Radio Network Controller* (RNC) per gestire l'insieme di celle costituenti l'intero scenario simulativo. La scelta operata è legata alla decisione di simulare i protocolli sull'interfaccia radio "Uu" tra gli UE e l'UTRAN. E' comunque possibile modificare il *software* sviluppato per creare una rete di accesso con più RNS.

Ogni cella fa capo ad un Node_B. Questi consente ad un UE di accedere alla rete e ridefinisce le funzioni tipiche delle *base station* dei sistemi di seconda generazione. In più però ogni Node_B è dotato di un'entità MAC_csh con il compito di gestire i canali condivisi all'interno della cella servita.

Le entità che compongono il singolo RNC sono:

- *il Radio Resource Control* (RRC): che esercita un pieno controllo di tutte le funzionalità dell'UTRAN necessarie per l'allocazione e la gestione delle

risorse radio ed effettua una vera e propria supervisione dei livelli dell'architettura dei protocolli radio.

- *i MAC_CSH*: che rappresentano una struttura di puntatori alle varie entità MAC_csh presenti nei Node_B. Servono per garantire una corrispondenza univoca tra RNC e i Node_B da quest'ultimo controllati;
- *la UMTS BTS_stack*: è una struttura dati responsabile della gestione delle pile protocollari presenti nell'UTRAN.

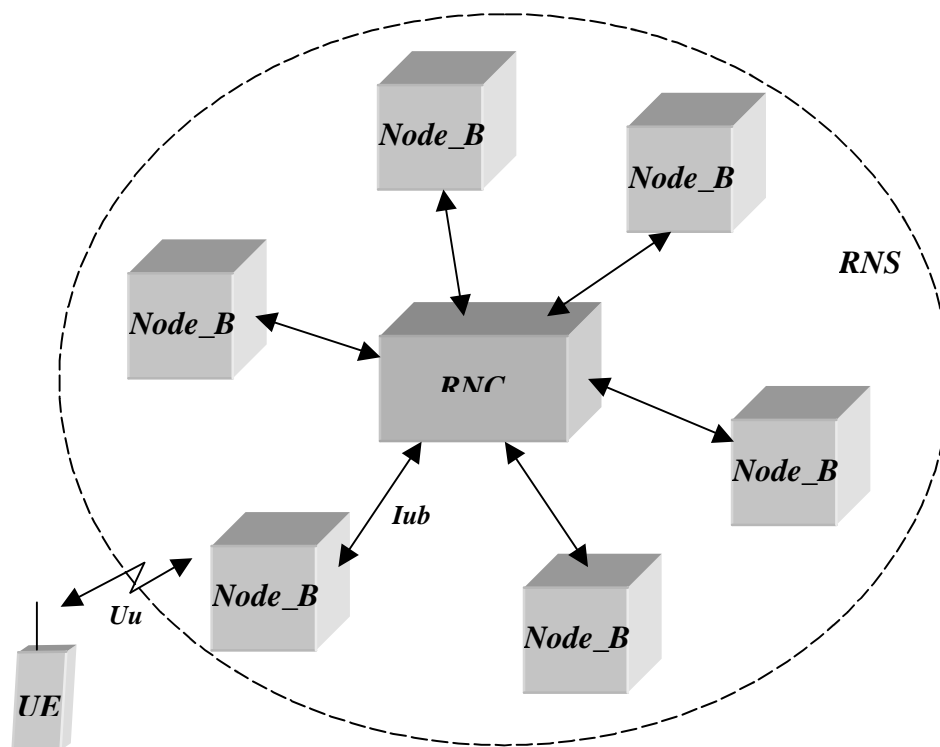


Figura 5.1 : Architettura di rete del simulatore

Per garantire l'interazione tra le pile protocollari simulate e i protocolli di livello trasporto (quali TCP o UDP), si è sovrapposta alla struttura della rete di accesso una parte di rete fissa, trascurando l'interfaccia "Iu" tra UTRAN e Core Network.

5.2 Architettura dei protocolli radio nel simulatore

I protocolli radio implementati nel simulatore (che riguardano la parte *wireless* del sistema UMTS spingendosi sino all'RNC) sono strutturati secondo la suddivisione illustrata nel capitolo 3 di questo scritto. Tale architettura è organizzata su più livelli protocollari e, all'interno di ogni livello più complesso, esiste un'ulteriore separazione in sottostrati corrispondente ad una precisa divisione funzionale.

In figura 5.2 e 5.3 si illustra il modello protocollare del simulatore rispettivamente nell'UE e nell'UTRAN.

I livelli che costituiscono la pila protocollare nel terminale mobile sono qui elencati.

- Il *MAC_MS* risulta composto dalle entità *MAC_d* e *MAC_c/sh* ed è collocato trasversalmente rispetto ai piani di controllo e d'utente. Questo è il primo dei due sottolivelli che formano lo strato *Data Link* e offre un'interfaccia di accesso radio ottimizzata per la trasmissione dei dati a pacchetto, mediante la multiplexazione statistica di più utenti su un set di canali comuni. I canali di trasporto in *uplink* simulati sono il RACH, il CPCH e il DCH.
- L'*Umts_RLC_block* rappresenta l'altro sottolivello dello strato 2 e assicura la trasmissione affidabile dell'informazione all'interno dell'UTRAN, offrendo un servizio di ritrasmissione per quei pacchetti che il livello 1 non è riuscito a consegnare correttamente a destinazione. Nel piano utente si può pensare l'*Umts_RLC_block* come un "contenitore" di entità RLC in grado di operare con modalità di trasmissione differenti in funzione del servizio che l'utente richiede. Questa soluzione permette ad un UE di instaurare più *Radio Bearers*, allocando un'entità RLC per ogni canale logico di traffico. Nel piano di controllo invece l'*Umts_RLC_block* non definisce alcuna entità RLC per trasmettere i dati di controllo del livello RRC verso l'UTRAN; si è infatti scelto di trasmettere la segnalazione mediante l'uso del puntatore alla pari entità. L'unico messaggio del blocco RRC che transita attraverso le pile protocollari e sull'interfaccia "Uu" è quello relativo all'instaurazione della connessione RRC. Tuttavia per come è stato modellato il livello RLC tale messaggio non viene però inoltrato attraverso l'UM RLC SAP (come previsto dagli standard), ma è passato direttamente al MAC sottostante. Sono invece implementate tutte le primitive che consentono al *Radio Resource Control* di interagire con lo strato RLC, attraverso il *Control SAP*.
- il *Radio_Resource_Control* è adibito a gestire e controllare i livelli inferiori MAC e RLC, nonché ad interagire con la pari entità nel RNC per conoscere le risorse di rete di cui può usufruire. Infatti il RRC, in funzione dei servizi richiesti dai livelli superiori e della momentanea disponibilità di risorse, alloca un *Radio Bearer* con caratteristiche in termini di *bit rate* e QoS tali da poter supportare il servizio richiesto. Una volta effettuata l'allocazione delle risorse, il RRC, attraverso le misure realizzate dal MAC e dal RLC, esegue un monitoraggio del traffico nell'UE e della qualità del canale radio. Se, mediante l'interazione con la pari entità nell'UTRAN, risulta necessario riorganizzare dinamicamente le risorse assegnate al terminale, interviene riconfigurando i livelli sottostanti. Il RRC fornisce i *Service Access Point GC*, *Nt* e *DC* ai livelli soprastanti, mentre definisce i *Control SAP* per interagire con i livelli sottostanti.

- l'*Umts_UPPER_layer* è definito sia nel piano di controllo che in quello d'utente. Nel primo esso fornisce le procedure adatte per far partire una richiesta di instaurazione di un servizio e per rinegoziare i parametri del servizio una volta instaurato. Nel secondo implementa le sole primitive di interfaccia verso il livello RLC, per permettere un'eventuale introduzione di alcuni modelli che emulino gli algoritmi propri del TCP e dell'UDP, senza modificare i livelli già creati. Nel piano d'utente l'*UPPER_layer* deve contenere tante entità TCP e/o UDP quante sono le entità RLC sottostanti. Attualmente le entità TCP e UDP contengono le sorgenti di traffico che simulano rispettivamente i servizi UDD e video, appartenenti a differenti classi di qualità di servizio. Questa modellazione consente di analizzare il comportamento del sistema con servizi diversi, anche nello stesso UE. I nomi dati a queste entità (TCP, UDP) sono legati alla futura introduzione accennata precedentemente.

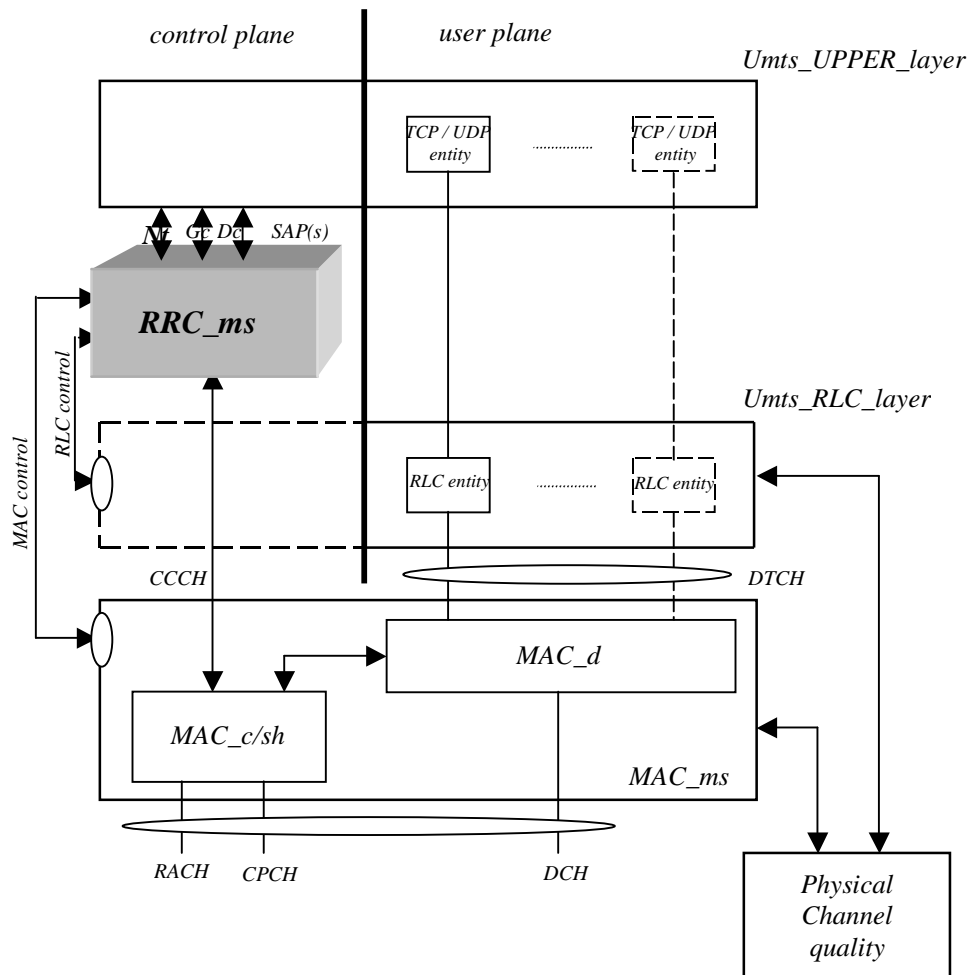


Figura 5.2 : Architettura dei protocolli radio lato UE

Il blocco *Physical Channel Quality* serve per simulare l'andamento della qualità del canale radio in funzione del tempo. Data la casualità del modello di mobilità dei terminali, è presente un blocco diverso per ogni UE. La scelta di introdurre un modello di canale trasmissivo permette di realizzare le procedure di livello RRC sulla stima della qualità del canale.

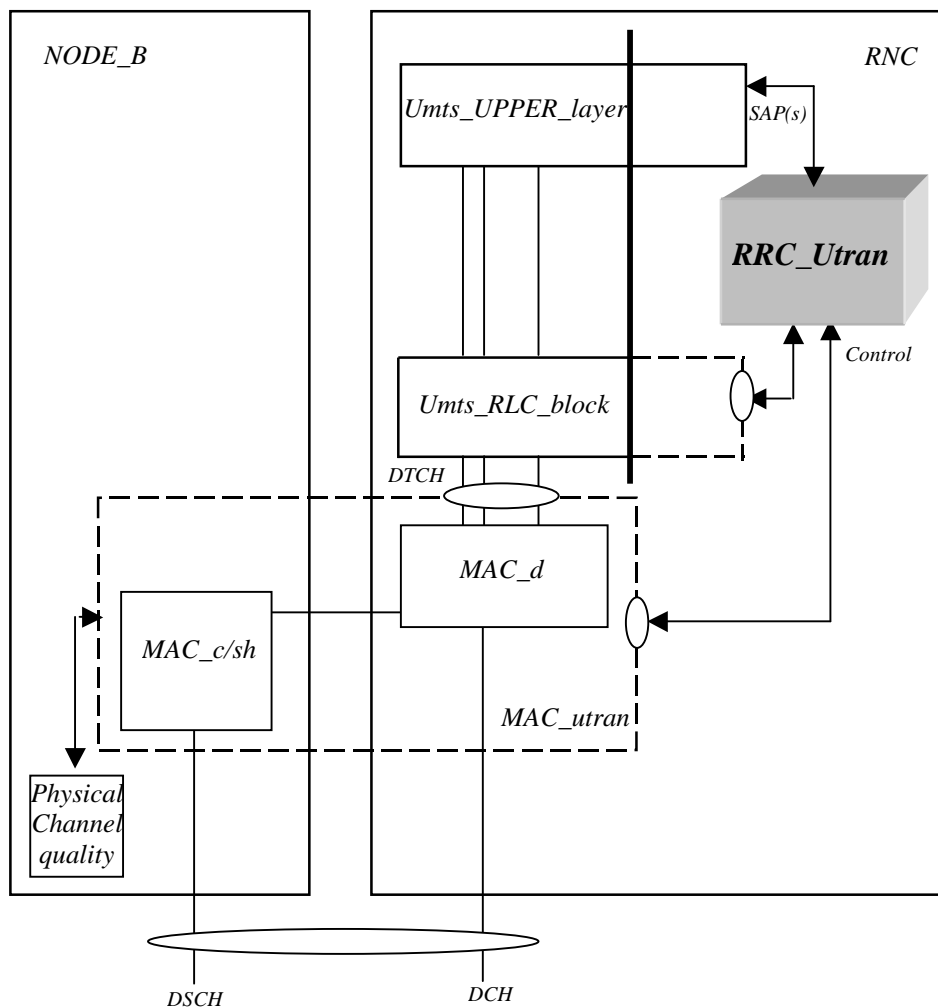


Figura 5.3 : Modello protocollare del simulatore lato UTRAN

Esiste una certa simmetria tra l'architettura protocollare nell'UE e nell'UTRAN. Ci sono però alcuni aspetti distintivi tra le due strutture che vale la pena di esaminare.

1. Nell'UTRAN sono definite tante pile protocollari quanti sono i mobili presenti nel simulatore. In particolare lo *stack* di ogni UE è in corrispondenza biunivoca con una particolare pila collocata nella rete d'accesso.
2. Durante una sessione, un mobile può spostarsi da una cella ad un'altra effettuando la procedura di *handover*; dal momento che ogni cella contiene un'entità MAC_c/sh responsabile della sua gestione, risulta evidente che l'UE in questione può essere servito da MAC_c/sh di versi durante il periodo di trasmissione dei dati. Pertanto la pila nell'UTRAN, duale al terminale, non è legata in modo statico ad una entità MAC_c/sh.

Nell'UTRAN si distinguono i livelli protocollari ora descritti.

- Il *MAC_UTRAN* risulta composto da varie entità MAC_csh e MAC_d. Ciò che differenzia, da un punto di vista strutturale, il livello MAC nell'UE e nell'UTRAN è il diverso legame tra queste entità. All'interno di ogni mobile vi è un solo MAC_d la cui gestione è delegata all'unico MAC_c/sh presente nel terminale. Nell'UTRAN invece vi sono tanti MAC_d quanti sono i mobili nel sistema e un MAC_c/sh per ogni cella della rete di accesso. In particolare ogni MAC_c/sh ha in carico l'insieme delle entità MAC_d, in corrispondenza di quei mobili che si trovano all'interno della cella da esso servita. I canali di trasporto gestiti sono il DSCH e il DCH.
- L'*Umts_RLC_block* realizza le funzionalità tipiche del livello *Radio Link*, in perfetta analogia alla definizione data per il blocco presente nel lato UE. Ci sono tanti RLC_block quanti sono i MAC_d nel RNC e ogni blocco contiene un numero di entità RLC pari al numero di *Radio Bearers* instaurati per ogni pila protocollare.
- il *Radio Resource Control* svolge una funzione di supervisione di tutte le pile protocollari presenti nell'UTRAN in cui è collocato e gestisce l'allocazione delle risorse radio dell'intero sistema. Per questo necessita di tutte le misure effettuate dal MAC e dal RLC sia negli UE che nell'UTRAN. Contiene al suo interno un vettore di puntatori alle pari entità RRC presenti nei vari UE.
- L'*Umts_UPPER_layer* realizza le primitive di interazione con i livelli RLC e RRC sottostanti. E' definito uno di questi blocchi per ciascuna pila protocollare e ogni blocco contiene tante entità TCP/UDP quante sono le entità RLC presenti nel RLC_block della medesima pila. Questo livello appartiene alla *Core Network*, ma è stato inserito all'interno del *Radio Network Controller* per consentire l'implementazione futura degli algoritmi di livello applicativo.

In ogni Node_B è inserito un modello markoviano che rappresenta la qualità del canale radio, di cui si è già parlato in precedenza. Tale modello riproduce l'andamento della qualità del mezzo trasmissivo in *downlink*.

5.3 Il livello RRC

I requisiti di questa tesi chiedono la realizzazione di un modello per rappresentare le funzionalità tipiche del livello RRC. Ad un primo livello di analisi risulta evidente associare alle entità RRC nei domini d'utente e di accesso radio due oggetti *software* distinti. Dallo studio degli standard però si nota una struttura funzionale comune tra le due entità. Pertanto conviene definire una superclasse *Umts_RRC* per raccogliere i comportamenti comuni ai due oggetti, mentre si riserva alle due classi derivate *Umts_RRC_ms* e *Umts_RRC_Utran* il compito di specializzare il comportamento della classe padre rispettivamente nei lati UE e UTRAN.

La relazione di generalizzazione-specializzazione tra le classi è rappresentata in figura 5.4.

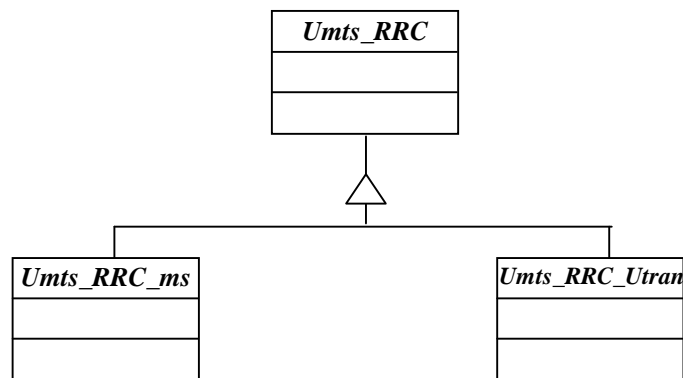


Figura 5.4 : Gerarchia di derivazione delle classi del livello RRC

5.3.1 Modello base del livello RRC

In questo sottoparagrafo si illustra l'architettura di base, che accomuna le entità RRC inserite nelle pile protocollari dell'UE e dell'UTRAN all'interno del simulatore. Gli elementi costitutivi di questa struttura, rappresentata in figura 5.5, vengono descritti nei punti che seguono.

5.3.1.1 Control Function Entity (CFE)

Il modello dello strato protocollare RRC definito nello standard [23] comprende quattro entità funzionali di controllo: la BCFE, la PNFE, la DCFE e la SCFE. Queste presentano caratteristiche differenti a seconda del dominio in cui sono posizionate: l'*User Equipment Domain* e l'*Infrastructure Domain*.

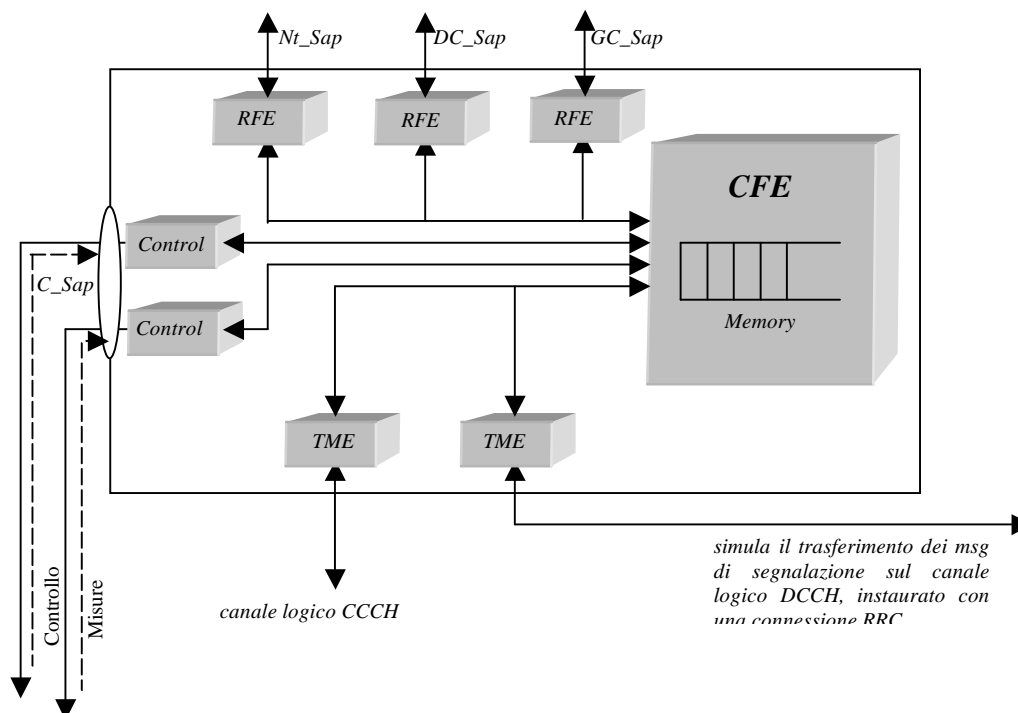


Figura 5.5 : Modello di base del livello RRC nel simulatore

Fissati gli obiettivi di:

1. realizzare un modello generale di *Radio Resource Control*, che sia al di sopra della suddivisione in domini;
2. semplificare al massimo la gestione delle procedure di livello RRC, con eventuali guadagni sui tempi di simulazione;

si è deciso di realizzare una *Control Function Entity* generica capace di eseguire tutte le funzionalità del livello in esame, indipendentemente dal lato (UE o UTRAN) in cui la CFE viene collocata.

La scelta effettuata porta ad utilizzare il *Pattern Command* per implementare la gestione delle procedure nello strato RRC. Tale *Pattern* è una soluzione *software* tipica dei linguaggi di programmazione orientati agli oggetti, come il C++. L'idea base di questa architettura è quella di incapsulare ogni richiesta in un oggetto e di definire una classe astratta *Command* che dichiari un'interfaccia generica per la gestione delle operazioni. Con oggetto si intende un'entità che raccoglie al suo interno sia la struttura dei dati sia le funzioni necessarie a specificarne il comportamento. Questa soluzione è giustificata dalle caratteristiche specifiche del *Pattern* citato:

1. disaccoppiamento tra l'oggetto che invoca l'operazione da quello che la deve eseguire;
2. possibilità di inviare richieste a degli oggetti, senza avere alcuna informazione sull'oggetto a cui deve essere mandata la richiesta o sul tipo di richiesta stesso;
3. possibilità di creare comandi completamente indipendenti dalle classi esistenti, implementando ogni cosa nel comando stesso senza delegare nulla alle altre entità;
4. semplicità nell'aggiungere nuovi *Command* senza modificare quanto già esiste;
5. possibilità di creare una *history list* dei comandi che sono stati o devono ancora essere eseguiti.

Le entità fondamentali che intervengono in questa architettura sono:

- *Command*: dichiara un'interfaccia per eseguire un'operazione;
- *Concrete Command*: può da un lato definire un semplice legame tra l'oggetto Receiver e un'azione; dall'altro implementare ogni cosa lui stesso senza affidare nulla al ricevitore;
- *Client (Application)*: crea un oggetto Concrete Command e gli associa il ricevitore;
- *Invoker*: dice al comando concreto di eseguire l'azione associata;
- *Receiver*: è l'oggetto che subisce l'azione e può conoscere o meno il modo di eseguire le operazioni associate alla richiesta.

Il modello RRC è implementato utilizzando le classi tipiche del *pattern Command*. In particolare le procedure di livello RRC diventano i *Concrete Command* dell'architettura descritta. Questi comandi sono pensati come oggetti estremamente intelligenti, in quanto implementano al loro interno l'azione associata alla procedura che rappresentano, senza delegare nulla al ricevitore. L'*Invoker* è invece rappresentato dalla *Control Function Entity*, che è in grado di gestire ogni tipo di procedura senza nulla sapere su chi sia il destinatario dell'operazione o sulla procedura stessa. L'entità RRC diventa al tempo stesso il *Client* e il *Receiver* del *pattern*. In figura 5.6 si illustra la relazione statica tra le classi del *pattern Command* mediante l'uso del diagramma delle classi, di cui si fornisce una breve descrizione. La classe *Umts_RRC* include al suo interno un oggetto della classe *Control Function Entity* mediante una relazione di contenimento di tipo stretto. Con aggregazione (o composizione) in senso stretto si intende che l'oggetto contenuto non ha una vita propria ma esiste in quanto parte dell'oggetto contenente. Questa relazione differisce dalla relazione di contenimento lasco, che invece indica l'indipendenza del ciclo di vita dell'oggetto contenuto dall'oggetto contenitore. Tornando al diagramma, la CFE a sua volta contiene:

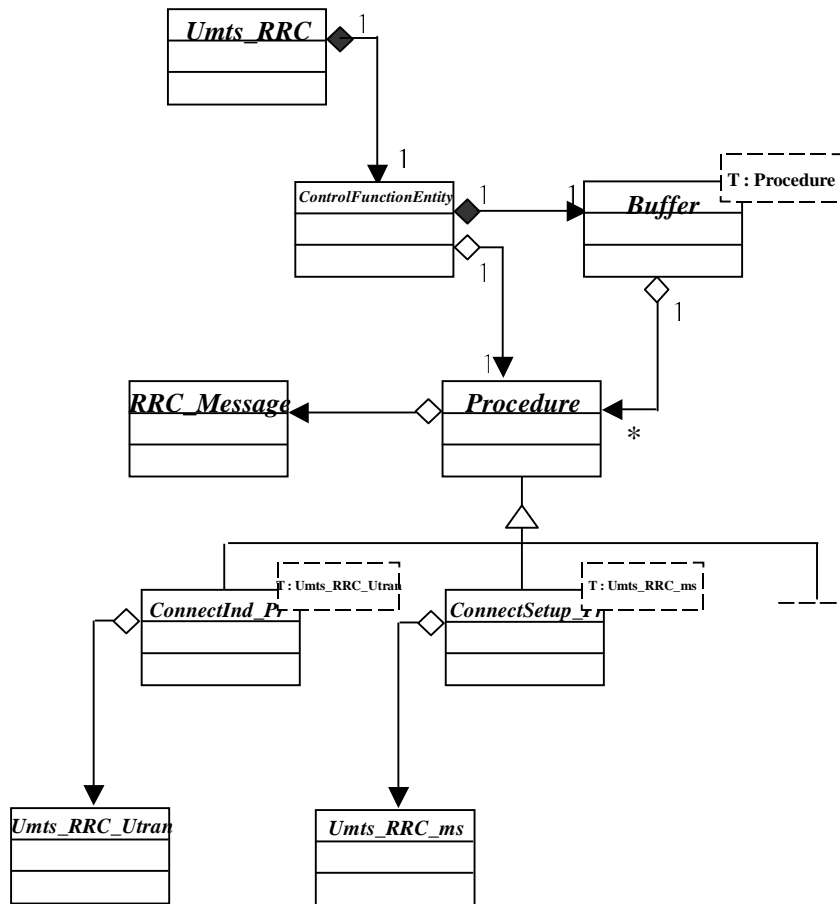


Figura 5.6 : Diagramma delle classi del pattern Command

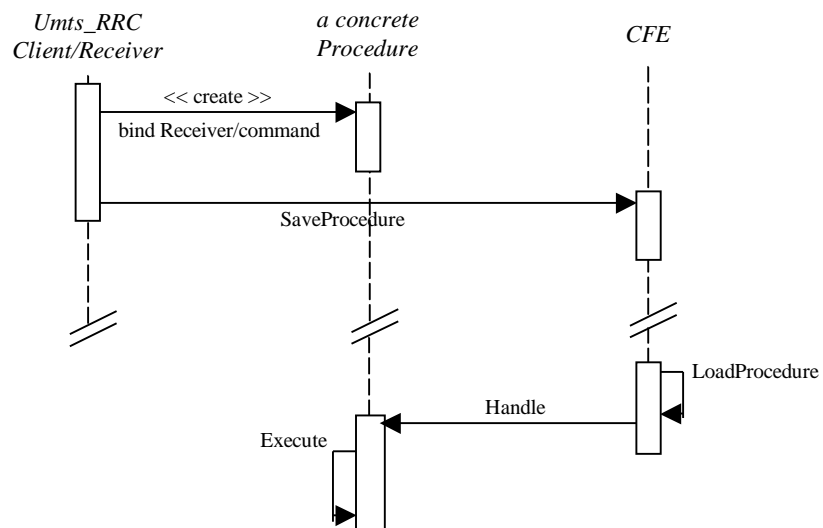


Figura 5.7 : Diagramma di interazione tra le classi del pattern Command

1. un *buffer* di operazioni secondo una relazione di contenimento stretto. Questo *buffer* è un oggetto della classe modello Buffer qui istanziata come classe di elementi derivati dalla gerarchia di derivazione avente *Procedure* come classe astratta fondamentale;
2. un puntatore alla procedura oggetto, di volta in volta estratta dal buffer, secondo un contenimento di tipo lasco.

L'interazione tra le entità appena citate è mostrata nel diagramma in figura 5.7. Quando il *Radio Resource Control* nell'UE o nell'UTRAN riceve dai livelli con cui interagisce un messaggio di segnalazione, crea la procedura oggetto, che il messaggio arrivato deve far iniziare, e associa al *Concrete Command* realizzato un puntatore al *Receiver*. Il ricevitore è in genere l'entità RRC stessa essendo questi il destinatario dell'azione. A questo punto la procedura viene passata all'entità funzionale di controllo contenuta nell'entità RRC. Quando la CFE viene processata, ordina alla procedura corrente di eseguire l'azione.

Poiché il passo temporale di avanzamento del simulatore è pari ad una trama UMTS, la CFE nel livello RRC viene schedulata ogni 10 ms. Per poter gestire più messaggi nel *Radio Resource Control* ad ogni intervallo temporale, è necessario inserire nella CFE un buffer (*memory*) per memorizzare le procedure associate ai messaggi. Questa soluzione consente all'entità di controllo di invocare le procedure secondo l'ordine cronologico con cui sono create.

5.3.1.2 Routing Function Entity (RFE)

La RFE è definita come l'entità responsabile dell'instradamento dei messaggi da/verso i livelli superiori al *Radio Resource Control*.

Nel simulatore questa entità risulta notevolmente semplificata. Essa infatti ha il solo compito di associare ai messaggi che giungono dall'UPPER_layer le procedure corrispondenti. L'instradamento invece dei messaggi che il RRC invia alle entità superiori è definita all'interno delle procedure stesse.

E' prevista una RFE per ogni *Service Access Point* con il livello superiore.

La *Routing Function Entity* nel simulatore è implementata come una funzione virtuale della classe *Umts_RRC*, in quanto il suo comportamento deve essere specializzato in funzione della collocazione dell'entità RRC nel lato UE o UTRAN.

5.3.1.3 Transfer Mode Entity (TME)

La TME, secondo lo standard [23], deve gestire la corrispondenza tra le entità funzionali di controllo di livello RRC con i SAP AM, TM e UM forniti dal livello RLC.

Nel simulatore però si è già sottolineato come i messaggi di livello RRC non transitino attraverso lo strato RLC, ma vengano o passati direttamente al MAC

sottostante o inviati all'entità RRC di pari livello mediante puntatore. Si può comunque ipotizzare che le informazioni indirizzate dal RRC alla pari entità passino:

- attraverso un canale logico CCCH, quando non è ancora stata instaurata una connessione RRC tra il mobile e l'UTRAN;
- attraverso un canale logico DCCH, una volta instaurata la connessione RRC.

La *Transfer Mode Entity* svolge un ruolo analogo alla RFE. Essa infatti deve associare ai messaggi di segnalazione che arrivano, rispettivamente dai canali logici CCCH e DCCH, le procedure opportune.

Nel modello base del RRC, la TME è definita come un metodo virtuale, la cui implementazione è da specializzare nelle entità RRC derivate a seconda che queste siano posizionate nel dominio d'utente o in quello di accesso radio.

5.3.1.4 Control SAP

I *MAC* e *RLC Control SAP* sono i punti di accesso che consentono al livello RRC di interagire con i rispettivi strati MAC e RLC. Attraverso questi il *Radio Resource Control* può:

- configurare le entità sottostanti;
- monitorare il volume di traffico sui canali di trasporto usati nel MAC;
- stimare la qualità del canale radio e sospendere di conseguenza la trasmissione dei dati e a livello MAC e a livello RLC.

5.3.2 Entità RRC lato UTRAN

Il *Radio Resource Control* nell'UTRAN è il solo responsabile della gestione delle risorse radio del sistema. Nel fare ciò esso è adibito a controllare i livelli inferiori MAC e RLC delle pile protocollari poste sia nel RNC sia negli UE. Mentre la supervisione degli *stack* protocollari nell'UTRAN è immediata (essendo collocati nello stesso dominio di rete in cui è situata l'entità in esame), il controllo delle pile dei terminali mobili può essere realizzata mediante l'interazione con una unità di pari livello denominata RRC_UE. Quest'ultima gestisce infatti gli strati della pila nel mobile in base ai comandi impartiti dall'UTRAN RRC e al tempo stesso trasmette alla rete tutte le misure effettuate nell'UE. Esiste quindi un rapporto di *master-slave* tra l'unità RRC dell'UTRAN e quella dell'UE (come mostrato in figura 5.8).

Il *Radio Resource Control* nel dominio di rete di accesso eredita mediante il meccanismo di ereditarietà del C++ la struttura di base (propria di ogni entità RRC) definita al paragrafo precedente. Inoltre sovrappone a tale architettura un nuovo

modello funzionale, che consente di realizzare il gestore di *Radio Resource Management* (ovvero l'insieme di tutte le procedure e algoritmi per la gestione interlivello delle risorse radio). Quest'ultima entità deve garantire che venga supportata la QoS richiesta dal *Radio Bearer* per erogare il servizio associatogli, allocando un numero adeguato di risorse radio. La figura 5.9 riproduce questo nuovo modello inserito nell'UTRAN RRC.

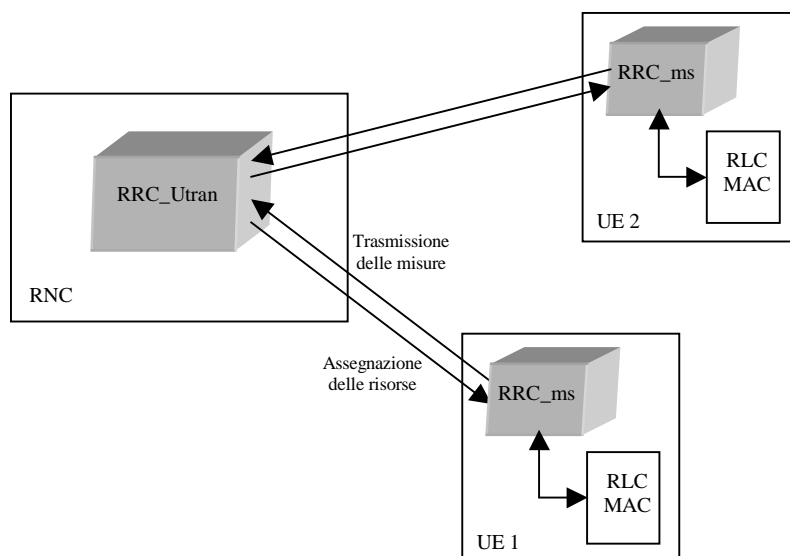


Figura 5.8 : Relazione tra entità RRC nell'UTRAN e nell'UE

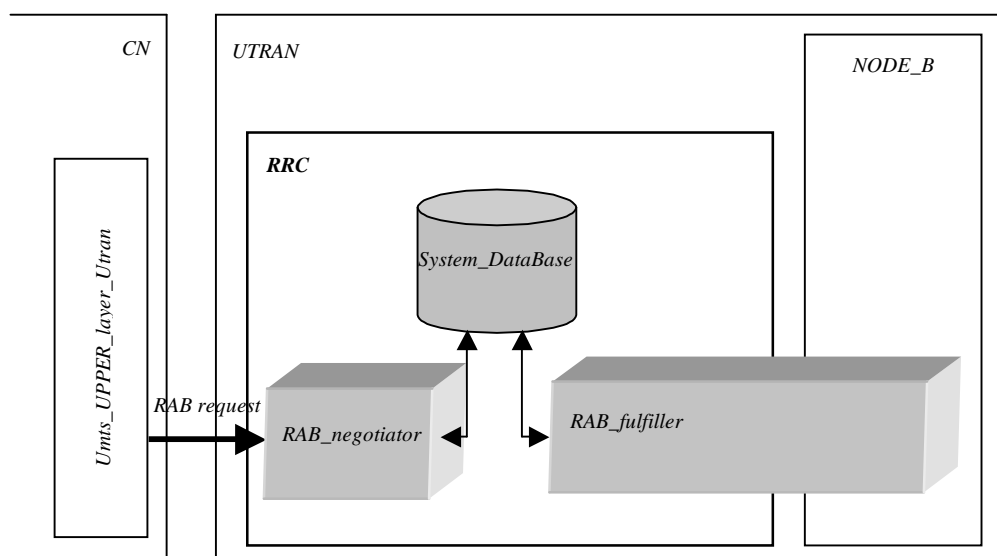


Figura 5.9 : Modello funzionale inserito nel RRC dell'UTRAN

5.3.2.1 Il gestore di Radio Resource Management (RRM)

Il concetto di Qualità di Servizio in UMTS poggia su una gerarchia di servizi di trasporto, dove il *Radio Access Bearer Service (RABS)* garantisce il trasporto dei dati attraverso l'interfaccia radio. Una descrizione più approfondita di questa architettura è riportata nel paragrafo 2.2 del Capitolo 2.

I servizi che si vogliono gestire nel simulatore realizzato appartengono a due classi di QoS distinte: quella *Streaming* e quella *Interactive*. Quando un utente richiede uno di questi servizi, il suo *Upper_layer* nella pila protocollare invia al RRC nell'UTRAN la richiesta di instaurare un *RAB service* con le caratteristiche opportune.

Lo *Streaming RABs* presenta attributi tali da garantire un certo *bit rate* e un ritardo di trasferimento contenuto. L'*Interactive RABs* invece presenta caratteristiche *best-effort*.

Il modello funzionale (figura 5.9), che nel livello RRC dell'UTRAN si occupa di gestire la QoS dei vari servizi, è composto da due entità:

1. il RAB negotiator;
2. il RAB fulfiller.

Questi due gestori necessitano entrambi di conoscere istante per istante le risorse radio disponibili nel sistema e il numero e la qualità dei RAB attivi. Si è quindi pensato di definire un *Database* di sistema, da loro condiviso, capace di memorizzare ogni tipo di informazione utile per le politiche di RRM.

Per quanto riguarda la progettazione ad oggetti delle due entità, si è deciso di realizzare una classe fondamentale astratta *RAB_handler* per raccogliere le caratteristiche comuni ai due gestori, e di specializzare invece il loro comportamento nelle rispettive classi derivate.

In figura 5.10 è rappresentata la gerarchia di derivazione delle classi dei gestori dei RAB.

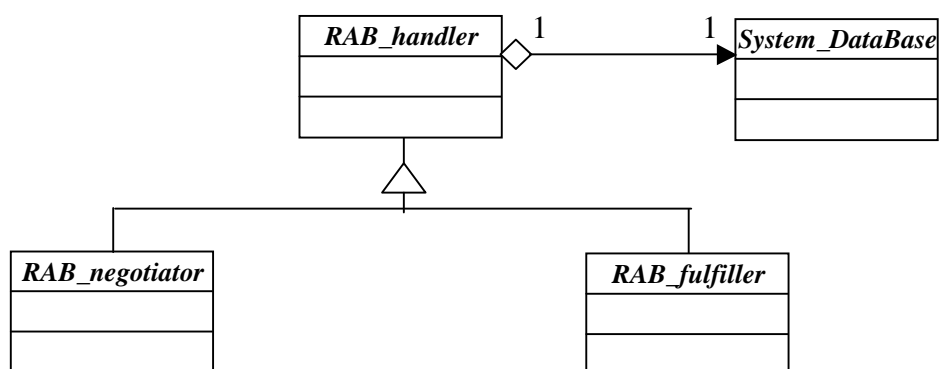


Figura 5.10 : Relazione di generalizzazione-specializzazione dei RAB handler

5.3.2.1.1 RAB negotiator

La richiesta di iniziare un servizio è gestita dal blocco in esame.

Il numero di RABs ammissibili in un sistema come quello UMTS ha un'influenza rilevante sulla qualità di servizio percepita dagli utenti. Il ruolo pertanto giocato da questa entità di ammettere o meno nuovi servizi è di fondamentale importanza per controllare la QoS.

I compiti del *RAB negotiator* sono:

- (a) Fissare, in funzione del servizio richiesto, i parametri di configurazione dei livelli MAC e RLC, quali:
 - lo *Spreading Factor*;
 - i canali logico e di trasporto;
 - la modalità di trasmissione a livello RLC;
 - la priorità di *scheduling* nel MAC.

Tali valori sono ricavati dall'interrogazione del *database* di sistema, al quale l'entità inoltra una richiesta di *Get_Configuration_Data*, specificando il servizio che si vuole ammettere.

- (b) Stimare la risorsa radio richiesta in termini di codice OVSF.
- (c) Verificare se il servizio può essere garantito in base alle informazioni dei punti (a) e (b). La politica di *Admission Control* implementata nel simulatore si basa sulla sola disponibilità del codice ortogonale. E' comunque possibile considerare nuove politiche basate per esempio sui livelli di potenza trasmessa richiesta dal servizio.

Se il RAB non può essere garantito, il *negotiator*, prima di respingere la richiesta di ammettere il nuovo servizio, cerca di rinegoziare o le risorse da assegnare al nuovo RAB o quelle dei RAB già instaurati nel simulatore. In particolare:

- se è richiesto uno *Streaming RAB*, si cerca di ridurre il *rate* di trasporto assegnato ad eventuali servizi *Interactive* già attivati; detta procedura continua fino a quando non si libera un codice capace di garantire il *bit rate* richiesto dal servizio. Se i traffici *Interactive*, precedentemente ammessi, raggiungono però un numero che non permette più di liberare codici con il *rate* desiderato, la domanda viene respinta.
- se il RAB richiesto è con caratteristiche *Interactive*, bisogna distinguere due sottocasi:
 - se il canale di trasporto a livello MAC è il DCH, si deve allocare un codice dedicato per il servizio per tutta la durata della sessione. Quando il codice non è però disponibile, si prova ad abbassare il *rate* degli altri

servizi *Interactive* già accettati, con il vincolo di continuare a garantire un codice per ogni servizio che sfrutta il DCH come canale. Se al termine di questa operazione si libera un codice, il nuovo servizio viene accettato. Diversamente la richiesta viene respinta.

- se il canale di trasporto a livello MAC è comune (CPCH, DSCH), allora il servizio è ammesso anche se attualmente non si dispone delle risorse necessarie.

Gli algoritmi implementati sono legittimati dal desiderio di evitare congestione del sistema, che causerebbe il rilascio anticipato dei RAB instaurati, prima che sia stato completato il trasferimento dei dati. Il meccanismo di bloccare a priori un servizio consente di garantire la QoS dei RAB già accettati, anche se può portare ad un decremento del traffico totale smaltito dal sistema.

5.3.2.1.2 RAB fulfiller

Dopo aver negoziato un certo numero di servizi di trasporto, il *Radio Resource Control* deve gestire i RAB instaurati. L'entità delegata per assolvere questo compito è il *RAB fulfiller*. Questi deve garantire la qualità di servizio per ogni connessione, allocando un numero adeguato di risorse radio (ovvero le sequenze ortogonali dell'albero dei codici).

Le funzionalità di tale entità sono eseguite in parte a livello di RNC e in parte a livello di *Node B*. Tra i compiti del *RAB fulfiller* si hanno:

- (a) la gestione del cambio di velocità del *Radio Bearer* in funzione dell'occupazione dei *buffer* nell'entità RLC associata al servizio in esame. Tale procedura si traduce nella ricerca di un codice OVSF che permetta al servizio di raggiungere il *bit rate* desiderato. Se la risorsa è disponibile, si termina con l'assegnazione del nuovo codice. Diversamente, la situazione rimane inalterata. Tale procedura è descritta nel capitolo 3 e viene richiamata al punto 5.3.3.4 di questo capitolo.
- (b) il coordinamento della trasmissione sul canale in *downlink* DSCH dei flussi dati appartenenti a servizi distinti. La modalità scelta per assegnare i codici ai vari flussi si basa su un algoritmo di schedulazione di tipo *Round Robin*.
- (c) l'assegnazione dei codici in base alla priorità dei servizi per permettere a flussi dati diversi di transitare sul canale comune a contesa in *uplink* PCPCH.

Per tutti gli algoritmi implementati valgono le considerazioni qui espresse.

- I servizi *Streaming* hanno una priorità superiore ai servizi *Interactive*. Pertanto in ogni procedura di allocazione delle risorse i primi hanno precedenza sui secondi.

- Nella ricerca dei codici per i servizi *Streaming* si applica un meccanismo di *pre-emption* delle risorse assegnate ai traffici *Interactive*. Quando infatti un servizio del primo tipo ha bisogno di un certo *rate* trasmissivo ma le sequenze di *spreading* per ottenerlo non sono più disponibili, si rallenta la velocità trasmissiva dei servizi *Interactive* per liberare un codice con le caratteristiche desiderate. Nel caso peggiore si tolgono codici al traffico *Interactive* per assegnarli a quello *Streaming*. Solo dopo che i servizi *Streaming* non hanno più bisogno di tali risorse, si riassegnano i codici ai servizi *Interactive*.

5.3.2.1.3 *System_DataBase*

System_DataBase è un oggetto *software* con il compito di memorizzare tutte le informazioni necessarie per gestire le risorse del sistema, configurare i livelli delle pile protocollari in base al servizio richiesto e conoscere il *bit rate* trama per trama di tutti i *Radio Bearer* instaurati.

Questa struttura dati è condivisa da entrambi i gestori dei RAB in quanto l'aggiornamento delle variabili in essa contenute deve essere visibile contemporaneamente al *negotiator* e al *fulfiller*. Non si pone però il problema di definire come sezione critica l'interazione con il *DataBase* essendo sequenziale il programma realizzato. Nell'elenco che segue si descrivono gli attributi contenuti nell'entità in esame.

Velocità di trasmissione e TFCS

Sono previsti due attributi nella classe *System_DataBase* per memorizzare le informazioni sui *rate* che il sistema può raggiungere e sull'insieme di combinazioni di *Transport Format* associati ai vari canali di trasporto. Tutti questi parametri sono ricavati dal file di ingresso "rate.dat". Le velocità di trasmissione previste dal sistema sono:

- 64 kbit/s;
- 144 kbit/s;
- 384 kbit/s.

Nello *standard* sono definite più velocità trasmissive, ma attualmente si è deciso di considerare nel simulatore solo quelle appena citate.

Parametri di configurazione di livello 2

Ogni servizio in UMTS è associato ad un certo RABS caratterizzato da determinate caratteristiche in termini di qualità di servizio. Per garantire tale QoS, si assegna a

ciascun RABs un certo numero di *Radio Bearer* (RB) opportunamente configurati. Attualmente, nel sistema realizzato, si instaura un RB per ogni *Radio Access Bearer Service*.

I parametri di configurazione del RB sono memorizzati nel *DataBase* di sistema e variano a seconda del servizio che si vuole instaurare. Questi attributi sono impostati dall'utente nel file dati "services.dat" e vengono registrati nei vettori *cmac_config_data* e *crlc_config_parameters* del *DataBase* al momento della creazione di quest'ultima entità. Questi vettori sono oggetti rispettivamente delle classi *CMAC_config_data* e *RLC_parameters*; hanno una lunghezza pari al numero delle tipologie di servizio che si possono simulare con il *software* realizzato.

Nella classe *CMAC_config_data* sono contenuti tutti i parametri necessari per configurare il *Radio Bearer* nel sottolivello MAC. In particolare si trovano le informazioni riguardanti:

- tipo e identificativo del canale di trasporto;
- priorità e identificativo del canale logico;
- *bit rate* da garantire (solo per i servizi che lo richiedono);
- *Transport Format Combination Set*.

Nella variabile *RLC_parameters* si memorizzano invece i parametri per caratterizzare l'entità di livello RLC. Poichè tali entità hanno attributi differenti a seconda della modalità di trasmissione scelta (TM, UM, AM), si è definita la classe astratta *RLC_parameters* come un'interfaccia generica per consentire una gestione equivalente dei parametri di tutte le entità di livello RLC, a prescindere dal loro modo di trattare le informazioni. In funzione del servizio, si legge dal file "services.dat":

- la modalità di trasmissione scelta;
- il tempo massimo di permanenza dei pacchetti nei *buffer* dell'entità RLC;
- il nome di un'altro file dati da cui ricavare ulteriori informazioni sull'entità RLC che si vuole creare.

Albero dei codici

Con la tecnica CDMA i segnali trasmessi sono sovrapposti sia in tempo sia in frequenza. La possibilità di distinguerli è ottenuta utilizzando codici ortogonali per ciascuno di essi. Essendo la velocità di trasmissione in aria costante e pari a 3.84 Mchip/s, tali sequenze devono avere lunghezza variabile al fine di trasmettere i segnali con diverse velocità di sorgente. Mentre i codici di uguale lunghezza risultano però tra loro ortogonali, quelli con *Spreading Factor* (SF) differente non necessariamente lo sono.

Per inserire nel simulatore i codici OVFS ed implementare i criteri di assegnazione delle sequenze secondo le regole definite negli standard, si è utilizzata una struttura ad albero binario.

Tale struttura è realizzata con un vettore gestito come albero binario da un'opportuna interfaccia di conversione. Quest'ultima infatti definisce una corrispondenza biunivoca tra le celle del vettore e i nodi dell'albero. I vantaggi legati a questa scelta comportano:

- una semplificazione di gestione dell'albero binario;
- un tempo di accesso unitario ai nodi dell'albero;
- una riduzione nell'occupazione di memoria dell'elaboratore.

L'albero dei codici OVFS che si ottiene è rappresentato in figura 5.11 e presenta le caratteristiche qui elencate.

- Ogni livello dell'albero definisce un fattore di *spreading* secondo la relazione seguente:

$$SF = 2^l, \quad l = 0, 1, \dots, (depth-1) \quad (5.1)$$

essendo l il numero corrispondente al livello in esame. Questa variabile può assumere valori compresi tra 0 e $(depth-1)$ se si ipotizza di numerare gli strati dell'albero partendo da zero e di indicare con $depth$ la profondità della struttura.

- Ogni livello contiene un numero di nodi pari al valore di SF che gli è associato.
- Ciascun nodo corrisponde ad un distinto codice OVFS. Indicando con $C_{l,m}$ il codice nella posizione m -esima del livello l dell'albero, esiste la seguente corrispondenza tra indice della cella nel vettore e nodo dell'albero:

$$index = 2^l - 1 + m \quad (5.2)$$

essendo $index$ l'indice di cella nel vettore. La relazione inversa, ricavata dalla (5.2), è invece data da:

$$\begin{aligned} l &= \lfloor \log_2 (index + 1) \rfloor, \quad \text{con } m = 0 \\ m &= index - 2^l + 1 \end{aligned} \quad (5.3)$$

- Poichè il numero delle sequenze a disposizione limita il numero dei segnali trasmessi simultaneamente, si vogliono realizzare alberi con *bit rate* diversi a parità di fattore di *spreading*. Per esempio si può definire un primo albero che associa per esempio al livello 2 un *rate_1* mentre un secondo albero che

fa corrispondere al medesimo livello un $rate_2$ diverso dal $rate_1$. Questo consente di gestire numeri variabili di codici per lo stesso $bit\ rate$. Tale soluzione è resa possibile se si fissa a priori un certo $rate\ minimo$ da associare all'ultimo livello dell'albero. Se l rappresenta lo strato l -esimo dell'albero, il $rate$ realizzato da l è pari a:

$$rate = min_rate * 2^{depth-l-1} \quad (5.4)$$

essendo min_rate il minimo rate trasmissivo garantito dalle foglie dell'albero e $depth$ la profondità di quest'ultimo.

Le considerazioni effettuate permettono di concludere che un albero di codici ortogonali è univocamente definito se si specificano:

1. la sua profondità ($depth$);
2. il minimo $bit\ rate$ da associare alle sue foglie.

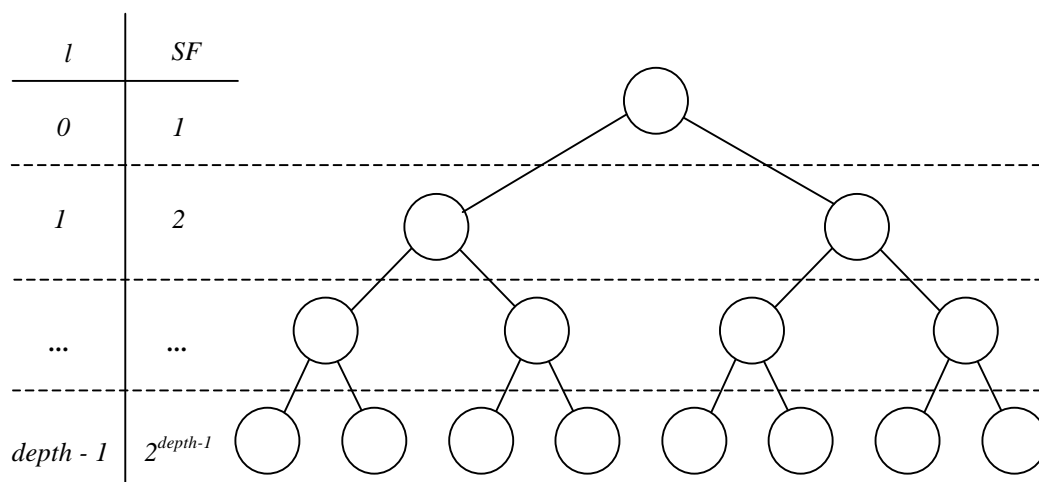


Figura 5.11 : Albero dei codici OVSF nel simulatore

La non correlazione tra i codici è garantita solo se sono verificate contemporaneamente le due condizioni seguenti:

- nessuna sequenza sul cammino dal codice, che si vuole assegnare, alla radice dell'albero è già riservata;
- nessun codice nel sotto-albero avente come radice la sequenza da assegnare è già prenotato.

Al fine di rispettare i requisiti di ortogonalità citati e differenziare le sequenze in base ai canali di trasporto per cui sono allocate, si è deciso di marcare i nodi dell'albero con sei possibili etichette:

1. *FREE*: il codice è libero e può ancora essere assegnato;
2. *RACH*: il codice può essere usato da quegli utenti che trasmettono in *uplink* usando il RACH come canale di trasporto;
3. *DEDICATED*: il codice è allocato in modo esclusivo ad un utente che usa un canale dedicato (DCH);
4. *COMMON*: la sequenza non è più libera ma è stata riservata per una futura allocazione. Può essere presa:
 - o da un utente che trasmette su un canale di trasporto comune (DSCH, CPCH);
 - o da uno che utilizza il DCH (ma anche il DSCH e CPCH) se ha bisogno momentaneamente del *rate* associato a tale codice. Permette quindi di gestire le procedure di cambio della velocità trasmissiva nel simulatore.
5. *TAKEN*: indica che un codice, marcato con *COMMON*, è stato finalmente allocato ad un utente che trasmette sul CPCH o sul DSCH. L'allocazione avviene in modo esclusivo per un numero di *frame* noto.
6. *NOT_AVAILABLE*: la risorsa non è allocata ma non può essere utilizzata perchè verrebbero meno le condizioni di non correlazione sopra descritte.

Il rilascio di un codice, dovuto o alla necessità di usare una sequenza con *rate* diverso o alla scadenza del tempo fissato per la sua allocazione, si distingue etichettando con *COMMON* il codice stesso. Una risorsa invece viene rimarcata con *FREE* solo quando il servizio che la sta usando ha terminato la sua sessione.

Il numero dei codici assegnati per la gestione del canale RACH è l'unico ad essere costante e fissato a priori nel file di configurazione dell'albero "bvsf_code.dat". Tale valore è espresso come numero in percentuale di foglie da marcare come *RACH*.

Al fine di consentire le operazioni di cambio *rate*, soprattutto in condizioni di sovraffollamento del sistema, è necessario contrassegnare come *COMMON* alcuni nodi dell'albero. Anche il numero di questi codici è espresso in percentuale nel file di ingresso e si riferisce a quelle celle del vettore che realizzano la massima velocità di trasmissione richiesta dal sistema.

Si riportano a questo punto alcuni esempi per chiarire meglio i concetti sopra esposti. Supponendo di impostare i seguenti parametri di ingresso nel file "bvsf_code.dat":

- profondità dell'albero uguale a 5;
- 120 kbit/s come velocità più bassa associata alle foglie della struttura;

- percentuale di codici destinati al RACH pari a 12,5% dei nodi dell'ultimo livello;
- 25% il numero in percentuale di codici *COMMON* con un *rate* uguale a quello massimo che il sistema vuole poter raggiungere (384 kbit/s);

si ottiene l'albero dei codici rappresentato in figura 5.11. Dal disegno si osserva come l'etichetta *NOT_AVAILABLE* prevalga sul marchio *COMMON* in caso di contesa di un nodo.

Un'assegnazione casuale dei codici con SF elevato potrebbe precludere l'uso di un gran numero di sequenze con basso SF. L'intento di massimizzare il numero di codici disponibili, soprattutto con basso fattore di *spreading*, ha quindi portato a definire il seguente algoritmo di allocazione e rilascio delle risorse. La ricerca di un codice disponibile per supportare la velocità desiderata avviene percorrendo il livello corrispondente dell'albero da destra verso sinistra. Quando invece si deve rilasciare il codice con un certo SF, si cerca la prima sequenza nello strato, associato allo SF in questione, procedendo da sinistra verso destra. Si è inoltre definito un algoritmo di ottimizzazione delle risorse per garantire al meglio l'intento anzidetto.

Questa funzione, richiamata in particolari istanti della simulazione, sposta le risorse già assegnate verso la destra della struttura mentre muove verso la sinistra dell'albero le sequenze che sono ancora o libere (*FREE*) o comuni (*COMMON*).

Ipotizzando di utilizzare l'albero in figura 5.12 come insieme di risorse disponibili, si vuole allocare un codice ad un utente che trasmette con un *rate* di 384 kbit/s su un canale di trasporto dedicato. In base agli algoritmi di allocazione, si ottiene la configurazione dell'albero illustrata in figura 5.13.

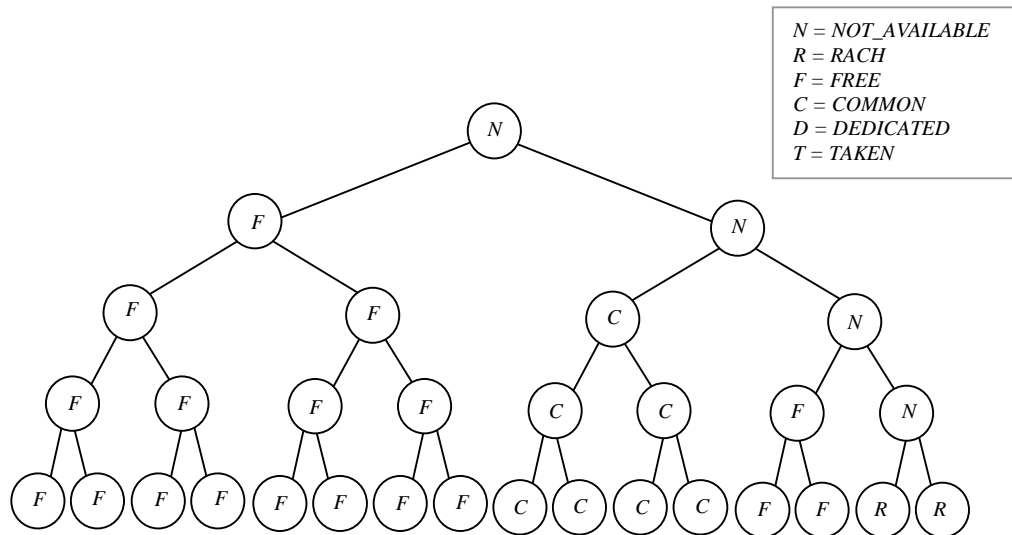


Figura 5.12 : Possibile configurazione di partenza dell'albero

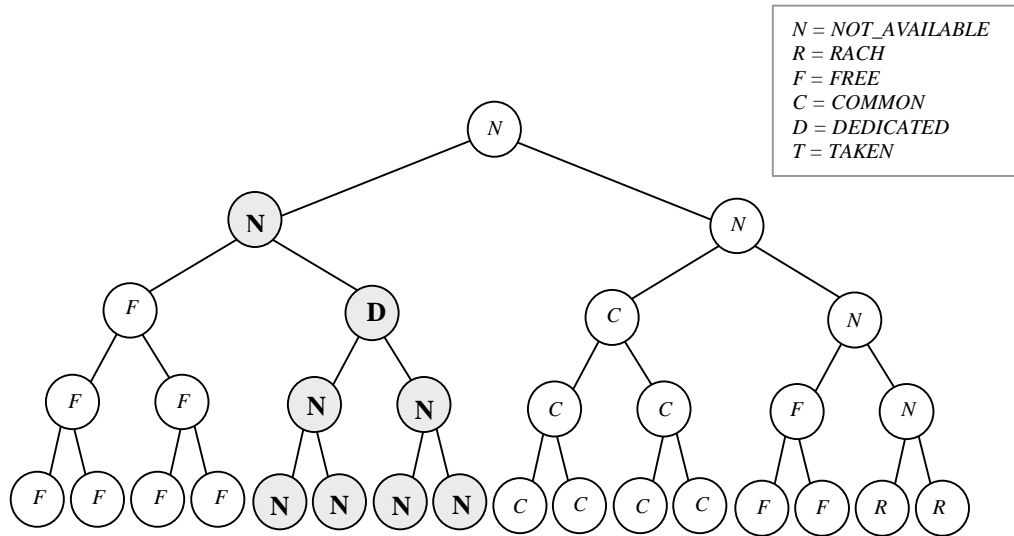


Figura 5.13 : Allocazione di un codice a 384 kbit/s per un canale dedicato

Se, ad un certo punto, il livello RRC ritiene necessario abbassare momentaneamente la velocità trasmissiva assegnata a quell'utente (ad esempio come conseguenza del monitoraggio dell'occupazione dei buffer di livello RLC) ad un *rate* di 64 kbit/s, l'albero assume la conformazione rappresentata in figura 5.14.

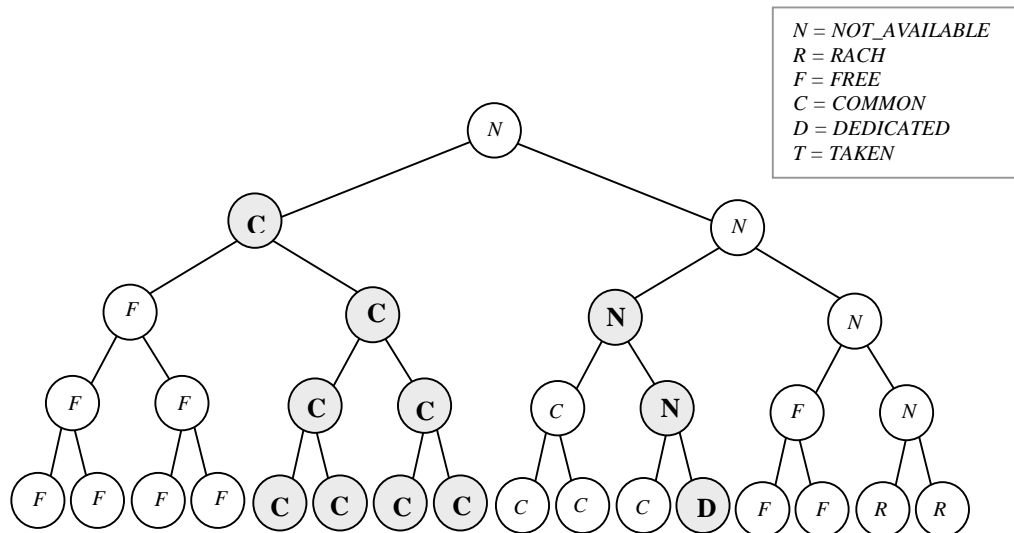


Figura 5.14 : Configurazione dell'albero in seguito al cambio della velocità

Supponendo infine che il *RAB negotiator* accetti la richiesta inviata da un UE di iniziare un servizio *Interactive*, assegnandogli una velocità trasmissiva di 144 kbit/s sul canale comune CPCH, l'albero dei codici assume l'aspetto di figura 5.15. Quando l'utente ottiene poi la possibilità di trasmettere sul canale CPCH grazie all'allocazione dedicata, per un numero di trame temporali fissato, di una sequenza ortogonale, un nodo **COMMON** corrispondente al *rate* desiderato viene marcato con **TAKEN** (figura 5.16).

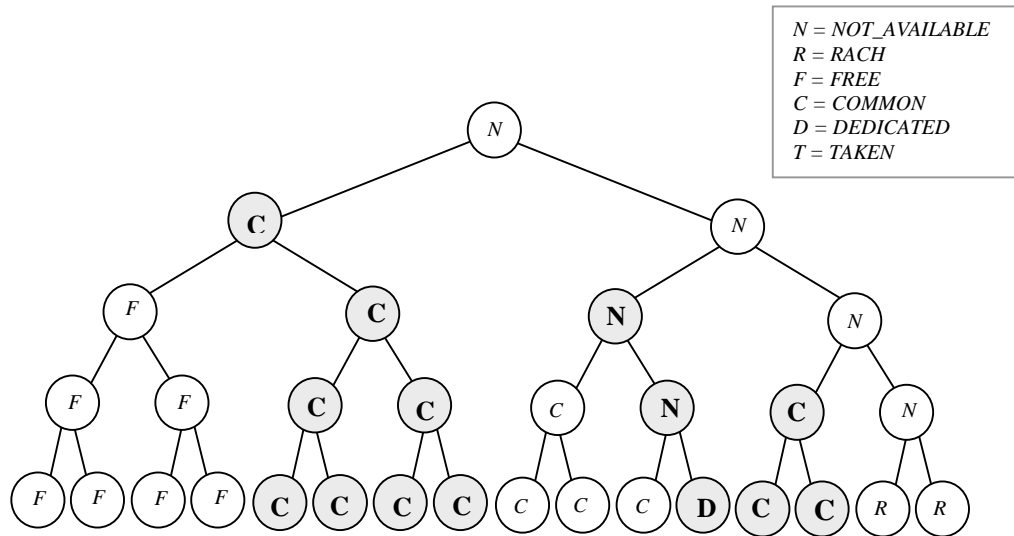


Figura 5.15 : Allocazione di un codice per un servizio su un canale comune

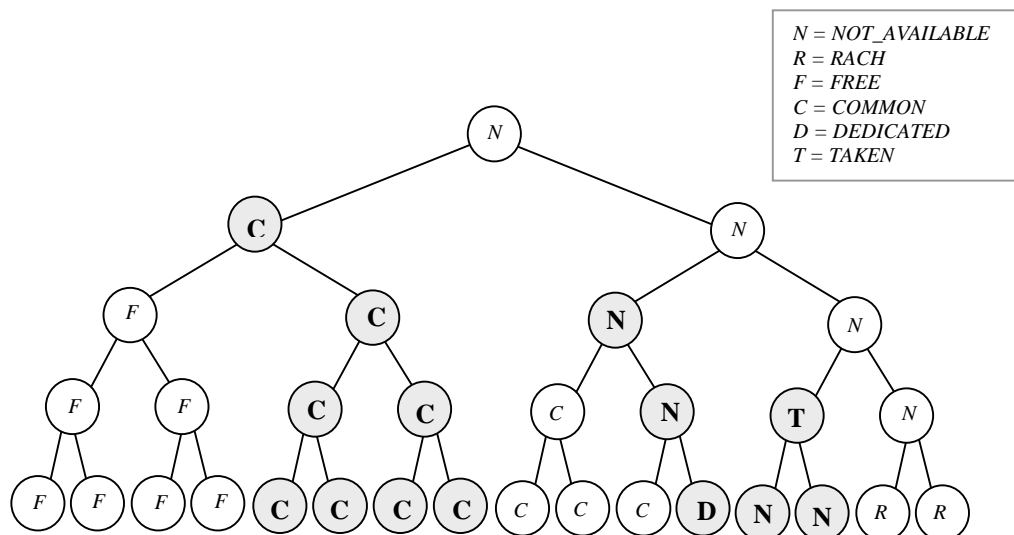


Figura 5.16 : Struttura dell'albero dopo che l'UE ha ottenuto il permesso di accedere al CPCH

5.3.2.2 Routing Function Entity nell'UTRAN

I metodi virtuali caratterizzano i sottotipi in una gerarchia di ereditarietà, legando dinamicamente durante l'esecuzione del programma il loro corpo alla funzione pubblica definita nella classe fondamentale mediante un riferimento.

Detto meccanismo consente quindi di specializzare la *Routing Function Entity* definita nel modello di base del *Radio Resource Control*, in funzione delle caratteristiche che la RFE presenta nel RNC o negli UE.

Nel RRC dell'UTRAN, esiste una sola di queste entità in corrispondenza del DC-SAP. In funzione dei messaggi di controllo che arrivano dagli *UPPER_layer* superiori, la RFE in questione deve creare le corrispondenti procedure:

- *TRAFFIC_VOLUME_MEASUREMENT*: è il messaggio che ordina al RFE di creare la procedura oggetto *MtrafficVolume_Pr* per configurare opportunamente le operazioni di monitoraggio del volume di traffico sui canali in *uplink* di quell'UE, identificato da un parametro di *input*;
- *TRAFFIC_VOLUME_MEASUREMENT_UTRAN*: deve creare la procedura *MtrafficVolumeUtran_Pr* per controllare il traffico in *downlink* nella pila protocollare del RNC in corrispondenza di quell'UE, il cui identificativo è ricevuto come ingresso;
- *INTRA_FREQUENCY_MEASUREMENT*: è l'ordine che genera la procedura *MintraFrequency_Pr* per far partire le misurazioni sulla stima della qualità del canale in *uplink* in quell'UE, specificato come argomento del metodo;
- *INTRA_FREQUENCY_MEASUREMENT*: è il messaggio che crea la procedura *MintraFrequencyUtran_Pr*, finalizzata ad avviare la procedura di stima del canale in *downlink* in un'opportuna cella del RNC.

5.3.2.3 Transfer Mode Entity nell'UTRAN

Ricordando quanto detto al punto 5.3.1.3 del presente capitolo, ci sono due *Transfer Mode Entity* nel RRC per gestire i messaggi provenienti rispettivamente dai canali logici CCCH e DCCH.

Nel *Radio Resource Control* lato UTRAN le suddette entità sono specializzate con i dettagli ivi descritti.

La TME associata al canale CCCH deve gestire i pacchetti spediti dagli UE sull'interfaccia radio e inoltrati alla presente entità per mezzo della primitiva *CMAC_Data_Ind* dal livello MAC sottostante. Il tipo di informazione contenuta nel messaggio permette alla TME di associargli la procedura oggetto corrispondente. Attualmente l'unica informazione di segnalazione qui gestita è quella legata alla richiesta di instaurare una connessione RRC tra l'UE mittente e l'UTRAN. L'oggetto *software* correlato a tale richiesta è denominato *ConnectInd_Pr*.

La TME collegata al canale logico DCCH deve invece gestire la corrispondenza tra i messaggi e le procedure seguenti:

- **RRC_CONNECTION_SETUP_COMPLETE**: è il messaggio spedito a tale entità dall'UE, come conseguenza della ricezione da parte di quest'ultimo del messaggio di CONNECTION_SETUP. La procedura associata a tale segnale è identificata dalla classe ConnectSetComplete_Pr.
- **RRC_CONNECTION_RELEASE_COMPLETE**: una volta che un utente riceve dalla rete un messaggio che lo informa di rilasciare la connessione RRC instaurata, l'UE invia come conferma di avvenuta ricezione un messaggio di RRC_CONNECTION_RELEASE_COMPLETE. La procedura oggetto creata per gestire questo messaggio è chiamata ConnectRelComplete_Pr.
- **MEASUREMENT_REPORT**: sono i messaggi che consegnano al RRC nell'UTRAN le misurazioni effettuate nei vari UE. L'oggetto per gestire questi dati di controllo appartiene alla classe derivata MReport_Pr.
- **RADIO_BEARER_RECONFIGURATION_COMPLETE**: è il riscontro spedito dall'UE, dopo che quest'ultimo ha ricevuto dalla rete un messaggio di RADIO_BEARER_RECONFIGURATION. Il TME associa al dato di RECONFIGURATION_COMPLETE la procedura denominata RBreconfComplete_Pr.

5.3.2.4 Control SAP del RRC nell'UTRAN

Il *MAC-Control-SAP* permette al *Radio Resource Control* nel RNC di interagire con il livello MAC delle pile protocollari contenute nell'UTRAN. In particolare l'unità di controllo di tale SAP deve associare ai messaggi, qui giunti e spediti dal MAC per mezzo della primitiva *CMAC_Measurement_Ind*, la procedura di *tvmResultUtran_Pr*. Tale oggetto *software* deve gestire i risultati del monitoraggio del traffico sui canali in *downlink*.

Il *RLC-Control-SAP* consente al *Radio Resource Control* di supervisionare il livello RLC degli *stack* nel *Radio Network Controller*. In particolare l'unità di controllo di questo punto di accesso deve associare alla stima della qualità del mezzo trasmissivo in *downlink* la procedura per coordinare lo stato delle entità RLC nelle pile protocollari, che usano tale mezzo per trasmettere. Lo stato delle entità dello strato controllato può corrispondere alla sospensione o alla riattivazione del trasferimento dei dati.

5.3.2.5 Reporting Criteria

Nel *Radio Resource Control* lato RNC sono definiti inoltre un insieme di attributi privati. Tra questi merita una breve spiegazione la variabile *criteria*. Essa è un vettore di puntatori ad oggetti della classe *ReportingCriteria*; la sua dimensione è pari al numero di canali di trasporto gestiti nel simulatore. L'identificativo del canale di trasporto viene adoperato per indicizzare il vettore. Ogni cella contiene i

parametri che caratterizzano le possibili modalità di consegna delle misure sul monitoraggio del traffico del corrispondente canale di trasporto. La modalità di trasmissione dei resoconti può essere periodica o ad eventi (consultare il punto 3.3.3.1 del capitolo 3).

Le variabili della classe ReportingCriteria sono:

- le soglie superiori e inferiori, che fissano il *range* dell'occupazione dei buffer all'interno del quale non si deve eseguire la consegna dei risultati con la modalità *Event_Trigger*;
- il tempo di *Pending* per la modalità ad eventi;
- il tempo di recapito delle misure per la modalità periodica.

Questi attributi, classificati in funzione del canale di trasporto, sono impostati nel file dati "criteria.dat", passato come ingresso al simulatore.

5.3.3 Entità RRC lato UE

Il *Radio Resource Control* nel dominio d'utente eredita la struttura della classe astratta *Umts_RRC* da cui deriva. A suddetta struttura si sovrappone una serie di attributi privati per gestire le procedure tipiche di tale entità. Queste variabili, sotto elencate, sono definite secondo le notazioni fornite nello standard [23].

- *t300*: è un oggetto di tipo T300, appartenente alla gerarchia di derivazione della superclasse *Timer*. Esso deve realizzare la funzionalità dell'omonimo *timer* prevista dallo standard (e descritta al punto 3.7.6 del capitolo 3).
- *V300* e *N300*: sono variabili intere che rappresentano rispettivamente il contatore del numero di ritrasmissioni del messaggio di RRC CONNECTION REQUEST e il limite superiore di ritrasmissioni che si possono effettuare dello stesso messaggio.
- *ue_id*: rappresenta l'identificativo d'utente. Tale numero viene assegnato al RRC nel momento in cui l'UE, che contiene tale entità, viene inserito nell'ambiente di simulazione.
- *status*: è un attributo di tipo enumerazione STATE_TRANSITIONS. Serve per memorizzare lo stato corrispondente in cui si trova l'entità RRC. In base agli stati previsti dallo standard, gli enumeratori (ovvero l'insieme di costanti simboliche costituenti l'enumerazione) sono: IDLE_MODE, CELL_DCH, CELL_FACH, CELL_PCH, URA_PCH.
- *M_identity*: è un vettore di puntatori ad oggetti, il cui tipo appartiene alla gerarchia di ereditarietà illustrata in figura 5.17 (*Measurement_Type* è la classe fondamentale astratta di questa struttura). Questo vettore contiene i parametri di gestione di tutte le misurazioni attualmente attive nell'UE. L'identificativo della misurazione, usata per indicizzare il vettore, individua

la cella contenente gli attributi di configurazione dell'operazione di misura a stessa. Nell'UE sono previste per ora solo due delle sette possibili misurazioni definite negli standard: una è il monitoraggio del volume di traffico, l'altra è la stima della qualità del canale.

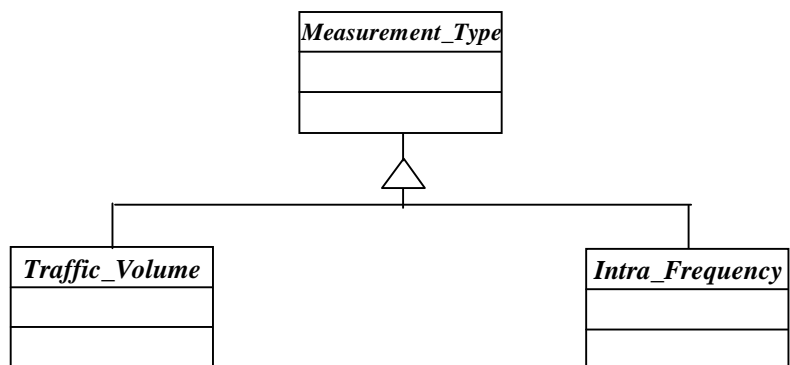


Figura 5.17 : Relazione tipo-sottotipo della classe astratta Measurement_Type

La classe Traffic_Volume definisce i parametri che l'entità RRC utilizza per configurare il sottolivello MAC al fine di implementare la procedura di controllo del volume di traffico. Tali attributi comprendono:

- *time_Interval*: rappresenta l'intervallo temporale intercorrente tra due misure successive, effettuate nel MAC;
- *num_trCh*: è l'identificativo del canale di trasporto che si deve monitorare;
- *validity*: indica lo stato di livello RRC in cui la misurazione può essere eseguita. Questa variabile è di tipo enumerazione M_VALIDITY, i cui enumeratori sono: ALL_STATES, ONLY_CELL_DCH, NO_CELL_DCH, NO_VALIDITY.
- *rep_Mode*: è un elemento di tipo enumerazione REPORTING_MODE. Definisce la modalità con cui devono essere spedite le misure all'entità RRC nell'UTRAN. Sono previste due tipologie di consegna delle misure: una periodica (PERIODICAL) e una ad eventi (EVENT_TRIGGER);
- *criteria*: contiene i parametri che consentono al MAC di determinare gli istanti di tempo in cui deve passare le misure eseguite al livello RRC soprastante.

La classe Intra_Frequency contiene invece i seguenti attributi, finalizzati a caratterizzare la procedura di stima della qualità del canale trasmissivo nel livello RLC.

- *time_Interval*: definisce il passo temporale con cui si deve campionare il canale trasmissivo;
- *rep_Mode*: analogamente a sopra, definisce la modalità di consegna (periodica o ad eventi) delle stime effettuate sulla qualità del canale.

5.3.3.1 Routing Function Entity nell'UE

Nel RRC dell'UE si specializza il comportamento della sola RFE associata al *Service Access Point Nt*. L'entità in esame fa corrispondere al messaggio di RRC_CONNECTION_REQUEST inviato dai livelli superiori la procedura di ConnectReq_Pr. Quest'ultima gestisce la spedizione della richiesta di instaurazione della connessione RRC sull'interfaccia radio.

5.3.3.2 Transfer Mode Entity nell'UE

La *Transfer Mode Entity* in relazione con il canale logico CCCH dovrebbe gestire i messaggi di controllo inviati sul canale FACH in *downlink* dalla rete e consegnati a tale entità per mezzo della primitiva CMAC_Data_Ind. Dal momento che gli algoritmi di gestione del canale di trasporto FACH non sono stati implementati, si consegnano i dati trasmessi su tale canale al RRC dell'UE destinatario con un ritardo casuale di distribuzione uniforme. Il compito di questa TME è quello di mettere in relazione i messaggi con le opportune procedure di livello RRC. Le corrispondenze ivi gestite sono quelle tra:

- il messaggio di RRC_CONNECTION_SETUP e la procedura ConnectSetup_Pr;
- il messaggio di RRC_CONNECTION_REJECT e la procedura di ConnectReject_Pr.

La TME associata al canale logico DCCH gestisce invece la relazione tra i messaggi e le procedure seguenti:

- *MEASUREMENT_CONTROL*: è il messaggio spedito dal RRC nell'UTRAN per impostare nell'UE i parametri di configurazione adeguati a realizzare una qualche misurazione. La procedura che deve gestire tale azione è denominata Mcontrol_Pr.
- *RRC_CONNECTION_RELEASE*: è l'informazione di controllo inviata dalla rete per avvisare l'utente che deve rilasciare la connessione RRC. La procedura oggetto associata a tale messaggio è ConnectRelInd_Pr.
- *RADIO_BEARER_RECONFIGURATION*: è il messaggio che informa l'utente che deve riconfigurare un opportuno *Radio Bearer*. La procedura RbreconfInd_Pr associata può comportare il cambio del rate trasmissivo, la

sospensione della trasmissione, lo *switch* da un canale di trasporto ad un altro, l'assegnazione di una nuova priorità di *scheduling* a livello MAC.

- *CHANGE_SOURCE_STATE*: è un messaggio non previsto negli standard ma qui implementato per consentire al RNC di modificare lo stato della sorgente *Interactive* (solo quando è utilizzata) nell'UE. La scelta del nuovo stato è compiuta dal *RAB negotiator* in base alle politiche di gestione dei servizi.

5.3.3.3 Control SAP nell'UE

I SAP di controllo permettono al *Radio Resource Control* di supervisionare i livelli sottostanti nella pila protocollare, secondo i parametri fissati dall'entità RRC del RNC.

Attraverso il *MAC-Control-SAP*, il RRC configura il livello MAC per mezzo delle primitive *CMAC-Config-Req* e *CMAC-Measurement-Req*. Riceve invece le misure realizzate dal sottolivello MAC utilizzando la primitiva *CMAC-Measurement_Ind*. Quest'ultima verifica se lo stato del RRC è abilitato ad inoltrare i risultati delle misurazioni alla rete e, in caso affermativo, crea la procedura di *tvmResult_Pr* per gestire la trasmissione delle stesse misure alla pari entità nell'UTRAN.

Il *RLC-Control-SAP* consente al livello in esame di interagire con lo strato sottostante RLC. Le primitive previste in questo punto di accesso sono:

- *CRLC-Config-Req*: per configurare i parametri del livello RLC;
- *CRLC_status_req*: per avviare le misurazioni sulla stima del canale;
- *CRLC-resume-Req*: per riabilitare alla trasmissione le entità del RLC precedentemente disattivate;
- *CRLC-suspend-Req*: per sospendere la trasmissione delle entità RLC;
- *CRLC-status-ind*: per passare al RRC la stima della qualità del canale. Essa crea la procedura *ifmResult_Pr*, che coordina la spedizione delle misure ricevute alla entità di pari livello posta nel RNC.

Una descrizione più approfondita di suddette primitive è fornita nel capitolo 3 di questa tesi.

5.3.4 Procedure di livello RRC nel simulatore

Le procedure di livello RRC definite nel simulatore riproducono fedelmente le omonime procedure tratte dallo standard [23] e descritte nel paragrafo 3.7.5 del capitolo 3.

L'obiettivo dei punti che seguono è finalizzato a mostrare l'uso combinato dei messaggi delle pari entità RRC e delle primitive inter-livello per sottolineare la divisione funzionale tra gli strati della pila protocollare.

5.3.4.1 Instaurazione della connessione RRC

La procedura seguente può essere pensata come conseguenza di due distinte situazioni:

- (a) i livelli superiori nel dominio di rete fissa trasmettono una richiesta di *paging* per avvisare l'utente selezionato del suo coinvolgimento in un trasferimento di dati;
- (b) i livelli superiori dell'UE necessitano una connessione con la rete per poter espletare un certo servizio (come una chiamata verso un altro utente).

Nel simulatore entrambe le condizioni (a) e (b) sono rappresentate dal medesimo evento *Init_Session_Ev*, creato per ogni utente al momento del suo inserimento nell'ambiente di simulazione. Detto evento ha il compito di fissare l'istante temporale in cui i livelli superiori dell'UE devono avvisare il *Radio Resource Control* sottostante di avviare la procedura di *RRC Connection Establishment*. Nel passare tale richiesta, l'*UPPER_Layer* deve specificare il tipo di servizio che vuole realizzare tramite la connessione. Questi può essere *Streaming* o *Interactive*.

I diagrammi di sequenza 5.18 e 5.19 illustrano l'interazione tra i livelli degli *stack* protocollari nell'UE e nell'UTRAN rispettivamente nei casi di connessione RRC conclusa con successo e fallita; si indicano in particolare le primitive di comunicazione, i messaggi scambiati e i canali di trasporto usati per trasferire i dati di controllo.

Il *Radio Resource Control* inizia la procedura di instaurazione della connessione inviando il messaggio di *RRC_Connection_Request* sul MAC SAP per il canale logico CCCH. La primitiva di comunicazione tra MAC e RRC utilizzata è la *CMAC_Data_Req*. Il MAC trasmette poi il messaggio di livello 3 sul canale di trasporto in *uplink* RACH. Tale messaggio è un oggetto *software* con i seguenti attributi:

- *UE_identity*: identificativo dell'utente che inoltra la richiesta di connessione. Serve per accedere alla pila duale a quella del mobile, situata all'interno dell'UTRAN;
- *Establishment_cause*: specifica il motivo della connessione, indicando il tipo di servizio richiesto.

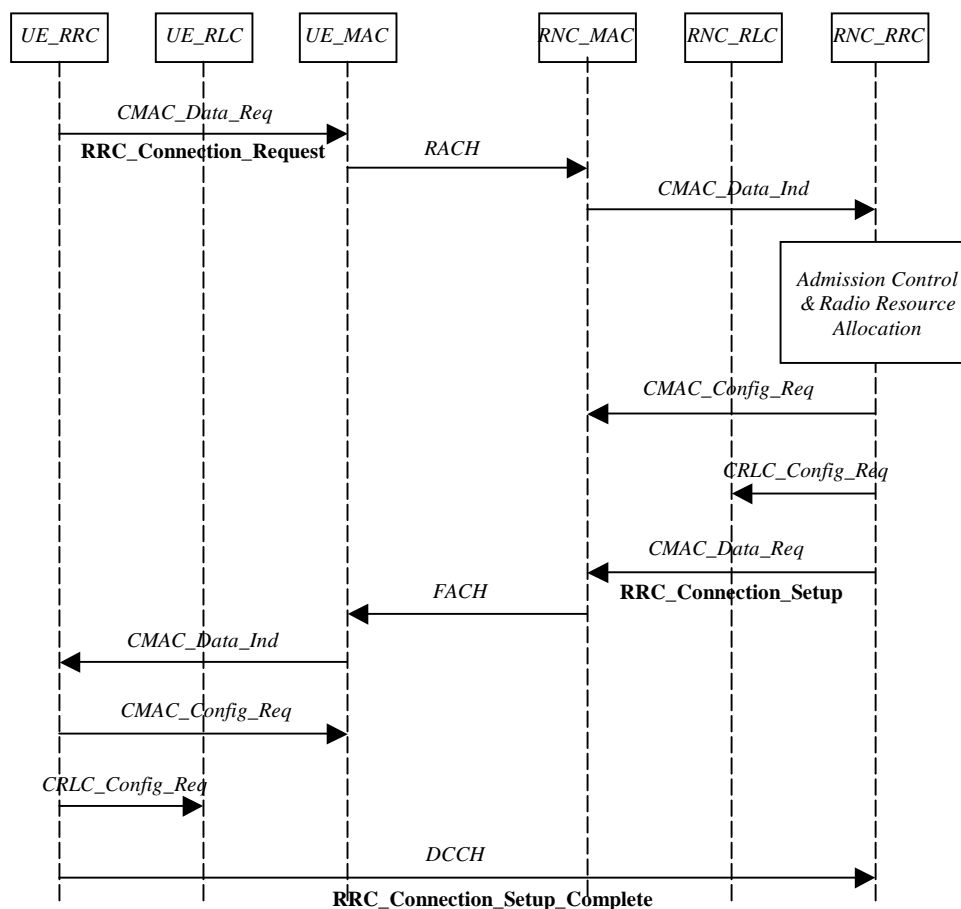


Figura 5.18 : Diagramma di sequenza di una connessione RRC conclusa con successo

Nel lato UTRAN, in seguito alla ricezione del messaggio di `RRC_Connection_Request`, il livello RRC realizza le opportune procedure di *Admission Control* e *Radio Resource Management*. Nel caso in cui la rete sia in grado di soddisfare le richieste dell'utente (figura 5.18), il RRC dell'UTRAN sceglie le risorse radio necessarie per configurare i livelli MAC e RLC sottostanti. Tra i vari parametri, alcuni sono finalizzati ad instaurare un *Signalling Radio Bearer*, che utilizzi il canale logico DCCH e di trasporto DCH, per lo scambio della segnalazione tra le pari entità RRC. Nel sistema realizzato, questo servizio di trasporto è emulato dal metodo `DCCH_transfer` della classe `UMTS_RRC`, che si limita ad usare il puntatore al pari livello nella pila duale. I parametri fissati vengono poi trasmessi all'UE per mezzo del messaggio di `RRC_Connection_Setup`, usando il MAC SAP per il canale logico CCCH. Il canale di trasporto usato dal `MAC_UTRAN` per rispondere all'utente è il FACH. Poichè tale canale non è implementato, si simula il trasferimento sul FACH con un ritardo di trasmissione prefissato.

Alla ricezione del messaggio di RRC_Connection_Setup, l'entità RRC nel mobile configura i livelli MAC e RLC per mezzo delle primitive CMAC_Config_Req e CRLC_Config_Req, usando i parametri letti dal messaggio. Inoltre entra in uno dei sottostati del macrostato *UTRA_RRC_Connected_Mode*, a seconda del canale di trasporto che la rete gli ha assegnato. Informa infine la rete dell'avvenuta ricezione del segnale di Setup, con un ultimo messaggio di RRC_Connection_Setup_Complete sul canale logico DCCH appena instaurato.

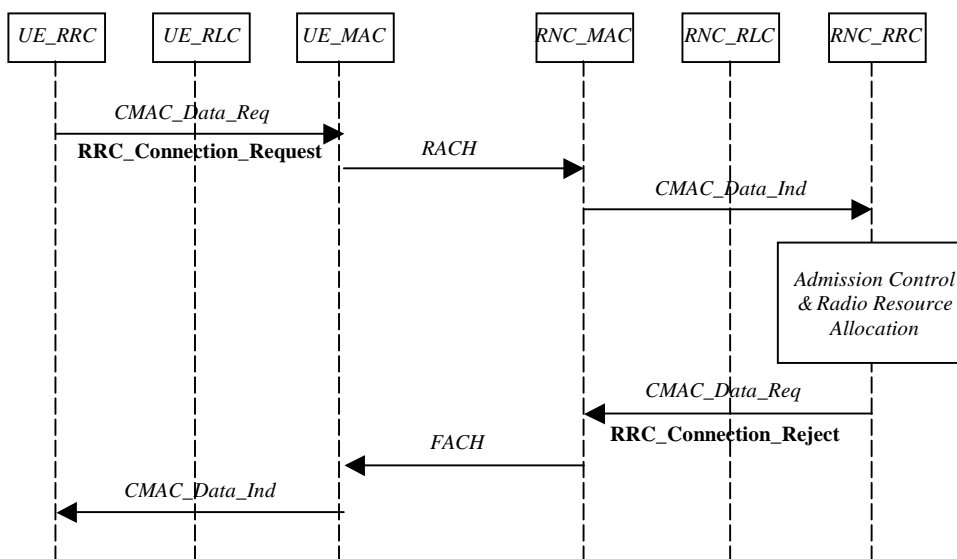


Figura 5.19 : Diagramma di sequenza sul rifiuto della connessione RRC

Se invece la rete non può stabilire la connessione con l'UE, per esempio, per mancanza di risorse, allora invia un messaggio di RRC_Connection_Reject, come si vede dal diagramma 5.19. Il RRC nel RNC trasferisce tale messaggio verso il *Radio Resource Control* dell'utente, utilizzando il MAC SAP per il canale logico CCCH; il MAC, a sua volta, lo inoltra sul FACH. Il dato di controllo RRC_Connection_Reject contiene i seguenti campi:

- *UE_identity*: identificativo dell'utente;
- *cause*: indica il motivo del rifiuto della connessione. E' un attributo del messaggio di tipo enumerazione REJECT_TYPE. Suoi possibili enumeratori sono: CONGESTION e UNSPECIFIED.
- *wait_time*: rappresenta l'intervallo temporale che l'UE deve attendere prima di ritrasmettere una nuova richiesta di RRC Connection Establishment, da quando riceve il messaggio in esame.

Alla ricezione di questo messaggio, l'UE si comporta secondo quanto descritto al punto 3.7.5.1 del capitolo 3.

5.3.4.2 Rilascio della connessione RRC

Quando tutti i servizi, riguardanti lo stesso UE, sono stati espletati (ovvero le sorgenti di traffico sono spente e non ci sono più dati nei buffer in attesa di essere trasmessi sull'interfaccia radio 'Uu'), la rete rilascia la connessione RRC precedentemente instaurata con quell'utente.

La figura 5.20 rappresenta l'interazione tra gli strati protocollari, con i messaggi di livello 3 coinvolti e le primitive di comunicazione impiegate.

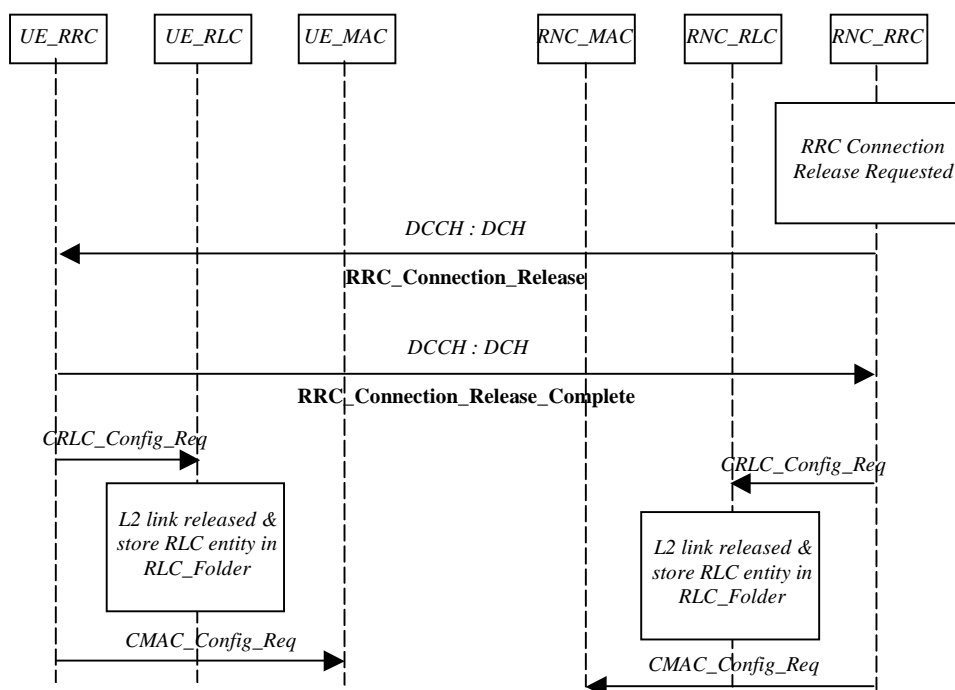


Figura 5.20 : Rilascio della connessione RRC

L'entità RRC nel RNC invia il messaggio di RRC_Connection_Release sul canale logico DCCH, instaurato in precedenza per trasportare la segnalazione. In seguito alla ricezione di questo messaggio, il RRC nell'UE è costretto ad abbattere la connessione RRC e a rilasciare le risorse di cui dispone. Prima di fare ciò, risponde alla rete inoltrando un messaggio di RRC_Connection_Release_Complete sul canale DCCH. A questo punto rilascia tutte le risorse di livello RLC e MAC, invocando le rispettive primitive CRLC_Config_Req e CMAC_Config_Req, e torna nello stato di IDLE_MODE. In particolare le entità RLC usate per il trasferimento dei dati d'utente non vengono distrutte, ma sono depositate in un opportuno

raccoglitore (denominato RLC_Folder) per consentire un loro futuro riutilizzo. Prima di essere immagazzinate, tutti gli attributi delle varie entità vengono reimpostati ai valori iniziali.

Tra i vantaggi di questa soluzione prevale la riduzione del tempo di simulazione. Infatti, quando un qualsiasi UE ha bisogno di nuove entità RLC, può prelevarle direttamente dal raccoglitore senza che siano create *ex-novo*.

Quando il *Radio Resource Control* nel RNC riceve il messaggio di RRC_Connection_Release_Complete, procede come la pari entità rilasciando tutte le risorse di livello 2. Anche in questo lato della rete, esiste un contenitore comune per raccogliere le entità di livello RLC nell'UTRAN.

5.3.4.3 Procedure per le misurazioni

Instaurata la connessione RRC tra UTRAN ed UE con gli opportuni *Radio Bearer* per gestire i servizi richiesti, il *Radio Resource Control* nel RNC deve poter riconfigurare, se necessario, tali RB per garantire la QoS dei servizi. Per l'attuazione di ciò, l'UTRAN_RRC deve eseguire una serie di misurazioni, delegando i livelli sottostanti sia nell'UE sia nell'UTRAN. In particolare, se il traffico è in *uplink*, è l'entità RRC nell'UE a gestire le misure, sotto il pieno controllo del RRC nell'UTRAN; se invece le misurazioni riguardano i canali in *downlink*, è l'UTRAN_RRC che le gestisce interamente. Infatti questi ordina direttamente agli strati sottostanti di monitorare per esempio il volume di traffico o la qualità del canale radio. Attraverso poi la consegna a detta entità delle misure realizzate dal MAC e dal RLC, la rete può provvedere a riorganizzare dinamicamente le risorse assegnate.

Delle sette diverse categorie di misurazione previste dagli standard (vedere paragrafo 3.7.5.4 del capitolo 3), si è implementato il monitoraggio del volume di traffico e la stima della qualità del mezzo trasmissivo (misurazione *intra-frequency*). Nei punti seguenti si tratta l'interazione tra dominio d'utente e dominio di rete per eseguire le misurazioni nell'UE.

5.3.4.3.1 Monitoraggio del volume di traffico

Il diagramma 5.21 illustra le interazioni tra le entità protocollari coinvolte nel realizzare questa procedura. I messaggi di livello 3 scambiati sono evidenziati in nero nella figura.

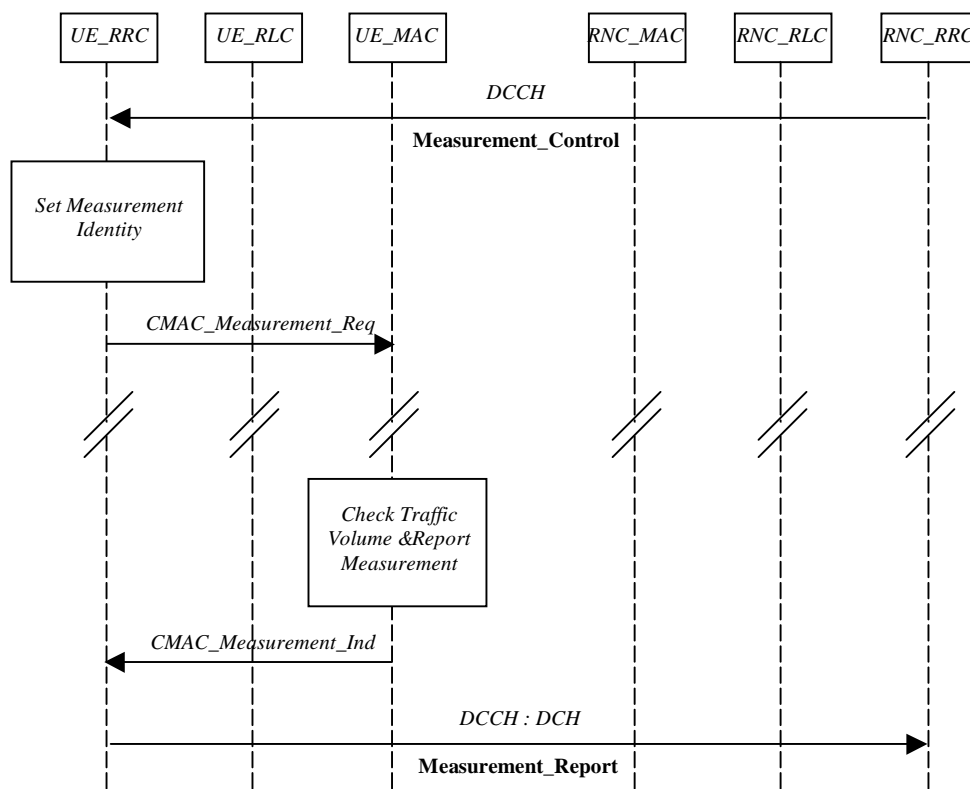


Figura 5.21 : Diagramma di sequenza per gestire il monitoraggio del traffico nell'UE

L'entità RRC nel *Radio Network Controller*, dopo aver letto dal *DataBase* di sistema i parametri di configurazione dei livelli protocollari nell'UE, adibiti a realizzare la misurazione in esame, invia al terminale un messaggio di *Measurement_Control* per mezzo del canale logico *DCCH*. Tale messaggio presenta i seguenti attributi:

- *M_id*: identificativo della misurazione da eseguire. In questo caso è *TRAFFIC_VOLUME_MEASUREMENT*;
- *command*: variabile di tipo enumerazione. I suoi enumeratori possono essere: *SETUP*, *MODIFY*, *ERASE*.
- *meas*: puntatore ad un oggetto della classe *Measurement_Type*, contenente i parametri per consentire al sottolivello designato (il *MAC*) di realizzare le misure.

Alla ricezione di suddetto messaggio, il RRC nell'UE memorizza nella variabile *Midentity* le caratteristiche della misurazione richiesta e configura il livello *MAC* sottostante per mezzo della primitiva *CMAC_Measurement_Req*. Il sottolivello 2 può adesso monitorare il volume di traffico sui vari canali di trasporto impiegati.

L'algoritmo di controllo seguito nelle misurazioni è quello definito al punto 3.3.3.1 del capitolo 3.

Nel momento in cui si verificano le condizioni per trasmettere le misure al livello RRC soprastante, il MAC invoca la primitiva `CMAC_Measurement_Ind` per interagire con il *Radio Resource Control* dell'UE. Quest'ultimo, ricevuti i risultati, li invia alla pari entità, trasmettendo il messaggio di `Measurement_Report` sul canale dedicato allo scambio della segnalazione di livello 3. `Measurement_Report` è un oggetto con i seguenti campi:

- *M_id*: identificativo della categoria della misurazione;
- *result*: puntatore ad un oggetto contenente i risultati delle misure (*Buffer Occupancy*, canale logico e di trasporto controllati).

5.3.4.3.2 Misurazione intra-frequency

Questa misurazione realizza la stima della qualità del mezzo trasmissivo. Il suo obiettivo è quello di gestire la trasmissione sul canale radio in funzione dello stato del mezzo trasmissivo. Non essendo implementato il livello fisico nella pila protocollare simulata, si è deciso di effettuare tale operazione nel livello RLC. Tale strato può essere sospeso o riattivato in base alle stime compiute. Anche il MAC campiona il canale per determinarne la qualità, ma esegue l'operazione in modo indipendente dal sottolivello RLC.

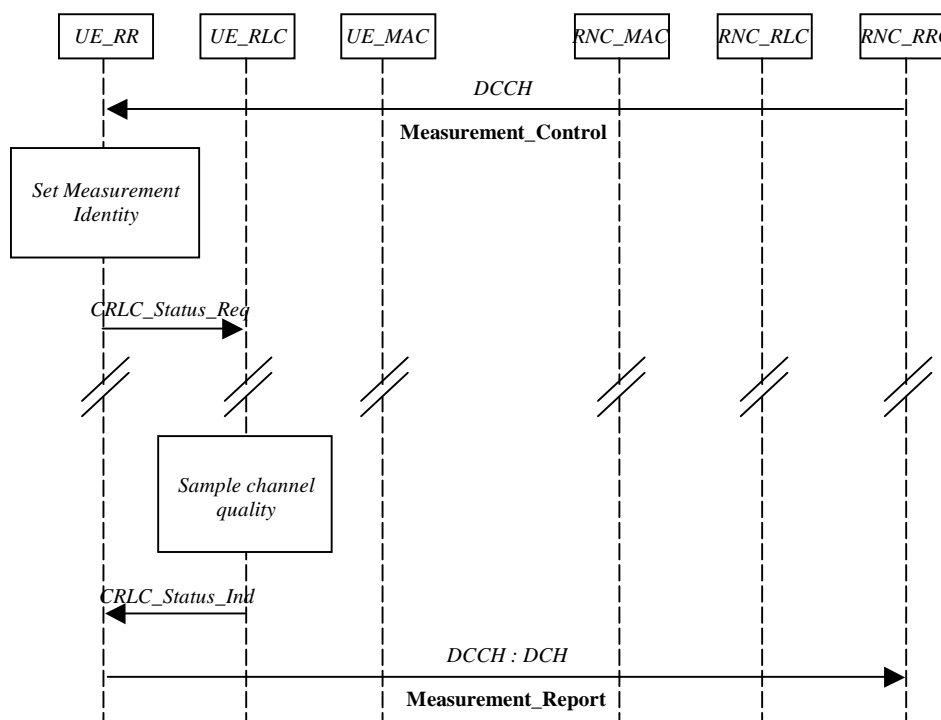


Figura 5.21 : Diagramma di sequenza per la procedura di intra-frequency

L'interazione tra i livelli è mostrata in figura 5.22. Si noti la stretta analogia nella sequenza delle operazioni con la procedura precedente.

L'UTRAN_RRC invia il messaggio di Measurement_Control con le informazioni atte a realizzare la procedura di stima del canale. Alla ricezione di tale messaggio, la pari entità RRC nell'UE registra nella variabile Midentity i parametri dell'operazione di *intra-frequency* e configura il livello RLC affinché stimi la qualità del canale radio. La frequenza di campionamento è un attributo specificato dalla rete e letto dall'utente nel messaggio. Ogni volta che al RRC nell'UE pervengono le misure realizzate dal RLC sul mezzo, il primo le inoltra al pari livello nel RNC affinché decida sul da farsi (sospendere o meno la trasmissione). Il messaggio di livello 3 spedito per trasmettere i risultati è sempre il Measurement_Report. Il puntatore *result* in esso contenuto indirizza adesso un oggetto *software* che immagazzina la stima del canale (GOOG o BAD) e l'identificativo della misurazione (INTRA_FREQUENCY_MEASUREMENT).

5.3.4.4 Riconfigurazione dei Radio Bearer

La procedura ora esaminata viene utilizzata dal RRC nell'UTRAN per riorganizzare le risorse radio nel sistema, in seguito alle misurazioni effettuate nei livelli protocollari dell'UE e dell'UTRAN. Con questa procedura si può:

- cambiare la velocità trasmissiva assegnata ad un particolare utente;
- sospendere o riattivare la trasmissione dei dati sui *Radio Bearer*;
- assegnare nuovi TFCS ai mobili in base al *rate* a quest'ultimi destinato;

In tale sottoparagrafo si analizza il caso particolare della sospensione della trasmissione dei servizi in *uplink* in un specifico UE. Il diagramma 5.22 mostra la sequenza delle operazioni, i messaggi scambiati e le primitive coinvolte.

Quando il canale radio, visto da un certo utente, passa dallo stato BAD a quello GOOD e l'UTRAN_RRC si accorge di questa transizione, invia all'UE interessato il messaggio di RadioBearer_Reconfiguration sul canale DCCH. Detto messaggio contiene il puntatore ad un oggetto della classe CMAC_config_data. Tra gli attributi di tale classe, descritta precedentemente in questo capitolo, esiste un elemento *action* per informare la pari entità su quale azione deve intraprendere. Tra i valori possibili di *action* troviamo:

- *SUSPEND*: per sospendere la trasmissione dei dati ponendo le entità RLC in un UE nello stato di SUSPEND;
- *RESUME*: per riprendere il trasferimento dei dati, riattivando le entità RLC opportune;
- *CHANGE_RATE*: per assegnare un *rate* trasmissivo diverso ad un particolare utente;

- *CHANGE_TFCS*: per cambiare il *set* delle combinazioni dei formati di trasporto.

Nel caso considerato, *action* assume il valore *SUSPEND*. In seguito alla ricezione del messaggio di *RadioBearer_Reconfiguration*, l'entità RRC sospende le entità RLC nello strato RLC_block che le contiene, per mezzo della primitiva *CRLC_Suspend_Req*. Occorre ricordare che questa primitiva può interrompere la trasmissione delle sole entità che operano nella modalità *acknowledged* e *unacknowledged*. Le entità *transparent* invece non possono essere sospese. Per confermare alla rete l'avvenuta sospensione, il RRC nell'UE risponde con il messaggio di *RadioBearer_Reconfiguration_Complete*.

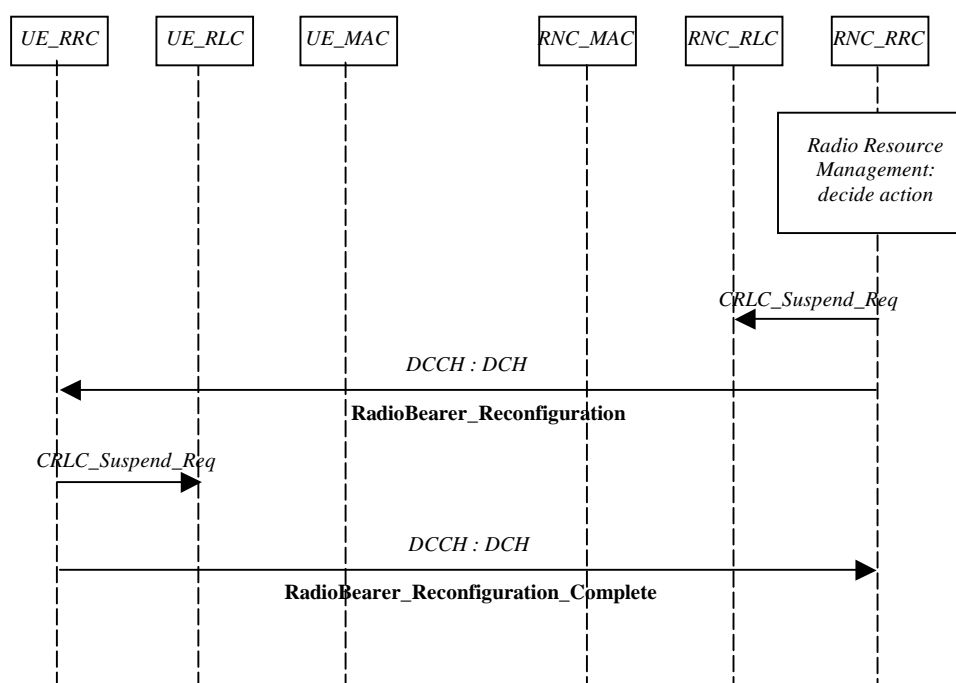


Figura 5.22 : Diagramma di interazione per la procedura di Radio Bearer Reconfiguration

5.3.4.5 Procedura di Handover

L'operazione di *handover* nel simulatore non è realizzata sfruttando le tipiche procedure di livello RRC previste dagli *standard*. Essa è attualmente simulata mediante l'uso di eventi opportuni che fissano per ogni utente l'istante di tempo necessario in cui i mobili devono cambiare il Node-B presso cui attestarsi. Fare riferimento al paragrafo 4.3 del capitolo 3 per una descrizione più dettagliata di tali eventi.

Le figure 5.23 e 5.24 rappresentano la struttura del sistema, prima e dopo la procedura di *handover* dell'UE numero 1.

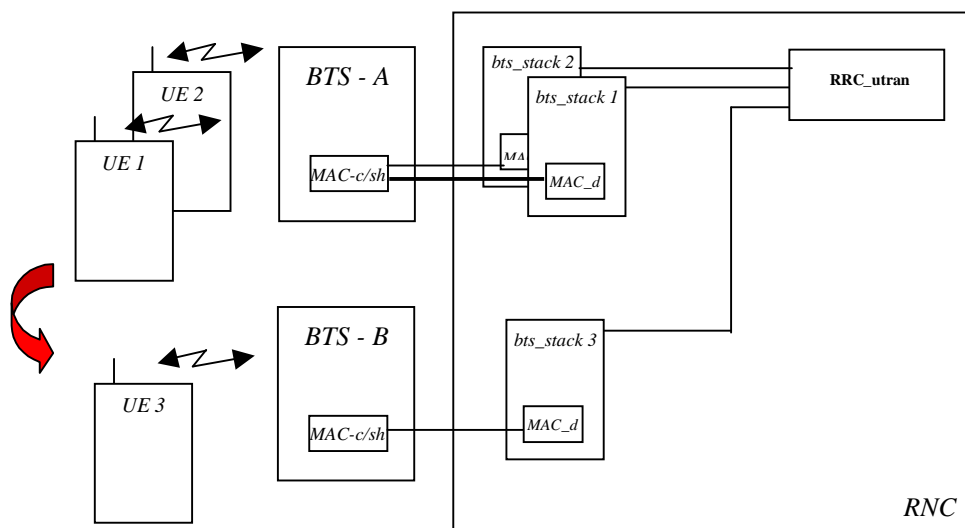


Figura 5.23 : Configurazione del sistema prima della procedura di handover

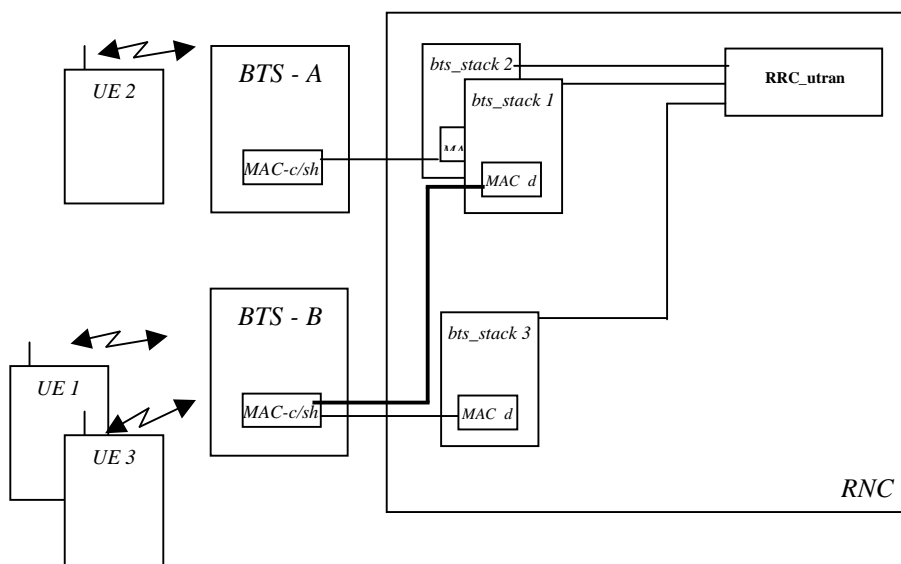


Figura 5.24 : Configurazione del sistema dopo la procedura di handover

5.4 Scelta del TFC nel MAC

In UMTS le risorse radio sono assegnate agli utenti dal *Radio Resource Control* dell'UTRAN. Il livello MAC è responsabile di gestire l'accesso di servizi differenti alle risorse destinategli dal livello RRC. Negli UE che supportano più di un servizio contemporaneamente, sono attivi tanti canali logici quanti sono i traffici con differenti requisiti di qualità di servizio. Nel MAC i canali logici sono messi in corrispondenza con i canali di trasporto. La tabella delle corrispondenze tra questi canali è fissata dallo strato RRC ed è realizzata dalla classe *RB_mapping_info* nel simulatore. E' possibile che più canali logici siano multiplati sullo stesso canale di trasporto (figura 5.25).

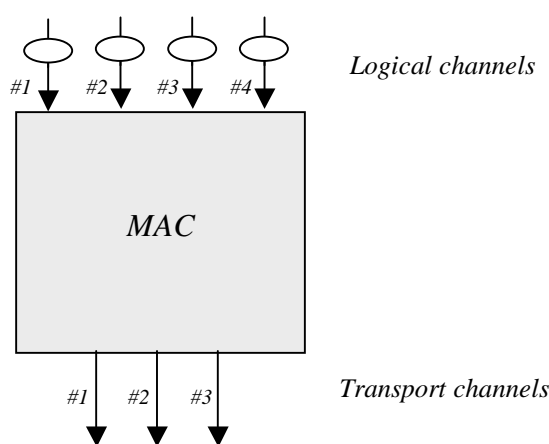


Figura 5.25 : Architettura dei canali logici e di trasporto in un UE

Un canale di trasporto definisce la modalità in cui il traffico, appartenente a canali logici distinti, viene processato e inviato al livello fisico. La più piccola unità di traffico che può essere trasmessa su un canale di trasporto è chiamata *Transport Block* (vedere paragrafo 3.2 del capitolo 3). Esso consiste di un blocco RLC più una intestazione di livello MAC. Ogni canale di trasporto è caratterizzato da una serie di attributi che definiscono il *Transport Format* associato al canale. Detti attributi possono essere semi-statici o dinamici. Questi ultimi in particolare fissano la dimensione del TB e possono variare ad ogni *Transmission Time Interval*. La loro variazione porta a definire più TFs per lo stesso canale di trasporto e a variare il *bit rate* istantaneo di quest'ultimo. Per ciascun canale di trasporto e ad ogni TTI quindi il MAC deve scegliere un appropriato TF. L'insieme dei *Transport Format* scelti per tutti i canali di trasporto viene detto *Transport Format Combination*, mentre l'insieme di tutti i TFC da cui il MAC deve estrarre quello più appropriato viene chiamato *Transport Format Combination Set*. Il TFCS è assegnato al MAC dal livello RRC.

Un importante compito del livello MAC è scegliere il TFC. Questa azione ha luogo ad ogni TTI e deve tenere in conto i seguenti parametri:

1. priorità dei canali logici;
2. carico corrente dei buffer RLC;
3. livello di potenza indicato dal livello fisico.

L'algoritmo di *TFC selection* implementato nel simulatore considera attualmente i soli primi due punti dell'elenco, non essendo ancora prevista alcuna limitazione sui requisiti di potenza. Con questo algoritmo i canali logici sono classificati secondo un ordine decrescente delle loro priorità e, a parità di precedenza, sono ordinati in base all'occupazione dei *buffer* a livello RLC. L'algoritmo di ordinamento utilizzato è l'*insertion sort*, che risulta tra i più efficienti nel caso in cui si debba ordinare un piccolo numero di elementi. Esso infatti poggia su una modalità di ordinamento *in loco*: i canali sono cioè risistemati dentro il vettore con al più un numero costante di essi memorizzati, ad ogni istante, in variabili ausiliarie fuori dal vettore. Questo consente di risparmiare risorse e velocizzare l'esecuzione. Dopo l'ordinamento dei canali logici, il MAC sceglie tra tutte le combinazioni dei formati di trasporto di cui dispone, quella che massimizza il numero di *Transport Block* trasmessi. In particolare il MAC assegna ai vari canali logici il numero di blocchi da trasmettere secondo il seguente algoritmo:

- (a) estrae dal vettore un canale logico, secondo l'ordine prefissato;
- (b) legge il numero di pacchetti che il *logical channel* in esame vuole trasmettere;
- (c) assegna al canale logico un numero di blocchi da inviare in funzione della richiesta fatta al punto (b), tenendo contemporaneamente presente la disponibilità del TF associato al canale di trasporto che lo stesso canale logico vuole utilizzare. Se la richiesta supera le risorse disponibili, il MAC gli assegna solo quelle che il TF può ancora supportare. Infine decrementa il numero dei blocchi disponibili nel TF di una quantità pari al numero delle risorse assegnate. Quando il numero nel TF considerato arriva a zero, vuol dire che su quel canale di trasporto non si possono più inviare blocchi;
- (d) riprende dal punto (a) fino a quando non ha scandito tutti i canali logici esistenti.