

TRABAJO ESPECIAL DE GRADO

**MODELO Y ANÁLISIS DEL HANDOVER VERTICAL DEL TIPO
“RANK AGGREGATION” BASADO EN CADENAS DE MARKOV
PARA REDES WIRELESS HETEROGENEAS**

Presentado ante la Ilustre
Universidad Central de Venezuela
por la Br. Mora M., Cynthia J.
para optar por el Título de
Ingeniero Electricista

Caracas, 2010

TRABAJO ESPECIAL DE GRADO

MODELO Y ANÁLISIS DEL HANDOVER VERTICAL DEL TIPO “RANK AGGREGATION” BASADO EN CADENAS DE MARKOV PARA REDES WIRELESS HETEROGENEAS

Prof. Guía: Zeldivar Bruzual

Presentado ante la Ilustre
Universidad Central de Venezuela
por la Br. Mora M., Cynthia J.
para optar por el Título de
Ingeniero Electricista

Caracas, 2010

CONSTANCIA DE APROBACIÓN

Caracas, 17 de mayo de 2010

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por la Bachiller Cynthia J. Mora M., titulado:

“MODELO Y ANALISIS DEL HANDOVER VERTICAL DE TIPO “RANK AGGREGATION” BASADO EN LAS CADENAS DE MARKOV PARA REDES WIRELESS HETEROGENEAS”

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Comunicaciones, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.

Prof. Luis Fernández
Jurado

Prof. Gerlis Caropresse
Jurado

Prof. Zeldivar Bruzual
Prof. Guía

DEDICATORIA

A las tres personas más importantes en mi vida, Freddy, Omaira y Angely. Nuestros logros siempre serán de los cuatro, donde sea que estemos.

RECONOCIMIENTOS Y AGRADECIMIENTOS

Al dottore ingegnere Gabriele Tamea, per il suo instancabile lavoro ogni giorno fino alle 19 nel laboratorio e oltre, per tutte le sue conoscenze e soprattutto per la pazienza, gracias!, sei una bravissima persona, come dice Ale, loro avranno la maggiore quantità ma voi italiani avete Tamea.

A mi familia por enseñarme a vivir, apoyarme en cualquier proyecto o momento y brindarme tantas oportunidades y cariño, a ustedes tres se los debo todo. A mis hermanas, primas y primos, que más que eso son mis grandes amigos. A mis hermosísimas tias y tios, gracias por el apoyo en el completo camino. A mis abuelitas gracias por el gran carino, y abuelito sabes que te recuerdo mucho, con Angel espero que me sigan cuidando. Si no los nombro a todos es porque somos 61 contando a las cusi cusi y los extranjeros.

Al professore Roberto Cusani per avermi dato l'opportunità di svolgere questa tesi.

A mi adorada escuela de la Universidad Central de Venezuela, y a la "Sapienza", a mis profesores. Como dije llegué siendo una niña y me voy sintiéndome toda una mujer.

A mi segunda familia, agradezco a Dios por habérmelos puesto en el camino, sin su apoyo no hubiese pasado tan feliz los momentos más duros de mi vida, que han pasado en Roma. Richard, Jose, Jesus, William, Antonio, Cath, Ruby, Tato y Joao, mis coinquilinos, mis hermanos, y las que no pagaban affitto pero estaban todos los días: Anas y Giorgia, que pesada, es broma. A Mer por la presión y por la seguridad que siempre me quisiste infundir, a Jesus y Carmen. A via Enea y Furio, Cabe (por no pelear conmigo ese mes), J, Reni y mi Nohe, mi

Ari, Luis, Daniel y Josh, los quiero mucho, espero poderlos ir a ver en el aula magna. Ponte Lungo, con este poco tiempo se ganaron mis gracias, mis queridos parásitos, Chris, Marcel, 2da mencion Giorgia; Ron (chicharron #1), fufi y mi mamá espero que estés orgullosa Su. A mis amigas, ustedes saben cuando agradezco las largas conversaciones y tanto cariño, las voy a extrañar, Paola, Pato, Vale, Mari, a los Bueno, siempre los seguiremos, a Luis y a mi gocho. A las viejas generaciones gracias por los consejos. Muchachos fueron los tres años más maravillosos de mi vida, solo gracias a ustedes!!!!... A celebrar...

A mis grandes amigas (no es que los primeros no lo sean M): Aliosha, Alelu, Mariana, Diana, Alexandra, Olga, Fefa, Luisana, espero mantener la amistad para siempre, gracias a sus madres tambien, no se me olvidan Suki, Nan, Eli, Erika, Mate, Gabys.

A mis grandes amigos, peque, chiro, Dave, Ricardo, Giul, Dan, Tom's, Jump, Moro, Tos, Rony, Herni, Merh, Serg, Miku, Edu, Elio (incluye a vane) y Landi...gracias por todo.

Ai tesori che spero non perdere dell'Italia, Marco, Ciccio, Sara, Serena, Riccardo, Leti, Alessandra, Ire, Stef... grazie dalla pasta. Venitemi a trovare.

A Marlene y a Nina, gracias por las arepitas y por el hogar!!

A tantas personas, no los olvido se me acaba la pagina.

Me copio de Jose, sería justo despues que me robó las láminas!... Grazie Dio, grazie Italia, forza Roma...

Mora M., Cynthia J.

**MODELO Y ANÁLISIS DEL HANDOVER VERTICAL DEL TIPO
“RANK AGGREGATION” BASADO EN LAS CADENAS DE
MARKOV PARA REDES WIRELESS HETEROGÉNEAS.**

**Prof. Guía: Zeldivar Bruzual. Tesis. Caracas. U.C.V. Facultad de Ingeniería.
Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción:
Comunicaciones. Institución: *Università degli Studi di Roma “La Sapienza”*.
2010. 124h**

Palabras Claves: Handover Vertical; Técnica Rank Aggregation; Cadenas de Markov.

Resumen. En esta tesis se diseñó un modelo que se concentra en la segunda fase del Handover vertical, la selección de la red más adecuada a la cual se quiere realizar el acceso, de un conjunto de redes wireless heterogéneas disponibles y el respectivo análisis de las prestaciones del algoritmo desarrollado. Para ello se utilizaron técnicas de decisión multi-atributo, específicamente el rank aggregation, basado en cadenas de Markov. De cada red, el móvil recibía medidas de valores de retardo, pérdida de paquetes y banda, a partir de las cuales creaba listas ordenadas de las redes y las agregaba, para obtener una lista final, y cada posición era introducida en las cadenas de Markov para determinar la matriz de probabilidades de transición, posteriormente el vector de probabilidades de estado estacionario a partir del cual se seleccionaba la red. El retardo fue estimado con el modelo de los sistemas de colas, los paquetes perdidos con el modelo de Gilbert y la banda aleatoriamente. Se realizó una modificación a la matriz de ranking original, haciéndola más sensible a los valores de los parámetros que ofrecía cada red. Se modificó una de las cadenas de Markov empleadas, para evitar valores de probabilidad de transición entre los estados no deseados y por último, se introdujo un vínculo a los parámetros respecto a la clase de servicio considerada, entre las cuales: conversacional, streaming, interactiva y background, dándoles un peso diverso dentro de las cadenas empleadas, dependiendo de la clase.

Mora M., Cynthia J.

MODELLO ED ANALISI DELL'HANDOVER VERTICALE DEL TIPO "RANK AGGREGATION" BASATO SULLE CATENE DI MARKOV PER RETI WIRELESS ETEROGENEE.

Prof. Guida: Zeldivar Bruzual. Tesis. Caracas. U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción: Comunicaciones. Institución: *Università degli Studi di Roma "La Sapienza"* 2010. 124h

Parole Chiavi: Handover Vertical; Técnica Rank Aggregation; Cadenas de Markov.

Riassunto. In questa tesi si è progettato un modello che si incentra nella seconda fase del'Handover verticale, la scelta della rete più adatta verso la quale si realizzerà lo spostamento, da un insieme di reti wireless eterogenee disponibili e anche si ha fatto l'analisi delle prestazioni dell'algoritmo sviluppato. Si sono impiegate tecniche di decisione multi-attributo, più specificamente il rank aggregation, basato su catene di Markov. Da ciascuna rete, il terminale mobile riceve le misure dei valori di ritardo, packet loss e banda, a partire delle quali si generano le liste ordinate delle reti e si combinano, per ottenere una lista finale, e ciascun ordine è introdotto nelle catene di Markov per determinare la matrice di probabilità di transizione, tutto ciò per calcolare il vettore di probabilità degli stati stazionari a partire del quale si seleziona la rete. Il ritardo è stato stimato col modello dei sistemi a coda, i packet loss col modello di Gilbert e la banda aleatoriamente. Si ha modificato la matrice dei ranking originale, rendendola più sensibile ai valori dei parametri che offre ogni rete. Si ha modificato una delle cadete di Markov impiegate, per evitare di assegnare valori di probabilità di transizione tra stati non desiderati y finalmente, si è introdotto un vincolo ai parametri rispetto alla classe di servizio considerata, tali classi sono state: conversacional, streaming, interactiva y background, assegnandoli un peso diverso dentro le catene impiegate, dipendendo della classe.



SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria

**Corso di Laurea Specialistica in Ingegneria delle
Telecomunicazioni**

Tesi di laurea:

**Modello ed Analisi dell'Handover Verticale
del tipo "Rank Aggregation" basato sulle Catene di Markov
per Reti Wireless Eterogenee**

Relatore:

Prof. Roberto Cusani

Co-relatore:

Dott. Ing. Gabriele Tamea

Laureanda:

Cynthia J. Mora M.

Anno accademico 2009-2010

SOMMARIO

INTRODUZIONE	12
CAPITOLO I	15
HANDOVER	15
I.1 INTRODUZIONE	15
I.2 HANDOVER	15
I.2.1 DEFINIZIONE GENERALE:	15
I.2.2 FASI DEL HANDOVER:	19
I.2.3 SUPPORTO AI VARI LIVELLO PROTOCOLLARI DELI'HANDOVER:	20
I.2.4 HANDOVER VERTICALE:	23
I.2.5 MEDIA INDEPENDENT HANDOVER:	25
CAPITOLO II	29
LA DECISIONE DELL'HANDOVER	29
II.1 INTRODUZIONE	29
II.2 CRITERI DI DECISIONE PER L'HANDOVER VERTICALE	29
II.3 TECNICHE DI DECISIONE	31
II.3.1 SINGOLA FUNZIONE	31
II.3.2 CENTRATO NELL'UTENTE:	33
II.3.3 MULTIPLI ATTRIBUTI: MADM MULTIPLE ATTRIBUTE DECISION MAKING:	34
II.3.4 FUZZY LOGIC:	40
II.3.5 SENSIBILITÀ AL CONTESTO:	42
STATO DELL'ARTE	45
RANK AGGREGATION BASATO SU CATENE DI MARKOV	45
CAPITOLO III:	51

RANK AGGREGATION BASATO SU CATENE DI MARKOV MODIFICATO	51
III.1 INTRODUZIONE	51
III.2 RANK AGGREGATION E TEORIA DEI GRAFI	52
III.3 MOTIVAZIONI ALLA PROPOSTA	53
III.4 MODIFICHE	54
CAPITOLO IV	66
RISULTATI E SIMULAZIONI	66
IV.1 INTRODUZIONE	66
IV.2 CASO D'USO	68
IV.3 SCELTA DELLA RETE MIGLIORE	71
IV.4 MISURA DELLA SENSIBILITÀ DEI PESI	73
IV.5 INFLUENZA DEL RANKING E DEI PARAMETRI NELLA SELEZIONE	76
IV.6 SCELTA DELLA RETE MIGLIORE CON CLASSE DI SERVIZIO	81
CONCLUSIONI	88
APPENDICE	90
RIFERIMENTI	120

INTRODUZIONE



Il crescente sviluppo delle comunicazioni mobili di future generazioni, dovuto principalmente alle tecnologie degli elementi coinvolti, al numero di utenti attivi sempre più alto ed alle esigenze di ciascuno, spinge gli operatori ad essere più competitivi, mantenendo le comunicazioni attive, con la migliore qualità possibile offrendo, minori costi, maggiore qualità di servizio e finalmente migliori prestazioni, con la premessa di ricevere sempre il corrispondente guadagno. Un concetto che inquadra perfettamente questo scenario, è il paradigma *Always Best Connected*, dove un terminale con le interfacce necessarie possa collegarsi a diverse reti cambiando sempre a quella che offre le migliori condizioni. Questa procedura di collegamento “intelligente” è denominata Handover, ed in particolare questo lavoro si focalizza sull’handover verticale, dove le reti alternative che si possono scegliere sono di tipo eterogenee, lasciando quelle omogenee per la modalità di handover orizzontale.

Sono molti gli studi di ricerca effettuati per conseguire la standardizzazione di tale procedura, ritenendo come principale requisito l’indipendenza dalla tecnologia che si va a scegliere, tuttavia ancora non si è raggiunto tale scopo. Al cercare in letteratura tali lavori si evidenzia come la decisione di gruppo è la caratteristica comune, molto risalente che incluso può essere considerata come principale limitazione nella selezione. Più specificamente il problema risiede in scegliere i parametri che comprendono tali insieme di decisione per ciascun scenario specifico e conseguentemente come combinarli in maniera più giusta ed adeguata.

Alcune delle tecniche di decisione multi attributo già sviluppate si possono adeguare perfettamente al handover verticale, infatti numerosi autori hanno impiegato quelle più popolari sfruttando i suoi vantaggi e modificandole a seconda dello scenario considerato.

In questo lavoro si è optato per una tecnica normalmente utilizzata nel contesto della ricerca nel web, dove tante liste di posizionamento sui siti web si combinano per creare una lista con un ordinamento finale, portando alla selezione del sito desiderato. La precedente procedura è chiamata Rank aggregation, non essendo tra le tecniche più popolari impiegate nel handover verticale, offre grandi vantaggi dal punto di vista della complessità ed in situazioni particolari esplicitate nei prossimi capitoli.

Riguardando sempre nell'ambito dell'handover verticale, un approccio già applicato nel rank aggregation sono le catene di Markov. Ogni stato della catena rappresenta una delle possibili reti alternative da selezionare ed ogni transizione è dominata dalla probabilità con cui avviene lo spostamento, che si basa ogni volta su ciascun parametro od attributo considerato.

Questa ricerca s'incentra sulla procedura di selezione di rete appena descritta modificandone tanto il ranking come le catene impiegate, ed aggiungendo un vincolo sulle particolari classi di servizio.

Il capitolo I spiega le motivazioni, i concetti e le caratteristiche generali del handover, le fasi e le diverse classifiche a seconda del tipo di controllo, le operazioni specifiche e gli scenari esaminati, descrive il supporto protocollare e gli avanzamenti o le basi dello standard finora conosciuti. Il capitolo II riguarda gli attributi ed i criteri di decisione considerati per l'handover, le loro caratteristiche e le classifiche, utilizzando questi criteri si analizzano le tecniche di decisione impiegate comunemente nella procedura, esplicitando per alcuni casi tutti i passi seguiti da ciascuna tecnica. Nel capitolo III si enumerano i vantaggi del rank aggregation e le motivazioni della proposta principale, in seguito si esplicitano le modifiche e si struttura la proposta in maniera formale descrivendo e giustificando ogni cambiamento, per finire in quella sezione si descrivono le quattro classi di QoS inclusi negli scenari considerati nelle prove realizzate. L'ultimo capitolo illustra la proposta presentando un esempio dettagliato di uno scenario realistico, mostra i risultati delle prove applicate agli algoritmi programmati in maniera da testare i benefici che questi coinvolgono ed evidenziare il loro comportamento in maniera grafica;

in una prima istanza si convalidano i pesi calcolati per ciascun parametro a seconda della classe di servizio in studio, successivamente si svolgono una serie di prove che riflettono graficamente l'importanza di ogni modifica eseguita ed ogni contributo aggiunto nella selezione finale.

CAPITOLO I

HANDOVER



I.1 INTRODUZIONE

Considerando uno scenario di comunicazione con utenti che godono di mobilità, che hanno a disposizione diversi punti di accesso da dove attaccarsi, con terminali dotati di interfacce che rendono possibile il collegamento con queste diverse reti di accesso, nella ricerca di mantenere sempre attive le connessioni e provando a raggiungere le migliori prestazioni tanto per gli utenti come per i gestori delle reti, si esegue la detta procedura di Handover. Questa procedura non è altro che lo spostamento di una determinata connessione ad un altro canale, cella o punto di accesso, causato dagli spostamenti fisici dei terminali o quando la qualità del collegamento scende al di sotto di una soglia minima.

I.2 HANDOVER

I.2.1 DEFINIZIONE GENERALE:

In generale, l'handover è una procedura impiegata nelle reti radiomobili, per assegnare ai terminali nuove risorse di traffico, da un insieme disponibile, alla fine di mantenere sempre tutte le connessioni attive.

L'handover può essere classificato secondo il suo scenario di mobilità in: Orizzontale, in cui lo spostamento si fa tra le celle di una stessa rete senza filo, per questo, si prende in considerazione la misura della forza del segnale ricevuto (*Received Signal Strength*) e si sceglie quella cella con maggior valore, che si suppone proporzionerà le

migliori prestazioni. Altri criteri di decisione nel caso specifico di reti cellulari, possono essere il rapporto segnale a interferenza il *Bit Error Rate*, tra altro.

Un secondo tipo di handover è quello verticale, eseguito in reti eterogenee cioè uno scenario dove ci sono diversi tipi di reti e per scegliere quella più conveniente, si prendono in considerazione tanti altri parametri a parte quelli del caso precedente. Questo handover può essere iniziato per convenienza, invece che solo per la necessità di mantenere la connessione, con motivazioni che sono inquadrare nel requisito di essere sempre *Always Best Connected ABC*, che vuol dire avere una migliore connessione alle applicazioni ovunque ed in qualsiasi momento ci sia.

Per raggiungere questa condizione desiderata: ABC, le tecniche di gestione per l'handover devono eseguire due importanti fasi: *Iniziazione dell'handover* dove si sceglie il momento appropriato per iniziare l'handover, minimizzando con questo il sovraccarico di segnalazione, evitando spostamenti non necessari e prevedendo sconnessioni, poi *Selezione della rete* di accesso più adatta per erogare uno specifico servizio tra quelli disponibili, che possa anche garantire la continuità del servizio. Tradizionalmente, per scegliere la rete migliore tra le n disponibili nella regione di studio, si considerano le metriche di handover, che sono misure relative alla qualità e caratteristiche del collegamento.

Come la motivazione principale dell'handover ricade nella gestione della mobilità in [1] si evidenzia una classifica in due tipi di handover. Il primo tipo detto Hard anche riferito come *pausa prima di far handover*, dove il terminale è collegato ad un solo punto di accesso alla volta, lasciando il punto di accesso attuale prima di attaccarsi ad un altro ed un secondo tipo: Soft anche riferito come *fare handover prima di pausa*, dove il terminale si collega a due punti di accesso per un po, prima di finalizzare la procedura. Per tutti e due i tipi, la condizione che deve essere rispettata è quella chiamata "*seamlessness*" nella quale l'utente non deve rendersi conto del trasferimento che si sta facendo tra le reti, cioè le sue prestazioni devono mantenersi al più possibili, costanti.

In [2] è stata mostrata un'altra classificazione dal punto di vista di chi inizia e chi controlla l'handover nelle sue diverse fasi (che saranno esplicitate più avanti) questo grazie a che la raccolta di informazione e le misure relative ai parametri, possono essere eseguite tramite meccanismi localizzati nei terminali oppure in una entità nella propria rete:

Nella fase di decisione:

- *MIHO Mobile Initiated HandOver*: Se è il terminale mobile chi prende la decisione, nella fase di iniziazione spiegata anteriormente.
- *NIHO Network Initiated HandOver*: Caso contrario in cui la rete ha tutta la responsabilità.
- *MAHO Mobile Assisted HandOver*: L'informazione circa il proprio stato e le misure fatte nel terminale mobile, sono passate alla rete per prendere la decisione.
- *NAHO Network Assisted HandOver*: L'informazione raccolta dalla rete è usata dal terminale, chi prende la decisione.

Nella fase di esecuzione:

- *NCHO Network Controlled HandOver*: è la rete chi ha un controllo primario dell'handover, misura tutti i parametri necessari e poi in base ai risultati prende la decisione e la esegue.
- *MCHO Mobile Controlled HandOver*: Il terminale mobile deve fare tutte le misure eppure prendere la decisione di handover.

In termini di segnalazione ed sforzi di elaborazione, i primi due approcci si comportano approssimativamente uguali, ma MIHO permette di distribuire lo sforzo di

elaborazione tra tutti i terminali mobili, al contrario che NIHO che lo centralizza tutto nella rete di accesso.

Per scegliere l'approccio preferito per l'iniziazione dell'handover [2], non è sufficiente sapere quale delle due coinvolge un maggiore sforzo per cumulare l'informazione nel rispettivo ente.

Se le risorse del collegamento radio sono scarse, la segnalazione deve essere mantenuta al livello minimo possibile e questa è direttamente influenzata dal trasferimento di informazione e le misure dei parametri dal terminale alla rete, tali come stato della batteria, posizione, velocità, così come dagli rispettivi aggiornamenti, cosa che non succede nel trasferimento dell'informazione di carico della rete, modelli di prezzo dei servizi, e altre che possono essere inviate in maniera Broadcast giacché sono le stesse per tutti i terminali, quindi da questo punto di vista si preferisce l'approccio MIHO.

Come alcuni criteri di decisione, che saranno descritte più avanti, possono essere inclusi nella procedura di decisione solo se la funzione di utilità (anche questa descritta posteriormente) è conosciuta, e per ragioni di privacy queste funzioni non devono essere trasferite alla rete, è più favorevole scegliere ancora una volta MIHO.

Dal punto di vista del fornitore di servizio è leggermente più ragionevole lasciare il controllo ai terminali mobili, specie rispetto dal controllo del carico, quindi in questo caso si sceglie NIHO.

Un altro vantaggio del MIHO è la possibilità di realizzare handover verso punti di accessi in diversi domini amministrativi portando a rate di accesso più economici e con maggiore QoS da altri fornitori.

In termini di sicurezza e possibilità di manipolazione si considerano come uguali NIHO e MIHO.

Per ultimo in termini di disponibilità è leggermente più desiderato MIHO, perché permette ai terminali mobili ripristinare le connessioni dopo che sono state interrotte.

In generale si crede che la libertà di selezione del fornitore conduce a realizzare handover iniziati dai terminali mobili, assistiti dalla rete e controllati da terminali, ma può portare alcuni benefici che siano iniziati e controllati i Soft Handover tra tanti punti di accesso ove possibili, in quanto questi sono mascherati dal terminale mobile e lasciare che un gruppo di celle appaiono come una cella virtuale di grandi dimensioni.

I.2.2 FASI DEL HANDOVER:

Molti lavori di ricerca descrivono l'handover in tre fasi principali [3],[4]:

I.2.2.1 Raccolta di informazione o Scoperta di Sistema: In questa fase si raccoglie tutta l'informazione necessaria perché ogni terminale mobile possa, tramite le sue interfacce scoprire tutte le reti presenti e i servizi che queste possono offrire, di conseguenza identificare la necessità di fare l'handover.

I.2.2.2 Decisione o Selezione di Sistema o Rete: Si decide se realizzare l'handover e si sceglie la rete di accesso più adatta verso dove spostarsi, d'accordo con diversi parametri, criteri e preferenze più avanti studiati, e si danno anche istruzioni alla seguente fase.

I.2.2.3 Esecuzione: Si esegue l'handover in maniera trasparente per il terminale, cambiando di canale o punto di accesso a quello deciso precedentemente. Questa fase anche include autenticazione, autorizzazione e il trasferimento d'informazione di contesto dell'utente.

Nella seguente figura presa da [1] vediamo i concetti principali coinvolti nella gestione e la procedura di handover:

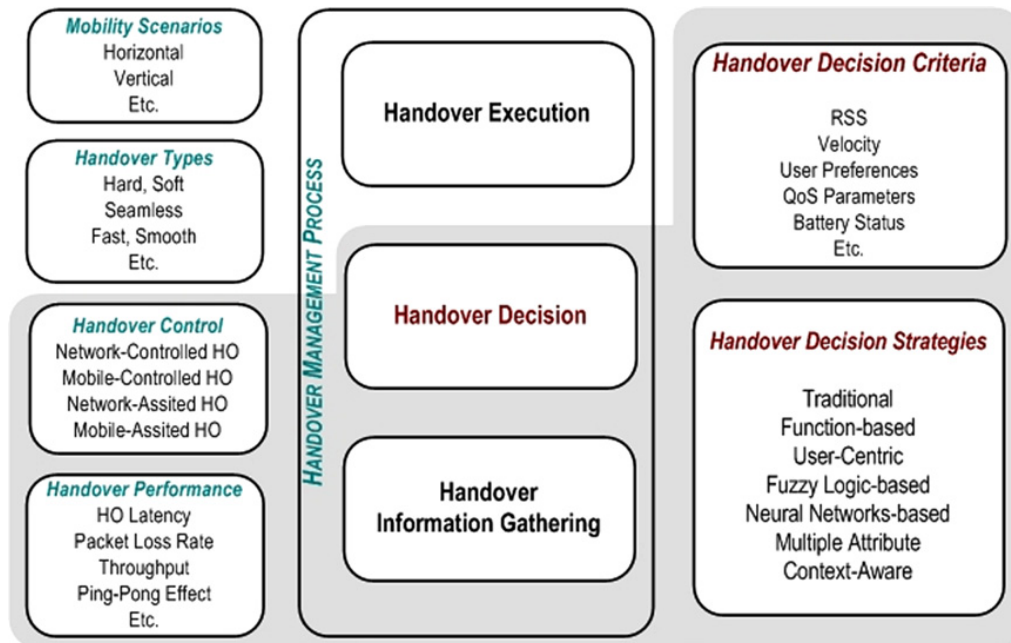


Fig. 1 Concetti di gestione di Handover

I.2.3 SUPPORTO AI VARI LIVELLO PROTOCOLLARI DELI'HANDOVER:

Si sono sviluppate tecniche per la gestione della mobilità che prendono i vantaggi delle tecnologie basate su IP, per poter fare l'handover in reti eterogenee, riducendo anche il carico di segnalazione e il ritardo dell'handover.

Tra le soluzioni per la gestione dell'handover espote in [5] che sono basati sui diversi strati, lo schema più popolare è quello per la gestione della mobilità Mobile IP: MIP descritto in [6].

Questo protocollo che consente la mobilità per internet globale e introduce tre entità funzionali: il *Home Agent* (HA), *Foreing Agent* (FA) e il nodo mobile (MN). Il protocollo è responsabile di indirizzare i pacchetti inviati per un Nodo Corrispondente CN ad un Nodo Mobile MN al suo indirizzo corrente, la procedura è la seguente:

- **Agent Discovery:** Per rilevare se MN si è trasferito ad una nuova rete di accesso, lui riceve messaggi non richiesti di *Agent Advertisement* inviati di ogni FA in maniera broadcasting. Anche MN può inviare messaggi di *Agent Solicitation* per conoscere l'eventuale presenza di agenti potenziali di mobilità.

- **Registration:** Quando MN capisce che è su una nuova rete, ottiene un nuovo indirizzo temporaneo CoA (Care-of-Address) chiedendo o ascoltando i messaggi di FA , tramite *Dynamic Host Configuration Protocol* (DHCP) o con *Point-to-Point Protocol* (PPP). Poi registra il CoA come il suo HA, impostando con questo un tunnel fino al punto finale della nuova CoA, e rimuovendo quello vecchio.

- **Routing and Tunneling:** Una volta impostato il nuovo tunnel , il HA invia i pacchetti destinati al MN, usando il nuovo CoA del MN. I pacchetti inviati dal CN al MN sono intercettati per il HA, che incapsula i pacchetti e gli invia per il tunnel al CoA del MN. Con un CoA di FA, i pacchetti incapsulati raggiungono il FA servendosi del MN, che decapsula i pacchetti e li invia al MN, come si mostra nella seguente figura.

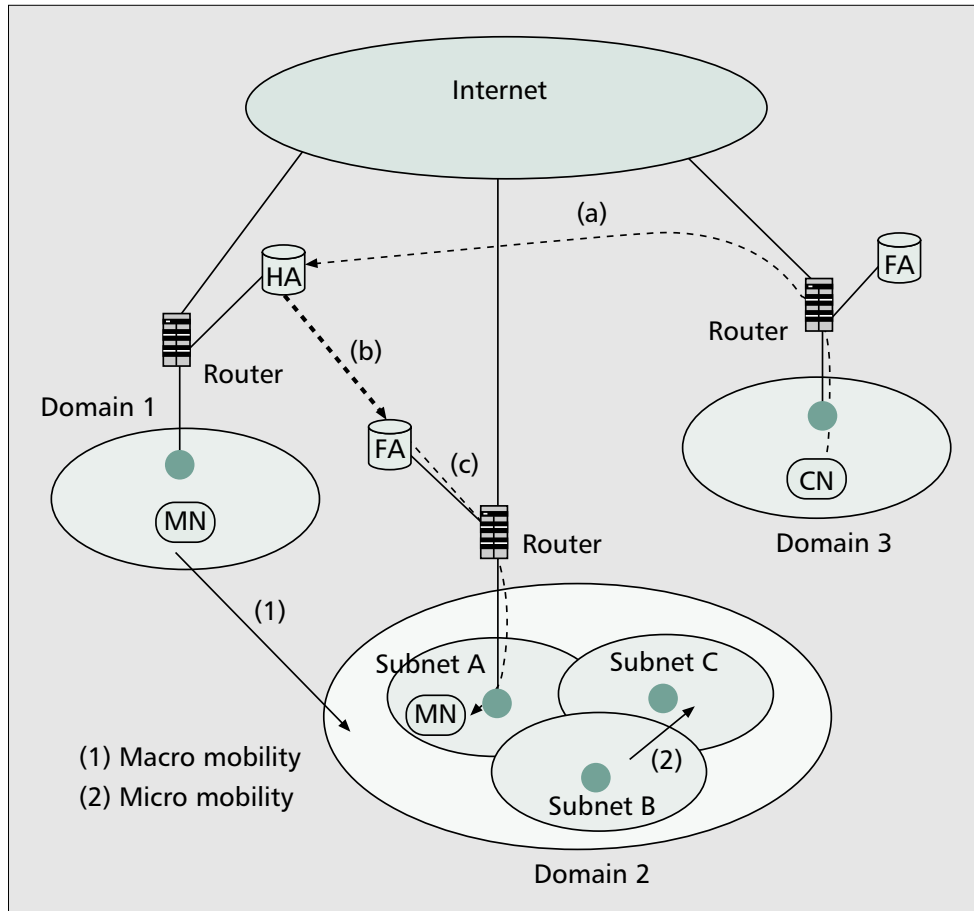


Fig. 2 Architettura Mobile IP

Un'estensione per MIP chiamata *IP Paging* è stata proposta in [7], dove il MN è permesso di entrare in modo inattivo per il risparmio della batteria per un periodo di tempo. In questo modo il sistema conosce l'ubicazione del MN con una certa accuratezza, definita per la *paging area* che è composta di diverse reti, quindi l'aggiornamento dell'ubicazione deve essere solo fatta quando si cambia di *paging area*. Il MN può anche disattivare alcuni dei suoi componenti per la stessa finalità di risparmio di batteria.

I pacchetti inviati da un CN al MN, sono prima intercettati per il HA e poi rilasciati al MN tramite il tunnel e quelli dal MN sono inviati direttamente al CN, questo istradamento triangolare si traduce in un problema significativo e che introduce un ritardo addizionale nella consegna dei pacchetti. Questo problema è stato risolto in [8] tramite un'ottimizzazione nell'istradamento, utilizzando un percorso diretto tra i MN e i suoi CN. I

CN mantengono una cache vincolata ai CoA dei MN e per inviare i pacchetti, prima controlla se c'è un ingresso di cache associato al MN, e così li invia direttamente, in caso di che non ci sia un ingresso cache associato, il CN invia i pacchetti seguendo la procedura basica di Mobile IP, cioè tramite il HA del MN.

In [9] si introduce un meccanismo per la mobilità chiamato Proxy Mobile IP (PMIP) che garantisce la mobilità di un utente terminale verso diversi reti di accesso. Questo metodo provvede gestione della mobilità basata sulla rete, senza la necessità d'interazioni tra gli utenti, e garantisce un ridotto sovraccarico di segnalazione e bassa latenza durante l'handover.

I.2.4 HANDOVER VERTICALE:

La industria delle comunicazioni wireless è evoluta molto in fretta, partendo da una semplice chiamata voce e messaggi di testo arrivando alla prossima generazione di servizi, apparecchi ed applicazioni. Grazie a tutto questo la voce, dati e multimedia mobili ad alte velocità sono diventati la norma nelle esigenze degli utenti.

Via a via la maggioranza delle tecnologie stanno evolvendo per fornire individualmente i servizi richiesti, però considerando che lo spettro di frequenza è costoso gli operatori di rete sono costretti ad appoggiarsi nelle reti eterogenee wireless per poter soddisfare le necessità degli utenti in qualsiasi ubicazione avendo il maggiore risparmio di risorse possibile.

Ma si deve considerare che per mantenere le connessioni senza interruzioni incluso nei momenti di esecuzione dell'handover è prevalente garantire comunicazioni con caratteristiche “*seamless*”, cioè che sia completamente trasparente tutta la procedura di handover per l'utente. Sfruttando così i vantaggi delle diversi tecnologie: cellulari, wireless, satellitari, proprio perché ciascuna è indirizzata specificamente ad una classe di utenti e di servizi. Tutto questo si può riassumere nel concetto di ABC (*Always Best*

Connected) cioè, di essere collegato sempre alla miglior tecnologia tra quelle disponibili, per sfruttare i parametri desiderati nel momento giusto, senza nessun intervento da parte dagli utenti, risparmiando la batteria dei terminali mobili e selezionando le reti con minori costi di connessione, ecc.

In questo senso non si parla più di handover ma di handover Verticale, che sostanzialmente cerca di raggiungere lo spostamento automatico tra le tecnologie disponibili nella regione di studio, portando questo al cambio di tecnologia nello strato Data Link impiegato per accedere alla rete.

Attualmente [10], ci sono disponibili apparecchi che sono capaci individualmente di supportare multipli tecnologie wireless, tali come GSM, WCDMA, WiFi, WiMAX o cdma2000. Tuttavia, è stato piuttosto impegnativo integrare servizi basati sul Web, voce su IP (VoIP), e-mail, video e TV, in multiple località, apparecchi, reti di accesso ed operatori.

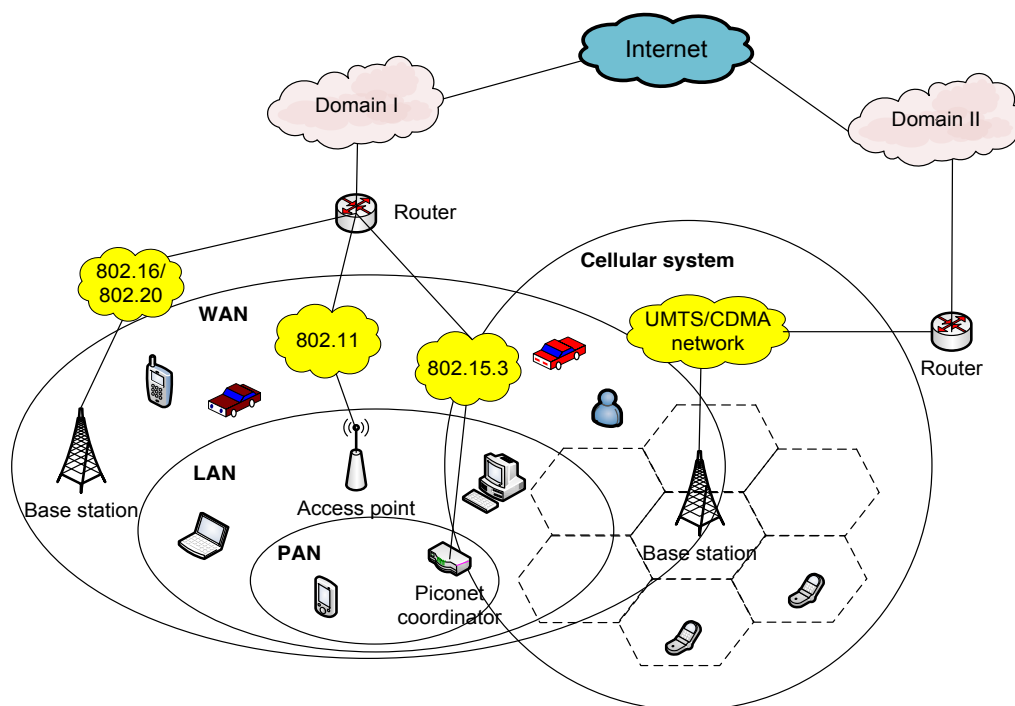


Fig. 3 Scenario di reti eterogenee

I.2.5 MEDIA INDEPENDENT HANDOVER:

Per le ragioni precedentemente esposte, alcuni lavori di ricerca suggeriscono di modificare le architetture di rete sottostanti dando luogo allo standard Media Independent Handover (MIH) proposto per il gruppo di lavoro IEEE 802.21.

Lo standard comprende una sorta di meccanismi che cercano di facilitare la coesistenza e interoperabilità tra reti eterogenee cioè con diverse tecnologie, perché gli utenti che si muovono tra di questi possano avere la desiderata universalità nell'accesso alle reti ed essere sempre connessi ovunque con le migliori prestazioni, considerando anche che questo processo sia trasparente cioè indipendente dalla tecnologia scelta ed anche indipendente da chi lo ha iniziato, rete o mobile. Tutto ciò [10] astruendo la "intelligenza" dallo strato di collegamento a strati superiori, creando un'interfaccia comune, che gestisce tutte le reti di accesso in maniera generica.

Questo standard [11] proporziona informazione per determinare il momento giusto per iniziare l'handover. Due bozze dallo standard sono stati prodotti e tanti protocolli di segnalazione sono stati proposti, ma continuano i lavori di ricerca per finalizzare la standardizzazione.

Lo Standard è composto di un blocco di MIH Function, che prevede tre servizi Media Independent:

MIES: *MI Event Service*: Copre la classificazione, il filtraggio e la segnalazione degli eventi corrispondenti ai cambiamenti dinamici delle caratteristiche, lo status e la qualità dei collegamenti. I *link Events* e *MIH Events*, possono essere classificati in 6 categorie nella seguente tabella come: amministrative, cambio di stato, parametro di collegamento, predittivo, collegamento sincrono e trasmissione del collegamento.

No	Event Type	Event Name	Description
1	State Change	Link Up	L2 Connection established
2	State Change	Link Down	L2 Connection is broken
3	Predictive	Link Going Down	L2 connection breakdown imminent
4	State Change	Link Detected	New L2 link has been found
5	State Change	Link Parameters Change	Change in specific link parameters has crossed pre-specified thresholds (link Speed, Quality metrics)
6	Administrative	Link Event Rollback	Change in specific link parameters has crossed pre-specified thresholds (link Speed, Quality metrics)
7	Link Transmission	Link SDU Transmit Status	Improve handover performance through local feedback as opposed to waiting for end-to-end notifications
8	Link Synchronous	Link Handover Imminent	L2 intra-technology handover imminent (subnet change). Notify Handover information without change in link state.
9	Link Synchronous	Link Handover Complete	Notify handover state

Tab. 1 Eventi nello strato di collegamento proposti.

MICS: *MI Command Service*: Consente agli utenti di gestire e controllare il comportamento dei collegamenti pertinenti all'handover e alla mobilità. Questi comandi sono specifici per la rete di accesso particolare che è stata usata. La seguente tabella mostra liste di MIH e i comandi di collegamento:

No	Command Name	MIHF \diamond MIHF	Client \diamond Network
1	MIH Handover Initiate		Initiates handovers and sends a list of suggested networks and suggested PoA (AP/BS).

2	MIH Handover Prepare	Network \diamond Network	This command is sent by MIHF on old network to MIHF on suggested new network . This allows the client to query for resources on new network and also allows to prepare the new network for
3	MIH Handover Commit	Client \diamond Network	In this case the client commits to do the handover based on selected choices for network and PoA.
4	MIH Handover Complete	Client \diamond Network Network \diamond Network	This is a notification from new network PoA to old network PoA that handover has been completed, new PoA has been established and any pending packets may now be forwarded to the new PoA.

Tab. 2 Comandi nel Handover

MIIS: *MI Information Service*: Proporziona l'informazione necessaria per i handover. Include database di accesso e recupero della disponibilità delle reti, informazione dello strato di collegamento, parametri e servizi delle reti servente e vicine, dagli utenti di queste e le politiche degli operatori (ad esempio mappe della distribuzione geografica delle reti ibride, costi dei servizi, funzionalità di QoS,...) per ottenere un ottimo collegamento iniziale alla rete decisa o un ottima ri-selezione di rete. Per prendere le decisioni di handover, questo servizio scambia dei IEs (*Information Elements*) in percorsi a due vie.

Information Element	Description	Comments
List of networks available	List all network types that are available given client location	E.g., 802.11, 802.16, GSM, GPRS/EDGE, UMTS networks
Location of PoA	Geographical Location, Civic address, PoA ID	E.g. GML format for LBS or network management purpose
Operator ID	Name of the network provider	E.g. Could be equivalent to Network ID.
Roaming Partners	List of direct roaming agreements	E.g. in form of NAIs or MCC +MNC

Cost	Indication of costs for service/network usage	E.g, Free/Not free or (flat rate, hourly, day or weekly rate)
Security	Link layer security supported	Cipher Suites and Authentication Methods, Technology specific, e.g. WEP in 802.11. 802.11i. PKM in
Quality of Service	Link QoS parameters	802 wide representation, application friendly
PoA Capabilities	Emergency Services, IMS Services, etc.	Higher Layer Services
Vendor Specific IEs	Vendor/Operator specific information	Custom information

Tab. 3 Elementi di informazione

Esiste un gestore di connessione e monitoraggio nel MIH, chi è situato tra l'applicazione ed il dispositivo radio per monitorare l'accesso, lo stato della rete e la disponibilità. Questo, interagisce con i livelli superiori in modo coerente per garantire handover *seamless* con ritardi minimi nelle differenti zone di copertura, con diverse velocità e livelli di qualità del servizio (QoS) offerti.

CAPITOLO II

LA DECISIONE DELL'HANDOVER



II.1 INTRODUZIONE

La decisione di handover [2] si basa su una varietà di criteri, che possono essere gli stessi per tutti i terminali mobili oppure possono essere scelti in funzione delle preferenze dell'utente o di altre politiche, che si combinano differentemente secondo diverse tecniche. Nei prossimi paragrafi si illustrerà lo stato dell'arte delle varie tecniche di decisione con un approfondimento sul rank aggregation che servirà come fondamento di ciò che verrà discusso nel Capitolo III.

II.2 CRITERI DI DECISIONE PER L'HANDOVER VERTICALE

Nell'handover verticale non è sufficiente prendere come criterio di decisione la sola potenza del segnale ricevuto per confrontarla con una soglia minima, invece si coinvolgono altri criteri come i costi monetari, le preferenze di utente, i servizi offerti e le condizioni di rete.

Il problema risiede nel combinare efficientemente una serie di parametri considerando la loro rilevanza nella specifica classe di servizio, tale approccio viene definito generalmente *processo di decisione di gruppo*.

I criteri più rilevanti [2] nelle reti mobili cellulari sono la qualità del collegamento in uplink ed in downlink, che possono essere misurati in termini di potenza del segnale ricevuto, rapporto segnale-interferenza, bit error rate, ecc. Includere tali criteri può portare benefici sia al fornitore che all'utente.

Un criterio è il carico di ciascun punto d'accesso, ovvero quanti utenti mobili stanno contemporaneamente utilizzando le risorse della rete di accesso. Dalla prospettiva del fornitore questa informazione è molto importante, giacché permette una gestione dinamica del carico: è possibile trasferire utenti da un punto di accesso molto carico ad uno che lo sia meno. In tal modo i punti di accesso dove il carico è stato ridotto possono servire in maniera più efficiente i nuovi terminali, ciò aumenta l'utilizzazione di tutta la rete di accesso e di conseguenza l'utilità del fornitore. Dal punto di vista del terminale mobile il comportamento precedentemente descritto permette un miglioramento della QoS della sua connessione.

Inoltre, gli utenti con terminali mobili hanno un forte interesse nel preservare la batteria, ovvero tendono a collegarsi a reti che siano più vicine geograficamente risparmiando potenza in trasmissione, od a cambiare ad una rete che sia più efficiente in termini di energia ma che offre una minore banda. Anche per motivi simili, un terminale può desiderare di rimanere agganciato allo stesso AP anche se altri criteri suggeriscono il contrario. In particolare includere questo criterio ed ottimizzare i consumi di energia, si contrappongono agli effetti di bilanciamento del carico, un comportamento che porti ad un aumento dell'utilità dell'utente può provocare una diminuzione di quella del fornitore.

Un altro criterio per la decisione di handover è la posizione corrente e la velocità del terminale mobile, che insieme alle misure di qualità del collegamento permettono una predizione più accurata del momento per inizializzare l'handover, in modo da migliorare la sincronizzazione nell'esecuzione.

Altri aspetti possono condurre ai terminali mobili a cambiare un punto di accesso verso quello controllato da un router differente, che può essere più adatto, ad esempio maggiore sicurezza, una QoS più affidabile, maggiore copertura o costi, in modo da avere il migliore rapporto costo-beneficio.

Consideriamo infine i criteri di decisione classificati in [1], nella seguente maniera:

- Correlati alla rete: Copertura, banda, latenza, perdita di pacchetti, ritardi, jitter, qualità del collegamento (RSS), BER, costi monetari.
- Correlati al terminale: Consumo della batteria, velocità.
- Correlati all'utente: profilo di utente, preferenze, informazione sulla localizzazione.
- Correlati ai servizi: QoS, capacità di servizio.

II.3 TECNICHE DI DECISIONE

II.3.1 SINGOLA FUNZIONE

Tenendo in considerazione alcune delle metriche precedentemente viste la decisione può essere basata su una funzione detta Score o di Costo, che dà una certa rilevanza a ciascuno dei parametri a seconda del caso e trova quindi dei compromessi tra loro. Questa funzione misura i benefici che si ottengono nell'eseguire l'handover verso una rete, valutato per tutte le n reti che coprono la zona di servizio. Una prima forma della funzione costo f_n è stata proposta in [12] per la rete n ed è la somma di funzioni pesate di parametri specifici normalizzati, in quel caso sono stati considerati la banda B_n , il consumo di potenza P_n , ed il costo monetario C_n , con funzione di normalizzazione $N(i)$:

$$f_n = w_b \cdot N\left(\frac{1}{B_n}\right) + w_p \cdot N(P_n) + w_c \cdot N(C_n) \quad \text{con} \quad \sum_i w_i = 1$$

In maniera più generale, considerando varie classi di servizio in [3], definiscono la funzione come:

$$f^n = \sum_s \sum_i w_{s,i} \cdot P_{s,i}^n \quad \text{con} \quad \sum_i w_i = 1$$

Essendo $p_{s,i}^n$, il costo del parametro i -esimo al esserci il servizio s nella rete n e $w_{s,i}$ il peso assegnato ad un parametro in uno determinato servizio e che può essere collegato al livello di importanza che l'utente assegna al particolare servizio.

Un fattore chiamato eliminazione E_s^n di rete è introdotto in [13] per rispecchiare l'incapacità di una rete per garantire i parametri di qualità di un determinato servizio richiesto.

$$E_s^n = \prod_i E_{s,i}^n$$

Indica se la rete n può garantire un minimo vincolo i per un servizio s , se non lo raggiunge questa rete non può essere più considerata come una target. Siccome questa procedura di cancellazione porta un alto costo, il fattore in questione prende un valore molto alto quando il vincolo non può essere raggiunto.

La scelta della rete target si fa alla fine tramite:

$$n_{opt} = \arg \min_n (C^n)$$

$$\text{con } C^n = \sum \prod_i E_{s,i}^n Q_s^n$$

Essendo C^n la funzione costo valutata per la rete n , e Q il fattore di QoS per un servizio s in una rete n .

II.3.2 CENTRATO NELL'UTENTE:

In questo caso si vogliono focalizzare altri criteri all'ora di fare l'handover, si impiega una politica che si basa sulle preferenze dell'utente in termini di costo e di QoS. In [9] si valuta l'impatto dei handover verticali e la sua frequenza su un insieme di applicazioni e servizi tipici dell'utente, provando a definire una metrica impiegata per trovare una soluzione che tenta di bilanciare il costo complessivo tra il costo dei handover verticali e i benefici che portano alle necessità attuali dell'utente. Quindi scegliendo la "miglior" rete ad un livello di utente ed essendo sempre questa scelta dinamica.

Il costo di fare l'handover a livello di rete si valuta in termini di latenza e lacune nella banda, inoltre la latenza dell'aggiornamento dell'ubicazione si considera come ritardo addizionale.

Si propongono due politiche di decisione, tra due reti WLAN e GPRS:

- Il terminale Mobile MT non lascia la connessione GPRS senza *blackout* di connessione.
- L'algoritmo cerca i punti di accesso WiFi con *blackout* di connessione.

La prima per soddisfare gli utenti che sono disposti a pagare per avere le connessioni più sicure possibili, e la seconda dando più importanza al costo che a la QoS. In seguito abbiamo la funzione costo che è stata definita:

$$C = \sum_{i=1}^n T_{N_i} c_{N_i}(h)$$

Questa espressione rappresenta il costo monetario per un utente per una data sessione di comunicazione. T_{N_i} è la somma degli intervalli di tempo di permanenza speso dall'utente

nella i -esima rete di accesso. C_{N_i} è la tariffa per unità di tempo (secondo) che l'operatore della i -esima rete di accesso fissa all'utente.

In [14] si misura il livello di soddisfazione dell'utente tramite una funzione d'utilità per servizi dati non in tempo reale. Per scegliere l'appropriata funzione d'utilità le metriche di decisione sono rischio d'atteggiamento degli utenti: *Neutral* (l'utente preferisce pagare ugualmente meno ed avere minor ritardo), *Seeking* (preferisce minor ritardo a risparmiare) e *Adverse* (preferisce essere sicuro di pagare meno). I risultati mostrano che il livello di soddisfazione all'impiegare una funzione d'utilità appropriata per una data preferenza di utente è altamente dipendente dalla dimensione del file.

II.3.3 MULTIPLI ATTRIBUTI: MADM MULTIPLE ATTRIBUTE DECISION MAKING:

Il problema della decisione nell'handover verticale, risiede principalmente nel scegliere una tecnologia fra un insieme di reti candidate, basandosi su diversi criteri od attributi caratteristici di ciascuna. Ed è qui dove si devono impiegare tecniche di selezione di gruppo essendo gli enti decisori, gli attributi. Tra questi metodi chiamati *Multiple Attribute Decision Making*, i più popolari sono:

- *SAW (Simple Additive Weighting)*: [15] è un metodo di pesatura molto semplice e probabilmente il più popolare. Nel contesto attuale assegna il punteggio complessivo di una rete candidata i come la somma di tutti i valori degli attributi, cioè, sommando le contribuzioni normalizzate di ciascuna metrica $r_{i,j}$ moltiplicate per il peso assegnato w_j della metrica j . La rete selezionata A_{SAW}^* è:

$$A_{SAW}^* = \arg \max_{i \in M} \sum_{j=1}^N w_j r_{ij}$$

Essendo ancora N il numero di parametri ed M il numero di reti candidate.

- *MEW (Multiplicative Exponent Weighting)*: [16] Anche questo è un metodo di assegnazione di punteggi. Impiega una matrice dove le righe rappresentano le alternative e le colonne sono gli attributi a considerare. Il punteggio delle candidate è determinato dal prodotto pesato degli attributi (o metriche) non essendo più necessaria la fase di normalizzazione fatta in SAW:

$$S_i = \prod_{j=1}^N x_{ij}^{w_j}$$

Essendo x_{ij} l'attributo j dell'alternativa i , e w_j il peso dall'attributo j , con la seguente condizione $\sum_{j=1}^N w_j = 1$. Come il punteggio ottenuto tramite questo metodo non ha

un limite superiore, è conveniente paragonarlo con il punteggio della soluzione ideale-positiva denotata come A^{**} , e come detto prima, scelta come quella con i migliori valori per ogni metrica. Il valore del rapporto tra queste due sarà:

$$R_i = \frac{\prod_{j=1}^N x_{ij}^{w_j}}{\prod_{j=1}^N (x_{ij}^{**})^{w_j}} \quad \text{dove } 0 \leq R_i \leq 1$$

La scelta sarà:

$$A_{MEW}^* = \arg \max_{i \in M} R_i$$

- *TOPSIS (Tecniche of Order Preference by Similarity to Ideal Solution)*: Si fa un approccio geometrico del problema della decisione, come un sistema di m punti in uno spazio n -dimensionale. Si basa sul principio di che l'alternativa scelta deve essere quella con una distanza minore alla "soluzione ideale-positiva" e con distanza maggiore alla "soluzione ideale-negativa". La soluzione ideale, normalmente è ottenuta usando i migliori valori di ogni metrica [16]. In [15] considerano c_i^* come relativa vicinanza (o somiglianza) della rete candidata i alla soluzione ideale. La rete scelta sarà:

$$A_{TOP}^* = \arg \max_{i \in M} c_i^*$$

- *AHP (Analytic Hierarchy Process)*: Definita in [17] come una teoria della misura che tramite confronti a coppie e basandosi sul giudizio di esperti (attributi) ottiene scale di priorità. Sono queste scale che convertono le misure intangibili in termini relativi. I paragoni sono realizzati utilizzando una scala di giudizi assoluti che rappresenta, quanto più un elemento domina un altro rispetto a un dato attributo.

La decisione si fa tramite i seguenti passi:

1. Definire il problema e determinare il tipo di conoscenza richiesta.
2. Decomporre il problema di decisione in diversi livelli di gerarchia, collocando nei livelli più alti l'obiettivo della decisione e scendendo ai livelli più bassi dove generalmente si trovano un insieme di alternative.
3. Costruire un insieme di matrici di confronto dove ogni elemento di livello superiore è utilizzato per confrontare con gli elementi di livello immediatamente inferiore ad esso.
4. Utilizzare le priorità ottenute dai confronti precedenti per pesare le priorità nel livello immediatamente inferiore. Allora per ogni elemento del livello sottostante aggiungere i propri valori pesati e ottenere la sua priorità globale. Continuare questa procedura fino al livello più basso.

In [16] si fa un approccio diverso di AHP, si scompone un problema in una gerarchia di semplici e più gestibili sottoproblemi. Questi sottoproblemi sono normalmente chiamati “fattori di decisione” e sono pesati d’accordo con i suoi dominanze al problema.

- *GRA (Grey Relational Analysis)*: è un metodo che analizza i gradi di rapporti per sequenze discrete. Una delle sequenze è scelta come di riferimento. Il rapporto “grey” tra la sequenza di riferimento e le altre può essere raggiunto calcolando il “*Grey Relational Coefficient*” (GRC) d’accordo con il livello di somiglianza e variabilità, posteriormente la sequenza con il maggiore GRC sarà quella scelta.

$$A_{GRA}^* = \arg \max_{i \in M} \Gamma_{0,i}$$

Essendo $\Gamma_{0,i}$ il GRC della rete i

Questo metodo è implementato con i seguenti tre passi: Normalizzazione dei dati, definizione della sequenza ideale e calcolo del GRC [18].

In [18] si combinano due di queste tecniche per trovare i compromessi tra preferenze di utenti, applicazione di servizio e condizioni di rete, per poi usarli nel meccanismo di selezione della rete ottima per utenti mobili. AHP è responsabile della pesatura dei parametri di QoS basati sulle preferenze di utente ed applicazioni di servizio. GRA costruisce un ranking delle reti alternative. La selezione viene realizzata tramite tre blocchi funzionali: “*collecting data*” raccoglie le preferenze di utente e le condizioni della rete, “*processing data*” che processa i dati basati sugli utenti tramite AHP e normalizza i dati basati sulla rete tramite GRA, e per ultimo “*making decision block*” che finalizza la procedura.

In [15] viene effettuato uno studio comparativo delle prestazioni ottenute con quattro diversi algoritmi di decisione di handover verticale descritti prima: MEW, SAW, TOPSIS e GRA, in quattro scenari caratterizzati da diversi classi di traffico. I risultati mostrano che

MEW, SAW e TOPSIS, forniscono prestazioni simili per tutti gli scenari, GRA fornisce una banda leggermente maggiore e minore ritardo per due scenari.

- *RANK AGGREGATION*:

Sebbene non è uno dei metodi più popolari tra quelli di decisione di gruppo, presenta dei vantaggi nell'applicarlo nella decisione di handover.

Il “ranking” consiste essenzialmente in una lista di elementi posizionati in ordine di preferenza, a seconda di uno o più criteri scelti [23]. Un esempio è il ranking $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$, secondo cui si preferisce l'elemento i invece che j se la posizione di i è maggiore che quella di j , $\pi(i) < \pi(j)$.

Si considera il problema di classificare un insieme di elementi in ingresso che li rappresenti in una “miglior” forma. Da qui nasce il *Rank Aggregation* che consiste nell'aggregazione di elenchi con risultati di classificazione provenienti da diversi giudici, in una nuova lista ordinata che prende in considerazione ciascuna delle precedenti. Quindi l'aspetto più importante sarebbe come aggregare queste multipli liste in una sola, per raggiungere il consenso.

Inizialmente Il rank aggregation fu studiato nella teoria della scelta sociale e applicata nella teoria delle elezioni, in cui ogni elettore fornisce le sue preferenze di un insieme di alternative e poi il sistema fornisce un ordine di preferenze singolo finale, basato su ciascuno degli elenchi di preferenze degli elettori. Alcune applicazioni sono gli ordinamenti degli alberghi o dei film, basati sui punteggi dati dagli utenti oppure per offrire raccomandazioni a questi basandosi su diversi criteri. Negli ultimi anni si è sparso per tante altre applicazioni come filtraggio di documenti, *meta-search*, per trovare *spam* nelle pagine web, ecc.

In [20] i metodi di rank aggregation sono ampiamente impiegati nell'area della ricerca nel web, dove l'ordinamento si fa sui siti, combinando i risultati da diverse funzioni di posizionamento basandosi su multipli criteri. Quindi il risultato è dato da un consenso ed un compromesso tra le individuali liste di posizioni ottenute a seconda di ogni criterio per ottenere l'ordinamento. Lo scenario ideale per il rank aggregation è quando ogni giudice (ogni criterio di tutto l'insieme) dà un completo ordine di tutte le alternative possibili. C. Dwork et al. impiegano come uno dei suoi principi le catene di Markov per combinare le liste parziali di posizioni.

Il rank aggregation può essere classificato in due categorie. La prima denominata *score-based*, dove dalle liste individuali di classificazione si prendono dei punteggi e la funzione di rank aggregation li usa per la classifica finale. La seconda categoria, chiamata *order-based*, necessita soltanto dell'ordine nelle liste individuali che poi saranno usati per la stessa funzione di rank aggregation.

Per trovare questo ordinamento desiderato che sia “vicino” all'ordinamento delle scelte dato in ingresso, in [23] si cerca una permutazione che minimizzi la distanza Kendall-tau tra gli ordinamenti in ingresso, definendosi questa distanza come la somma del numero di coppie di elementi che stanno in ordine diversi fra gli ordinamenti in ingresso e quelli in uscita. Se gli ordinamenti di partenza sono permutazioni, cioè se nell'esempio di scelta sociale, ciascun elettore dà un ordinamento completo delle alternative, il problema è conosciuto come *Kemeny Rank Aggregation*, che corrisponde alla media geometrica degli ingressi. Più precisamente, date due permutazioni π, σ di $\{1, \dots, n\}$ Ailon [24] definisce la distanza Kendall-tau come:

$$K(\sigma, \pi) = \sum_{i=1}^n \sum_{j=1}^n 1\{(\pi(i) < \pi(j)) \& (\sigma(i) > \sigma(j))\}$$

Essendo $1\{\cdot\}$ la funzione indicatore. Dato un insieme di permutazioni π_1, \dots, π_k di $\{1, \dots, n\}$, il problema *Kemeny Rank Aggregation* cerca una permutazione σ che minimizza il numero di disaccordi a coppie con le permutazioni in ingresso:

$$\frac{1}{k} \sum_{l=1}^k K(\pi_l, \sigma)$$

Possiamo inoltre considerare il problema di *Rank Aggregation Parziale*. Un ordinamento parziale di V è una funzione $\pi : V \rightarrow \{1, \dots, |V|\}$, dove la funzione π non deve essere una funzione uno ad uno, in parole più povere un ordinamento parziale è un ordinamento con legami.

II.3.4 FUZZY LOGIC:

I sistemi basati su *fuzzy logic* sono flessibili e capaci di lavorare con data non precisa e quindi si possono usare per modellare funzioni non lineari con una complessità arbitraria. Precisamente come può lavorare con questo tipo di data è largamente impiegata negli algoritmi di handover che usano parametri di diverse natura per prendere la decisione.

In [25] è presentata la seguente figura, dove si mostra un diagramma a blocco del concetto di logica diffusa, descrivendo la metodologia e criteri usati nella procedura di handover.

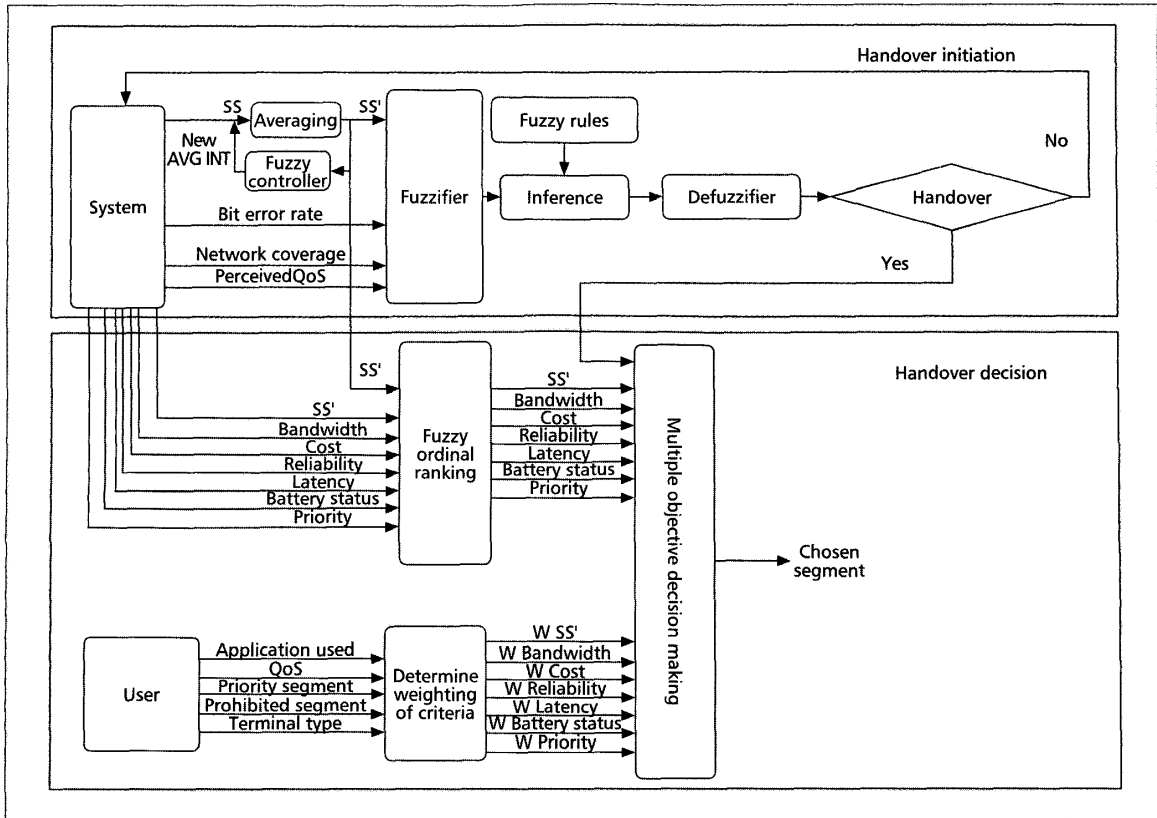


Fig. 4 Diagramma a blocco del concetto di logica diffusa nel algoritmo di handover.

Per iniziare l'handover, l'algoritmo è suddiviso in tre fasi:

In prima fase i parametri del sistemi sono messi in un "fuzzifier" che trasformerà le misure in tempo reale in insieme *fuzzy*, questi insieme contengono degli elementi che hanno un grado variabile di adesione e che quindi anche possono appartenere a più di un insieme. I valori di adesione sono ottenuti mappando i valori ottenuti per un particolare parametro su una funzione di adesione. Questa funzione è una curva o riga che definisce come ciascun dato o valore è mappato su un valore di adesione.

Nella seconda fase gli insieme diffusi vanno in ingresso ad un motore di inferenza, dove sono applicati una serie di regole *fuzzy*, che possono essere definite come una serie di possibili scenari utilizzando regole *if-then* che decidono proprio quando è necessario fare l'handover. Gli insieme di decisione possono dividersi in: Sì (Y), Probabilmente sì (PY),

Probabilmente no (PN), No (N), e come detto prima, i valori possono cadere in più di una categoria contemporaneamente.

In una terza fase, chiamata “*Defuzzificazione*” si trasformano gli insieme di decisione *fuzzy* risultanti in quantità precise.

II.3.5 SENSIBILITÀ AL CONTESTO:

Secondo [14], l’informazione sulla sensibilità del contesto può aiutarci ad ottimizzare le decisioni di handover, dove si tiene in conto non solo la qualità del segnale ma anche la conoscenza sul contesto circostante dei terminali e delle reti. Gli autori hanno proposto un approccio integrato e flessibile per la gestione del contesto, un’architettura che comprende una piattaforma programmabile e componenti per la gestione di contesto distribuito, in nodi di rete e dispositivi mobili e anche uno schema di diffusione di servizio per i servizi di rete.

L’informazione di contesto rilevante per l’algoritmo di decisione di handover è stata classificata prima come criteri per la decisione.

In [26] si presenta una soluzione per la decisione con sensibilità al contesto basata sul metodo già definito precedentemente AHP, una architettura con una categorizzazione analitica del contesto e un algoritmo di decisione di handover dettagliato. Loro dividono l’informazione contestuale in due insieme, quella che rimane statica come la descrizione degli apparecchi, reti, applicazioni e i suoi requisiti di QoS e l’altra che cambia dinamicamente, informazione attuale degli utenti e delle reti, come per esempio localizzazione, parametri di QoS della rete anche include *Impending Network Profile* (INP) che indica a quale rete è più probabile fare handover. Questa informazione cambia dinamicamente quando l’ubicazione dell’utente o la QoS della rete cambia.

L'architettura proposta in quel lavoro include un magazzino di contesto che raccoglie, gestisce e valuta l'informazione di contesto, un amministratore di adattabilità che prende le decisioni sugli adattamenti ante i cambi nel contesto e dei *proxies* che sono responsabili dell'esecuzione dell'handover, ogni rete ha il suo proprio *proxy*. La seguente figura mostra l'architettura proposta:

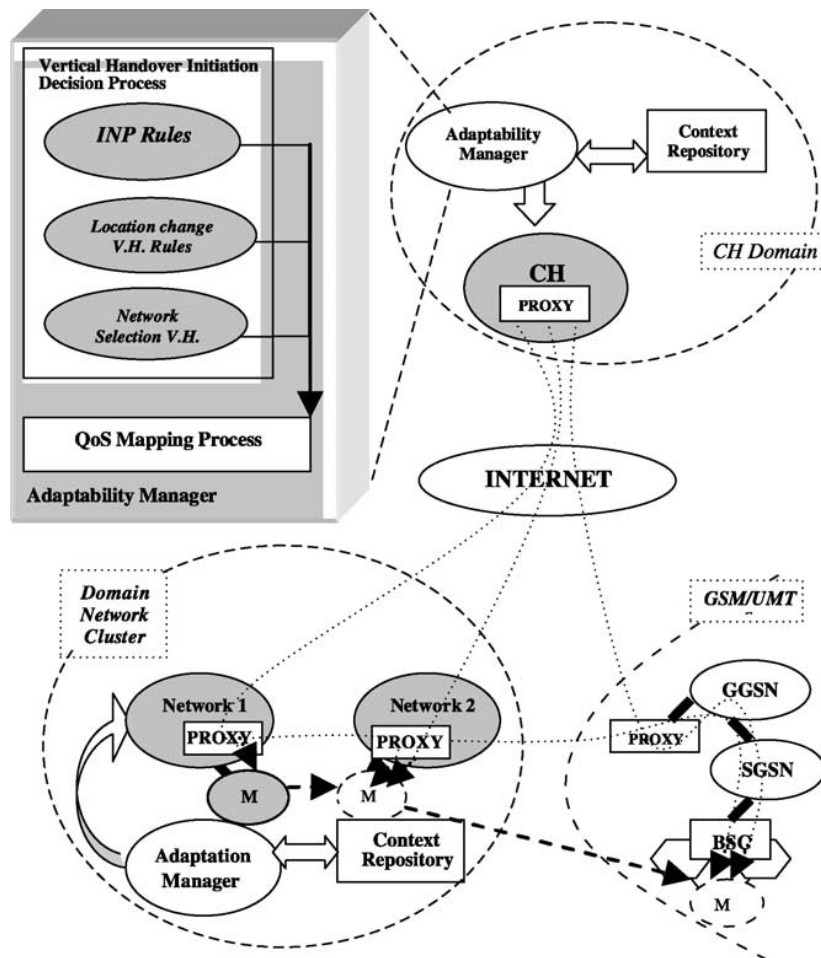


Fig. 5 Architettura di Handover Verticale

L'amministratore di adattabilità è l'incaricato del processo di decisione di quando invocare l'operazione di handover e su quale rete, questo lo fa valutando i cambi nella localizzazione dell'utente (entrando o lasciando l'area di copertura di una rete in particolare) e la QoS della rete attuale e quelle alternative. Questa procedura è basata su

regole e deve soddisfare multipli obiettivi come massimizzare le preferenze di apparecchi di utente, raggiungere il maggior livello di banda per applicazioni e minimizzare la perdita di pacchetti, ritardo, jitter e cancellare le fluttuazioni di banda che possono incidere sulle applicazioni.

Uno svantaggio in questo caso è che la raccolta di informazione di contesto è fatta in un singolo punto: il magazzino di contesto, e quindi è un punto di debolezza che richiede una comunicazione frequente tra il terminale e la rete, risultando quindi un sovraccarico elevato nel collegamento radio.

Il vantaggio della soluzione proposta in [14] rispetto a quella precedentemente esposta, consiste in una gestione del contesto avanzato, dove questa informazione è distribuita a più di un magazzino di contesto. Ed è così come lo schema di distribuzione minimizza il tempo di decisione di handover.

In [22] si fa una classifica del contesto in:

- Contesto dell'utente che è composto di informazione sulla sua localizzazione, modalità mobile, preferenze e istruzioni, in cui è favorevole fare handover. In informazione sulla localizzazione si classifica il tipo di ambiente come urbano, suburbano o rurale, la modalità mobile è tipo *indoor*, con alta velocità e nomadi, dove l'handover è avviato per i cambiamenti nella direzione e velocità del mobile, le preferenze dell'utente indicano costi monetari e/o energia che preferisce l'utente, e per ultimo gli istruzioni che specificano una rete desiderata per un utente.

- Contesto di rete è composto di informazione di carico, risorse disponibili, *throughput* e livello di sicurezza. Quest'ultimo è uno degli elementi chiavi nella decisione e non può essere sostituito, quindi anche se una rete ha buone prestazioni, se il suo livello di sicurezza non raggiunge una soglia minima, questa deve essere scartata.

STATO DELL'ARTE

RANK AGGREGATION BASATO SU CATENE DI MARKOV



Nella categoria di rank aggregation basato sull'ordine, troviamo questo approccio basato sulle catene di Markov, in cui si assume che si può generare una catena di Markov tra le entità che costituiscono gli elenchi di classifica, e che i rapporti di ordine rappresentano le transizioni nella catena [21].

Come usato in [22], considerando multipli fattori di decisione, ognuno determina una lista di posizioni, e la combinazione di queste produce un'ultima lista con una "migliore" classifica. Quindi l'handover verticale con multipli fattori di decisione può essere visto come un problema di *rank aggregation*. Allora, un punto chiave è assumere di avere disponibilità di queste liste per ogni fattore, infatti lo scenario ideale per il rank aggregation è quello dove ciascun giudice (parametro di decisione) fornisce un ordine completo di tutte le alternative nell'universo delle alternative possibili. Si presentano anche dei vantaggi per quei casi dove si hanno a disposizione solo le liste parziali di ordine.

In generale, possiamo dire che una catena di Markov omogenea è composta da un insieme di stati $S = \{1, 2, \dots, N\}$ ed è descritta da una matrice di probabilità $M [N \times N]$ non negativa, stocastica che ha come elementi le probabilità di transizioni. Data una distribuzione di probabilità di uno stato attuale della catena, la distribuzione dello stato seguente è data dal prodotto di quella distribuzione attuale per la matrice M . Considerando alcune condizioni favorevole il sistema raggiungerà una distribuzione di probabilità stazionaria, che è un punto fisso dove la distribuzione degli stati non cambia e può essere rappresentata dal suo autovettore y :

$$yM = \lambda y$$

Che dà un ordinamento di y , e un ordinamento naturale anche su M .

In [20] è proposto un metodo generale per ottenere una aggregazione iniziale di liste parziali usando catene di Markov. Gli stati corrispondono ai candidati ad essere classificati, le probabilità di transizione dipendono dalle liste parziali date e l'ordine della catena di Markov è l'ordine aggregato.

Le probabilità di transizione degli stati si possono determinare da conoscenze previe disponibili, e si possono anche sfruttare le connessioni della catena per inferire l'ordinamento nel caso in cui non si hanno a disposizione. Quindi la catena dà un punto di vista più generale delle classifiche di tutti i candidati.

Quindi alla fine, la “miglior” rete è selezionata in base a quella che ha il valore di probabilità stazionaria maggiore.

In [20], ma sempre lavorando nel contesto della ricerca nel web, si propongono quattro catene di Markov specifiche: MC_1 , MC_2 , MC_3 e MC_4 si definiscono le matrici di transizioni corrispondenti, e si dà una descrizione generale della propria catena, essendo le alternative l'insieme di pagine ordinate da varie motori di ricerca. Considerando per tutte la pagina attuale: P ,

MC_1 : Lo stato seguente è scelto uniformemente da un insieme di tutte le pagine che hanno un ranking maggiore (o uguale) che P per alcuni motori di ricerca che hanno posizionato P , cioè, dall'insieme: $\cup_i \{Q \mid \tau_i(Q) \leq \tau_i(P)\}$. L'idea principale è ad ogni passo, spostarsi di una pagina ad una migliore, e lasciando una probabilità di $1/j$ di rimanere nella stessa pagina, dove j è approssimativamente la posizione media della pagina attuale.

MC_2 : Lo stato seguente è scelto prima selezionando uniformemente un ranking τ da tutte le liste parziali τ_1, \dots, τ_k che contengono P , e poi scegliendo una pagina Q

uniformemente dell'insieme: $\{Q \mid \tau(Q) \leq \tau(P)\}$. Questa catena ha conto del fatto che si hanno disponibili alcune liste ordinate, non un confronto a coppie tra tutte le pagine, e generalizza la media geometrica analoga al metodo Borda. Per liste complete seleziona le pagine con media geometrica minore nei ranking.

MC_3 : Lo stato seguente è scelto prima selezionando uniformemente un ranking τ da tutte le liste parziali τ_1, \dots, τ_k che contengono P , poi scegliendo uniformemente una pagina Q che è stata ordinata da τ . Se $\tau(Q) < \tau(P)$ si sposta verso Q , altrimenti rimane in P . Questa è una generalizzazione del metodo Borda. Si considera naturalmente che in ciascun stato P , la probabilità di rimanere in P è approssimativamente la frazione di “concorsi a coppie” (con le altre pagine), che P ha “vinto”, una misura che si assomiglia molto a quella Borda.

MC_4 : Lo stato seguente è scelto selezionando una pagina Q uniformemente dall'unione di tutte le pagine ordinate per i motori di ricerca. Se $\tau(Q) < \tau(P)$ per la maggioranza delle liste in cui si trovano entrambe P e Q , quindi si sposta verso Q , altrimenti rimane in P . Questa catena generalizza il suggerimento di Copeland di ordinare le alternative per il numero maggiore di concorsi che hanno vinto.

In [23] considerano che la catena più generale e forse la “migliore” di quelle esposte in [7] è MC_4 , in cui cominciando in uno stato generico p e scegliendo uniforme e casualmente un altro stato q , se secondo la maggioranza delle liste di classificazione in ingresso si preferisce lo stato q , cioè $\tau_q < \tau_p$, quindi ci spostiamo su di esso con una

probabilità di transizione $P(p, q) = \frac{1}{n}$, e altrimenti $P(p, q) = 1 - \sum_{p \neq q} P(p, q)$. Sotto certi

condizioni questo processo ha un'unica distribuzione limite x che soddisfa $x = xP$, loro propongono che gli elementi $x(i)$ siano organizzati in elementi non crescenti.

F. Schalekamp et al. [23] propongono di utilizzare un “random jump” cioè con probabilità $\delta > 0$ scegliere un elemento Random, e muoversi a questo secondo le condizioni imposte da MC4, questo per assicurare che il processo abbia un'unica distribuzione limite x . Insomma propongono un metodo MC4Approx nel quale si comincia con un vettore y , e si ordinano gli elementi d'accordo con $x' = yP_n$.

W. Ying et al. [22] astraggono queste catene e questi metodi per adattarli allo scenario che interessa a questo lavoro. Specificamente prendono in considerazione un insieme di reti candidate: $P = \{p_1, \dots, p_N\}$ ed un insieme di fattori di decisione: $Q = \{q_1, \dots, q_M\}$ dove N è il numero di reti candidate ed M il numero di fattori di decisione. Per ciascun fattore di decisione si ottiene una lista di posizioni come un ordinamento per P , cioè $\tau_q = [p_1^q \geq p_2^q \geq \dots \geq p_N^q]$ dove il simbolo \geq rappresenta un rapporto di ordine nel sottoinsieme P_{sub} , e $\tau_q(p)$ denota la posizione della rete p nel ranking ottenuto dal fattore q , in τ_q , un elemento che sia “preferito” da altri cioè che ha un ranking maggiore, ha un numero di posizione più basso nella lista.

Di seguito si descrivono tutti i passi dei due metodi basati su due diverse catene di Markov descritti in quel lavoro:

1. Normalizzazione dei fattori di decisione:

A ciascun fattore di decisione è dato un peso denotato da: w_q che è normalizzato

alla somma di tutti i fattori di decisione, con il vincolo: $\sum_{q \in Q} w_q = 1$

2. Costruzione delle matrici di probabilità di transizione delle catene di Markov pesate MC:

a) Si inizializza la $N \times N$ matrice di transizione di probabilità $MC = \{mc_{ij}\}$ con tutti gli elementi uguali a zero, in cui mc_{ij} rappresenta la probabilità di transizione da p_i a p_j .

b) Per ciascun $\tau_q, q \in Q$, MC è aggiornata della seguente forma per i due metodi diversi:

WMC1:

i) Per ciascun mc_{ij} in MC, aggiorna:

$$mc_{ij} = mc_{ij} + \frac{w_q}{\tau_q(p_j)} \quad \text{se } p_i, p_j \in P \text{ e } \tau_q(p_j) \geq \tau_q(p_i)$$

ii) Ripetere il passo precedente finché tutti i $\tau_q, q \in Q$ siano stati testati.

WMC2:

i) Per ciascun mc_{ij} in MC, aggiorna:

$$mc_{ij} = mc_{ij} + \frac{w_q}{N} \quad \text{se } p_i, p_j \in P \text{ e } \tau_q(p_j) > \tau_q(p_i)$$

e poi $mc_{ij} = mc_{ij} + \frac{N - \tau_q(p_j) + 1}{N} w_q \quad \text{se } p_i, p_j \in P \text{ e } \tau_q(p_i) = \tau_q(p_j)$

iii) Ripetere il passo precedente finché tutti i $\tau_q, q \in Q$ siano stati testati.

3. Calcolo del vettore di distribuzione di probabilità stazionaria SD:

$$SD = SD \times MC$$

Dove $SD = \{sd_1, \dots, sd_N\}$ e sd_n è l'elemento per la rete p_n .

4. Selezione della rete favorita p_γ :

$$\gamma = \arg \max_n sd_n$$

Loro raccomandano assegnare i pesi dei fattori di decisione d'accordo con la classe di servizio che si intende coinvolgere, considerando di conseguenza diverse classi di servizio, questo perché come si è visto, la pesatura nella procedura precedente è stata assegnata uniformemente cioè $1/M$ dipendendo soltanto dal numero di parametri considerati.

CAPITOLO III: RANK AGGREGATION BASATO SU CATENE DI MARKOV MODIFICATO



III.1 INTRODUZIONE

Per risolvere i problemi negli ordinamenti, molti degli algoritmi di approssimazione ed euristiche che sono stati proposti in letteratura sono di tipo: Posizionali, dove si cerca una permutazione in cui la posizione di ciascun elemento è vicino alla “posizione media” degli elementi delle permutazioni in ingresso; Ordinamento per confronti, dove si usano rapporti di confronti per ordinare gli elementi e questi rapporti non sono necessariamente transitivi, cioè diversi algoritmi di confronto possono produrre diversi ranking; Di ricerca locale, dove si muovono i singoli vertici alla volta se questo migliora il valore obiettivo oppure per ultimo gli algoritmi Ibridi, combinando i primi due tipi.

Essendo il problema rank aggregation NP-hard, proprio perché per il calcolo di soluzioni ottime basate sull'approccio Kemeny è NP-hard, nessuno degli algoritmi proposti individualmente garantiscono di trovare una soluzione ottimale e diversi algoritmi danno diversi soluzioni. In [23] si fornisce evidenze teorica ed sperimentale di che al combinare questi approcci, gli algoritmi ottenuti sono superiori a quelli individuali in termini di prestazioni, infatti provvedono tre raccomandazioni per quali combinazioni di metodi impiegare basandosi nelle preferenze degli utenti per ottenere algoritmi con tre livelli di velocità diverse.

Nella teoria della complessità computazionale NP-hard è una classe di problemi che sono, informalmente “Almeno difficile quanto il più difficile dei problemi in NP”, essendo NP: non-deterministico tempo-polinomiale. Un problema H [27] è NP-hard se e solo se esiste un problema NP-completo L che può essere risolto in tempo polinomiale per una

macchina oracolo con un oracolo per H , cioè un algoritmo che può chiamare una macchina oracolo come un sottoprogramma per risolvere H , e risolve L in tempo polinomiale se questa chiamata è fatta in un solo passo. Questo indica che un problema NP-hard è quello per cui incluso teoricamente, non esiste un algoritmo intelligente che possa dare una soluzione semplice o veloce, invece l'unica forma di trovarla è tramite un'analisi esaustiva, computazionalmente intensiva in cui si devono testare tutti i possibili risultati.

III.2 RANK AGGREGATION E TEORIA DEI GRAFI

Il problema di rank aggregation [23] è un caso speciale del problema *feedback arc set* (pesato), perché nel caso classico si usa solo informazione dell'ordinamento desiderato relativo di coppie di elementi per trovare l'ordinamento desiderato, mentre che per il rank aggregation si possiede informazione aggiuntiva sulla forma degli ordinamenti in ingresso completi, dove i metodi posizionali sono gli unici che usano questa informazione.

Un problema connesso al rank aggregation è quello di *feedback arc set* in tornei, essendo un torneo un grafo completo ed orientato $G = (V, A)$, dove V sono i vertici e A gli archi, in cui si cerca l'insieme più piccolo di archi A' tali che $(V, A \setminus A')$ è aciclico, che equivale a dire che si desidera trovare una permutazione dei vertici π che minimizza il numero di archi di ritorno, che sono quelli orientati da destra a sinistra se i vertici sono ordinati d'accordo con π .

Una generalizzazione di questo è il problema *feedback arc set* pesato, dove dato un insieme di vertici V ed un peso non negativo $w_{(i,j)}$ per ciascuna coppia ordinata (i,j) , si cerca di trovare la permutazione π che minimizza: $\sum_{i,j:\pi(i)<\pi(j)} w_{(j,i)}$. I pesi soddisfano i

vincoli di probabilità se $w(i,j) + w(j,i) = 1$ per tutti $i, j \in V$, e la disuguaglianza triangolare se $w(i,j) + w(j,k) \geq w(i,k)$ per tutti $i, j, k \in V$.

Rank aggregation è un caso speciale del problema *feedback arc set* pesato, dal momento in cui possiamo dire: $w(i,j) = \frac{1}{k} \sum_{l=1}^k 1\{\pi_l(i) < \pi_l(j)\}$ e allora il problema

feedback arc set pesato cerca un σ che minimizza:

$$\sum_{i,j:\sigma(i)<\sigma(j)} \frac{1}{k} \sum_{l=1}^k 1\{\pi_l(i) < \pi_l(j)\} = \frac{1}{k} \sum_{l=1}^k K(\pi_l, \sigma)$$

Si può verificare semplicemente che i pesi soddisfano la disuguaglianza triangolare e nel caso di rank aggregation completo inoltre soddisfano i vincoli di probabilità.

Anche le catene di Markov hanno una definizione nella teoria dei grafi perché queste M definiscono un grafo pesato con n nodi tali che il peso nell'arco (u,v) è dato da M_{uv} .

III.3 MOTIVAZIONI ALLA PROPOSTA

Nello studio di implementazione fatto per Dwork et al. [20] sono stati studiati e confrontati tre metodi chiamati combinazione lineare, conteggio Borda e Catena di Markov nell'area della ricerca nel web, e si è ottenuto che quegli ultimi sono estremamente efficienti e superano le prestazioni dei primi due, in quanto integra applicazioni euristiche dipendenti. Vale la pena sottolineare che anche se è stato normalmente impiegato nell'area della ricerca nel web, questa efficienza rende il rank aggregation basato su catene di Markov molto attraente per applicarlo al handover verticale con fattori di decisione multipli. Questo metodo è stato ispirato in quello Copeland che può essere visto come un

metodo ibrido, si dice ciò giacché la diagonale in ingresso della matrice di transizione è esattamente $1/n$ volte i punteggi Copeland degli elementi. Essi propongono l'aggregazione di diverse motori di ricerca come una forma per combattere lo spam ed ottenere risultati di ricerca più robusti, hanno definito una catena di Markov sui risultati della ricerca e li hanno ordinati in base alle rispettive distribuzioni di probabilità stazionaria.

Le catene proposte in [20] e la forma di costruzione sono state descritte precedentemente e vale dire che un vantaggio molto risalente che si ottiene al applicarle è proprio la diminuzione della complessità di calcolo. In generale per costruire una delle proposte catene di Markov e calcolare la loro distribuzione di probabilità stazionaria si impiega approssimativamente $\Theta(n^2k + n^3)$ tempo. Tuttavia, in pratica se si calcola la matrice di transizione in $\Theta(n^2k)$ tempo, alcune poche iterazioni bastano per calcolare la distribuzione stazionaria, ma in tutte quelle proposte in [20] si impiega circa $O(nk)$ tempo per l'elaborazione previa, poi è possibile simulare ogni paso in $O(k)$ tempo, così al simulare la catena in $O(n)$ passi, si deve essere capace di campionare piuttosto efficientemente la distribuzione stazionaria. Usualmente per identificare solo alcune candidate che hanno le distribuzioni stazionarie maggiori, si impiega $O(nk)$ tempo, il quale è di molto discosto a quello nel caso del problema NP-hard. Rappresentando quindi la riduzione della complessità computazionale un'altra spinta per impiegare queste catene, e più precisamente quelle utilizzate in [22].

III.4 MODIFICHE

A seguito delle ragioni precedentemente esposte, in questo lavoro sono state selezionate quelle euristiche, modificandole con la finalità di adattare allo scenario già descritto.

A continuazione si presenta un diagramma a blocchi con tutti i passi ed aggiustamenti applicati al ranking ed alle catene scelte in questo lavoro e successivamente si spiega in dettaglio la struttura e funzionalità di ciascun blocco:

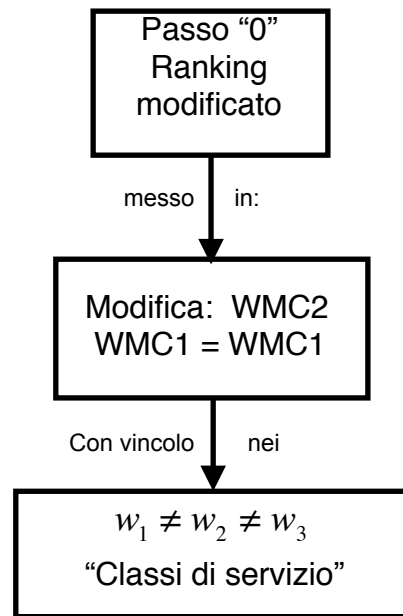


Fig. 6 Diagramma di Blocco: Proposta.

Più precisamente nel concetto del ranking si fondono le classi descritte precedentemente: *score-based* e *order-based*, come un passo preliminare all'applicazione delle catene di Markov che si modificano anche loro, non solo nella propria equazione bensì al includere un vincolo di peso a ciascun parametro di decisione dipendendo dalla classe di servizio che si desidera ricevere.

La modifica al ranking nel primo blocco è stata implementata in un passo detto "0", e consiste essenzialmente in prendere i M parametri scelti, con $M=3$, che sono: ritardo, packet loss e banda, che sono i giudici nella decisione dei gruppi oppure analoghi ai motori di ricerca nell'area del web-search, per creare dai loro valori raccolti nella fase di scoperta delle reti, una lista delle alternative ordinate, quindi si avranno tre ranking. Queste liste nel caso convenzionale di ranking sarebbero semplici permutazioni, cioè ciascuna rete avrebbe un ordine denotato da un numero intero, ma in questa proposta si pretende assegnare un

valore numerico decimale ad ognuna delle alternative, dipendendo direttamente dal valore che la rete offre per quello specifico parametro.

Con la procedura anteriormente descritta si cerca applicare un ordinamento più “giusto” delle alternative, fattore che inciderà posteriormente sulla probabilità di scegliere la miglior alternativa. Il ranking modificato per ciascun parametro è descritto di seguito:

Per i primi due parametri, ritardo e packet loss:

1. Si calcola un fattore che chiameremo *Diff*, che rappresenta la partizione che corrisponderebbe a ciascun livello esatto nel ranking, nel caso in cui i parametri sarebbero distribuiti uniformemente in tutto il rango compreso tra il parametro con valore massimo e quello con valore minimo. L'equazione corrispondente è la seguente:

$$Diff = \frac{P_{MAX} - P_{MIN}}{N - 1}$$

2. Al parametro col valore minimo gli si assegna l'ordine numero 1, perché sarà quello più desiderato:

$$r(P_{MIN}) = 1$$

3. Al parametro col valore massimo gli si assegna l'ordine numero N (quantità di reti scelta), perché sarà quello meno desiderato:

$$r(P_{MAX}) = N$$

4. Ai restanti parametri gli si assegna un ordine con valore d'accordo alla seguente equazione:

$$r(P_i) = \frac{P_i - P_{MIN}}{Diff} + 1$$

Per la Larghezza di Banda, si modificano i passi 2, 3 e 4, perché i parametri con valori maggiori e quelli con i valori minori hanno un senso totalmente opposto a quello considerato per il caso anteriore:

1. Il fattore *Diff* mantiene la stessa equazione.
2. Al parametro col valore minimo gli si assegna l'ordine numero N , perché sarà quello meno desiderato:

$$r(P_{MIN}) = N$$

3. Al parametro col valore massimo gli si assegna l'ordine numero 1 (quantità di reti scelta), perché sarà quello più desiderato:

$$r(P_{MAX}) = 1$$

4. Ai restanti parametri gli si assegna un ordine con valore d'accordo alla seguente equazione:

$$r(P_i) = \frac{P_{MAX} - P_i}{Diff} + 1$$

Questo valore sarà poi inserito direttamente nell'equazione che determina la probabilità di transizione di stati nella catena di Markov, che in questo ambito consiste nella probabilità di transizione tra le reti considerate, allora il valore del ranking è reso molto importante.

Risulta importante aggiungere che in uno scenario dove ci sono soltanto due reti, l'algoritmo di ranking appena descritto non conduce a nessuna differenza nella posizione finale aggregata.

Le catene usate in questo lavoro, sono quelle identificate prima come **WMC1** e **WMC2**, ma nel secondo blocco del diagramma si modifica quest'ultima catena perché come noto nella sua struttura c'è stata una differenziazione quando gli ordini degli stati sono uguali e quando non lo sono. La modifica consiste essenzialmente in applicare la stessa equazione per entrambi i casi, la motivazione principale è il fatto che la struttura originale produceva una probabilità di transizione di stati maggiore quando i ranking erano uguali, cioè determinava una probabilità maggiore di rimanere nello stesso stato anche se l'ordine fosse il minore possibile, più specificamente N , e una probabilità di transizione di

stati minore nonostante che l'ordine dello stato verso cui si voleva andare fosse stato quello maggiore, cioè l .

Si richiamano di seguito le strutture delle due catene, con la modifica prima esplicitata.

WMC1:

i) Per ciascun mc_{ij} in MC, aggiorna:

$$mc_{ij} = mc_{ij} + \frac{w_q}{\tau_q(p_j)} \quad \text{se } p_i, p_j \in P \text{ e } \tau_q(p_j) \geq \tau_q(p_i)$$

WMC2:

i) Per ciascun mc_{ij} in MC, aggiorna:

$$mc_{ij} = mc_{ij} + \frac{N - \tau_q(p_j) + 1}{N} w_q \quad \text{se } p_i, p_j \in P \text{ e } \tau_q(p_j) \geq \tau_q(p_i)$$

Nel terzo blocco si introduce un vincolo molto risalente nelle stesse catene, sulla pesatura dei diversi parametri che in una prima istanza erano uniformi. In questo passo si intende gerarchizzare i parametri secondo la loro importanza, tenendo conto essenzialmente la classe di servizio in cui si sta lavorando, in pratica gli si assegnano pesi diversi allo stesso attributo per le varie classi di servizio considerate.

Le classi di Servizio scelte in questo lavoro sono: *Conversational*, *Streaming*, *Interactive* e *Background*. Il Working Group propone queste quattro classi QoS e li descrive più in dettaglio [28].

Un fattore rilevante tra questi classi sarebbe la sensibilità del traffico rispetto al ritardo. La classe *Conversational* è molto sensibile al ritardo mentre che *Background* è la meno sensibile di tutte. Le prime due classi menzionate sono principalmente usate per portare flussi di traffico real-time. Servizi *Conversational* real-time, come la telefonia

video, sono le applicazioni più sensibili al ritardo. Classi *Interactive* e *Background* sono principalmente destinate ad essere utilizzate da applicazioni Internet tradizionali come WWW, e-mail, Telnet, FTP e Telegiornali, e dovuto ai requisiti di ritardo più sensibili, rispetto alle classi *Conversational* ed *Streaming*, entrambe forniscono un migliore tasso d'errore grazie alla codifica di canale e alla ritrasmissione. La differenza principale tra le classe *Interactive* e *Background* è che la prima è usata come il proprio nome lo dice, per applicazioni interattive, come Email o navigazione Web, mentre che la seconda è destinata a traffico *Background*, ad esempio, scaricare Email in *background* o scaricare documenti in *background*. La capacità di risposta delle applicazioni interattive è assicurata al separare le applicazioni *interactive* e *background*. Il traffico nella classe *Interactive* ha una priorità maggiore in *scheduling* che nel traffico della classe *Background*, così le applicazioni di quest'ultima utilizzano le risorse di trasmissione solo quando le applicazioni interattive non ne hanno bisogno. Questa sensibilità al ritardo previamente descritta è molto importante in ambienti wireless ove la larghezza di banda è primordiale in confronto con le reti fisse.

A continuazione si presenta una tabella riassuntiva delle diverse classi di servizio BWA (Broadband Wireless Access) estratta da [28]:

Traffic class	Conversational class conversational RT	Streaming class streaming RT	Interactive class Interactive best effort NRT	Background Background best effort NRT
Fundamental characteristics	<ul style="list-style-type: none"> • Preserve time relation (variation) between information entities of the stream • Conversational pattern (stringent and low delay) 	<ul style="list-style-type: none"> • Preserve time relation (variation) between information entities of the stream 	<ul style="list-style-type: none"> • Request response pattern • Preserve payload content 	<ul style="list-style-type: none"> • Destination is not expecting the data within a certain time • Preserve payload content
Example of the application	- voice	- streaming video	- Web browsing	- background download of emails

Tab. 4 Classi QoS BWA

In [29] si sono proposte due metodi per la assegnazione dei pesi degli attributi, il primo basato nei requisiti di ciascuna applicazione o classe di servizio, dove il profilo di QoS di ciascun servizio è applicabile a tutti gli utenti ed il secondo metodo basato nel profilo di sottoscrizione dell'utente descritto dall'operatore di rete, in questo caso il profilo di qualità di servizio è applicabile a tutti gli abbonati di quello specifico servizio. I pesi applicati agli attributi che ci interessano per tre diversi esempi di classi di servizio considerati in quel lavoro sono riportati di seguito:

$$\text{Voce: VoIP: } ABW = 0.2, D = 0.3, L = 0.15 \quad \sum w_q = 0.65$$

Normalizzandoli alla somma: $\sum w_q = 1$ risultano i seguenti pesi:

$$ABW = 0.3077, D = 0.4615, L = 0.2308$$

$$\text{Streaming: } ABW = 0.2, D = 0.1, L = 0.05 \quad \sum w_q = 0.35$$

Normalizzandoli: $ABW = 0.5714, D = 0.2857, L = 0.1429$

$$\text{Web Search: } ABW = 0.15, D = 0.05, L = 0.1 \quad \sum w_q = 0.30$$

Normalizzandoli: $ABW = 0.5, D = 0.1667, L = 0.3333$

In [15] e [16] per determinare i pesi w_q si usa il metodo dell'autovettore, e i risultati sono usati posteriormente nel metodo AHP. Si ha bisogno delle risposte ad una serie di confronti a coppie di metriche, che sono ottenute usando la scala fondamentale AHP [17]:

Intensità di dominanza	DEFINIZIONE
1	Indifferenza
3	Moderata preferenza
5	Forte preferenza
7	Preferenza molto forte
9	Estrema preferenza
2,4,6,8	Giudizi di preferenza intermedi
reciproci 1/2, 1/3,ecc.	Grado di dominanza di A_i su A_j

Tab. 5 Scala AHP

Tutta la procedura è esplicitata in [30], e si riporta nei seguenti paragrafi.

Se B è definita come una matrice di confronti AHP, quindi per risolvere il sistema $Bw = n_{\max} w$ (dove n_{\max} è l'autovalore maggiore di B) può essere ottenuto il vettore di importanza o priorità w . Per costruire la matrice AHP ricordiamo, i criteri: $C_q, q = 1,2,3$ essendo C_1 : Packet Loss, C_2 : Ritardo, C_3 : Larghezza di banda, e le classi di servizio: $A_l, l = 1,2,3,4$ con A_1 : *Conversational*, A_2 : *Streaming*, A_3 : *Interactive*, A_4 : *Background*.

La Matrice sarà costruita per ciascuna alternativa della seguente maniera:

A_i	C_1	C_2	C_3
C_1	1	c_1/c_2	c_1/c_3
C_2	c_2/c_1	1	c_2/c_3
C_3	c_3/c_1	c_3/c_2	1

Fig. 7 Matrice AHP

Come si può vedere la matrice è simmetrica con diagonale principale unitaria ed il numero di confronti dipende dalle dimensioni della matrice AHP (k=3):

$$\# \text{ conf} = \frac{k(k-1)}{2} = 3$$

Le risultanti quattro matrici mostrate in [15], togliendo un criterio che non ci interessa, sono le seguenti:

Conversational	Packet Loss	Ritardo	Banda
Packet Loss	1	1/9	1
Ritardo	9	1	9
Banda	1	1/9	1

Tab. 6 Matrice AHP per traffico *Conversational*

Streaming	Packet Loss	Ritardo	Banda
Packet Loss	1	1/5	1/9
Ritardo	5	1	1/5
Banda	9	5	1

Tab. 7 Matrice AHP per traffico *Streaming*

Interactive	Packet Loss	Ritardo	Banda
Packet Loss	1	5	5
Ritardo	1/5	1	1
Banda	1/5	1	1

Tab. 8 Matrice AHP per traffico *Interactive*

Background	Packet Loss	Ritardo	Banda
Packet Loss	1	9	5
Ritardo	1/9	1	1/5
Banda	1/5	5	1

Tab. 9 Matrice AHP per traffico Background

Con le matrici si applica il metodo dell'autovalore tramite i seguenti passi:

1. Si calcola l'autovalore principale n_{\max}^l della matrice dei confronti binari:

$$\text{Conversational: } n_{\max}^1 = 3$$

$$\text{Streaming: } n_{\max}^2 = 3.1171$$

$$\text{Interactive: } n_{\max}^3 = 3$$

$$\text{Background: } n_{\max}^4 = 3.1171$$

2. Si calcola l'autovettore principale e si normalizza con la somma dei valori:

$$\text{Conversational: } V_1 = \{0.1098, 0.9879, 0.1098\}$$

$$V_{1,NORM} = \{0.0909, 0.8181, 0.0909\}$$

$$\text{Streaming: } V_2 = \{0.0759, 0.2699, 0.9599\}$$

$$V_{2,NORM} = \{0.0581, 0.2067, 0.7352\}$$

$$\text{Interactive: } V_3 = \{0.9623, 0.1925, 0.1925\}$$

$$V_{3,NORM} = \{0.7143, 0.1429, 0.1429\}$$

$$\text{Background: } V_4 = \{0.9599, 0.0759, 0.2699\}$$

$$V_{4,NORM} = \{0.7352, 0.0581, 0.2067\}$$

Alla fine si è così ottenuto l'ordinamento dei pesi w_q^l con $\sum_q w_q^l = 1, \forall l$, ricordando

che q si riferisce a ciascun parametro e l ad ogni classe. Quindi risulta:

Conversational: $w_q^1 = \{0.0909, 0.8181, 0.0909\}$

Streaming: $w_q^2 = \{0.0581, 0.2067, 0.7352\}$

Interactive: $w_q^3 = \{0.7143, 0.1429, 0.1429\}$

Background: $w_q^4 = \{0.7352, 0.0581, 0.2067\}$

Ma prima di considerare giusti i pesi calcolati, bisogna fare una validazione per vedere se i confronti binari delle matrici sono corretti. Quindi si realizza la misura di consistenza proposta da Saaty, e si verifica se i giudizi sui criteri sono coerenti e quindi accettabili. Il metodo prevede i seguenti passaggi:

1. Si calcola l'Indice di Consistenza:

$$IC_l = \frac{n_{\max}^l - k}{k - 1}$$

Ottenendosi: $IC_1 = 0$, $IC_2 = 0.0585$, $IC_3 = 0$, $IC_4 = 0.0585$

2. Si calcola il Rapporto di Consistenza:

$$RC_l = \frac{IC_l}{ICR_l}$$

L'Indice di Rapporto di Consistenza dipende da k e si ottiene dalla seguente tabella:

k	1	2	3	4	5	6	7	8	9
ICR	0	0	0,58	0,9	1,12	1,24	1,32	1,41	1,45

Tab. 10 Valori ICR in funzione delle dimensioni della matrice AHP.

I valori risultanti sono: $RC_1 = 0$, $RC_2 = 0.1009$, $RC_3 = 0$, $RC_4 = 0.1009$

3. Si confronta il RC calcolato col RC teorico, funzione di k: $RC_{TEOR} = 0.1$

$$RC_l < RC_{TEOR}$$

Quindi possiamo dire che praticamente in tutti i casi in studio le valutazioni sono coerenti perciò i pesi accettabili e saranno utilizzati per tutto il lavoro qui svolto.

Si richiamano ancora le catene di Markov impiegate ma questa volta con il vincolo delle classi di servizio intrinseco nei pesi degli attributi:

WMC1:

i) Per ciascun mc_{ij} in MC, aggiorna:

$$mc_{ij} = mc_{ij} + \frac{w_q^l}{\tau_q(p_j)} \quad \text{se } p_i, p_j \in P \text{ e } \tau_q(p_j) \geq \tau_q(p_i)$$

WMC2:

i) Per ciascun mc_{ij} in MC, aggiorna:

$$mc_{ij} = mc_{ij} + \frac{N - \tau_q(p_i) + 1}{N} w_q^l \quad \text{se } p_i, p_j \in P \text{ e } \tau_q(p_i) \geq \tau_q(p_j)$$

CAPITOLO IV

RISULTATI E SIMULAZIONI



IV.1 INTRODUZIONE

Nei seguenti paragrafi si convalidano tutti i cambiamenti ed i contributi aggiunti ai metodi di base scelti, esplicitati nel capitolo precedente. Per mostrare tali risultati si è adoperato lo strumento di calcolo MATLAB versione 7.1.

E' importante sottolineare che per costruire gli scenari desiderati, è stato necessario generare i parametri scelti, tra i quali: ritardo, packet loss e banda nella maniera descritta in seguito.

Si è definito il ritardo modellandolo come un sistema a coda di tipo M/M/1, ovvero il tempo di attesa in coda di un pacchetto in una rete con un rate di servizio μ ed un rate di arrivo λ , sono stati generati aleatoriamente usando come unico vincolo $\mu > \lambda$.

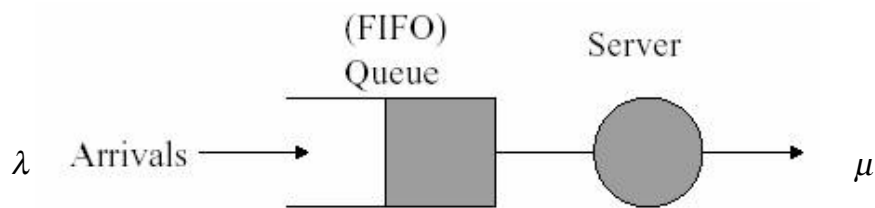


Fig. 8 Sistema a Coda

Quindi:

$$D = \frac{1}{\mu - \lambda}$$

Per generare il packet loss si è utilizzato il Modello di Gilbert, che consiste in una catena di Markov a due stati:

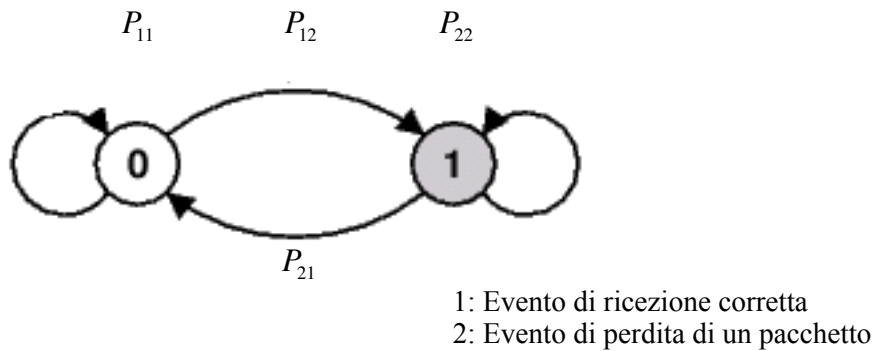


Fig. 9 Modello di Gilbert

$$\begin{cases} P_{11} + P_{12} = 1 \\ P_{21} + P_{22} = 1 \end{cases}$$

Ottenendo le seguenti probabilità di stato stazionario:

$$P_1 = \frac{P_{21}}{P_{21} + P_{12}} \qquad P_2 = \frac{P_{12}}{P_{21} + P_{12}}$$

Dove P_2 è la probabilità di perdere un pacchetto. Si è considerato per le probabilità di transizione degli stati, i seguenti valori tipici [31]:

$$P_{12} \in [1 \times 10^{-3}, 1 \times 10^{-5}]$$

$$P_{21} \in [0.05, 0.3]$$

La banda è generata in modo aleatorio da una distribuzione uniforme.

Nella sezione IV.2 si realizza uno studio statistico sull'algoritmo di selezione di rete per vedere i vantaggi che questo offre. In seguito, in IV.3 si misura la sensibilità dei pesi dei parametri calcolati precedentemente nei casi previsti. In IV.4 si verifica come al variare del ranking delle reti varia anche la scelta (elemento chiave di questa tesi) e si ripete lo stesso però al mutare dei valori dei parametri. In fine si evidenziano graficamente i parametri "ottimi" delle reti scelte per il caso di classi di servizio multiple.

IV.2 CASO D'USO

Per capire meglio tutto quanto si è proposto nel capitolo precedente a continuazione si presenta un esempio in cui si sceglie la rete “migliore” prima, di forma generica e poi per ciascuna classe di servizio, a partire dai parametri raccolti di ogni rete.

Nella seguente tabella si presentano i valori di ritardo, packet loss e banda disponibile per sei diverse reti: UMTS1: U_1 , UMTS2: U_2 , WLAN1: WL_1 , WLAN2: WL_2 , WiMax1: WM_1 , WiMax2: WM_2 .

	R	L	ABW
U_1	35	70	0,5
U_2	30	80	0,6
WL_1	100	20	1
WL_2	140	18	1,5
WM_1	60	15	2,5
WM_2	70	20	3

Tab. 11 Parametri di sei reti.

Si sono ottenuti tramite semplici confronti tre vettori ordinati da ciascun parametro, riportati nella seguente matrice di ranking R:

$$R = \begin{pmatrix} 2 & 5 & 6 \\ 1 & 6 & 5 \\ 5 & 3 & 4 \\ 6 & 2 & 3 \\ 3 & 1 & 2 \\ 4 & 4 & 1 \end{pmatrix}$$

Si è evidenziato come per il secondo parametro la terza e la sesta rete potevano scambiare le sue posizioni, perché presentavano gli stessi valori ed è qui dove si presenta la debolezza del metodo.

Si sono applicate le equazioni pertinenti, la matrice di ranking modificato, che è il primo passo dello schema a blocco presentato nel capitolo scorso:

$$R_{\text{mod}} = \begin{pmatrix} 1,2273 & 5,2308 & 6 \\ 1 & 6 & 5,8 \\ 4,1818 & 1,3846 & 5 \\ 6 & 1,2308 & 4 \\ 2,3636 & 1 & 2 \\ 2,8182 & 1,3846 & 1 \end{pmatrix}$$

Si sono considerati i pesi degli attributi uniformi per il caso generico, si sono introdotti i ranking nelle catene di Markov e si è calcolato il vettore di stati stazionari per ciascuna catena:

$$SD_1 = \{0.1167, 0.1646, 0.0909, 0.1228, 0.2612, 0.2438\}$$

$$SD_2 = \{0.1148, 0.1359, 0.1052, 0.1318, 0.2784, 0.2339\}$$

Dai vettori si vede come entrambi i metodi hanno scelto la rete numero 5 cioè WM_1 , proprio perché questa aveva il maggior valore di probabilità di stato stazionario, anzi l'ordinamento delle reti secondo questo vettore è lo stesso per tutti e due i metodi: 5,6,2,4,1,3.

Vale la pena dire che i valori di questo esempio sono anche stati introdotti nell'algoritmo del metodo TOPSIS, esplicitato nel secondo capitolo e si è selezionata la rete numero 6, ma con un ordinamento: 6,5,3,4,1,2, molto simile a quello ottenuto con il metodo in studio, questa differenza si deve ai criteri che prendono i metodi come metriche per la scelta.

Poi si è applicato il metodo da cui si partiva inizialmente e si sono ottenuti i seguenti due vettori:

$$SD_1 = \{0.0914, 0.2164, 0.0680, 0.0447, 0.3004, 0.2791\}$$

$$SD_2 = \{0.1117, 0.1819, 0.0948, 0.0581, 0.2937, 0.2598\}$$

In questo esempio la rete scelta è stata la stessa che per il caso precedente, cambiando solo gli ordinamenti per entrambe catene: 5,6,2,1,3,4. Ma questo non succede sempre e come si vedrà nelle prossime sezioni la decisione può cambiare secondo il metodo proposto.

Quando si sono coinvolte le quattro classi di servizio considerate, come visto prima per i metodi delle catene di Markov, e per TOPSIS moltiplicando semplicemente i parametri per i pesi corrispondenti a ciascuna classe, si sono ottenuti i seguenti vettori di stati stazionari che poi hanno condotto alla seguente selezione di reti:

$$\text{Classe Conversational: } SD_1 = \{0.1572, 0.4999, 0.0495, 0.0530, 0.1290, 0.1114\}$$

$$SD_2 = \{0.1791, 0.4191, 0.0590, 0.0579, 0.1652, 0.1196\}$$

Scelta la rete 2 e TOPSIS sceglie 5.

$$\text{Classe Streaming: } SD_1 = \{0.0964, 0.1400, 0.0414, 0.0535, 0.1764, 0.4922\}$$

$$SD_2 = \{0.0962, 0.1157, 0.0461, 0.0655, 0.2457, 0.4309\}$$

Tutti e tre metodi hanno scelto la rete 6.

$$\text{Classe Interactive: } SD_1 = \{0.0724, 0.0882, 0.0741, 0.1189, 0.4869, 0.1595\}$$

$$SD_2 = \{0.0773, 0.0773, 0.0839, 0.1245, 0.4754, 0.1616\}$$

Tutti e tre metodi hanno scelto la rete 5.

Classe Background: $SD_1 = \{0.0336, 0.0399, 0.0657, 0.1125, 0.5490, 0.1993\}$

$SD_2 = \{0.0363, 0.0356, 0.0732, 0.1198, 0.5412, 0.1938\}$

I metodi delle catene hanno scelto la rete 5 e TOPSIS la rete 6.

IV.3 SCELTA DELLA RETE MIGLIORE

Per verificare i vantaggi della proposte di questo lavoro si sono generati mille scenari con diversi parametri. In ciascun scenario si è applicato l'algoritmo di selezione di rete tramite i metodi delle catene di Markov con ranking presente in letteratura ed anche quelli proposti sia con modifica alla catena sia con modifica alla procedura di ranking. I risultati ottenuti visibili nella seguente figura mostrano che nel 74,4% dei casi MC1 sceglie una rete diversa da quella scelta da MC1 originale. Tale valore è molto simile al 74,2% per MC2 ed MC2 originale.

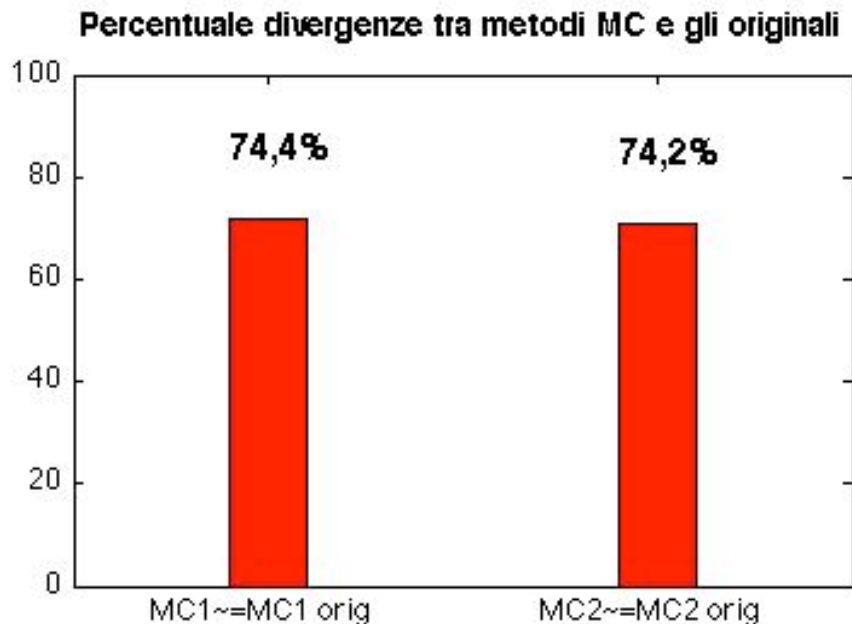


Fig. 10 Differenza nella selezione della rete tra MC1 ed MC2 (originali e modificati).

Il risultato appena mostrato lascia intuire una differenza di tipo prestazionale, infatti reti diverse avranno in genere parametri diversi. Nel seguito mostriamo come la discordanza nella scelta si rifletta nelle prestazioni offerte all'utente, in particolare consideriamo i valori medi e la deviazione standard dei parametri delle reti scelte da MC1 ed MC2 originali e modificati per tutti i mille casi.

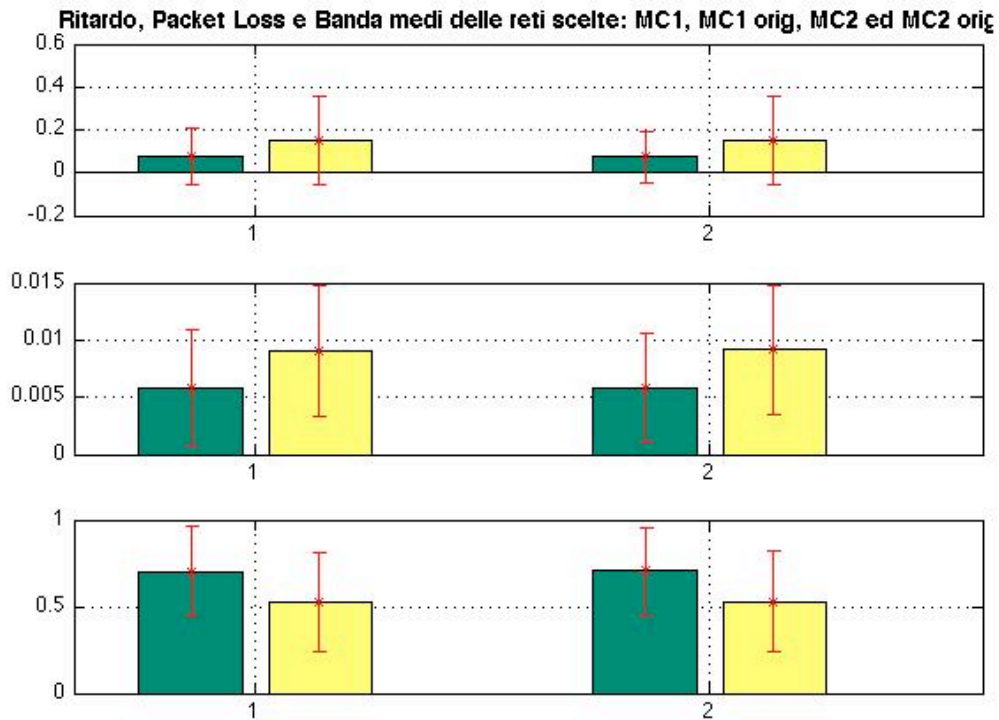


Fig.11 Valori medi e deviazione standard dei parametri, delle reti scelte con MC1 ed MC2 originali e modificati.

Dalla Fig. 11 si evince chiaramente come per tutti i parametri, in media, gli algoritmi modificati in questo lavoro scelgono reti migliori in quanto queste superano le prestazioni delle reti scelte dai metodi di partenza. Precisamente i valori ottenuti e le loro rispettive deviazioni standard sono mostrati nella seguente tabella.

	MC1	MC1 originale	MC2	MC2 originale
Ritardo	0,0728±0,1304	0,152±0,2055	0,0681±0,1191	0,1467±0,2044
Packet Loss	0,006±0,005	0,0092±0,0056	0,0057±0,0048	0,0092±0,0056
Banda	0,6897±0,2615	0,5335±0,2903	0,6935±0,2473	0,5384±0,2901

Tab. 12 Valori medi dei e deviazione standard parametri scelti per i diversi metodi

Quindi si può dire che MC1 sceglie reti con un ritardo che in media è minore del 52,11%, rispetto a quello che presentano le reti scelte dal metodo MC1 originale. Lo stesso succede per MC2 però in tale caso con una differenza del 53,58%.

Nel caso del packet loss, per MC1 il miglioramento ottenibile corrisponde ad un 34,78%. MC2 sceglie reti con un packet loss medio minore del 38.04% di quello delle reti scelte dal metodo MC2 di partenza.

Le reti scelte per MC1 offrono una banda media maggiore del 29.28% di quella offerta per le reti scelte da MC1 originale. Per MC2 la banda media è maggiore di un 28,80% dalla banda media di MC2 originale.

IV.4 MISURA DELLA SENSIBILITÀ DEI PESI

Nella sezione corrente si presentano i risultati di uno studio di sensibilità dei pesi che si sono calcolati nel capitolo precedente per le diverse classi di servizio, ovvero una valutazione della loro stabilità, rispetto alle variazioni dei parametri.

A questo scopo si considera uno scenario costituito da due reti. Dai parametri raccolti da ciascuna si prende il valore “assoluto” che non è altro che i loro pesi calcolati nel capitolo precedente. Questo valore serve di riferimento nel presente studio. Si sono

impiegati valori di parametri molto simili in maniera che non determinassero la scelta, ma si i loro pesi.

Per ciascuna classe si è fissato il valore di uno dei parametri considerati e si sono variati i restanti due, questo con la finalità di vedere i cambiamenti nel ranking che influisce direttamente nella scelta della rete. Si sono presi gli intervalli in cui non cambia l'ordinamento denominati *Absolute Top*: AT, ed a partire di questo si è calcolato il fattore chiamato *Percentage Top* PT, percentuale che il peso deve subire perché cambi la rete favorita:

$$PT = (AT/Valore\ assoluto)$$

Il seguente diagramma illustra meglio i concetti appena descritti:

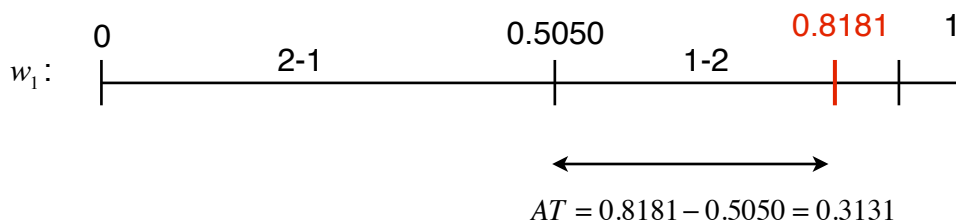


Fig. 12 Absolute Top per il ritardo.

I valori risultanti per ogni classe di servizio sono raccolti nelle seguenti tabelle:

Conversational:

Peso	Valore assoluto	Intervalli	Ranking	AT	PT(%)	Sensibilità
w_1 : Ritardo	0,8181	0 → 0,5050 0,5050 → 0,9091	2,1 1,2	0,3131	38	2,61
w_2 Packet loss	0,0909	0 → 0,4040 0,4040 → 0,9091	1,2 2,1	0,3131	344	0,2903
w_3 : Banda	0,0909	0,0909 → 0,4040 0,4040 → 0,9091	1,2 2,1	0,3131	344	0,2903

Tab. 13 Sensibilità del ranking ai valori dei pesi per la classe Conversational.

Nella classe conversational il valore del ritardo dovrebbe aumentare di un 38% perché si scelga una rete diversa, che comunque è una percentuale abbastanza elevata. Il packet loss dovrebbe aumentare di un 344%, stessa percentuale che dovrebbe diminuire la banda perché cambi l'ordinamento, sono dunque poco sensibili alle variazioni.

Streaming:

Peso	Valore assoluto	Intervalli	Ranking	AT	PT(%)	Sensibilità
w_1 : Ritardo	0,2067	0 → 0,4947 0,4947 → 0,9419	2,1 1,2	0,2880	139	0,7177
w_2 Packet loss	0,0581					
w_3 : Banda	0,7352	0 → 0,4947 0,4947 → 0,9419	1,2 2,1	0,2880	39	2,55

Tab. 14 Sensibilità del ranking ai valori dei pesi per la classe Streaming.

Per il traffico streaming il packet loss è considerato un criterio robusto, condizione intuibile dal peso così basso che possiede proprio perché dovrebbe aumentare di molto per influire significativamente nella selezione. Il ranking in questo caso è molto più sensibile alle variazioni del valore della banda che a quelle del ritardo.

Interactive:

Peso	Valore assoluto	Intervalli	Ranking	AT	PT(%)	Sensibilità
w_1 : Ritardo	0,1429	0 → 0,4934 0,4934 → 0,8571	2,1 1,2	0,3505	245	0,4082
w_2 Packet loss	0,7143	0 → 0,4934 0,4934 → 0,8571	1,2 2,1	0,3507	49	2,0408
w_3 : Banda	0,1429					

Tab. 15 Sensibilità del ranking ai valori dei pesi per la classe Interactive.

La banda è definita robusta, e la classe interactive è molto più sensibile alle variazioni del packet loss che a quelle del ritardo, come si lascia intuire dai suoi valori assoluti.

Background:

Peso	Valore assoluto	Intervalli	Ranking	AT	PT(%)	Sensibilità
w_1 : Ritardo	0,0581	0 → 0,4968 0,4968 → 0,7933	2,1 1,2	0,4387	755	0,1324
w_2 Packet loss	0,7352	0 → 0,2965 0,2965 → 0,7933	1,2 2,1	0,4387	59	1,6759
w_3 : Banda	0,2067					

Tab. 16 Sensibilità del ranking ai valori dei pesi per la classe Background.

Anche in questo caso la banda è definita robusta, e l'ordinamento è poco sensibile alle variazioni nel ritardo, essendolo un po più nel caso del ritardo.

IV.5 INFLUENZA DEL RANKING E DEI PARAMETRI NELLA SELEZIONE

Per vedere graficamente come piccole variazioni nel ranking delle reti influiscono nella selezione, in questa sezione si è creato uno scenario in cui si considerano tre reti e solo due liste di ranking di partenza, ovvero si hanno solo due parametri. Sono stati definiti due fattori $\delta_1 \in [-0.999, 0.999]$ e $\delta_2 \in [-0.999, 0.999]$, usati nella matrice di ranking che è introdotta negli algoritmi già visti. Più precisamente la matrice di ranking è allora definita come:

$$R = \begin{pmatrix} 1 & 2 + \delta_1 \\ 2 + \delta_2 & 1 \\ 3 & 3 \end{pmatrix}$$

Come si vede si è fissata la terza rete sempre nella posizione tre, per essere consistenti con la forma in cui si è calcolata inizialmente la matrice in studio a partire dai valori dei parametri, dove si fissavano le posizioni delle reti estremi: 1 ed N, e poi si calcolavano i ranking intermedi.

Si sono eseguite cento prove, a partire da cento diverse matrici R, calcolando le rispettive matrici di probabilità di transizione MC1 ed MC2, ed i vettori di stati stazionari.

Le seguenti due figure mostrano le probabilità di scelta al variare dei parametri definiti prima: δ_1 e δ_2 , rispettivamente per i metodi MC1 ed MC2.

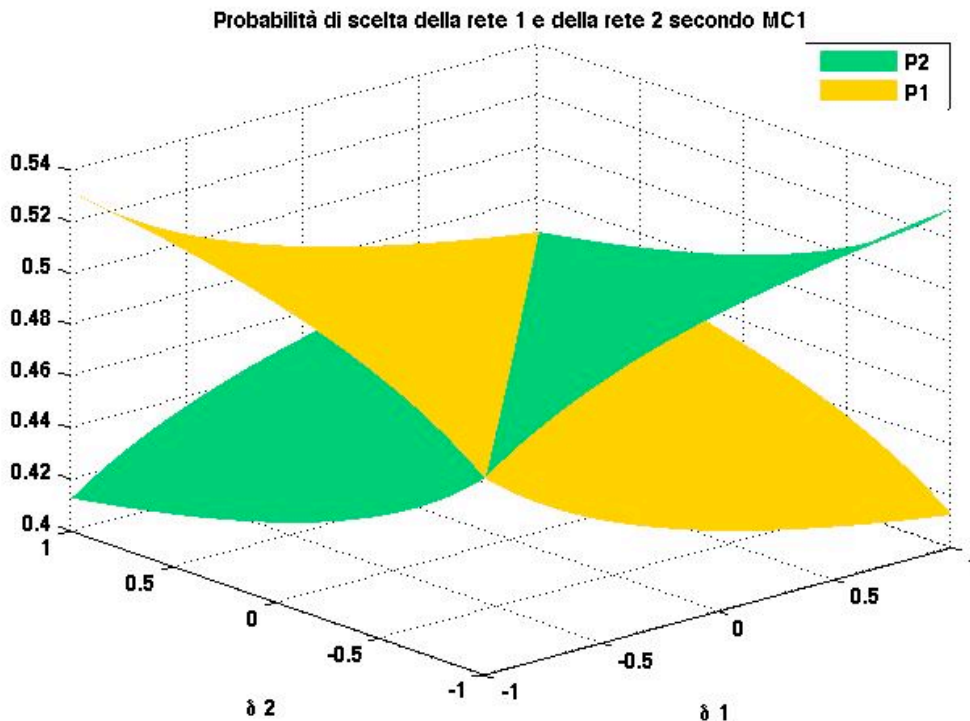


Fig. 13 Probabilità di scelta delle reti 1 e 2 secondo MC1 al variare il ranking tramite δ_1 ed δ_2 .

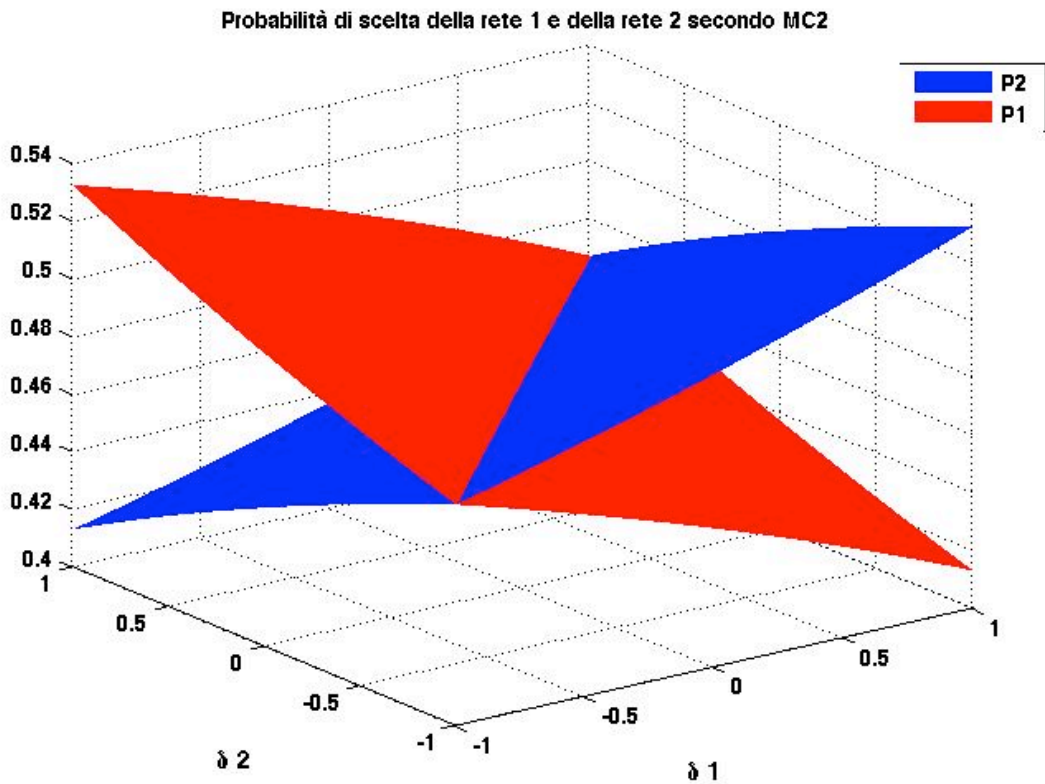


Fig. 14 Probabilità di scelta delle reti 1 e 2 secondo MC2 al variare il ranking tramite δ_1 ed δ_2 .

Dalle figure precedenti si può vedere, come ci si poteva attendere, che al variare dei delta per valori $\delta_1 > \delta_2$, entrambi i metodi scelgono la rete 2 e complementariamente per $\delta_1 < \delta_2$, scelgono la rete 1. La massima differenza tra le probabilità si ha nei casi limiti dei delta cioè in cui: $\delta_1 = -0.999, \delta_2 = 0.999$ che si traduce in:

$$R = \begin{pmatrix} 1 & 1.001 \\ 2.999 & 1 \\ 3 & 3 \end{pmatrix} \text{ e quindi con MC1 si sceglie la rete 1 con una probabilità}$$

$P_1 = 0.5203$ avendo la seconda $P_2 = 0.4128$, allora in questo caso specifico la rete 1 è selezionata con una probabilità 26.04% maggiore di quella di selezionare la rete 2. E per il caso di MC2: $P_1 = 0.5317$ e $P_2 = 0.4128$, con una differenza del 28.80%. Si hanno probabilità uguali in tutta la retta $\delta_1 = \delta_2$.

Questo studio si può anche estendere al caso in cui variano i parametri delle reti. Per vedere l'influenza si tengono in conto solo due parametri ad esempio banda e ritardo con valori normalizzati, per tre reti. La matrice dei parametri costituita da tre righe che rappresentano le tre reti e due colonne dei parametri, è aggiornata in ciascuno dei cento scenari dipendendo anche questa volta di due fattori chiamati delta primo: δ_1' e δ_2' .

$$PARAM = \begin{pmatrix} 1 & 0.5 + \delta_1' \\ 0.5 + \delta_2' & 0 \\ 0 & 1 \end{pmatrix}$$

Seguendo gli stessi ragionamenti del caso precedente, si sono ottenute due figure ognuna per ciascun metodo MC1 ed MC2, con due curve corrispondenti alle probabilità di scelta delle reti.

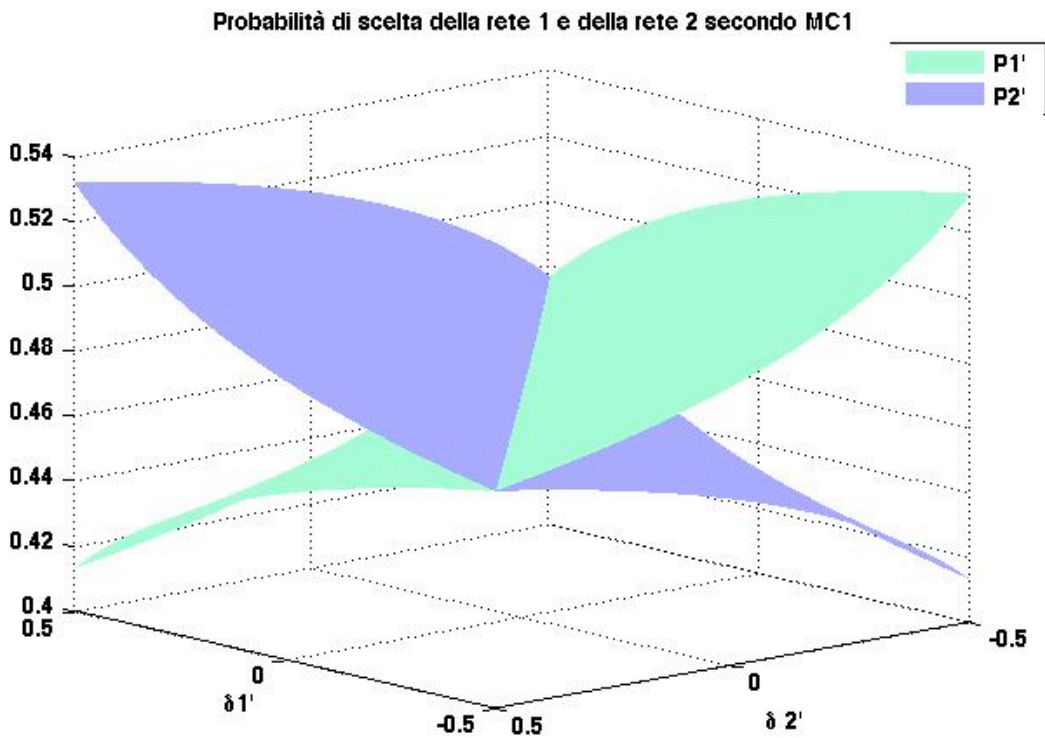


Fig. 15 Probabilità di scelta delle reti 1 e 2 secondo MC1 al variare i parametri tramite δ_1' ed δ_2' .

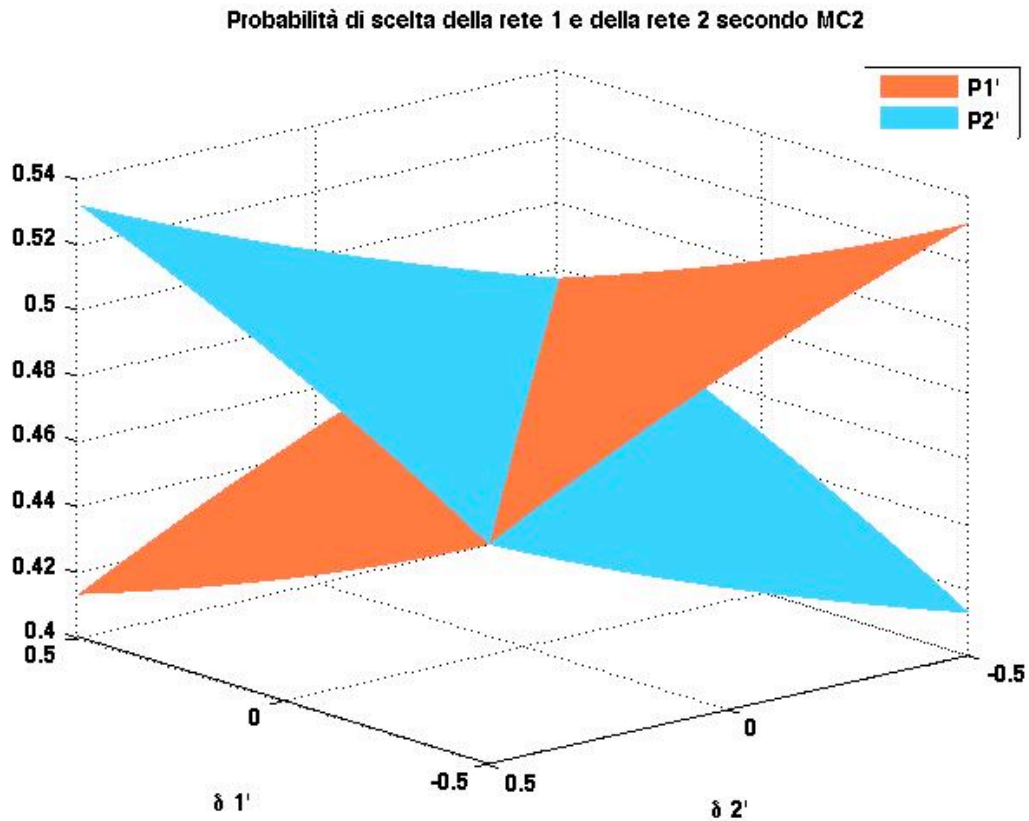


Fig. 16 Probabilità di scelta delle reti 1 e 2 secondo MC1 al variare i parametri tramite δ_1' ed δ_2' .

Dalle due figure precedenti si può concludere che per entrambe i metodi MC1 ed MC2 l'unica zona di incertezza nella scelta della rete è la retta $\delta_1' = -\delta_2'$ perché viene sempre scelta la rete 1 per valori di $\delta_1' < -\delta_2'$, e la rete 2 nel caso contrario. Le probabilità di scelta delle reti si discostano man mano che cresce questa differenza tra i δ' .

Si presenta una tabella riassuntiva con i casi limiti, ovvero valori dei rispettivi delta per cui si presenta il maggiore discostamento tra le probabilità di scelta, per entrambi i metodi.

Per MC1:

δ'_1	δ'_2	P_1	P_2	Relazione	% differenza
-0,499	-0,499	0,5316	0,4129	$\delta'_1 < -\delta'_2 \Rightarrow P_1 > P_2$	22,33%
0,499	0,499	0,4129	0,5316	$\delta'_1 > -\delta'_2 \Rightarrow P_2 > P_1$	22,33%

Tab. 13 Valori estremi di probabilità di scelta delle reti e percentuale di differenza per MC1.

Per MC2:

δ'_1	δ'_2	P_1	P_2	Relazione	% differenza
-0,499	-0,499	0,5316	0,4128	$\delta'_1 < -\delta'_2 \Rightarrow P_1 > P_2$	22,34%
0,499	0,499	0,4128	0,5316	$\delta'_1 > -\delta'_2 \Rightarrow P_2 > P_1$	22,34%

Tab. 14 Valori estremi di probabilità di scelta delle reti e percentuale di differenza per MC2.

IV.6 SCELTA DELLA RETE MIGLIORE CON CLASSE DI SERVIZIO

In questa sezione si analizzano i valori dei parametri delle reti scelte per ogni classe di servizio, che sono stati pesati diversamente per ciascuna, come è stato esplicitato nella proposta. Lo scenario è composto da sei reti con parametri diversi e dal momento che MC1 ed MC2 presentano circa le stesse prestazioni e scelgono le stesse reti, allora i risultati si mostrano solo nel caso di MC1.

Le seguenti tre figure mostrano i valori di ritardo, packet loss e banda per sei reti, non necessariamente diverse, scelte con i seguenti criteri: La prima con un parametro “ottimo” detto di riferimento, più specificamente ci sono tre casi diversi di reti con ritardo minimo, packet loss minimo oppure banda massima. La seconda rete scelta con il metodo MC1 con pesatura uniforme dei parametri. La terza scelta per la classe di servizio Conversational e le rimanenti tre per le classi streaming, interactive e background.

La seguente figura è composta da tre parti rappresentanti di ciascun parametro. E il parametro di riferimento in questo caso è il ritardo.

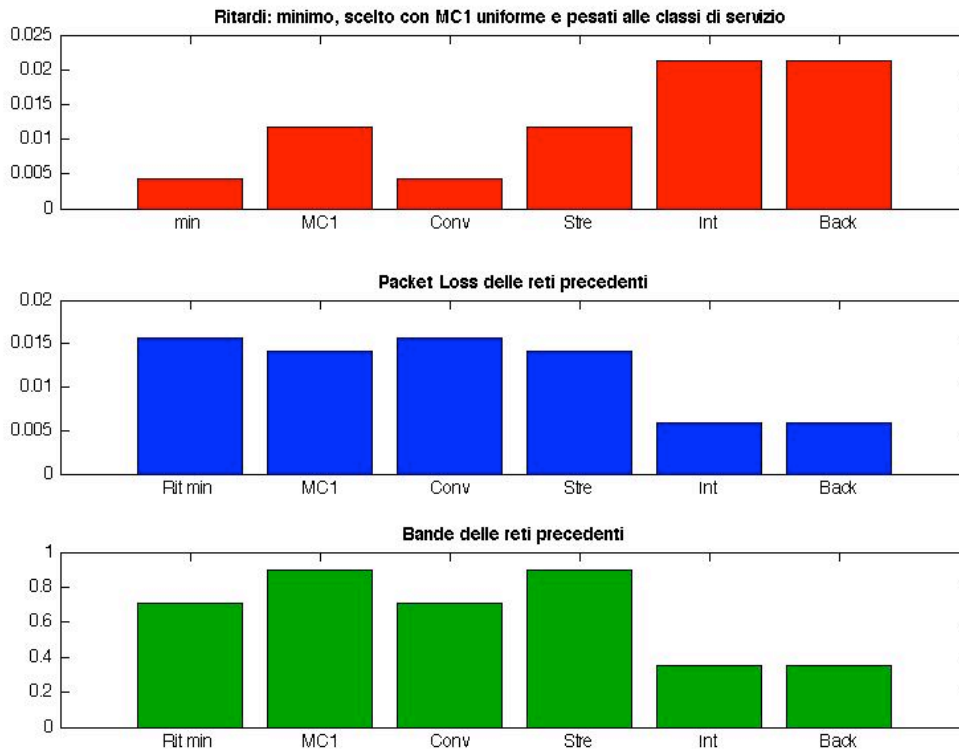


Fig. 17 Parametri scelti secondo diversi criteri, riferimento al ritardo minimo.

Analizzando il grafico è evidente come la rete con ritardo minore presenta valori svantaggiosi per il packet loss ed anche per la banda, essendo la stessa rete scelta per la classe conversational dove si pesa di più il ritardo in paragone con gli altri parametri. Si vede come le classi interactive e background scelgono la stessa rete perché i suoi pesi sono distribuiti in maniera molto simile.

La seguente figura è schematizzata uguale alla precedente, ma questa volta il parametro di riferimento è il packet loss.

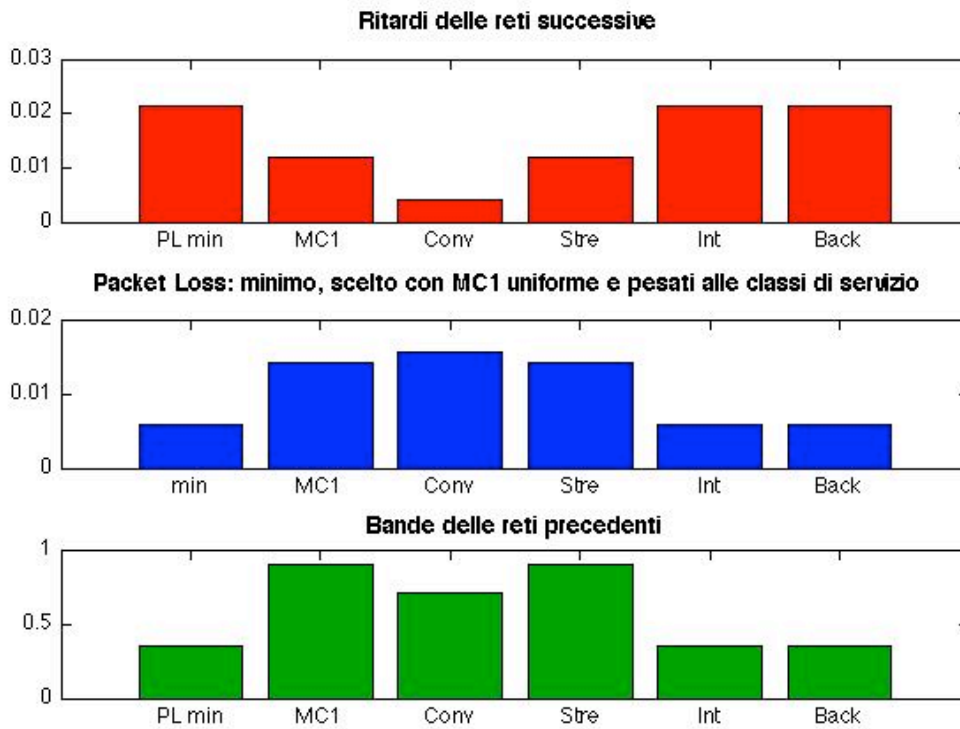


Fig. 18 Parametri scelti secondo diversi criteri, riferimento al packet loss minimo.

Anche in questo caso si vede come la rete che offre un parametro ottimo, cioè il packet loss minimo, presenta svantaggi con rispetto agli altri. Ancora una volta le classi di servizio interactive e background scelgono la stessa rete, che è proprio quella con il packet loss minore, per essere il suo peso maggiore tra gli altri.

Nella figura successiva il parametro di riferimento è la banda.

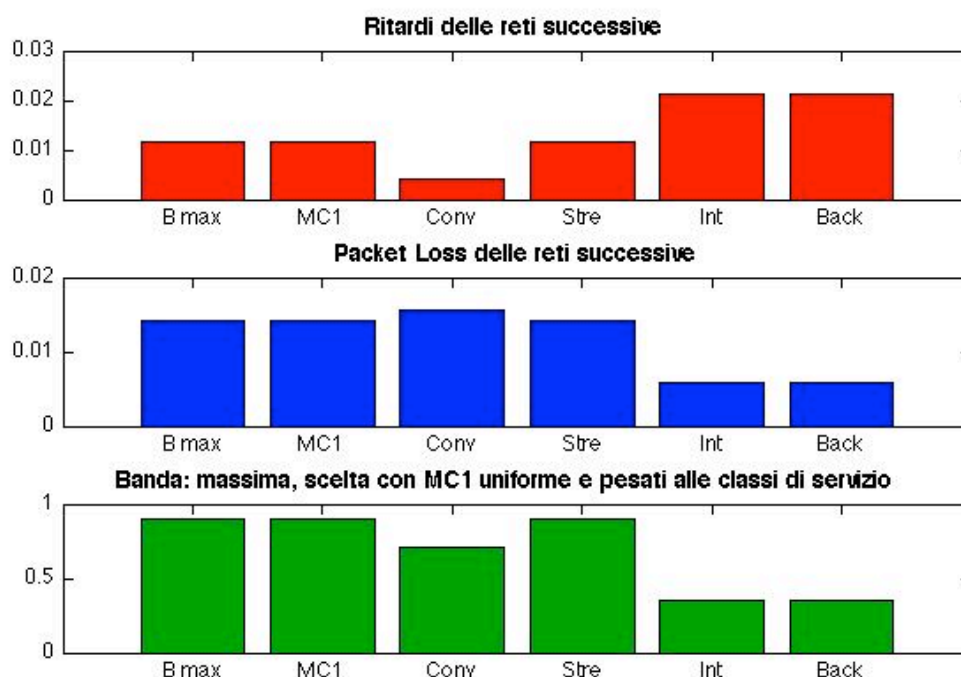


Fig. 19 Parametri scelti secondo diversi criteri, riferimento al packet loss minimo.

In questo caso specifico, la rete scelta per il suo valore di banda coincide con quella scelta da MC1, proprio perché presenta dei valori di packet loss e ritardo che non sono i maggiori, ovvero presenta dei vantaggi qualsiasi sia il parametro. La rete scelta per la classe streaming è quella con la banda massima per essere il parametro più rilevante per questa.

Riassumendo, dai risultati precedenti si possono arrivare a diverse conclusioni importanti. Il risultato più evidente è che per ciascuna classe di servizio si sceglie una rete diversa a seconda del valore del parametro più rilevante. In genere, la rete che offre il valore ottimo in un parametro, non necessariamente è vantaggiosa rispetto agli altri, ma il metodo MC1 sempre sceglie quella che presenta i parametri migliori in maniera uniforme. Infine un fattore molto determinante nella selezione a parte le proprie catene e la matrice di ranking, sono i pesi già studiati.

Nelle prossime tre figure, si cerca di studiare uno scenario di multiservizio con due reti. A questo scopo si sono definiti tre fattori corrispondenti alla differenza tra i valori dei parametri di ciascuna rete: $\delta_1 \in [-0.999, 0.999]$, $\delta_2 \in [-0.999, 0.999]$, $\delta_3 \in [-0.999, 0.999]$, variano in questo rango proprio per considerare tutti i valori possibili che possono assumere i parametri. Le corrispondenti equazioni:

$$\delta_1 = R_1 - R_2, \text{ Differenza rispetto al ritardo.}$$

$$\delta_2 = PL_1 - PL_2, \text{ Differenza rispetto al packet loss.}$$

$$\delta_3 = B_1 - B_2, \text{ Differenza rispetto alla banda.}$$

Questa differenza incide direttamente sulla scelta della rete, e dal momento che si sono coinvolti i pesi dei parametri, si sono anche considerate le classi di servizio.

In ciascuna delle seguenti figure si mostra la scelta della rete 1 nella parte sinistra, e della rete 2 nella parte destra, per i casi in cui si considera l'erogazione di una sola classe di servizio, di due, tre e di tutte simultaneamente, per tanto, i due grafici mostrati in ciascuna figura sono complementari.

Nella figura 20 si è fissato il valore del δ_3 a -0.999, essendo la seconda rete più attraente dal punto di vista della banda.

Selezione rete 1 o 2, multipli classi $B_1 < B_2$

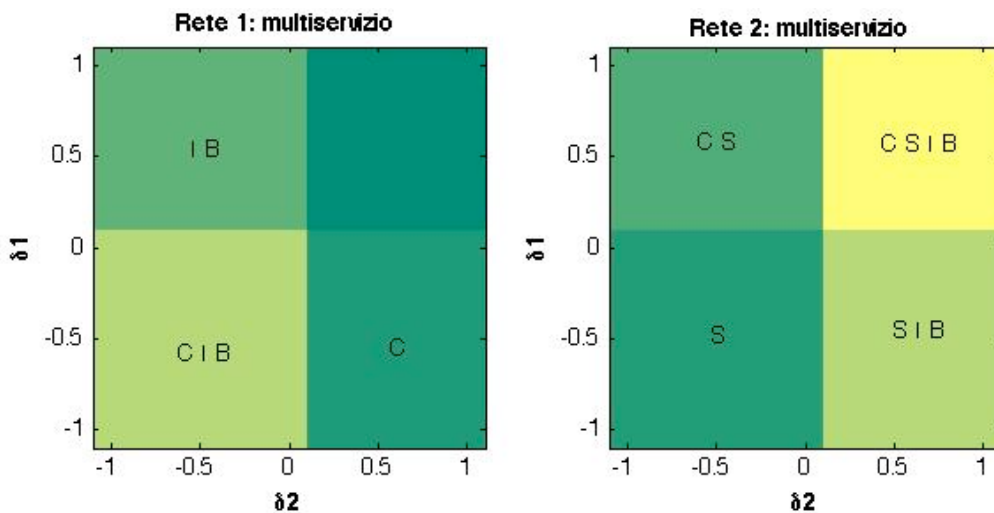


Fig. 20 Scelta reti scenario multiservizio, $\delta_3 = -0.999 \Rightarrow B_1 < B_2$

Come si sperava, valori negativi dei delta inducono alla scelta della rete 1, e valori positivi favoriscono alla seconda. Proprio per il valore del delta riferito alla banda, la rete 2 è preferita per tutte le classi nella zona dei restanti delta positivi.

Nella figura 21 si è fissato il valore del δ_3 a 0, dunque i valori di banda per entrambi i reti sono uguali.

Selezione rete 1 o 2, multipli classi $B_1 = B_2$

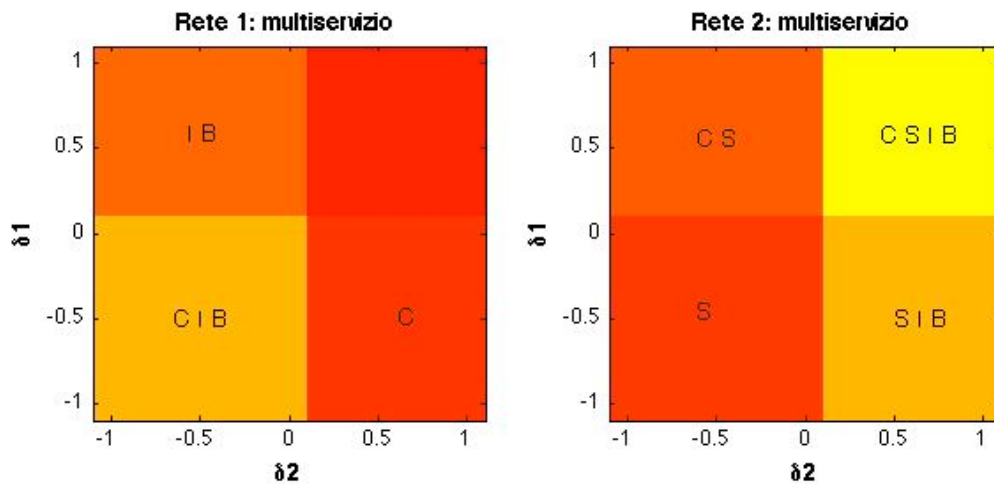


Fig. 21 Scelta reti scenario multiservizio, $\delta_3 = 0 \Rightarrow B_1 = B_2$

Anche in questo caso, i grafici sono complementari, e la rete 2 è preferita per erogare tutti i quattro classi di servizio per i delta positivi, cioè dove i parametri ottimi li presentano quella rete.

Nella figura 22 si è fissato il valore del δ_3 a 0.999, essendo la prima rete più attraente dal punto di vista della banda.

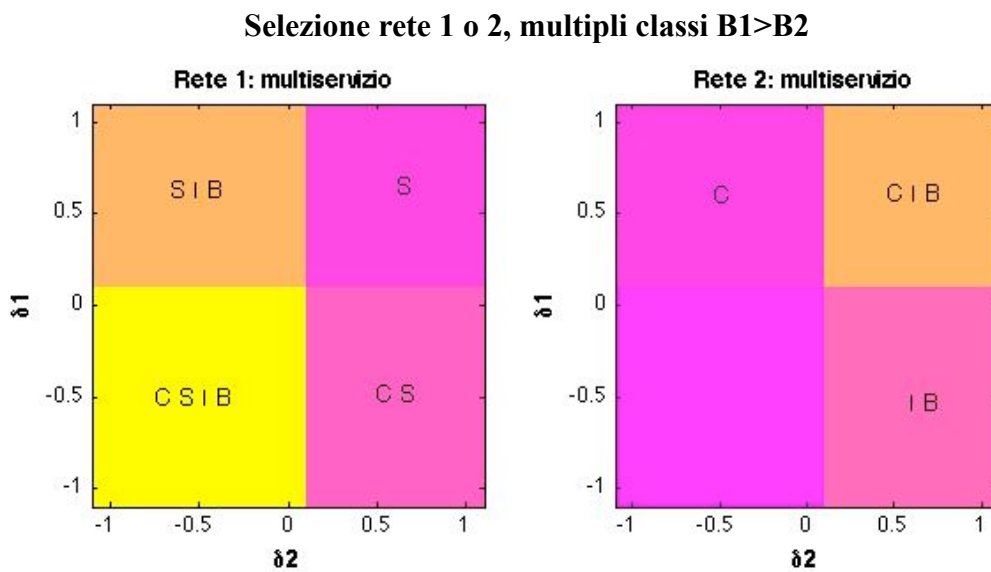


Fig. 22 Scelta reti scenario multiservizio, $\delta_3 = 0.999 \Rightarrow B_1 > B_2$

Per essere la banda della rete 1 maggiore che quella della 2, la prima risulta scelta per il multiservizio per tutti i valori dei delta minori da zero, cioè quando presenta i parametri ottimi, anche senza importare la pesatura, come è logico.

L'asse corrispondente a ciascun parametro determina anche quali specifici classi si assegnano a la rete in studio, dipendendo dalla rilevanza di quel parametro a quella classe.

CONCLUSIONI



La completa connettività è la sfida più grande nell'ambito delle comunicazioni qualsiasi sia lo scenario e le sue condizioni quindi le procedure di handover diventano più importanti man mano che cresce la disponibilità delle diverse tecnologie. Tali tecnologie offrono una varietà di prestazioni che possono adeguarsi ai requisiti degli utenti ed allo specifico servizio richiesto, non essendo più l'unica o principale necessità la comunicazione voce, bensì il trasferimento di file, la ricerca nel web, la messaggistica istantanea.

Finora molti sono gli avanzamenti ottenuti in materia di handover verticale ed i contributi crescono ogni giorno, ma non si è raggiunta la desiderata standardizzazione proprio perché la dipendenza della decisione può variare molto a seconda degli attributi, tecnologie, numero di utenti, tecniche, tipo di servizio ed infine dello scenario considerato, e risulta evidente la difficoltà o pratica impossibilità di considerare tutti gli esistenti scenari.

La presente ricerca si è focalizzata in tre premesse che si ritengono molto importanti nella tecnica specifica impiegata, handover verticale basato sul rank aggregation con un approccio nelle catene di Markov, che sono: migliorare le liste ordinate, strutturandole in maniera più sensibile e diretta ai parametri pervenuti da ciascuna alternativa, in secondo luogo aumentare la sensibilità delle catene impiegate a questo ranking, in maniera di sfruttare più ampiamente quello che rappresenta ogni posizione, ed il costo aggiuntivo che porta uno spostamento tra reti, e come ultima premessa si è ritenuta l'importanza di ciascun parametro studiato alla relativa classe di servizio considerata, tramite la pesatura. Tutto ciò si è messo negli algoritmi di decisione, che in definitiva risultano più sensibili a ciascuna premessa nominata.

Confrontando i lavori precedenti con i risultati ottenuti in tutte le prove già descritte, si dimostra come questi ultimi possono ritenersi soddisfacenti, dal momento in cui la selezione fornisce parametri che in media superano quelli forniti dai metodi di partenza. D'altro canto essi si adeguano in maniera molto sensibile al tipo di servizio richiesto includendo anche scenari con multipli classi di servizi, in riscontro alle modifiche eseguite.

Per i prossimi lavori si suggerisce di includere un insieme più ampio di attributi considerando anche parametri centrati sul terminale (risparmio di batteria) o sull'utente (costo di servizio). Dal momento che l'algoritmo sviluppato non presenta modifiche rispetto al ranking nei casi in cui ci sono soltanto due reti alternative, si consiglia applicarlo quando esistono più reti nella regione in studio, e si lascia a lavori futuri lo sviluppo di un nuovo ranking specifico per i tali casi. Si consiglia inoltre introdurre un costo nell'esecuzione della procedura di handover introducendo una probabilità minore nel passaggio ad uno stato diverso.

APPENDICE

```

% Calcolo del vettore di probabilità degli Stati Stazionari

function [SD]=calc_SD(M)

N = size (M,1);
M1=M;
M2=zeros(N,N);

while (M1~=M2)
    M1=M1*M;
    M2=M1*M;
end

SD=((1/N)*ones(1,N))*M1;

% Calcola il vettore di probabilità degli Stati Stazionari per ciascuna
% Classe di Servizio

function [SD_TC]=calc_SD_TC(M)

[N,N,T] = size (M);
SD_TC = zeros(N,T);

for l = 1:T
    L(:,:)=M(:,:,l);
    L1(:,:)=M(:,:,l);
    L2(:,:)= L1*L1;
    while (L1~=L2)
        L1=L1*L;
        L2=L1*L;
    end
    SD_TC(:,l)=((1/N)*ones(1,N))*L1;
end

% Aggiornamento della matrice di probabilità di transizione tra le diverse
% reti

function [MC1] = calcMC1(r1)

[N,M] = size (r1);

MC1 = zeros(N,N); % Inizializzo la matrice di transizione di catene di Markov
pesate MC
w = 1/M; % Peso dei parametri

for q=1:M
    for i=1:N
        for j=1:N
            if r1(i,q) >= r1(j,q)
                MC1(i,j) = MC1(i,j) + w/(r1(j,q));
            end
        end
    end
end
end
```

```

% Normalizzazione della matrice MC1 dovuto al ranking

row_MC1 = sum(MC1,2);

for i=1:N
    MC1(i,:) = MC1(i,:)/row_MC1(i);
end

% Prende in ingresso la matrice dei ranking e calcola la matrice di
% probabilità di transizione secondo il primo metodo delle catene de
% Markov, ma questa volta considerando i pesi dei parametri per ogni classe
% di servizio. Essendo Conversazionale (1), Streaming (2), Background (3)

function [MC1_TC] = calcMC1_TC(r1,w)

[N,M] = size (r1);
T = 4; %Numero di classi di servizio presse in considerazione
MC1_TC = zeros(N,N,T); % Inizializzo la matrice di transizione di catene di Markov
pesate MC

for l=1:T
    for q=1:M
        for i=1:N
            for j=1:N
                if r1(i,q) >= r1(j,q)
                    MC1_TC(i,j,l) = MC1_TC(i,j,l) + w(l,q)/(r1(j,q));
                end
            end
        end
    end
end

% Normalizzazione della matrice MC1 dovuto al ranking

row_MC1 = zeros(N,T);

for l=1:T
    row_MC1(:,l) = sum(MC1_TC(:, :, l),2);
    for i=1:N
        MC1_TC(i, :, l) = MC1_TC(i, :, l)/row_MC1(i,l);
    end
end

% Aggiornamento della matrice di probabilità di transizione tra le diverse
% reti

function [MC2] = calcMC2(r1)

[N,M] = size (r1);

MC2 = zeros(N,N); % Inizializzo la matrice di transizione di catene di Markov
pesate MC
w = 1/M; % Peso per ogni parametro

for q=1:M
    for i=1:N
        for j=1:N
            if r1(i,q)>=r1(j,q)

```

```

                MC2(i,j)=MC2(i,j)+(N-r1(j,q)+1)*(w/N);
            end
        end
    end
end

row_MC2 = sum(MC2,2);

for i=1:N
    MC2(i,:) = MC2(i,:)./row_MC2(i);
end

% ALGORITMO ORIGINALE: Aggiornamento della matrice di probabilità di transizione
tra le diverse reti

function [MC2] = calcMC2_ini(r1)

[N,M] = size (r1);

MC2 = zeros(N,N); % Inizializzo la matrice di transizione di catene di Markov
pesate MC
w = 1/M; % Peso per ogni parametro

for q=1:M
    for i=1:N
        for j=1:N

            if r1(i,q)>r1(j,q)
                MC2(i,j)=MC2(i,j)+(N-r1(j,q)+1)*(w/N);
            elseif r1(i,q)==r1(j,q)
                MC2(i,j)=MC2(i,j)+(N-r1(j,q)+1)*(w/N);
            end
        end
    end
end

row_MC2 = sum(MC2,2);

for i=1:N
    MC2(i,:) = MC2(i,:)./row_MC2(i);
end

% Prende in ingresso la matrice dei ranking e calcola la matrice di
% probabilità di transizione secondo il secondo metodo delle catene de
% Markov, ma questa volta considerando i pesi dei parametri w per ogni classe
% di servizio. Essendo Conversazionale (1), Streaming (2), Background (3)

function [MC2_TC] = calcMC2_TC(r1,w)

[N,M] = size (r1);
T = 4; %Numero di classi di servizio presse in considerazione
MC2_TC = zeros(N,N,T); % Inizializzo la matrice di transizione di catene di Markov
pesate MC

for l=1:T
    for q=1:M
        for i=1:N
            for j=1:N
                if r1(i,q) >= r1(j,q)

```

```

                MC2_TC(i,j,l) = MC2_TC(i,j,l)+(N-r1(j,q)+1)*(w(l,q)/N);
            end
        end
    end
end

% Normalizzazione della matrice MC1 dovuto al ranking
row_MC2 = zeros(N,T);

for l=1:T
    row_MC2(:,l) = sum(MC2_TC(:,:,l),2);
    for i=1:N
        MC2_TC(i,:,l) = MC2_TC(i,:,l)/row_MC2(i,l);
    end
end

ris = zeros(1000,5);
div1 = 0;
div2 = 0;
n_prov = 1000;

Rit_MC1 = zeros(1,n_prov);
Rit_MC2 = zeros(1,n_prov);
Rit_MC1_ini = zeros(1,n_prov);
Rit_MC2_ini = zeros(1,n_prov);

PL_MC1 = zeros(1,n_prov);
PL_MC2 = zeros(1,n_prov);
PL_MC1_ini = zeros(1,n_prov);
PL_MC2_ini = zeros(1,n_prov);

B_MC1 = zeros(1,n_prov);
B_MC2 = zeros(1,n_prov);
B_MC1_ini = zeros(1,n_prov);
B_MC2_ini = zeros(1,n_prov);

for s=1:n_prov

    Parametri;
    r = r_mod (Param);
    r_ini = r_mod_ini(Param);
    [N,M] = size (r);
    SD = (1/N)*ones(1,N);

    % Selezione la rete favorita secondo primo metodo
    MC1 = calcMC1(r);
    % Selezione la rete favorita secondo il secondo metodo
    MC2 = calcMC2(r);

    % Selezione la rete favorita secondo primo metodo INIZIALE
    MC1_ini = calcMC1(r_ini);
    % Selezione la rete favorita secondo il secondo metodo INIZIALE
    MC2_ini = calcMC2_ini(r_ini);

    % Calcolo il vettore degli Stati Stazionari
    SD_1 = calc_SD(MC1);
    SD_2 = calc_SD(MC2);

```

```

% Calcolo il vettore degli Stati Stazionari. INIZIALE
SD_1_ini = calc_SD(MC1_ini);
SD_2_ini = calc_SD(MC2_ini);

% Seleziono la rete con maggiore valore di sd
[Fav_1, F_1] = max (SD_1);
[Fav_2, F_2] = max (SD_2);

% Seleziono la rete con maggiore valore di sd. INIZIALE
[Fav_1_ini, F_1_ini] = max (SD_1_ini);
[Fav_2_ini, F_2_ini] = max (SD_2_ini);
[CC,fav_TOPSIS] = TOPSIS(Param');

ris(s,1)=F_1;
ris(s,2)=F_2;
ris(s,3)=F_1_ini;
ris(s,4)=F_2_ini;
ris(s,5)=fav_TOPSIS;

if ris(s,1)~=ris(s,3)
    div1 = div1 +1; % Caso in cui la rete scelta da MC1 è diversa da MC1_ini
end
if ris(s,2)~=ris(s,4)
    div2 = div2 +1; % Caso in cui la rete scelta da MC1 è diversa da MC1_ini
end

Rit_MC1(s) = D(F_1);
Rit_MC2(s) = D(F_2);
Rit_MC1_ini(s) = D(F_1_ini);
Rit_MC2_ini(s) = D(F_2_ini);

PL_MC1(s) = L(F_1);
PL_MC2(s) = L(F_2);
PL_MC1_ini(s) = L(F_1_ini);
PL_MC2_ini(s) = L(F_2_ini);

B_MC1(s) = ABW(F_1);
B_MC2(s) = ABW(F_2);
B_MC1_ini(s) = ABW(F_1_ini);
B_MC2_ini(s) = ABW(F_2_ini);

end

Y=[div1 div2];
bar(Y,0.2)

B_med_MC1 = mean(B_MC1);
B_dev_MC1 = std(B_MC1);
B_med_MC1_ini = mean(B_MC1_ini);
B_dev_MC1_ini = std(B_MC1_ini);
Band_MC1 = [B_med_MC1 B_med_MC1_ini];
B_med_MC2 = mean(B_MC2);
B_dev_MC2 = std(B_MC2);
B_med_MC2_ini = mean(B_MC2_ini);
B_dev_MC2_ini = std(B_MC2_ini);
Band_MC2 = [B_med_MC2 B_med_MC2_ini];
Banda_med = [Band_MC1;Band_MC2];
Banda_dev = [B_dev_MC1 B_dev_MC1_ini B_dev_MC2 B_dev_MC2_ini];

PL_med_MC1 = mean(PL_MC1);

```

```

PL_dev_MC1 = std(PL_MC1);
PL_med_MC1_ini = mean(PL_MC1_ini);
PL_dev_MC1_ini = std(PL_MC1_ini);
Pack_MC1 = [PL_med_MC1 PL_med_MC1_ini];
PL_med_MC2 = mean(PL_MC2);
PL_dev_MC2 = std(PL_MC2);
PL_med_MC2_ini = mean(PL_MC2_ini);
PL_dev_MC2_ini = std(PL_MC1_ini);
Pack_MC2 = [PL_med_MC2 PL_med_MC2_ini];
PackL_med = [Pack_MC1;Pack_MC2];
PackL_dev = [PL_dev_MC1 PL_dev_MC1_ini PL_dev_MC2 PL_dev_MC2_ini];

Rit_med_MC1 = mean(Rit_MC1);
Rit_dev_MC1 = std(Rit_MC1);
Rit_med_MC1_ini = mean(Rit_MC1_ini);
Rit_dev_MC1_ini = std(Rit_MC1_ini);
Ritar_MC1 = [Rit_med_MC1 Rit_med_MC1_ini];
Rit_med_MC2 = mean(Rit_MC2);
Rit_dev_MC2 = std(Rit_MC2);
Rit_med_MC2_ini = mean(Rit_MC2_ini);
Rit_dev_MC2_ini = std(Rit_MC2_ini);
Ritar_MC2 = [Rit_med_MC2 Rit_med_MC2_ini];
Ritardo_med = [Ritar_MC1;Ritar_MC2];
Ritardo_dev = [Rit_dev_MC1 Rit_dev_MC1_ini Rit_dev_MC2 Rit_dev_MC2_ini];

figure (11)
subplot (3,1,1)
bar (Ritardo_med,'group')
hold on
errorbar([0.86 1.14 1.86 2.14],[Ritar_MC1 Ritar_MC2],Ritardo_dev,'xr')
grid on
title 'Ritardo, Packet Loss e Banda medi delle reti scelte: MC1, MC1 orig, MC2 ed
MC2 orig'
colormap winter
subplot (3,1,2)
bar (PackL_med,'group')
hold on
errorbar([0.86 1.14 1.86 2.14],[Pack_MC1 Pack_MC2],PackL_dev,'xr')
grid on
colormap autumn
subplot (3,1,3)
bar (Banda_med,'group')
colormap summer
hold on
errorbar([0.86 1.14 1.86 2.14],[Band_MC1 Band_MC2],Banda_dev,'xr')
grid on

% Generazione parametro Ritardo, mediante un sistema a coda M/M/1

% generazione aleatoria dei rate di arrivi e partenze

function [D] = delay

miu = rand(1); % Rate di partenze
lambda = rand(1); % Rate di arrivi

while (miu <= lambda)
    lambda = rand(1);
end

```

```

D = 1/(miu - lambda);

% Studio della selezione della rete in funzione della distanza tra i
% parametri, considerando il caso di multiservizio, tre parametri e tre
% reti, ma la terza con posizione 3 per ciascun parametro

clear
w= [0.8181 0.0909 0.0909; 0.2067 0.0581 0.7352; 0.1429 0.7143 0.1429; 0.0581 0.7352
0.2067]; % Pesi dei parametri
Y1 = zeros(11,11);
Y2 = zeros(11,11);
Y3 = zeros(11,11);
Y4 = zeros(11,11);
Y5 = zeros(11,11);
Y6 = zeros(11,11);

delta_1 = linspace (-1,1,11);
delta_2 = linspace (-1,1,11);
delta_3 = linspace (-1,1,11);

for i=1:length(delta_1)
    for j=1:length(delta_2)
        for k=1:length(delta_3)
            if (delta_1(i)>0)
                r(1:2,1)=[2 1]';
            else
                r(1:2,1)=[1 2]';
            end
            if (delta_2(j)>0)
                r(1:2,2)=[2 1]';
            else
                r(1:2,2)=[1 2]';
            end
            if (delta_3(k)>0)
                r(1:2,3)=[1 2]';
            else
                r(1:2,3)=[2 1]';
            end
            MC1_TC = calcMC1_TC(r,w);
            SD_1_TC = calc_SD_TC(MC1_TC);
            [Fav_1, F_1] = max (SD_1_TC);
            fav(i,j,k,1:4)=F_1;
        end
    end
end

% Caso in cui è preferita una stessa rete per tutte le classi di servizio per
% tutti i valori dei delta

C_1 = (fav(:,:,,1:4)==1); % Scelta la rete 1
C_2 = (fav(:,:,,1:4)==2); % Scelta la rete 2

% Valuto per tre valori di delta_3 diversi:

%% Delta 3 = -1

Coinc_1_a1 = (C_1(:,:,1,1))&(C_1(:,:,1,2))&(C_1(:,:,1,3))&(C_1(:,:,1,4)); % Scelta
la rete 1 per tutti e quattro le classi

```



```

Coinc_2_a1 = (C_2(:, :, 1, 1)) & (C_2(:, :, 1, 2)) & (C_2(:, :, 1, 3)) & (C_2(:, :, 1, 4)); % Scelta
la rete 2 per tutti e quattro le classi

Coinc_1_b1 = (C_1(:, :, 1, 1)) & (C_1(:, :, 1, 2)) & (C_1(:, :, 1, 3)); % Scelta la rete 1 per
le classi conversational streaming e interactive
Coinc_2_b1 = (C_2(:, :, 1, 1)) & (C_2(:, :, 1, 2)) & (C_2(:, :, 1, 3)); % Scelta la rete 2 per
le classi conversational streaming e interactive

Coinc_1_c1 = (C_1(:, :, 1, 1)) & (C_1(:, :, 1, 2)) & (C_1(:, :, 1, 4)); % Scelta la rete 1 per
le classi conversational streaming e background
Coinc_2_c1 = (C_2(:, :, 1, 1)) & (C_2(:, :, 1, 2)) & (C_2(:, :, 1, 4)); % Scelta la rete 2 per
le classi conversational streaming e background

Coinc_1_d1 = (C_1(:, :, 1, 2)) & (C_1(:, :, 1, 3)) & (C_1(:, :, 1, 4)); % Scelta la rete 1 per
le classi streaming interactive e background
Coinc_2_d1 = (C_2(:, :, 1, 2)) & (C_2(:, :, 1, 3)) & (C_2(:, :, 1, 4)); % Scelta la rete 2 per
le classi streaming interactive e background

Coinc_1_e1 = (C_1(:, :, 1, 1)) & (C_1(:, :, 1, 3)) & (C_1(:, :, 1, 4)); % Scelta la rete 1 per
le classi conversational interactive e background
Coinc_2_e1 = (C_2(:, :, 1, 1)) & (C_2(:, :, 1, 3)) & (C_2(:, :, 1, 4)); % Scelta la rete 2 per
le classi conversational interactive e background

Coinc_1_f1 = (C_1(:, :, 1, 1)) & (C_1(:, :, 1, 2)); % Scelta la rete 1 per le classi
conversational e streaming
Coinc_2_f1 = (C_2(:, :, 1, 1)) & (C_2(:, :, 1, 2)); % Scelta la rete 2 per le classi
conversational e streaming

Coinc_1_g1 = (C_1(:, :, 1, 1)) & (C_1(:, :, 1, 4)); % Scelta la rete 1 per le classi
conversational e background
Coinc_2_g1 = (C_2(:, :, 1, 1)) & (C_2(:, :, 1, 4)); % Scelta la rete 2 per le classi
conversational e background

Coinc_1_h1 = (C_1(:, :, 1, 1)) & (C_1(:, :, 1, 3)); % Scelta la rete 1 per le classi
conversational e interactive
Coinc_2_h1 = (C_2(:, :, 1, 1)) & (C_2(:, :, 1, 3)); % Scelta la rete 2 per le classi
conversational e interactive

Coinc_1_i1 = (C_1(:, :, 1, 2)) & (C_1(:, :, 1, 4)); % Scelta la rete 1 per le classi
streaming e background
Coinc_2_i1 = (C_2(:, :, 1, 2)) & (C_2(:, :, 1, 4)); % Scelta la rete 2 per le classi
streaming e background

Coinc_1_j1 = (C_1(:, :, 1, 2)) & (C_1(:, :, 1, 3)); % Scelta la rete 1 per le classi
streaming e interactive
Coinc_2_j1 = (C_2(:, :, 1, 2)) & (C_2(:, :, 1, 3)); % Scelta la rete 2 per le classi
streaming e interactive

Coinc_1_k1 = (C_1(:, :, 1, 3)) & (C_1(:, :, 1, 4)); % Scelta la rete 1 per le classi
interactive e background
Coinc_2_k1 = (C_2(:, :, 1, 3)) & (C_2(:, :, 1, 4)); % Scelta la rete 2 per le classi
interactive e background

Sing_1_l1= (C_1(:, :, 1, 1)); % Scelta la rete 1 per la classe conversational
Sing_2_l1= (C_2(:, :, 1, 1)); % Scelta la rete 2 per la classe conversational

Sing_1_m1= (C_1(:, :, 1, 2)); % Scelta la rete 1 per la classe streaming
Sing_2_m1= (C_2(:, :, 1, 2)); % Scelta la rete 2 per la classe streaming

Sing_1_n1= (C_1(:, :, 1, 3)); % Scelta la rete 1 per la classe interactive
Sing_2_n1= (C_2(:, :, 1, 3)); % Scelta la rete 2 per la classe interactive

```

```

Sing_1_o1= (C_1(:, :, 1, 4)); % Scelta la rete 1 per la classe background
Sing_2_o1= (C_2(:, :, 1, 4)); % Scelta la rete 2 per la classe background

% Rappresentazione grafica per delta_3 = -1 => Banda_1 < Banda_2 :
% favorisce di molto la scelta per la rete 2 in streaming e un pò di meno
% in Background (dovuto ai pesi)

%assegnazione dei valori di colori quando si sceglie la rete 1 per: le
%diverse combinazioni di classi di servizio

for i=1:11
    for j=1:11
        if (Coinc_1_a1(i,j)==0)

            if (Coinc_1_b1(i,j)==0)&&(Coinc_1_c1(i,j)==0)&&(Coinc_1_e1(i,j)==0)&&
(Coinc_1_f1(i,j)==0)&&(Coinc_1_g1(i,j)==0)&&(Coinc_1_h1(i,j)==0)&&(Sing_1_l1(i,j)
==1)
                Y1(i,j)=30;
            end
            if (Coinc_1_b1(i,j)==0)&&(Coinc_1_c1(i,j)==0)&&(Coinc_1_d1(i,j)==0)&&
(Coinc_1_f1(i,j)==0)&&(Coinc_1_i1(i,j)==0)&&(Coinc_1_j1(i,j)==0)&&(Sing_1_m1(i,j)
==1)
                Y1(i,j)=32;
            end
            if (Coinc_1_b1(i,j)==0)&&(Coinc_1_d1(i,j)==0)&&(Coinc_1_e1(i,j)==0)&&
(Coinc_1_h1(i,j)==0)&&(Coinc_1_j1(i,j)==0)&&(Coinc_1_k1(i,j)==0)&&(Sing_1_n1(i,j)
==1)
                Y1(i,j)=34;
            end
            if (Coinc_1_c1(i,j)==0)&&(Coinc_1_d1(i,j)==0)&&(Coinc_1_e1(i,j)==0)&&
(Coinc_1_g1(i,j)==0)&&(Coinc_1_i1(i,j)==0)&&(Coinc_1_k1(i,j)==0)&&(Sing_1_o1(i,j)
==1)
                Y1(i,j)=36;
            end

            if (Coinc_1_b1(i,j)==0)&&(Coinc_1_c1(i,j)==0)&&(Coinc_1_f1(i,j)==1)
                Y1(i,j)=70;
            end
            if (Coinc_1_c1(i,j)==0)&&(Coinc_1_e1(i,j)==0)&&(Coinc_1_g1(i,j)==1)
                Y1(i,j)=72;
            end
            if (Coinc_1_b1(i,j)==0)&&(Coinc_1_e1(i,j)==0)&&(Coinc_1_h1(i,j)==1)
                Y1(i,j)=74;
            end
            if (Coinc_1_c1(i,j)==0)&&(Coinc_1_d1(i,j)==0)&&(Coinc_1_i1(i,j)==1)
                Y1(i,j)=76;
            end
            if (Coinc_1_b1(i,j)==0)&&(Coinc_1_d1(i,j)==0)&&(Coinc_1_j1(i,j)==1)
                Y1(i,j)=78;
            end
            if (Coinc_1_d1(i,j)==0)&&(Coinc_1_e1(i,j)==0)&&(Coinc_1_k1(i,j)==1)
                Y1(i,j)=80;
            end

            if Coinc_1_b1(i,j)==1
                Y1(i,j)=164;
            end
            if Coinc_1_c1(i,j)==1
                Y1(i,j)=166;
            end
        end
    end
end

```

```

        end
        if Coinc_1_d1(i,j)==1
            Y1(i,j)=168;
        end
        if Coinc_1_e1(i,j)==1
            Y1(i,j)=170;
        end
    else
        Y1(i,j)=255;
    end
end
end
end

```

%assegnazione dei valori di colori quando si sceglie la rete 2 per: le
%diverse combinazioni di classi di servizio

```

for i=1:11
    for j=1:11
        if (Coinc_2_a1(i,j)==0)

            if (Coinc_2_b1(i,j)==0)&&(Coinc_2_c1(i,j)==0)&&(Coinc_2_e1(i,j)==0)&&
(Coinc_2_f1(i,j)==0)&&(Coinc_2_g1(i,j)==0)&&(Coinc_2_h1(i,j)==0)&&(Sing_2_l1(i,j)
==1)
                Y2(i,j)=30;
            end
            if (Coinc_2_b1(i,j)==0)&&(Coinc_2_c1(i,j)==0)&&(Coinc_2_d1(i,j)==0)&&
(Coinc_2_f1(i,j)==0)&&(Coinc_2_i1(i,j)==0)&&(Coinc_2_j1(i,j)==0)&&(Sing_2_m1(i,j)
==1)
                Y2(i,j)=32;
            end
            if (Coinc_2_b1(i,j)==0)&&(Coinc_2_d1(i,j)==0)&&(Coinc_2_e1(i,j)==0)&&
(Coinc_2_h1(i,j)==0)&&(Coinc_2_j1(i,j)==0)&&(Coinc_2_k1(i,j)==0)&&(Sing_2_n1(i,j)
==1)
                Y2(i,j)=34;
            end
            if (Coinc_2_c1(i,j)==0)&&(Coinc_2_d1(i,j)==0)&&(Coinc_2_e1(i,j)==0)&&
(Coinc_2_g1(i,j)==0)&&(Coinc_2_i1(i,j)==0)&&(Coinc_2_k1(i,j)==0)&&(Sing_2_o1(i,j)
==1)
                Y2(i,j)=36;
            end

            if (Coinc_2_b1(i,j)==0)&&(Coinc_2_c1(i,j)==0)&&(Coinc_2_f1(i,j)==1)
                Y2(i,j)=70;
            end
            if (Coinc_2_c1(i,j)==0)&&(Coinc_2_e1(i,j)==0)&&(Coinc_2_g1(i,j)==1)
                Y2(i,j)=72;
            end
            if (Coinc_2_b1(i,j)==0)&&(Coinc_2_e1(i,j)==0)&&(Coinc_2_h1(i,j)==1)
                Y2(i,j)=74;
            end
            if (Coinc_2_c1(i,j)==0)&&(Coinc_2_d1(i,j)==0)&&(Coinc_2_i1(i,j)==1)
                Y2(i,j)=76;
            end
            if (Coinc_2_b1(i,j)==0)&&(Coinc_2_d1(i,j)==0)&&(Coinc_2_j1(i,j)==1)
                Y2(i,j)=78;
            end
            if (Coinc_2_d1(i,j)==0)&&(Coinc_2_e1(i,j)==0)&&(Coinc_2_k1(i,j)==1)
                Y2(i,j)=80;
            end
        end
    end
end

```

```

        if Coinc_2_b1(i,j)==1
            Y2(i,j)=164;
        end
        if Coinc_2_c1(i,j)==1
            Y2(i,j)=166;
        end
        if Coinc_2_d1(i,j)==1
            Y2(i,j)=168;
        end
        if Coinc_2_e1(i,j)==1
            Y2(i,j)=170;
        end
    else
        Y2(i,j)=255;
    end
end
end

figure (5)
title('Cado \delta3=-1 => B1<B2')
subplot(1,2,1)
imagesc(delta_1(:),delta_2(:),Y1);colormap(summer)
caxis([0 255])
xlabel('\delta1')
ylabel('\delta2')
title('Rete 1: multiservizio')
subplot(1,2,2)
imagesc(delta_1(:),delta_2(:),Y2);colormap(summer)
caxis([0 255])
xlabel('\delta1')
ylabel('\delta2')
title('Rete 2: multiservizio')

%% Delta 3 = 0

Coinc_1_a2 = (C_1(:,:,6,1))&(C_1(:,:,6,2))&(C_1(:,:,6,3))&(C_1(:,:,6,4)); % Scelta
la rete 1 per tutti e quattro le classi
Coinc_2_a2 = (C_2(:,:,6,1))&(C_2(:,:,6,2))&(C_2(:,:,6,3))&(C_2(:,:,6,4)); % Scelta
la rete 2 per tutti e quattro le classi

Coinc_1_b2 = (C_1(:,:,6,1))&(C_1(:,:,6,2))&(C_1(:,:,6,3)); % Scelta la rete 1 per
le classi conversational streaming e interactive
Coinc_2_b2 = (C_2(:,:,6,1))&(C_2(:,:,6,2))&(C_2(:,:,6,3)); % Scelta la rete 2 per
le classi conversational streaming e interactive

Coinc_1_c2 = (C_1(:,:,6,1))&(C_1(:,:,6,2))&(C_1(:,:,6,4)); % Scelta la rete 1 per
le classi conversational streaming e background
Coinc_2_c2 = (C_2(:,:,6,1))&(C_2(:,:,6,2))&(C_2(:,:,6,4)); % Scelta la rete 2 per
le classi conversational streaming e background

Coinc_1_d2 = (C_1(:,:,6,2))&(C_1(:,:,6,3))&(C_1(:,:,6,4)); % Scelta la rete 1 per
le classi streaming interactive e background
Coinc_2_d2 = (C_2(:,:,6,2))&(C_2(:,:,6,3))&(C_2(:,:,6,4)); % Scelta la rete 2 per
le classi streaming interactive e background

Coinc_1_e2 = (C_1(:,:,6,1))&(C_1(:,:,6,3))&(C_1(:,:,6,4)); % Scelta la rete 1 per
le classi conversational interactive e background
Coinc_2_e2 = (C_2(:,:,6,1))&(C_2(:,:,6,3))&(C_2(:,:,6,4)); % Scelta la rete 2 per
le classi conversational interactive e background

```

```

Coinc_1_f2 = (C_1(:, :, 6, 1)) & (C_1(:, :, 6, 2)); % Scelta la rete 1 per le classi
conversational e streaming
Coinc_2_f2 = (C_2(:, :, 6, 1)) & (C_2(:, :, 6, 2)); % Scelta la rete 2 per le classi
conversational e streaming

Coinc_1_g2 = (C_1(:, :, 6, 1)) & (C_1(:, :, 6, 4)); % Scelta la rete 1 per le classi
conversational e background
Coinc_2_g2 = (C_2(:, :, 6, 1)) & (C_2(:, :, 6, 4)); % Scelta la rete 2 per le classi
conversational e background

Coinc_1_h2 = (C_1(:, :, 6, 1)) & (C_1(:, :, 6, 3)); % Scelta la rete 1 per le classi
conversational e interactive
Coinc_2_h2 = (C_2(:, :, 6, 1)) & (C_2(:, :, 6, 3)); % Scelta la rete 2 per le classi
conversational e interactive

Coinc_1_i2 = (C_1(:, :, 6, 2)) & (C_1(:, :, 6, 4)); % Scelta la rete 1 per le classi
streaming e background
Coinc_2_i2 = (C_2(:, :, 6, 2)) & (C_2(:, :, 6, 4)); % Scelta la rete 2 per le classi
streaming e background

Coinc_1_j2 = (C_1(:, :, 6, 2)) & (C_1(:, :, 6, 3)); % Scelta la rete 1 per le classi
streaming e interactive
Coinc_2_j2 = (C_2(:, :, 6, 2)) & (C_2(:, :, 6, 3)); % Scelta la rete 2 per le classi
streaming e interactive

Coinc_1_k2 = (C_1(:, :, 6, 3)) & (C_1(:, :, 6, 4)); % Scelta la rete 1 per le classi
interactive e background
Coinc_2_k2 = (C_2(:, :, 6, 3)) & (C_2(:, :, 6, 4)); % Scelta la rete 2 per le classi
interactive e background

Sing_1_l2= (C_1(:, :, 6, 1)); % Scelta la rete 1 per la classe conversational
Sing_2_l2= (C_2(:, :, 6, 1)); % Scelta la rete 2 per la classe conversational

Sing_1_m2= (C_1(:, :, 6, 2)); % Scelta la rete 1 per la classe streaming
Sing_2_m2= (C_2(:, :, 6, 2)); % Scelta la rete 2 per la classe streaming

Sing_1_n2= (C_1(:, :, 6, 3)); % Scelta la rete 1 per la classe interactive
Sing_2_n2= (C_2(:, :, 6, 3)); % Scelta la rete 2 per la classe interactive

Sing_1_o2= (C_1(:, :, 6, 4)); % Scelta la rete 1 per la classe background
Sing_2_o2= (C_2(:, :, 6, 4)); % Scelta la rete 2 per la classe background

% Rappresentazione grafica per delta_3 = 0 => Banda_1 = Banda_2 :

%assegnazione dei valori di colori quando si sceglie la rete 1 per: le
%diverse combinazioni di classi di servizio

for i=1:11
    for j=1:11
        if (Coinc_1_a2(i,j)==0)

            if (Coinc_1_b2(i,j)==0)&&(Coinc_1_c2(i,j)==0)&&(Coinc_1_e2(i,j)==0)&&
(Coinc_1_f2(i,j)==0)&&(Coinc_1_g2(i,j)==0)&&(Coinc_1_h2(i,j)==0)&&(Sing_1_l2(i,j)
==1)
                Y3(i,j)=30;
            end
            if (Coinc_1_b2(i,j)==0)&&(Coinc_1_c2(i,j)==0)&&(Coinc_1_d2(i,j)==0)&&
(Coinc_1_f2(i,j)==0)&&(Coinc_1_i2(i,j)==0)&&(Coinc_1_j2(i,j)==0)&&(Sing_1_m2(i,j)
==1)
                Y3(i,j)=32;
            end
        end
    end
end

```

```

        if (Coinc_1_b2(i,j)==0)&&(Coinc_1_d2(i,j)==0)&&(Coinc_1_e2(i,j)==0)&&
(Coinc_1_h2(i,j)==0)&&(Coinc_1_j2(i,j)==0)&&(Coinc_1_k2(i,j)==0)&&(Sing_1_n2(i,j)
==1)
            Y3(i,j)=34;
        end
        if (Coinc_1_c2(i,j)==0)&&(Coinc_1_d2(i,j)==0)&&(Coinc_1_e2(i,j)==0)&&
(Coinc_1_g2(i,j)==0)&&(Coinc_1_i2(i,j)==0)&&(Coinc_1_k2(i,j)==0)&&(Sing_1_o2(i,j)
==1)
            Y3(i,j)=36;
        end

        if (Coinc_1_b2(i,j)==0)&&(Coinc_1_c2(i,j)==0)&&(Coinc_1_f2(i,j)==1)
            Y3(i,j)=70;
        end
        if (Coinc_1_c2(i,j)==0)&&(Coinc_1_e2(i,j)==0)&&(Coinc_1_g2(i,j)==1)
            Y3(i,j)=72;
        end
        if (Coinc_1_b2(i,j)==0)&&(Coinc_1_e2(i,j)==0)&&(Coinc_1_h2(i,j)==1)
            Y3(i,j)=74;
        end
        if (Coinc_1_c2(i,j)==0)&&(Coinc_1_d2(i,j)==0)&&(Coinc_1_i2(i,j)==1)
            Y3(i,j)=76;
        end
        if (Coinc_1_b2(i,j)==0)&&(Coinc_1_d2(i,j)==0)&&(Coinc_1_j2(i,j)==1)
            Y3(i,j)=78;
        end
        if (Coinc_1_d2(i,j)==0)&&(Coinc_1_e2(i,j)==0)&&(Coinc_1_k2(i,j)==1)
            Y3(i,j)=80;
        end

        if Coinc_1_b2(i,j)==1
            Y3(i,j)=164;
        end
        if Coinc_1_c2(i,j)==1
            Y3(i,j)=166;
        end
        if Coinc_1_d2(i,j)==1
            Y3(i,j)=168;
        end
        if Coinc_1_e2(i,j)==1
            Y3(i,j)=170;
        end
    else
        Y3(i,j)=255;
    end
end
end
end

```

%assegnazione dei valori di colori quando si sceglie la rete 2 per: le
%diverse combinazioni di classi di servizio

```

for i=1:11
    for j=1:11

        if (Coinc_2_a2(i,j)==0)

            if (Coinc_2_b2(i,j)==0)&&(Coinc_2_c2(i,j)==0)&&(Coinc_2_e2(i,j)==0)&&
(Coinc_2_f2(i,j)==0)&&(Coinc_2_g2(i,j)==0)&&(Coinc_2_h2(i,j)==0)&&(Sing_2_l2(i,j)
==1)

```

```

        Y4(i,j)=30;
    end
    if (Coinc_2_b2(i,j)==0)&&(Coinc_2_c2(i,j)==0)&&(Coinc_2_d2(i,j)==0)&&
(Coinc_2_f2(i,j)==0)&&(Coinc_2_i2(i,j)==0)&&(Coinc_2_j2(i,j)==0)&&(Sing_2_m2(i,j)
==1)
        Y4(i,j)=32;
    end
    if (Coinc_2_b2(i,j)==0)&&(Coinc_2_d2(i,j)==0)&&(Coinc_2_e2(i,j)==0)&&
(Coinc_2_h2(i,j)==0)&&(Coinc_2_j2(i,j)==0)&&(Coinc_2_k2(i,j)==0)&&(Sing_2_n2(i,j)
==1)
        Y4(i,j)=34;
    end
    if (Coinc_2_c2(i,j)==0)&&(Coinc_2_d2(i,j)==0)&&(Coinc_2_e2(i,j)==0)&&
(Coinc_2_g2(i,j)==0)&&(Coinc_2_i2(i,j)==0)&&(Coinc_2_k2(i,j)==0)&&(Sing_2_o2(i,j)
==1)
        Y4(i,j)=36;
    end

    if (Coinc_2_b2(i,j)==0)&&(Coinc_2_c2(i,j)==0)&&(Coinc_2_f2(i,j)==1)
        Y4(i,j)=70;
    end
    if (Coinc_2_c2(i,j)==0)&&(Coinc_2_e2(i,j)==0)&&(Coinc_2_g2(i,j)==1)
        Y4(i,j)=72;
    end
    if (Coinc_2_b2(i,j)==0)&&(Coinc_2_e2(i,j)==0)&&(Coinc_2_h2(i,j)==1)
        Y4(i,j)=74;
    end
    if (Coinc_2_c2(i,j)==0)&&(Coinc_2_d2(i,j)==0)&&(Coinc_2_i2(i,j)==1)
        Y4(i,j)=76;
    end
    if (Coinc_2_b2(i,j)==0)&&(Coinc_2_d2(i,j)==0)&&(Coinc_2_j2(i,j)==1)
        Y4(i,j)=78;
    end
    if (Coinc_2_d2(i,j)==0)&&(Coinc_2_e2(i,j)==0)&&(Coinc_2_k2(i,j)==1)
        Y4(i,j)=80;
    end

    if Coinc_2_b2(i,j)==1
        Y4(i,j)=164;
    end
    if Coinc_2_c2(i,j)==1
        Y4(i,j)=166;
    end
    if Coinc_2_d2(i,j)==1
        Y4(i,j)=168;
    end
    if Coinc_2_e2(i,j)==1
        Y4(i,j)=170;
    end
else
    Y4(i,j)=255;
end
end
end

figure (6)
title('Cado \delta3=0 => B1=B2')
subplot(1,2,1)
imagesc(delta_1(:),delta_2(:),Y3);colormap(autumn)
caxis([0 255])
xlabel('\delta1')

```

```

ylabel('\delta2')
title('Rete 1: multiservizio')
subplot(1,2,2)
imagesc(delta_1(:),delta_2(:),Y4);colormap(autumn)
caxis([0 255])
xlabel('\delta1')
ylabel('\delta2')
title('Rete 2: multiservizio')

%% Delta 3 = 1

Coinc_1_a3 = (C_1(:,:,11,1))&(C_1(:,:,11,2))&(C_1(:,:,11,3))&(C_1(:,:,11,4)); %
Scelta la rete 1 per tutti e quattro le classi
Coinc_2_a3 = (C_2(:,:,11,1))&(C_2(:,:,11,2))&(C_2(:,:,11,3))&(C_2(:,:,11,4)); %
Scelta la rete 2 per tutti e quattro le classi

Coinc_1_b3 = (C_1(:,:,11,1))&(C_1(:,:,11,2))&(C_1(:,:,11,3)); % Scelta la rete 1
per le classi conversational streaming e interactive
Coinc_2_b3 = (C_2(:,:,11,1))&(C_2(:,:,11,2))&(C_2(:,:,11,3)); % Scelta la rete 2
per le classi conversational streaming e interactive

Coinc_1_c3 = (C_1(:,:,11,1))&(C_1(:,:,11,2))&(C_1(:,:,11,4)); % Scelta la rete 1
per le classi conversational streaming e background
Coinc_2_c3 = (C_2(:,:,11,1))&(C_2(:,:,11,2))&(C_2(:,:,11,4)); % Scelta la rete 2
per le classi conversational streaming e background

Coinc_1_d3 = (C_1(:,:,11,2))&(C_1(:,:,11,3))&(C_1(:,:,11,4)); % Scelta la rete 1
per le classi streaming interactive e background
Coinc_2_d3 = (C_2(:,:,11,2))&(C_2(:,:,11,3))&(C_2(:,:,11,4)); % Scelta la rete 2
per le classi streaming interactive e background

Coinc_1_e3 = (C_1(:,:,11,1))&(C_1(:,:,11,3))&(C_1(:,:,11,4)); % Scelta la rete 1
per le classi conversational interactive e background
Coinc_2_e3 = (C_2(:,:,11,1))&(C_2(:,:,11,3))&(C_2(:,:,11,4)); % Scelta la rete 2
per le classi conversational interactive e background

Coinc_1_f3 = (C_1(:,:,11,1))&(C_1(:,:,11,2)); % Scelta la rete 1 per le classi
conversational e streaming
Coinc_2_f3 = (C_2(:,:,11,1))&(C_2(:,:,11,2)); % Scelta la rete 2 per le classi
conversational e streaming

Coinc_1_g3 = (C_1(:,:,11,1))&(C_1(:,:,11,4)); % Scelta la rete 1 per le classi
conversational e background
Coinc_2_g3 = (C_2(:,:,11,1))&(C_2(:,:,11,4)); % Scelta la rete 2 per le classi
conversational e background

Coinc_1_h3 = (C_1(:,:,11,1))&(C_1(:,:,11,3)); % Scelta la rete 1 per le classi
conversational e interactive
Coinc_2_h3 = (C_2(:,:,11,1))&(C_2(:,:,11,3)); % Scelta la rete 2 per le classi
conversational e interactive

Coinc_1_i3 = (C_1(:,:,11,2))&(C_1(:,:,11,4)); % Scelta la rete 1 per le classi
streaming e background
Coinc_2_i3 = (C_2(:,:,11,2))&(C_2(:,:,11,4)); % Scelta la rete 2 per le classi
streaming e background

Coinc_1_j3 = (C_1(:,:,11,2))&(C_1(:,:,11,3)); % Scelta la rete 1 per le classi
streaming e interactive
Coinc_2_j3 = (C_2(:,:,11,2))&(C_2(:,:,11,3)); % Scelta la rete 2 per le classi
streaming e interactive

```



```

Coinc_1_k3 = (C_1(:, :, 11, 3)) & (C_1(:, :, 11, 4)); % Scelta la rete 1 per le classi
interactive e background
Coinc_2_k3 = (C_2(:, :, 11, 3)) & (C_2(:, :, 11, 4)); % Scelta la rete 2 per le classi
interactive e background

Sing_1_l3 = (C_1(:, :, 11, 1)); % Scelta la rete 1 per la classe conversational
Sing_2_l3 = (C_2(:, :, 11, 1)); % Scelta la rete 2 per la classe conversational

Sing_1_m3 = (C_1(:, :, 11, 2)); % Scelta la rete 1 per la classe streaming
Sing_2_m3 = (C_2(:, :, 11, 2)); % Scelta la rete 2 per la classe streaming

Sing_1_n3 = (C_1(:, :, 11, 3)); % Scelta la rete 1 per la classe interactive
Sing_2_n3 = (C_2(:, :, 11, 3)); % Scelta la rete 2 per la classe interactive

Sing_1_o3 = (C_1(:, :, 11, 4)); % Scelta la rete 1 per la classe background
Sing_2_o3 = (C_2(:, :, 11, 4)); % Scelta la rete 2 per la classe background

% Rappresentazione grafica per delta_3 = 1 => Banda_1 > Banda_2 :

% assegnazione dei valori di colori quando si sceglie la rete 1 per: le
% diverse combinazioni di classi di servizio

for i=1:11
    for j=1:11

        if (Coinc_1_a3(i,j)==0)

            if (Coinc_1_b3(i,j)==0) && (Coinc_1_c3(i,j)==0) && (Coinc_1_e3(i,j)==0) &&
(Coinc_1_f3(i,j)==0) && (Coinc_1_g3(i,j)==0) && (Coinc_1_h3(i,j)==0) && (Sing_1_l3(i,j)
==1)
                Y5(i,j)=30;
            end
            if (Coinc_1_b3(i,j)==0) && (Coinc_1_c3(i,j)==0) && (Coinc_1_d3(i,j)==0) &&
(Coinc_1_f3(i,j)==0) && (Coinc_1_i3(i,j)==0) && (Coinc_1_j3(i,j)==0) && (Sing_1_m3(i,j)
==1)
                Y5(i,j)=32;
            end
            if (Coinc_1_b3(i,j)==0) && (Coinc_1_d3(i,j)==0) && (Coinc_1_e3(i,j)==0) &&
(Coinc_1_h3(i,j)==0) && (Coinc_1_j3(i,j)==0) && (Coinc_1_k3(i,j)==0) && (Sing_1_n3(i,j)
==1)
                Y5(i,j)=34;
            end
            if (Coinc_1_c3(i,j)==0) && (Coinc_1_d3(i,j)==0) && (Coinc_1_e3(i,j)==0) &&
(Coinc_1_g3(i,j)==0) && (Coinc_1_i3(i,j)==0) && (Coinc_1_k3(i,j)==0) && (Sing_1_o3(i,j)
==1)
                Y5(i,j)=36;
            end

            if (Coinc_1_b3(i,j)==0) && (Coinc_1_c3(i,j)==0) && (Coinc_1_f3(i,j)==1)
                Y5(i,j)=70;
            end
            if (Coinc_1_c3(i,j)==0) && (Coinc_1_e3(i,j)==0) && (Coinc_1_g3(i,j)==1)
                Y5(i,j)=72;
            end
            if (Coinc_1_b3(i,j)==0) && (Coinc_1_e3(i,j)==0) && (Coinc_1_h3(i,j)==1)
                Y5(i,j)=74;
            end
            if (Coinc_1_c3(i,j)==0) && (Coinc_1_d3(i,j)==0) && (Coinc_1_i3(i,j)==1)
                Y5(i,j)=76;
            end
        end
    end
end

```

```

    if (Coinc_1_b3(i,j)==0)&&(Coinc_1_d3(i,j)==0)&&(Coinc_1_j3(i,j)==1)
        Y5(i,j)=78;
    end
    if (Coinc_1_d3(i,j)==0)&&(Coinc_1_e3(i,j)==0)&&(Coinc_1_k3(i,j)==1)
        Y5(i,j)=80;
    end

    if Coinc_1_b3(i,j)==1
        Y5(i,j)=164;
    end
    if Coinc_1_c3(i,j)==1
        Y5(i,j)=166;
    end
    if Coinc_1_d3(i,j)==1
        Y5(i,j)=168;
    end
    if Coinc_1_e3(i,j)==1
        Y5(i,j)=170;
    end
else
    Y5(i,j)=255;
end
end
end
end

```

%assegnazione dei valori di colori quando si sceglie la rete 2 per: le
%diverse combinazioni di classi di servizio

```

for i=1:11
    for j=1:11

        if (Coinc_2_a3(i,j)==0)

            if (Coinc_2_b3(i,j)==0)&&(Coinc_2_c3(i,j)==0)&&(Coinc_2_e3(i,j)==0)&&
(Coinc_2_f3(i,j)==0)&&(Coinc_2_g3(i,j)==0)&&(Coinc_2_h3(i,j)==0)&&(Sing_2_l3(i,j)
==1)
                Y6(i,j)=30;
            end
            if (Coinc_2_b3(i,j)==0)&&(Coinc_2_c3(i,j)==0)&&(Coinc_2_d3(i,j)==0)&&
(Coinc_2_f3(i,j)==0)&&(Coinc_2_i3(i,j)==0)&&(Coinc_2_j3(i,j)==0)&&(Sing_2_m3(i,j)
==1)
                Y6(i,j)=32;
            end
            if (Coinc_2_b3(i,j)==0)&&(Coinc_2_d3(i,j)==0)&&(Coinc_2_e3(i,j)==0)&&
(Coinc_2_h3(i,j)==0)&&(Coinc_2_j3(i,j)==0)&&(Coinc_2_k3(i,j)==0)&&(Sing_2_n3(i,j)
==1)
                Y6(i,j)=34;
            end
            if (Coinc_2_c3(i,j)==0)&&(Coinc_2_d3(i,j)==0)&&(Coinc_2_e3(i,j)==0)&&
(Coinc_2_g3(i,j)==0)&&(Coinc_2_i3(i,j)==0)&&(Coinc_2_k3(i,j)==0)&&(Sing_2_o3(i,j)
==1)
                Y6(i,j)=36;
            end

            if (Coinc_2_b3(i,j)==0)&&(Coinc_2_c3(i,j)==0)&&(Coinc_2_f3(i,j)==1)
                Y6(i,j)=70;
            end
            if (Coinc_2_c3(i,j)==0)&&(Coinc_2_e3(i,j)==0)&&(Coinc_2_g3(i,j)==1)
                Y6(i,j)=72;
            end
        end
    end
end

```

```

        if (Coinc_2_b3(i,j)==0)&&(Coinc_2_e3(i,j)==0)&&(Coinc_2_h3(i,j)==1)
            Y6(i,j)=74;
        end
        if (Coinc_2_c3(i,j)==0)&&(Coinc_2_d3(i,j)==0)&&(Coinc_2_i3(i,j)==1)
            Y6(i,j)=76;
        end
        if (Coinc_2_b3(i,j)==0)&&(Coinc_2_d3(i,j)==0)&&(Coinc_2_j3(i,j)==1)
            Y6(i,j)=78;
        end
        if (Coinc_2_d3(i,j)==0)&&(Coinc_2_e3(i,j)==0)&&(Coinc_2_k3(i,j)==1)
            Y6(i,j)=80;
        end

        if Coinc_2_b3(i,j)==1
            Y6(i,j)=164;
        end
        if Coinc_2_c3(i,j)==1
            Y6(i,j)=166;
        end
        if Coinc_2_d3(i,j)==1
            Y6(i,j)=168;
        end
        if Coinc_2_e3(i,j)==1
            Y6(i,j)=170;
        end
    else
        Y6(i,j)=255;
    end
end
end
end

```

```

figure (7)
title('Cado \delta3=1 => B1>B2')
subplot(1,2,1)
imagesc(delta_1(:),delta_2(:),Y5);colormap(spring)
caxis([0 255])
xlabel('\delta1')
ylabel('\delta2')
title('Rete 1: multiservizio')
subplot(1,2,2)
imagesc(delta_1(:),delta_2(:),Y6);colormap(spring)
caxis([0 255])
xlabel('\delta1')
ylabel('\delta2')
title('Rete 2: multiservizio')

```

```

% Generazione della distribuzione del Packet Loss tramite il modello di
% Gilbert-Elliot. Stato 0= ricevere correttamente i bit. Stato 1= ricevere
% i bit con errori. h= probabilità di trasmissione corretta essendo nello
% stato 1, valori tipici tra 0 e 0.5

```

```

function [PacketLoss] = genPacketLoss

```

```

P12_max = 1e-3;
P12_min = 1e-5;
P21_min = 0.05;
P21_max = 0.3;

```

```

P1 = P12_max/(P21_min+P12_max);
P2 = P12_min/(P21_max+P12_min);

alt = P1-P2;
PacketLoss = P2+alt*rand;

% Generazione dei parametri di ciascuna rete. Nella matrice Param ho per
% ciascuna riga i valori di ritardo (D), pacchetti persi (L) e banda
% permessa (ABW) per ogni rete considerata.

N=6; % Numero di reti scelto

for i=1:N

    D(i) = delay;
    L(i) = genPacketLoss;
    ABW(i) = rand;

end

del=sum(D);
D = D/del; % Normalizzazione del ritardo

Param = [D;L;ABW]; % Matrice con i valori dei parametri per ciascuna rete candidata

% Verifica grafica della scelta dei parametri

% Parametri che devo avere in ingresso: D,L,ABW

clear;
Parametri;
T=4;
r = r_mod (Param);
[N,M] = size (r);
w= [0.8181 0.0909 0.0909; 0.2067 0.0581 0.7352; 0.1429 0.7143 0.1429; 0.0581 0.7352
0.2067];
% Pesi dei parametri per ciascuna classe di servizio

% Matrice di probabilità di transizioni secondo MC1, senza e considerando
% le classi di servizio

MC1 = calcMC1(r);
MC1_TC = calcMC1_TC(r,w);

% Matrice di probabilità di transizioni secondo MC2, senza e considerando
% le classi di servizio

MC2 = calcMC2(r);
MC2_TC = calcMC2_TC(r,w);

% Calcolo il vettore di Distribuzione Stazionaria

SD_1 = calc_SD(MC1);
SD_2 = calc_SD(MC2);

SD_1_TC = calc_SD_TC(MC1_TC);
SD_2_TC = calc_SD_TC(MC2_TC);

```

```

% Selezione della rete con maggior valore di sd.

[Fav_1, F_1] = max (SD_1);
[Fav_2, F_2] = max (SD_2);
%[CC_fav, fav_TOPSIS] = TOPSIS(Param');

% Selezione della rete con maggior valore di sd per ciascuna classe di
% servizio.

[Fav_1_TC, F_1_TC] = max (SD_1_TC);
[Fav_2_TC, F_2_TC] = max (SD_2_TC);
%[CC_fav_TC, fav_TOPSIS_TC] = TOPSIS_TC(Param',w);

% Calcolo dei valori dei parametri "ottimi" e degli altri parametri per
% quella stessa rete

% Delay minimo
[D_min,I_D_min] = min(Param(1,:)); % Rete col minore valore di ritardo
L_retDmin = Param(2,I_D_min); % Packet Loss per quella rete
B_retDmin = Param(3,I_D_min); % Banda per quella rete

% Packet Loss minimo
[L_min,I_L_min] = min(Param(2,:)); % Rete col minor valore di Packet Loss
D_retLmin = Param(1,I_L_min); % Ritardo per quella rete
B_retLmin = Param(3,I_L_min); % Larghezza per Banda per quella rete

% Banda massima
[B_max,I_B_max] = max(Param(3,:)); % Rete con maggior valor di Banda
D_retBmax = Param(1,I_B_max); % Ritardo per quella rete
L_retBmax = Param(2,I_B_max); % Packet Loss per quella rete

% Valori dei parametri della rete scelta secondo MC1

D_MC1 = Param(1,F_1);
L_MC1 = Param(2,F_1);
B_MC1 = Param(3,F_1);

% Valori dei parametri delle rete scelte secondo il metodo MC1 per ciascuna
% classe di servizio, e anche di quella scelta senza differenziare classe
% di servizio

% Conversational
D_MC1_conv = Param(1,F_1_TC(1));
L_MC1_conv = Param(2,F_1_TC(1));
B_MC1_conv = Param(3,F_1_TC(1));

% Streaming
D_MC1_stre = Param(1,F_1_TC(2));
L_MC1_stre = Param(2,F_1_TC(2));
B_MC1_stre = Param(3,F_1_TC(2));

%Interactive
D_MC1_inter = Param(1,F_1_TC(3));
L_MC1_inter = Param(2,F_1_TC(3));
B_MC1_inter = Param(3,F_1_TC(3));

% Background

```

```

D_MC1_back = Param(1,F_1_TC(4));
L_MC1_back = Param(2,F_1_TC(4));
B_MC1_back = Param(3,F_1_TC(4));

```

```

% Rappresentazione grafica: Confronto tra i parametri scelti dal metodo,
% quelli medi e quelli "ottimi"

```

```

D_MIN = [D_min D_MC1 D_MC1_conv D_MC1_stre D_MC1_inter D_MC1_back];
L_Dmin = [L_retDmin L_MC1 L_MC1_conv L_MC1_stre L_MC1_inter L_MC1_back];
B_Dmin = [B_retDmin B_MC1 B_MC1_conv B_MC1_stre B_MC1_inter B_MC1_back];

```

```

D_Lmin = [D_retLmin D_MC1 D_MC1_conv D_MC1_stre D_MC1_inter D_MC1_back];
L_MIN = [L_min L_MC1 L_MC1_conv L_MC1_stre L_MC1_inter L_MC1_back];
B_Lmin = [B_retLmin B_MC1 B_MC1_conv B_MC1_stre B_MC1_inter B_MC1_back];

```

```

D_Bmax = [D_retBmax D_MC1 D_MC1_conv D_MC1_stre D_MC1_inter D_MC1_back];
L_Bmax = [L_retBmax L_MC1 L_MC1_conv L_MC1_stre L_MC1_inter L_MC1_back];
B_MAX = [B_max B_MC1 B_MC1_conv B_MC1_stre B_MC1_inter B_MC1_back];

```

```

figure (5)
subplot (3,1,1)
bar (D_MIN)
title 'Ritardi: minimo, scelto con MC1 uniforme e pesati alle classi di servizio '
colormap winter
subplot (3,1,2)
bar (L_Dmin) %,'XTickLabel',{'L (R min)','L MC1','L conv','L stre','L inter','L
back'})
title 'Packet Loss della rete con ritardo minimo, MC1 uniforme e pesati alle classi
di servizio'
colormap autumn
subplot (3,1,3)
bar (B_Dmin) %,'XTickLabel',{'B (R min)','B MC1','B conv','B stre','B inter','B
back'})
title 'Banda della rete con ritardo minimo, MC1 uniforme e pesati alle classi di
servizio'
colormap summer

```

```

figure (6)
subplot (3,1,1)
bar (D_Lmin) %,'XTickLabel',{'R (L min)','R MC1','R conv','R stre','R inter','R
back'})
title 'Ritardi della rete con packet loss minimo, MC1 uniforme e pesati alle classi
di servizio'
colormap winter
subplot (3,1,2)
bar (L_MIN) %,'XTickLabel',{'L min','L MC1','L conv','L stre','L inter','L back'})
title 'Packet Loss: minimo, scelto con MC1 uniforme e pesati alle classi di
servizio'
colormap autumn
subplot (3,1,3)
bar (B_Lmin) %,'XTickLabel',{'B (L min)','B MC1','B conv','B stre','B inter','B
back'})
title 'Banda della rete con packet loss minimo, MC1 uniforme e pesati alle classi
di servizio'
colormap summer

```

```

figure (7)
subplot (3,1,1)

```

```

bar (D_Bmax) %'XTickLabel',{ 'R (B max)', 'R MC1', 'R conv', 'R stre', 'R inter', 'R
back'})
title 'Ritardi della rete con banda massima, MC1 uniforme e pesati alle classi di
servizio'
colormap winter
subplot (3,1,2)
bar (L_Bmax) %'XTickLabel',{ 'L (B max)', 'L MC1', 'L conv', 'L stre', 'L inter', 'L
back'})
title 'Packet Loss della rete con banda massima, MC1 uniforme e pesati alle classi
di servizio'
colormap autumn
subplot (3,1,3)
bar (B_MAX) %'XTickLabel',{ 'B max', 'B MC1', 'B conv', 'B stre', 'B inter', 'B back'})
title 'Banda: massima, scelta con MC1 uniforme e pesati alle classi di servizio'
colormap summer

```

```

% Rappresentazione grafica dei miglioramenti al ranking, variando il
% parametro delta che ci da lo spostamento delle reti dai
% valori discreti del ranking.

```

```

delta_1= linspace (-0.999,0.999,100); % spostamenti dalla seconda posizione nel
ranking della prima rete secondo il secondo parametro
delta_2= linspace (-0.999,0.999,100); % spostamenti dalla seconda posizione nel
ranking della seconda rete secondo il primo parametro

```

```

N=3; % Esempio con tre reti
M=2; % Esempio con due parametri

```

```

for i = 1:length(delta_1)
    for j = 1:length(delta_2)

        r = [1 (2+delta_1(i));(2+delta_2(j)) 1;3 3];
        MC1 = calcMC1(r);
        MC2 = calcMC2(r);
        SD_1 = calc_SD(MC1); % Vettore di probabilità degli stati stazionari
secondo MC1
        SD_2 = calc_SD(MC2); % Vettore di probabilità degli stati stazionari
secondo MC2
        P_1_MC1(i,j) = SD_1(1); % tutte le probabilità di scegliere la rete 1
secondo MC1
        P_1_MC2(i,j) = SD_2(1); % tutte le probabilità di scegliere la rete 1
secondo MC2
        P_2_MC1(i,j) = SD_1(2); % tutte le probabilità di scegliere la rete 2
secondo MC1
        P_2_MC2(i,j) = SD_2(2); % tutte le probabilità di scegliere la rete 2
secondo MC2
    end
end

```

```

figure (1);
mesh(delta_1,delta_2,P_1_MC1(1:100,1:100));figure(gcf);
xlabel('\Delta 1')
ylabel('\Delta 2')
text (100,51,P_1_MC1(100,1), 'P1>P2')
title('Probabilità di scelta della rete 1 e della rete 2 secondo MC1')
hold on
mesh(delta_1,delta_2,P_2_MC1(1:100,1:100));figure(gcf);
text (51,100,P_2_MC1(1,100), 'P2>P1')
grid on

```

```

figure (2);
mesh(delta_1,delta_2,P_1_MC2(1:100,1:100));figure(gcf);
xlabel('\Delta 1')
ylabel('\Delta 2')
text (100,51,P_1_MC2(100,1),'P1>P2')
title('Probabilità di scelta della rete 1 e della rete 2 secondo MC2')
hold on
mesh(delta_1,delta_2,P_2_MC2(1:100,1:100));figure(gcf);
text (51,100,P_2_MC2(1,100),'P2>P1')
grid on

% Rappresentazione grafica delle probabilità di selezione di una rete per
% il caso tre reti, due parametri: Banda e ritardo, secondo la variazione
% nella sensibilità dei parametri

delta_1_prima= linspace (-0.499,0.499,100); % variazioni della banda per la seconda
rete

delta_2_prima= linspace (-0.499,0.499,100); % variazioni del ritardo per la prima
rete

N=3; % Esempio con tre reti
M=2; % Esempio con due parametri

SD = (1/N)*ones(1,N);

for i = 1:length(delta_1_prima)
    for j = 1:length(delta_2_prima)

        r = [1 2+2*delta_1_prima(i);(2-2*delta_2_prima(j)) 1;3 3];
        MC1 = calcMC1(r);
        MC2 = calcMC2(r);
        SD_1 = calc_SD(MC1); % Vettore di probabilità degli stati stazionari
secondo MC1
        SD_2 = calc_SD(MC2); % Vettore di probabilità degli stati stazionari
secondo MC2
        P_1_MC1_prima(i,j) = SD_1(1); % tutte le probabilità di scegliere la rete 1
secondo MC1
        P_1_MC2_prima(i,j) = SD_2(1); % tutte le probabilità di scegliere la rete 1
secondo MC2
        P_2_MC1_prima(i,j) = SD_1(2); % tutte le probabilità di scegliere la rete 2
secondo MC1
        P_2_MC2_prima(i,j) = SD_2(2); % tutte le probabilità di scegliere la rete 2
secondo MC2
    end
end

figure (3);
mesh(delta_1_prima,delta_2_prima,P_1_MC1_prima(1:100,1:100));figure(gcf);
xlabel('\Delta 1''')
ylabel('\Delta 2''')
title('Probabilità di scelta della rete 1 e della rete 2 secondo MC1')
hold on
mesh(delta_1_prima,delta_2_prima,P_2_MC1_prima(1:100,1:100));figure(gcf);
grid on

figure (4);
mesh(delta_1_prima,delta_2_prima,P_1_MC2_prima(1:100,1:100));figure(gcf);
xlabel('\Delta 1''')

```



```

ylabel('\Delta 2''')
title('Probabilità di scelta della rete 1 e della rete 2 secondo MC2')
hold on
mesh(delta_1_prima,delta_2_prima,P_2_MC2_prima(1:100,1:100));figure(gcf);
grid on

% Crea la matrice dei ranking modificata che prende in ingresso i
% vettori con i valori dei parametri: Ritardo (R), Pacchetti persi (L) e Larghezza
di
% Banda (ABW), per ciascuna rete.

function [r] = r_mod (Param)

[M,N] = size (Param); % Numero di reti candidate
r = zeros(N,M);
Param1 = transpose (Param);
flag=0;

for j=1:M

    [Max,ind_M] = max(Param1(:,j));
    [Min,ind_m] = min(Param1(:,j));
    Diff = (Max-Min)/(N-1);

    for s=2:N
        if (Param1(s,j)~=Param1(s-1,j))
            flag=1;
        end
    end
    if flag==0
        for m=1:N
            r(m,j)=1;
        end
    end
    end

    for i=1:N
        if (j==3)&&(flag==1)
            r(ind_m,j) = N;
            r(ind_M,j) = 1;
            if ((i ~= ind_m) && (i ~= ind_M))
                r(i,j) = ((Max - Param1(i,j))/Diff) + 1;
            end
        end
        elseif (j~=3)&&(flag==1)
            r(ind_m,j) = 1;
            r(ind_M,j) = N;
            if ((i ~= ind_m) && (i ~= ind_M))
                r(i,j) = ((Param1(i,j) - Min)/Diff) + 1;
            end
        end
    end
end

end

function r_ini=r_mod_ini(Param)

Param1=Param';
[rit,ret1]=sort(Param1(:,1));
[pac,ret2]=sort(Param1(:,2));
[band,ret3]=sort(Param1(:,3),'descend');

```

```

r_ini =[ret1 ret2 ret3];

% Questa funzione prende in ingresso i valori attesi dei minimi e dei
% massimi per ogni parametro, e i valori attesi della rete sotto studio
% per ogni parametro per poi calcolarne la distanza di questa rete dal
% punto migliore (massimo) e da quello peggiore (minimo) e finalmente
% tirare fuori un numero che rappresenta un detto coefficiente di vicinanza
% tramite il quale si vanno poi a fare il ranking delle rete disponibili

function CC = Ranking_TOPSIS(R,Mmax,Mmin)

Distmax = sqrt(sum((R - Mmax).^2));
Distmin = sqrt(sum((R - Mmin).^2));

CC = Distmin/(Distmax + Distmin);

% Parametri che devo avere in ingresso: D,L,ABW,C

clear;
Parametri;
r = r_mod (Param);
[N,M] = size (r);
SD = (1/N)*ones(1,N);

% Selezione la rete favorita secondo primo metodo
MC1 = calcMC1(r);

% Selezione la rete favorita secondo il secondo metodo
MC2 = calcMC2(r);

% Calcolo il vettore degli Stati Stazionari
SD_1 = calc_SD(MC1);
SD_2 = calc_SD(MC2);

% Seleziono la rete con maggiore valore di sd

[Fav_1, F_1] = max (SD_1);
[Fav_2, F_2] = max (SD_2);
[CC,fav_TOPSIS] = TOPSIS(Param');

fprintf ('la rete migliore secondo il primo metodo MC1 è : %i \n', F_1)
fprintf ('la rete migliore secondo il secondo metodo MC2 è : %i \n', F_2)
fprintf ('la rete migliore secondo TOPSIS è : %i \n', fav_TOPSIS)

clear;
Parametri;
r_ini=r_mod_ini(Param);
[N,M] = size (r);
SD = (1/N)*ones(1,N);

% Selezione la rete favorita secondo primo metodo
MC1 = calcMC1(r);

% Selezione la rete favorita secondo il secondo metodo
MC2 = calcMC2_ini(r);

% Calcolo il vettore degli Stati Stazionari
SD_1 = calc_SD(MC1);
SD_2 = calc_SD(MC2);

```

```

% Seleziono la rete con maggiore valore di sd

[Fav_1, F_1] = max (SD_1);
[Fav_2, F_2] = max (SD_2);
[CC,fav_TOPSIS] = TOPSIS(Param');

fprintf ('la rete migliore secondo il primo metodo MC1 è : %i \n', F_1)
fprintf ('la rete migliore secondo il secondo metodo MC2 è : %i \n', F_2)
fprintf ('la rete migliore secondo TOPSIS è : %i \n', fav_TOPSIS)

% parametri che devo avere in ingresso: D,L,ABW,C

clear;
Parametri;
T=4;
r = r_mod (Param);
[N,M] = size (r);
w= [0.8181 0.0909 0.0909; 0.2067 0.0581 0.7352; 0.1429 0.7143 0.1429; 0.0581 0.7352
0.2067]; % Pesi dei parametri

% Selezione la rete favorita secondo primo metodo

MC1_TC = calcMC1_TC(r,w);

% Selezione la rete favorita secondo il secondo metodo

MC2_TC = calcMC2_TC(r,w);

% Calcolo il vettore di Distribuzione Stazionaria

SD_1_TC = calc_SD_TC(MC1_TC);
SD_2_TC = calc_SD_TC(MC2_TC);

% Seleziono la rete con maggiore valore di sd per ciascuna classe di
% servizio

[Fav_1, F_1] = max (SD_1_TC);
[Fav_2, F_2] = max (SD_2_TC);
[CC_TC,fav_TOPSIS_TC] = TOPSIS_TC(Param',w);

fprintf ('la rete migliore per il traffico Conversational, secondo il primo metodo
MC1 è : %i \n', F_1(1))
fprintf ('la rete migliore per il traffico Conversational, secondo il secondo
metodo MC2 è : %i \n', F_2(1))
fprintf ('la rete migliore per il traffico Conversational, secondo TOPSIS_TC è : %i
\n', fav_TOPSIS_TC(1))

fprintf ('la rete migliore per il traffico Streaming, secondo il primo metodo MC1
è : %i \n', F_1(2))
fprintf ('la rete migliore per il traffico Streaming, secondo il secondo metodo MC2
è : %i \n', F_2(2))
fprintf ('la rete migliore per il traffico Streaming, secondo TOPSIS_TC è : %i \n',
fav_TOPSIS_TC(2))

fprintf ('la rete migliore per il traffico Interactive, secondo il primo metodo MC1
è : %i \n', F_1(3))

```

```

fprintf ('la rete migliore per il traffico Interactive, secondo il secondo metodo
MC2 è : %i \n', F_2(3))
fprintf ('la rete migliore per il traffico Interactive, secondo TOPSIS_TC è : %i
\n', fav_TOPSIS_TC(3))

fprintf ('la rete migliore per il traffico Background, secondo il primo metodo MC1
è : %i \n', F_1(4))
fprintf ('la rete migliore per il traffico Background, secondo il secondo metodo
MC2 è : %i \n', F_2(4))
fprintf ('la rete migliore per il traffico Background, secondo TOPSIS_TC è : %i
\n', fav_TOPSIS_TC(4))

% Studio della sensibilità dei diversi pesi per ogni classe di servizio,
% considerando solo due reti

% clear;
Param = [0.2833 0.2978; 0.0020 0.0017; 0.8810 0.9075];
T = 4;
r = r_mod (Param);
[N,M] = size (r);

w11 = linspace(0,1-0.0909,100); % Conv. variando ritardo lasciando fissi Packet
Loss e poi Banda

w23_a = linspace(0,1-0.0581,100); % Stream. variando Banda fissando Packet Loss
w21_b = linspace(0,1-0.0581,100); % Stream. variando Ritardo fissando Packet Loss

w32_a = linspace(0,1-0.1429,100); % Interac. variando packet loss fissando il banda
w31_b = linspace(0,1-0.1429,100); % Interac. variando ritardo fissando banda

w42_a = linspace(0,1-0.2067,100); % Back. variando Packet Loss fissando Banda
w41_b = linspace(0,1-0.2067,100); % Back. variando Packet Loss fissando Ritardo

%% Caso A

fprintf('Caso A: Servizio conversazionale con banda fissa, streaming con ritardo
fisso e background con banda fissa \n')

for i=1:length(w11)

    w= [w11(i) 0.0909 (1-w11(i)-0.0909); (1-w23_a(i)-0.0581) 0.0581 w23_a(i);
(1-0.1429-w32_a(i)) w32_a(i) 0.1429; 1-w42_a(i)-0.2067 w42_a(i) 0.2067];

    % Calcolo della Matrice di probabilità di transizioni: 1mo metodo
    MC1_TC = calcMC1_TC(r,w);

    % Calcolo della Matrice di probabilità di transizioni: 2do metodo
    %MC2_TC = calcMC2_TC(r,w);

    % Calcolo il vettore di Distribuzione Stazionaria
    SD_1_TC = calc_SD_TC(MC1_TC);
    %SD_2_TC = calc_SD_TC(MC2_TC);

    % Ordinamento discendente delle reti per ciascuna classe di servizio
    [SD_1_TC_ord,ord_1] = sort(SD_1_TC,'descend');

    if (i==1)

```

```

        ord_1_ant_conv = ord_1(:,1);
        fprintf('\nPrimo ordinamento servizio conversazionale: %i %i \n con w1= %g,
w2= %g, w3= %g \n',ord_1_ant_conv,w(1,1),w(1,2),w(1,3))
        ord_1_ant_stre = ord_1(:,2);
        fprintf('Primo ordinamento servizio streaming: %i %i \n con w1= %g, w2= %g,
w3= %g \n',ord_1_ant_stre,w(2,1),w(2,2),w(2,3))
        ord_1_ant_int = ord_1(:,3);
        fprintf('Primo ordinamento servizio interactive: %i %i \n con w1= %g, w2=
%g, w3= %g \n',ord_1_ant_int,w(3,1),w(3,2),w(3,3))
        ord_1_ant_back = ord_1(:,4);
        fprintf('Primo ordinamento servizio background: %i %i \n con w1= %g, w2=
%g, w3= %g \n\n',ord_1_ant_back,w(4,1),w(4,2),w(4,3))
    else
        ord_1_act_conv = ord_1(:,1);
        ord_1_act_stre = ord_1(:,2);
        ord_1_act_int = ord_1(:,3);
        ord_1_act_back = ord_1(:,4);
        if (ord_1_act_conv ~= ord_1_ant_conv)
            fprintf ('cambio ordine servizio conversazionale: %i %i \n con w1= %g,
w2= %g, w3= %g \n',ord_1_act_conv,w(1,1),w(1,2),w(1,3))
        end
        if (ord_1_act_stre ~= ord_1_ant_stre)
            fprintf ('cambio ordine servizio streaming: %i %i \n con w1= %g, w2= %g,
w3= %g \n',ord_1_act_stre,w(2,1),w(2,2),w(2,3))
        end
        if (ord_1_act_back ~= ord_1_ant_back)
            fprintf ('cambio ordine servizio background: %i %i \n con w1= %g, w2=
%g, w3= %g \n\n', ord_1_act_back,w(4,1),w(4,2),w(4,3))
        end
        if (ord_1_act_int ~= ord_1_ant_int)
            fprintf ('cambio ordine servizio interactive: %i %i \n con w1= %g, w2=
%g, w3= %g \n', ord_1_act_int,w(3,1),w(3,2),w(3,3))
        end
        ord_1_ant_conv = ord_1_act_conv;
        ord_1_ant_stre = ord_1_act_stre;
        ord_1_ant_int = ord_1_act_int;
        ord_1_ant_back = ord_1_act_back;
    end
end
end

%% Caso B

fprintf('\n Caso B: Servizio conversazionale con packet loss fisso, streaming con
packet loss fisso e background con ritardo fisso')

for i=1:length(w11)

    w= [w11(i) 1-w11(i)-0.0909 0.0909;w21_b(i) 0.0581 1-w21_b(i)-0.0581;w31_b(i) 1-
w31_b(i)-0.1429 0.1429;w41_b(i) 1-w41_b(i)-0.2067 0.2067];

    % Calcolo della Matrice di probabilità di transizioni: 1mo metodo
    MC1_TC = calcMC1_TC(r,w);

    % Calcolo della Matrice di probabilità di transizioni: 2do metodo
    %MC2_TC = calcMC2_TC(r,w);

    % Calcolo il vettore di Distribuzione Stazionaria
    SD_1_TC = calc_SD_TC(MC1_TC);
    %SD_2_TC = calc_SD_TC(MC2_TC);

    % Ordinamento discendente delle reti per ciascuna classe di servizio

```

```

[SD_1_TC_ord,ord_1] = sort(SD_1_TC,'descend');

if (i==1)
    ord_1_ant_conv = ord_1(:,1);
    fprintf('\nPrimo ordinamento servizio conversazionale: %i %i \n con w1= %g,
w2= %g, w3= %g \n',ord_1_ant_conv,w(1,1),w(1,2),w(1,3))
    ord_1_ant_stre = ord_1(:,2);
    fprintf('Primo ordinamento servizio streaming: %i %i \n con w1= %g, w2= %g,
w3= %g \n',ord_1_ant_stre,w(2,1),w(2,2),w(2,3))
    ord_1_ant_int = ord_1(:,3);
    fprintf('Primo ordinamento servizio interactive: %i %i \n con w1= %g, w2=
%g, w3= %g \n',ord_1_ant_int,w(3,1),w(3,2),w(3,3))
    ord_1_ant_back = ord_1(:,4);
    fprintf('Primo ordinamento servizio background: %i %i \n con w1= %g, w2=
%g, w3= %g \n\n',ord_1_ant_back,w(4,1),w(4,2),w(4,3))
else
    ord_1_act_conv = ord_1(:,1);
    ord_1_act_stre = ord_1(:,2);
    ord_1_act_int = ord_1(:,3);
    ord_1_act_back = ord_1(:,4);
    if (ord_1_act_conv == ord_1_ant_conv)
    else
        fprintf ('cambio ordine servizio conversazionale: %i %i \n con w1= %g,
w2= %g, w3= %g \n',ord_1_act_conv,w(1,1),w(1,2),w(1,3))
    end
    if (ord_1_act_stre == ord_1_ant_stre)
    else
        fprintf ('cambio ordine servizio streaming: %i %i \n con w1= %g, w2= %g,
w3= %g \n',ord_1_act_stre,w(2,1),w(2,2),w(2,3))
    end
    if (ord_1_act_int == ord_1_ant_int)
    else
        fprintf ('cambio ordine servizio interactive: %i %i \n con w1= %g, w2=
%g, w3= %g \n',ord_1_act_int,w(3,1),w(3,2),w(3,3))
    end
    if (ord_1_act_back == ord_1_ant_back)
    else
        fprintf ('cambio ordine servizio background: %i %i \n con w1= %g, w2=
%g, w3= %g \n\n', ord_1_act_back,w(4,1),w(4,2),w(4,3))
    end
    ord_1_ant_conv = ord_1_act_conv;
    ord_1_ant_stre = ord_1_act_stre;
    ord_1_ant_int = ord_1_act_int;
    ord_1_ant_back = ord_1_act_back;
end
end

% Ranking tramite TOPSIS classico

function [CC,fav_TOPSIS] = TOPSIS(Param1)

[N,M] = size(Param1);
Som = sum(Param1);

for j = 1:M
    for i = 1:N
        Param1_norm (i,j) = Param1(i,j)/Som(j);
    end
end
end

```

```

Mo_max = [0 0 1];
Mo_min = [1 1 0];

for i = 1:N
    CC(i) = Ranking_TOPSIS (Param1_norm(i,:),Mo_max,Mo_min);
end

[CC_fav,fav_TOPSIS] = max (CC);

% Ranking tramite TOPSIS con pesatura dei parametri d'accordo con la classe
% di servizio trattata

function [CC_fav_TC,fav_TOPSIS_TC] = TOPSIS_TC(Param1,w)

[N,M] = size(Param1);
Som = sum(Param1);
T=4;

for l=1:T
    for j = 1:M
        for i = 1:N
            Param1_norm (i,j,l) = (Param1(i,j)*w(l,j))/Som(j);
        end
    end
end

Mo_max = [0 0 1];
Mo_min = [1 1 0];
CC_TC = zeros(N,T);

for l=1:T
    for i = 1:N
        CC_TC (i,l) = Ranking_TOPSIS (Param1_norm(i,:,l),Mo_max,Mo_min);
    end
end

[CC_fav_TC,fav_TOPSIS_TC] = max (CC_TC);

```

RIFERIMENTI



[1] M. Kassar, B. Kervella and G. Pujolle, “An overview of vertical handover decision strategies in heterogeneous wireless networks,” *Computer Communications* (2008), Doi:10. 1016/j.comcom.2008.01.044.

[2] F. Zdarsky and J. Schmitt, “Handover in mobile communication networks: who is in control anyway?”, in: *Proceedings of the 30th Euromicro Conference*, 2004, pp. 205-212.

[3] J. McNair and F. Zhu, “Vertical handoffs in fourth-generation multinet network environments”, *IEEE Wireless Communications*, vol. 11, no. 3, pp. 8 - 15, June 2004.

[4] W. Chen, J.Liu, and H.Huang, “An Adaptive Scheme for Vertical Handoff in Wireless Overlay Networks”, in: *Proceedings of ICPADS’04, Newport Beach, C.A. July 2004*.

[5] I. Akyildiz, J. Xie and S. Mohanty, “A survey of mobility management in next-generation all-IP-based Wireless System”, *IEEE Wireless Communications* 11 (4) (2004) 16-28.

[6] C. E. Perkins, “IP Mobility Support for IPv4”, *IEEE Communications Magazine* 40 (2002) 66-82.

[7] H. Haverinen and J. Malinen, “Mobile IP Regional Paging,” Internet draft, IETF, draft-haverinen-mobileip-reg-paging-00.txt, June 2000.

[8] C. E. Perkins and D. B. Johnson, "Route Optimization in Mobile IP," Internet draft, IETF, draft-ietf-mobileip-optim-11.txt, Sep. 2001.

[9] A. Calvagna and G. di Modica, "A User-Centric Analysis of Vertical Handovers", in: *Proceedings of the second ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, 2004, pp. 137-146.

[10] Media Independent Handover : WHITE PAPER. InterDigital. [Online]

[11] IEEE 802.21 Media Independent Handover Working Group. [Online]. Available: <http://www.ieee802.org/21/>

[12] H. J. Wang, R. H. Katz and J. Giese, "Policy-enabled handoffs across heterogeneous wireless networks," in: *Proceedings of second IEEE workshop on Mobile Computing Systems and Applications 1999*, pp. 51 - 60.

[13] F. Zhu, J. McNair, "Optimizations for vertical handoff decision algorithms", in: *IEEE Wireless Communications and Networking Conference, 2004 (WCNC 2004)*, vol. 2, 2004, pp. 867-872.

[14] Q. Wei, K. Farkas, C. Prehofer, P. Mendes, B. Plattner, "Context-aware handover using active network technology", *Computer Networks* 50 (15) (2006) 2855-2872.

[15] E. Stevens-Navarro and V. W. S. Wong, "Comparison between vertical handoff decision algorithms for heterogeneous wireless networks", in: *Proceedings IEEE VTC 2006-Spring*, pp. 947 - 951.

[16] C.L. Hwang and K. Yoon, "Multiple Attribute Decision Making: An introduction", Springer, Berlin 1981.

[17] T. L. Saaty, “Decision Making with the Analytic Hierarchy Process” Int. J. Services Sciences, 2008.

[18] Q. Song and A. Jamalipour, “A Network Selection Mechanism for Next Generation Networks,” in: *Proceedings ICC 2005*, Korea.

[19] J. L. Deng “Introduction to Grey System Theory”, The Journal of Grey System, vol. 1, no. 1, pp. 1-24,1989.

[20] C. Dwork, R. Kumar, M. Naor and D. Sivakumar, “Rank Aggregation Methods for the Web”, in: *Proceedings of the 10th International World Wide Web Conference*. 2001, 613- 622.

[21] Yu-Ting Liu, Tie-Yan Liu, Tao Qin, Zhi-Ming Ma and Hang Li, “Supervised Rank Aggregation”, WWW 2007.

[22] Ying, W., Jun, Y., Yun, Z, Gen, L, “Vertical Handover Decision in An Enhanced Media Independent Handover Framework”, IEEE WCNC 2008, Las Vegas, USA.

[23] F. Schalekamp and A. van Zuylen, “Rank aggregation: Together we’re strong”, in: *Proceedings of 11th ALENEX*, SIAM (2009), pp. 38–51.

[24] N. Ailon. “Aggregation of partial rankings, p-ratings and top-m lists”, In *SODA ’07: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 415-424. SIAM, 2007.

[25] P. Chan, R. Sheriff, Y. Hu, P. Conforto, C. Tocci, “Mobility management incorporating fuzzy logic for a heterogeneous IP environment”, *IEEE Communications Magazine* 39 (12) (2001) 42–51.

[26] S. Balasubramaniam, J. Indulska, “Vertical Handover supporting pervasive computing in future wireless networks”, *Computer Communications* 27, 2004 708-719.

[27] NP-hard. Wikipedia.

[28] IEEE 802.16 Broadband Wireless Access Working Group. [Online]. Available: <http://www.ieee802.org/16/sysreq/>

[29] F. Bari, V. C. M. Leung, “Automated network selection in a heterogeneous wireless network environment”, in: *Proceedings Network-2007*, pp. 34-40, 2007.

[30] L. Miconi. “Metodo Topsis”. Torino, 26 giugno 2009. [Online].

[31] V. Cuperman, “An upper bound for the error probability on the gilbert channel”. *IEEE Transactions on Communication Technology* 17 (1969) 532–535.