

POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in Ingegneria delle Telecomunicazioni

Dipartimento di Elettronica, Informazione e Bioingegneria



Estensione e valutazione delle prestazioni del protocollo RPL a supporto
di funzionalità di matching distribuito su reti di sensori visuali.

Relatore:

Prof. Matteo Cesana

Tesi di Laurea di:
Fabrizio CASONE
Matr. 754661

Anno Accademico 2013-2014

Alla mia famiglia

Indice

Abstract.....	1
Introduzione.....	2
1 - Visual Sensor Networks.....	4
1.1 Definizione.....	4
1.2 Caratteristiche.....	4
1.2.1 Volume Dati.....	4
1.2.2 Risorse.....	5
1.2.3 Processing.....	5
1.2.4 Real Time.....	5
1.2.5 Localizzazione.....	6
1.2.6 Sincronizzazione.....	6
1.2.7 Memoria.....	6
1.3 Applicazioni.....	7
1.3.1 Sorveglianza.....	7
1.3.2 Monitoraggio Ambientale.....	8
1.3.3 Smart Homes.....	8
1.3.4 Telepresenza.....	8
2 – Riconoscimento di immagini.....	9
2.1 Local Visual Features.....	9
2.2 Scale Invariant Feature Transform (SIFT).....	10
2.2.1 SIFT Detector.....	10
2.2.2 SIFT Descriptor.....	10
2.3 Speeded Up Robust Features (SURF).....	11
2.3.1 SURF Detector.....	11
2.3.2 SURF Descriptor.....	12
2.4 Binary Robust Invariant Scalable Keypoints (BRISK).....	12
2.4.1 BRISK Detector.....	12
2.4.2 BRISK Descriptor.....	13
2.5 Image Retrieval e Riconoscimento di Oggetti.....	13
2.5.1 Feature Matching.....	13
2.5.2 Ricerca del Nearest Neighbor.....	14
2.6 Matching Distribuito.....	14
3 – Estensione del protocollo RPL	15
3.1 RPL.....	15
3.2 RPL - Caratteristiche.....	16
3.2.1 Auto-Configurazione.....	16
3.2.2 Self-Healing.....	16
3.2.3 Loop avoidance and detection.....	16
3.2.4 Indipendenza e Trasparenza.....	16
3.2.5 Edge Router Multipli.....	16
3.2.6 QoS.....	17
3.3 Funzionamento.....	17
3.4 Funzione Obiettivo Proposta.....	19
3.5 RPL: messaggi di controllo.....	19
3.5.1 DODAG Information Solicitation (DIS).....	20

3.5.2 DODAG Information Object (DIO).....	20
3.5.3 Destination Advertisement Object (DAO).....	22
3.5.4 Destination Advertisement Object – Ack (DAO-ACK).....	22
3.6 Costruzione del DODAG.....	23
3.7 Modifiche Apportate e Sviluppo.....	25
3.8 Modifica della Funzione Obiettivo.....	25
4 - Valutazione Prestazioni.....	27
4.1 Scelta del software.....	27
4.2 Contiki OS e la situazione iniziale.....	28
4.3 Simulazioni con le reti di controllo.....	30
4.4 Simulazioni e Raccolta dei Dati.....	31
4.5 Risultati.....	32
4.5.1 Risultati di RPL senza modifica.....	32
4.5.2 Risultati di RPL modificato.....	33
4.5.2 Risultati di RPL modificato.....	33
4.5.2.1 Connessioni totali.....	33
4.5.2.2 Ritardo Medio.....	38
4.5.2.3 Accuratezza.....	39
4.5.3 Analisi Ritardo-Accuratezza (l'impatto della funzione obiettivo).....	42
5 - Conclusioni.....	43
Bibliografia.....	45
Appendice.....	47

Indice Figure, Tabelle e Grafici

Tabella 1: Principali Applicazioni delle VSN.....	7
Figura 1: Creazione del descrittore SIFT.....	10
Figura 2: Rete RPL con tre DODAG divisi in due Istanze separate.....	18
Figura 3: Struttura dei messaggi di controllo RPL.....	19
Figura 4: Formato dei messaggi DIO.....	20
Figura 5: Formato di messaggi DAO.....	22
Figura 6: Operazioni di un router RPL in un DODAG.....	24
Tabella 2: Elementi chiave dei simulatori RPL disponibili.....	27
Tabella 3: Tipologie dei Motes Disponibili in Cooja.....	29
Figura 7: Sky mote e relative informazioni all'interno di Cooja.....	30
Figura 8: Esempio di topologia di rete utilizzata.....	30
Grafico 1: Ritardo Medio in reti che utilizzano RPL senza modifica.....	32
Grafico 2: Percentuale Connessioni a buon fine in reti che utilizzano RPL senza modifica	33
Grafico 3: Percentuali Delle Connessioni a buon fine in reti con 24 Nodi, 2 Server e Coefficiente settato a 50.....	34
Grafico 4: Percentuali Delle Connessioni a buon fine in reti con 24 Nodi, 2 Server e Coefficiente settato a 75.....	34
Grafico 5: Percentuali delle Connessioni a buon fine nelle diverse topologie con Coeff. 50	35
Grafico 6: Percentuali delle Connessioni a buon fine nelle diverse topologie con Coeff. 75	35
Grafico 6: Percentuali delle Connessioni a buon fine nelle diverse topologie con Coeff. 75	35
Tabella 4: Legenda dei grafici per riconoscere le topologie.....	36
Grafico 7: Differenziale delle percentuali delle Connessioni a buon fine nelle reti con modifica rispetto a quelle di riferimento.....	37
Tabella 5: Differenziale delle percentuali delle Connessioni a buon fine nelle reti con modifica rispetto a quelle di riferimento.....	37
Grafico 8: Differenziale del Ritardo Medio in rete Rispetto ai dati di Riferimento.....	38
Tabella 6: Differenziale del Ritardo Medio in rete Rispetto ai dati di Riferimento.....	38
Grafico 9: Accuratezza media raggiunta dalle connessioni nelle varie topologie di rete.....	39
Tabella 7: Accuratezza media raggiunta dalle connessioni nelle varie topologie di rete.....	40
Tabella 8: Distribuzione Accuratezza 7030 Coeff.50.....	40
Tabella 9: Distribuzione Accuratezza 8020 Coeff.50.....	40
Tabella 10: Distribuzione Accuratezza 9010 Coeff.50.....	40
Tabella 11: Distribuzione Accuratezza 7030 Coeff.75.....	41
Tabella 12: Distribuzione Accuratezza 8020 Coeff.75.....	41
Tabella 13: Distribuzione Accuratezza 9010 Coeff.75.....	41
Grafico 10: Accuratezza Raggiunta in relazione Al Ritardo Medio in rete nelle varie topologie analizzate.....	42

Abstract

Una "visual sensor network" è una rete costituita da quelle che vengono denominate "autonomous cameras": delle foto o videocamere che vengono utilizzate per acquisire immagini dell'ambiente circostante e che sono in grado di eseguire varie operazioni su di esse utilizzando le componenti elettroniche con le quali sono equipaggiate; possiamo considerarle come dei veri e propri piccoli computer in grado di comunicare tra loro.

Negli ultimi anni si sono sviluppati vari metodi per il riconoscimento di oggetti all'interno immagini, questi si basano maggiormente sull'estrazione di caratteristiche principali anche dette feature e dal confronto di queste con un database creato in precedenza, questo permette di abbinare un insieme di determinate feature ad un oggetto che sarà quindi presente nella foto in esame.

Questi processi possono essere dispendiosi dal punto di vista energetico e vengono quindi generalmente svolti su terminali fissi alimentati dalla rete elettrica.

L'obiettivo di questo studio è quello di esplorare la possibilità di suddividere il database delle feature in varie parti che possano essere installate anche all'interno dei nodi stessi, decentrando quindi l'operazione di riconoscimento da un computer esterno verso l'interno della rete di sensori.

Questo tipo di approccio, se risultasse vantaggioso, potrebbe fornire diversi benefici: prima di tutto significherebbe un risparmio monetario, visto che un "mote" o elemento di rete è di gran lunga meno costoso di un computer, potrebbe essere conveniente investire in due o tre nodi se questi riuscissero a sostituire un computer in maniera soddisfacente; in secondo luogo avremmo senza dubbio un risparmio dal punto di vista del tempo impiegato nel riconoscimento non dovendo più arrivare fino ad un computer fisso all'esterno della rete per il riconoscimento ma rimanendo all'interno; ci sarebbe quindi anche un ovvio taglio dei tempi di comunicazione.

L'approccio valutato in questo studio si concentra sulla modifica del protocollo di rete, e quindi su come i vari pacchetti in rete prima decidono e poi arrivano alla loro destinazione tenendo conto di come il database per il riconoscimento delle feature è stato suddiviso ed installato nella rete stessa.

Introduzione

Una delle applicazioni principali per le quali vengono utilizzate le Visual Sensor Networks è quella del riconoscimento di immagini; questa viene effettuata utilizzando vari protocolli che si basano tutti sull'estrazione di "feature" che vengono poi confrontate con un database precompilato per ottenere la massima corrispondenza possibile e quindi riconoscere l'oggetto in esame.

Eseguire questo tipo di operazioni in singoli punti esterni alla rete può essere troppo dispendioso dal punto di vista del ritardo, del volume dei dati da trasmettere e dal punto di vista economico; per questi motivi viene proposto un tipo di matching distribuito che utilizza solo porzioni del database installate su nodi interni alla rete che quindi diventa autonoma nell'esecuzione dell'applicazione di riconoscimento.

Ovviamente il posizionamento ed il frazionamento del database nella rete diventano molto importanti per ottenere delle buone prestazioni nel caso in esame; in questo lavoro si prende in considerazione l'implementazione di una modifica al protocollo di rete RPL che tenga conto appunto della suddivisione del database.

RPL è un protocollo basato su Distance Vector e sul concetto di Grafo Diretto Aciclico (o DAG), in pratica, nel caso in esame, RPL costruisce dei grafi orientati (chiamati DODAG) le cui radici sono i nodi/server in cui viene installata una percentuale del database; poi RPL deve tener conto dell'informazione sulla percentuale per creare il resto dell'albero.

Per rendere possibile questo tipo di decisione viene modificata la funzione obiettivo di RPL, cioè la funzione che, analizzando determinate metriche, decide come i nodi si uniscono al DODAG e, di conseguenza, come viene modificato il routing all'interno della rete stessa.

La modifica proposta utilizza la nuova metrica della "percentuale del database nella radice" e la invia ai nodi della rete attraverso i messaggi di controllo di RPL per poi utilizzarla in una combinazione lineare tra la vecchia funzione obiettivo e la nuova informazione.

Una volta implementata la modifica al protocollo sono state eseguite diverse simulazioni su reti di varie dimensioni modificando il numero di server, il frazionamento del database ed il bilanciamento della nuova funzione obiettivo.

I risultati delle simulazioni hanno mostrato come su reti di piccole dimensioni con pochi

server questa modifica non sia vantaggiosa ma, quando il numero di nodi e di server inizia ad aumentare, si cominciano ad ottenere grossi vantaggi per il ritardo sperimentato in rete, il numero delle comunicazioni andate a buon fine e soprattutto come aumenti in media la percentuale del database che viene raggiunta dalle singole richieste dei nodi interni alla rete.

In conclusione questo tipo di modifica si è dimostrata vantaggiosa in tutti i campi analizzati (ritardo, connessioni e percentuale raggiunta) e non risulta essere un appesantimento del carico energetico per i nodi perchè utilizza campi pre-esistenti all'interno dei messaggi RPL per funzionare.

1 - Visual Sensor Networks

In questo capitolo si introduce il concetto di Visual Sensor Network come caso specifico di una Wireless Sensor Network, se ne descrivono le componenti e le caratteristiche principali e si esplorano i possibili campi di utilizzo di questa tipologia di reti.

1.1 Definizione

Il miglioramento della tecnologia utilizzata per realizzare fotocamere e quindi sensori ottici ha avuto come risultato un abbassamento dei prezzi degli stessi, permettendo la nascita delle "Visual Sensor Networks" o VSN [1]: reti di sensori equipaggiate con sensori visuali che rappresentano un upgrade dalle WSN(wireless sensor networks) classiche.

Questi nuovi nodes, chiamati "camera nodes", sono dotati di sensori di immagine, processore, memoria quando necessaria e un modulo per la comunicazione wireless; all'interno di una VSN questi elementi rendono i nodi capaci di comunicare e creare un sistema distribuito per analizzare immagini ed estrarre dati ricchi di informazioni in maniera autonoma.

1.2 Caratteristiche

1.2.1 Volume Dati

Una delle principali differenze tra VSN e WSN sta nella tipologia dei dati trattati: normalmente una rete di sensori analizza dati monodimensionali, mentre in una VSN le immagini sono percepite come un segnale bidimensionale restituito dal sensore ottico.

Oltre alla differenza di base tra i segnali che rappresentano i dati c'è anche da considerare la fondamentale diversità del modello da applicare ai sensori: mentre tipicamente un sensore raccoglie dati dall'area circostante, quelli raccolti da una fotocamera sono completamente direzionali e rappresentano un'immagine in due dimensioni di uno spazio tridimensionale chiamato Campo Visivo (Field of View o FoV).

1.2.2 Risorse

Come nelle WSN, operando a batteria, i mote sono soggetti ad un vincolo molto stringente sul consumo energetico [2][5]; nelle VSN questo vincolo è ancora più pesante viste le grandi quantità di dati da gestire inerenti alle applicazioni che lavorano con immagini.

La vita media di un nodo dipende da molti fattori: il tempo in cui viene acceso il sensore ottico, i tempi di utilizzo della cpu, della memoria e del trasmettitore; tutti questi sono fattori da tenere in considerazione quando si vogliono sviluppare applicazioni all'interno di una VSN.

1.2.3 Processing

Se un nodo può eseguire operazioni complesse sulle immagini catturate in maniera locale questo si traduce in un risparmio sui dati da trasmettere utilizzando il modulo wireless, ovviamente la capacità di processing ha un suo costo di base per quanto riguarda l'utilizzo di batteria e di realizzazione per il mote, che diventa quindi più dispendioso.

A seconda dell'applicazione che si vuole realizzare e dei costi avremo quindi la possibilità di avere camera nodes più "intelligenti" e capaci di trattare le immagini con algoritmi più o meno complessi.

Uno sviluppo ulteriore può essere la collaborazione dei nodi all'interno della VSN con la finalità dell'analisi delle immagini, minimizzando, in questo caso, la necessità di elaborazione esterna alla rete di sensori.

1.2.4 Real Time

La maggior parte delle applicazioni per VSN richiede una componente real time imponendo così un vincolo molto stringente sul ritardo massimo tra la fotocamera e l'utente [4].

In questo caso la "pressione" ricadrà sul modulo wireless che dovrà soddisfare le esigenze e fornire qualche forma di assicurazione sotto forma di correzione e o protezione degli errori oltre che modificare o tenere sotto controllo il tipo di routing utilizzato e la banda totale occupata.

1.2.5 Localizzazione

Nella maggior parte delle applicazioni per VSN oltre all'elaborazione delle immagini è richiesta l'esatta posizione dalla quale queste provengono, rendendo necessaria la localizzazione del camera node e del suo orientamento; questa informazione viene in genere ottenuta tramite il processo di calibrazione del nodo stesso che fornisce all'utente i parametri necessari; per eseguire queste operazioni vengono utilizzati set di informazioni o punti conosciuti tra le varie camere, quando questi non sono disponibili gli unici dati possibili sono quelli della posizione relativa tra i nodi stessi.

1.2.6 Sincronizzazione

La sincronizzazione tra i nodi diventa importante quando le immagini devono essere messe in relazione con l'istante di tempo nel quale sono state acquisite ed in questo caso vengono utilizzati protocolli di sincronizzazione per reti di sensori wireless.

1.2.7 Memoria

In linea teorica, come abbiamo già visto, un camera node produce una quantità di dati maggiore rispetto al mote tradizionale e può essere quindi conveniente immagazzire gli stessi al posto di trasmetterli tutti.

La memoria installata sul mote diventa quindi molto importante e può servire per mantenere immagini che possono essere analizzate in un momento successivo o sostituite nel momento in cui perdono di importanza.

1.3 Applicazioni

TABLE 1: Applications of visual sensor networks.

General application	Specific application
Surveillance	Public places
	Traffic
	Parking lots
	Remote areas
Environmental monitoring	Hazardous areas
	Animal habitats
	Building monitoring
Smart homes	Elderly care
	Kindergarten
Smart meeting rooms	Teleconferencing
	Virtual studios
Virtual reality	Telepresence systems
	Telereality systems

Tabella 1: Principali Applicazioni delle VSN

Il rapido sviluppo delle VSN ha portato a trovare numerose applicazioni [3] in cui utilizzarle : come si evince dalla *Tabella 1* abbiamo in generale quattro tipologie di applicazione :

1.3.1 Sorveglianza

La sorveglianza è stata l'attività principale di applicazioni "camera-based" per molto tempo, soprattutto per monitorare aree pubbliche come aeroporti e stazioni; in generale richiede l'utilizzo di un alto numero di videocamere che forniscono dei video stream dai quali raccogliere informazioni importanti tramite una grande quantità di processing e risorse umane che rendono queste operazioni molto costose, sia dal punto di vista del tempo che della vulnerabilità agli errori.

Attualmente la tecnologia delle VSN sta essendo utilizzata per migliorare la situazione, provando a realizzare l'estrazione delle informazioni dalle immagini localmente sulle videocamere diminuendo così l'ammontare di traffico dati necessario; inoltre, vista la natura stessa delle VSN che si basano su tecnologia wireless ed hanno applicazioni sensibili al consumo energetico, queste nuove reti di sensori rappresentano il futuro per implementare la sorveglianza anche in luoghi in cui mancano le infrastrutture che sono invece indispensabili per utilizzare il vecchio paradigma: per esempio non è più necessario

avere una grande capacità computazionale in un grosso server centrale che assorbe grandi quantità di risorse.

Questa tipologia di reti è particolarmente adattabile alle dinamiche ambientali, è autonoma e può rispondere a richieste dell'utente per raccogliere immagini immediate dal punto di vista desiderato fornendo così le informazioni necessarie dalle aree specifiche richieste in remoto.

1.3.2 Monitoraggio Ambientale

Le VSN possono essere utilizzate per monitorare aree remote ed inaccessibili per grandi periodi di tempo.

Ovviamente in queste situazioni è necessario operare con un'alta efficienza energetica per prolungare la vita dell'applicazione il più possibile; in molti casi le videocamere sono combinate con altre topologie di sensori per creare reti eterogenee in modo da poter attivare le videocamere utilizzando delle soglie dipendenti da altri fattori ambientali raccolti dalla rete.

1.3.3 Smart Homes

Ci sono situazioni in cui è richiesto un controllo costante delle persone, come per esempio in ospedali o nei casi di persone con particolari disabilità; in questi casi le VSN possono essere utilizzate per monitoraggio continuo e, mediante l'utilizzo di algoritmi intelligenti, la rete può fornire informazioni sulla necessità di cure riguardanti un paziente o avvertire in caso di emergenze o comportamenti anomali.

1.3.4 Telepresenza

La telepresenza può essere realizzata utilizzando sensori audio e video e può servire in varie applicazioni: possiamo per esempio rendere possibili degli "smart meetings" usando le VSN per permettere una partecipazione in remoto, oppure possiamo utilizzare le reti di sensori video nei musei e gallerie d'arte, le cui stanze possono essere "coperte" da una VSN che restituisca uno stream video a chiunque voglia visitare questi luoghi in remoto, fornendo così all'utente un punto di vista da ogni possibile nodo della rete.

2 – Riconoscimento di immagini

In questo capitolo viene spiegato il funzionamento di una delle applicazioni più diffuse per le reti VSN: il riconoscimento di immagini e di oggetti attraverso l'estrazione delle principali caratteristiche presenti, che vengono poi comparate con gli elementi all'interno di un database per scegliere quale oggetto abbinare all'immagine iniziale [6].

2.1 Local Visual Features

Una "Visual Feature" rappresenta una parte "interessante" all'interno di un'immagine che può coincidere con una particolare struttura dal punto di vista della rappresentazione nel dominio dei pixel; dei buoni candidati ed esempi possono essere lati, angoli e strutture particolari.

Per rilevare queste caratteristiche (o feature appunto) esistono vari algoritmi che vengono chiamati detectors.

Questi algoritmi cercano particolari feature attorno a punti chiave (keypoints) nelle immagini; dopo la loro individuazione le feature locali vengono rappresentate da un vettore anche chiamato descriptor (o descrittore) che codifica la zona dei pixel attorno al keypoint in un vettore numerico.

Gli algoritmi che generano questi vettori agiscono in modo che il loro output sia il più robusto possibile rispetto alle diverse trasformazioni possibili per un'immagine, come il cambio del punto di vista o lo zoom o la rotazione e cambio di illuminazione.

La robustezza è ovviamente un elemento critico per implementare algoritmi di riconoscimento efficienti che possano operare sotto condizioni visive differenti.

Di seguito troviamo una veloce introduzione sui tre principali algoritmi utilizzati per l'estrazione e codifica delle feature.

2.2 Scale Invariant Feature Transform (SIFT)

Questo algoritmo è considerato lo standard per l'estrazione delle feature ed è sia invariante rispetto alle variazioni di scala che robusto nella creazione dei descrittori; SIFT è in grado di estrarre un alto numero di feature ben localizzate e distinte che possono poi essere utilizzate per operazioni di riconoscimento in maniera affidabile [7].

2.2.1 SIFT Detector

Il detector di SIFT è composto da tre stage consecutivi:

- "Scale-space extrema detection": i keypoint sono individuati in tutte le posizioni all'interno dell'immagine utilizzando tutte le scale possibili utilizzando una serie di filtri a cascata.

- "Localizzazione dei keypoint": per ogni zona candidata viene applicato un modello per determinare accuratamente la scala e la precisa posizione di ogni keypoint.

- "Assegnamento dell'orientamento": come ultimo step, ad ogni keypoint viene assegnato un orientamento basato sulla direzione del gradiente locale nell'immagine.

In questo modo viene garantita l'invariabilità rispetto alla posizione, alla rotazione e alla scala.

2.2.2 SIFT Descriptor

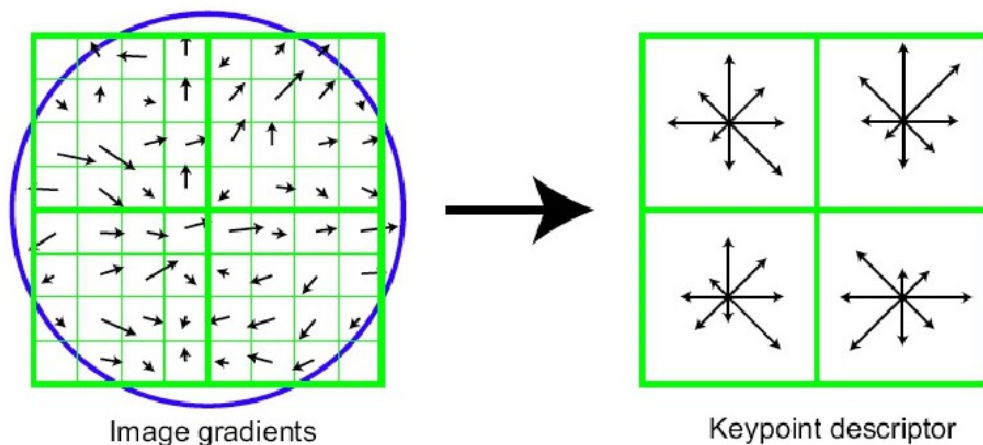


Figura 1: Creazione del descrittore SIFT

La fase di ricerca ha fornito ogni keypoint con una posizione, una scala ed un orientamento; questi parametri definiscono un sistema di coordinate 2D nelle quali definire la zona in oggetto in maniera invariante rispetto alle trasformazioni dell'immagine.

A questo punto deve essere calcolato un descrittore per questa zona in maniera altamente distintiva e possibilmente invariante rispetto ad altri tipi di trasformazioni come la variazione di illuminazione.

Per prima cosa i gradienti di magnitudine e l'orientamento nell'immagine vengono campionati attorno al keypoint e tutte le coordinate vengono ruotate utilizzando l'orientamento del keypoint stesso per implementare invarianza rispetto alla rotazione; poi una funzione gaussiana è utilizzata per assegnare pesi ad ogni punto di campionamento per evitare cambi improvvisi nel descrittore abbinati a piccoli cambi di posizione.

A questo punto il descrittore viene costruito creando istogrammi di orientamento su sezioni di campionamento grandi 4x4 come possiamo vedere in *Figura 1*.

2.3 Speeded Up Robust Features (SURF)

L'algoritmo Speeded Up Robust Features (SURF) [8] approssima le prestazioni di SIFT provando ad essere più veloce e leggero da eseguire.

L'aumento di velocità è raggiunto utilizzando tecniche di "integral imaging" per le convoluzioni all'interno del detector e semplificando gli schemi esistenti per costruire il descrittore.

2.3.1 SURF Detector

Il detector di SURF si basa su di una approssimazione della matrice Hessiana grazie alle sue alte performance in termini di accuratezza e complessità computazionale.

L'utilizzo della matrice Hessiana implica convoluzioni con derivate del secondo ordine di funzioni Gaussiane e può essere molto costoso; per ridurre la complessità SURF approssima queste convoluzioni con una serie di "box filter" (filtri a scatola) che può essere utilizzata in maniera computazionalmente efficace utilizzando appunto le "immagini integrali".

2.3.2 SURF Descriptor

Il primo passo nell'estrazione del descrittore consiste nel centrare una regione quadrata attorno al keypoint orientata lungo la direzione dominante e la cui dimensione è proporzionale alla scala del keypoint stesso.

La regione è poi divisa in piccole sotto regioni 4x4; in ogniuna di esse sono presi 25 campioni e per ogni campione viene applicata la trasformata di Haar, a questo punto le risposte vengono sommate su ogni sottoregione per formare le componenti del vettore delle feature.

Il descrittore finale è ottenuto concatenando i vari feature vectors di tutte le sottoregioni ottenendo così un vettore finale composto da 64 elementi.

L'invarianza rispetto al contrasto viene ottenuta in modo simile a SIFT normalizzando i valori del descrittore.

2.4 Binary Robust Invariant Scalable Keypoints (BRISK)

Binary Robust Invariant Scalable Keypoints (BRISK) [9] è un nuovo tipo di algoritmo che è stato creato pensando in modo specifico ad architetture a basso consumo energetico.

Il focus principale è stato centrato sul creare algoritmi che siano drasticamente più economici dal punto di vista computazionale rispetto quelli precedenti e che contemporaneamente mantengano un'alta qualità nelle performance.

2.4.1 BRISK Detector

L'elemento chiave che garantisce la velocità in BRISK sta nell'applicazione di un nuovo "space-scale corner detector": dovendo soddisfare la richiesta di efficienza computazionale, il detector di BRISK è ispirato al popolare rivelatore di angoli (corner detector) "Fast Accelerated Segment Test" (FAST) che è molto efficiente come base per l'estrazione delle feature; uno dei problemi di questo approccio però è che FAST non produce keypoint che siano invarianti rispetto alla scala utilizzata, in BRISK questo gap viene colmato dalla ricerca degli angoli non solo nell'immagine piana ma anche in quella scalata.

2.4.2 BRISK Descriptor

Il Descrittore di BRISK è estremamente veloce da calcolare poichè si basa sul concetto di "descrittore binario": questi sono assemblati con stringhe di bit ottenute dal confronto dell'intensità dei pixel che richiede un piccolo sforzo computazionale.

Per la formazione del descrittore normalizzato rispetto alla scala e alla rotazione BRISK applica un pattern di campionamento ruotato attorno al keypoint, il binary descriptor è poi costruito comparando l'intensità di coppie di punti a breve distanza.

2.5 Image Retrieval e Riconoscimento di Oggetti

Le Visual Feature locali sono utilizzate come base per molte applicazioni visive; in particolare dimostrano delle eccellenti performance quando vengono usate per tecniche di riconoscimento e ripresa della immagini.

Applicazioni come il riconoscimento di oggetti e paesaggi, richieste di immagini e simili sono solo alcune possibilità in cui possono essere utilizzate le visual features con successo.

2.5.1 Feature Matching

Il caso principale è quello del riconoscimento: in cui un'immagine, che proviene da un sensore visuale, contiene un oggetto che si vuole identificare utilizzando le feature estratte dall'immagine stessa.

Il processo di riconoscimento utilizza un database di immagini di riferimento dove per ogni immagine presente sono estratte ed etichettate tutte le visual features presenti (ovviamente questa operazione viene eseguita offline in precedenza).

A questo punto l'idea è quella di identificare all'interno del database l'immagine più simile all'immagine che si sta analizzando; questa similitudine può essere trovata realizzando un matching (corrispondenza) delle feature presenti nell'immagine con tutte quelle delle immagini presenti nel database e poi scegliendo quella che restituisce il più alto numero di corrispondenze.

Nonostante il processo sembri abbastanza semplice si possono implementare numerosi miglioramenti affinché questo processo abbia le performance migliori possibili.

2.5.2 Ricerca del Nearest Neighbor

Per decidere se un feature nell'immagine di partenza corrisponde ad una all'interno del database viene eseguita una ricerca detta del "nearest neighbor" (letteralmente del più vicino).

Considerato il numero di feature che possono essere contenute in un'immagine analizzata o all'interno del database il numero di confronti da eseguire può diventare molto alto e quindi una ricerca che prenda in considerazione tutte le immagini può essere a sua volta troppo costosa sia dal punto di vista computazionale che dal punto di vista del ritardo; per ovviare a questi problemi vengono utilizzate varie soluzioni: si possono implementare per esempio delle approssimazioni dell'algorithmo del nearest neighbor come il "kd-tree" (KDT) o le "hierarchical k-means" (HKM) oppure algoritmi come il "locality sensitive hashing"(LSH).

2.6 Matching Distribuito

In ogni caso comunque le operazioni di matching si basano tutte sull'utilizzo di un database per riconoscere le feature presenti nelle immagini; diventa quindi molto importante il suo posizionamento all'interno di una rete VSN che abbia questo tipo di scopi, una delle soluzioni proposte è quella di non installarlo completamente su di un singolo nodo, che poi si dovrà fare carico di tutte le operazioni di riconoscimento, ma dividerlo in sezioni differenti su vari nodi realizzando una rete VSN che operi appunto un matching distribuito sulla rete.

L'utilizzo di questo tipo di soluzioni richiede ovviamente una riflessione sull'impatto generato sul traffico nella rete stessa e sui protocolli che essa utilizza; per questo motivo in questo lavoro abbiamo analizzato e modificato i protocolli di rete delle VSN per renderli più "consapevoli" o comunque adattabili alle operazioni di matching per il quale la rete stessa può essere stata costruita.

3 – Estensione del protocollo RPL

In questo capitolo viene analizzato uno dei principali protocolli di rete utilizzati nelle reti di sensori e, dopo averne spiegato le caratteristiche principali, si introduce la proposta di una modifica che potrebbe essere vantaggiosa per lo scenario analizzato in questo studio.

3.1 RPL

RPL [10] o "Routing Protocol for Low power and lossy networks" è un protocollo di routing che è stato sviluppato per reti di sensori LLN (low power and lossy networks appunto), cioè reti di sensori (quindi sensibili dal punto di vista del consumo energetico) caratterizzate da link non particolarmente robusti con un alto tasso di "link outage" ed errori per pacchetto.

RPL è un protocollo che si basa su Distance-Vector (DV) e sul source routing, è pensato per operare al di sopra di differenti link-layer, quali IEEE 802.15.4 PHY ed i MAC layer sottostanti.

Viene utilizzato su reti di tipo collection-base cioè reti nelle quali i nodi inviano periodicamente le loro misurazioni ad un nodo centrale o collection point.

Questo protocollo è basato sul concetto di Grafo Diretto Aciclico (Directed Acyclic Graph o DAG); il DAG definisce una struttura simile ad un albero che stabilisce le rotte tra i nodi all'interno della rete LLN.

Il DAG è più di un semplice albero poiché un nodo potrebbe associarsi a più di un parent all'interno del DAG stesso contrariamente a quanto avviene nelle tradizionali strutture dove un nodo può avere un singolo parent.

In realtà RPL organizza i nodi in quelli che vengono chiamati DODAG o Destination Oriented DAG cioè grafi orientati dove le destinazioni sono quelle che in genere permettono un accesso ad internet e agiscono come radice del DAG stesso.

3.2 RPL - Caratteristiche

3.2.1 Auto-Configuration

Dovendo essere compatibile con Ipv6 una LLN che sfrutta RPL possiede tutte le caratteristiche di base di IP e quindi può sfruttare la dynamic discovery di path e destinazioni utilizzando il meccanismo di Neighbor Discovery.

3.2.2 Self-Healing

RPL è in grado di adattarsi a cambiamenti logici nella topologia della rete ed a rotture dei nodi; questa è una delle caratteristiche principali di questo protocollo dovendo essere utilizzato in reti che sono caratterizzate da questo tipo di problematiche (le LLN appunto). Il meccanismo utilizzato è quello della scelta di nodi parent multipli per eliminare o almeno diminuire il rischio di fallimento.

3.2.3 Loop avoidance and detection

I DAG creati sono aciclici di natura poiché un nodo all'interno del DAG deve avere un grado (o rank) maggiore di tutti i suoi parent. Come misura ulteriore di prevenzione RPL include meccanismi reattivi per rilevare eventuali loop in caso di un cambio nella topologia della rete: quando questo avviene RPL fa iniziare dei meccanismi di recupero locali o globali a seconda della necessità.

3.2.4 Indipendenza e Trasparenza

Esattamente come IP, RPL è stato creato per operare sopra diversi link layer; questo protocollo è in grado di operare in reti con forti restrizioni o in congiunzione con dispositivi che abbiano requisiti molto stringenti, in pratica RPL è completamente indipendente dal data-link layer e quindi trasparente rispetto ad esso.

3.2.5 Edge Router Multipli

Vista la possibilità di costruire DAG multipli in una rete RPL dove ogni DAG ha una sua "radice" o "root", un nodo può appartenere a più istanze e quindi a più DAG

contemporaneamente, assumendo in ognuno un ruolo diverso, garantendo in questo modo una grossa capacità di load balancing alla rete stessa.

3.2.6 QoS

Visto che RPL è un protocollo che utilizza metriche e restrizioni è possibile implementare con esso anche della QoS (Quality of Service): a questo scopo vengono definiti ed utilizzati vari parametri:

Node Energy Object: una restrizione che serve ad evitare nodi che abbiano una bassa riserva di energia

Hop Count Object: una metrica che semplicemente conta il numero di hop in uno specifico path.

Link Throughput / Latency / Reliability / Color Object: che sono tutte metriche che permettono di controllare lo stato di un link per gestire il traffico all'interno della rete in maniera consapevole ai fini di mantenere una determinata QoS.

3.3 Funzionamento

Rpl definisce tre tipologie di nodi:

Low Power and Lossy Border Routers (LBRs): sono le radici dei DODAG, rappresentano il punto di raccolta dei dati all'interno della rete e possiedono la capacità di creare dei DAG.

Gli LBR agiscono anche come gateway (o edge router) tra internet e la LLN.

Router: sono dispositivi che possono generare e smistare traffico, non hanno la capacità di creare DAG ma si uniscono a quelli già esistenti.

Host: sono dispositivi che hanno la capacità di generare traffico ma non di smistarlo.

Ad ogni nodo all'interno di un DODAG viene assegnato un grado (o rank) che è definito come la "posizione individuale relativa agli altri nodi e alla radice del DODAG".

Il rank è un numero intero che di fatto rappresenta la posizione all'interno del DODAG stesso; questo numero aumenta andando dalla radice verso l'esterno, in pratica nodi con posizioni più "alte" nell'albero o più vicini alla radice avranno rank più bassi rispetto a nodi più lontani.

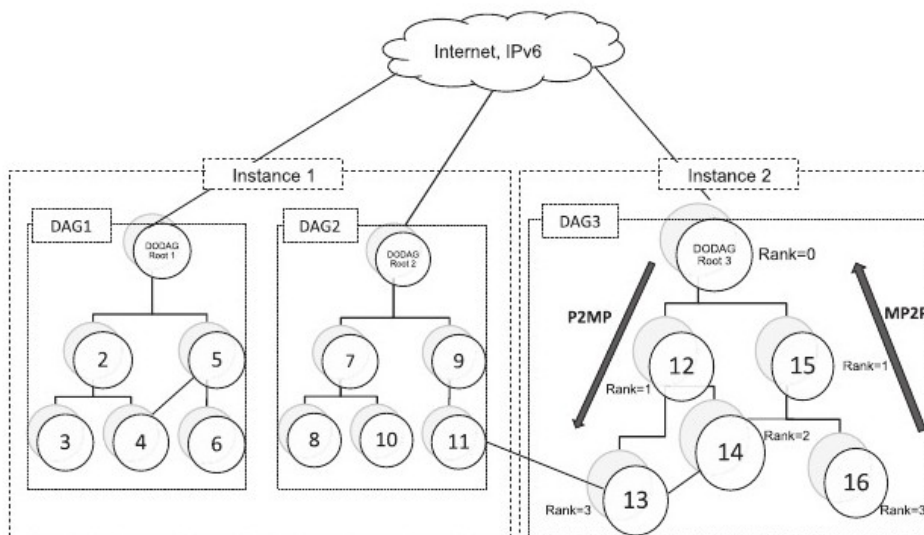


Figura 2: Rete RPL con tre DODAG divisi in due Istanze separate

Come si vede in *Figura 2* l'architettura di un DODAG è simile a quella di una topologia cluster-tree dove tutto il traffico viene raccolto nella "radice"; la differenza tra le due topologie sta nel fatto che in RPL un nodo può essere associato non solo al parent ma anche ai nodi fratelli (siblings) con lo stesso rank.

Il grado in RPL viene utilizzato per rilevare ed evitare eventuali loop e permette ai nodi di distinguere tra i nodi parent e siblings all'interno del DODAG, infatti i nodi mantengono una lista dei possibili parent e siblings con i quali sostituire l'attuale parent se ci dovessero essere dei problemi.

Nel processo di costruzione del DODAG ogni nodo identifica un set di possibili parent sulla strada per la radice del DAG e si associa ad un parent preferito che viene scelto utilizzando una "Funzione Obiettivo"; questa funzione definisce come i nodi che utilizzano RPL traducono determinate metriche in decisioni sul routing all'interno della rete.

3.4 Funzione Obiettivo Proposta

La funzione obiettivo è responsabile per il calcolo del rank in base a specifiche metriche, come delay e qualità del link, e determina le restrinzioni da applicare per ottimizzare il routing.

Il design di una Funzione Obiettivo efficiente è ancora di grande interesse per la ricerca e finora sono state proposte solo alcune soluzioni:

-quella che sarà poi utilizzata per gli scopi di questo lavoro utilizza l'ETX o Expected Number of Transmission, cioè il numero di trasmissioni necessarie in media per trasmettere con successo un pacchetto su un determinato link all'interno del path, e decide in base a questa metrica come creare il DODAG e quindi instradare i pacchetti.

-un'altra funzione, chiamata "Objective Function 0" (Of0) [11], è completamente trasparente rispetto a metriche riguardanti i link e si basa solamente su informazioni trasportate all'interno dei pacchetti RPL com per esempio il rank dei nodi in esame.

3.5 RPL: messaggi di controllo

I messaggi RPL sono registrati come un nuovo tipo di messaggi di controllo di ICMPv6 e sono composti da un header di ICMPv6 che contiene tre campi: type, code e checksum; mentre la seconda parte contiene un messaggio base ed un determinato numero di opzioni come si vede in *Figura 3*.

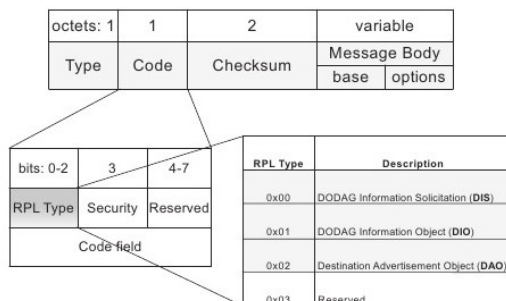


Figura 3: Struttura dei messaggi di controllo RPL

Le tipologie di messaggi di controllo per RPL sono quattro [12]:

3.5.1 DODAG Information Solicitation (DIS)

I messaggi DIS sono utilizzati per sollecitare un l'invio di un DODAG Information Object da un nodo RPL.

Questi messaggi possono essere utilizzati per sondare i nodi in un DODAG vicino.

Attualmente il formato dei messaggi DIS contiene dei campi con flag non specificati e riservati per utilizzo futuro.

3.5.2 DODAG Information Object (DIO)

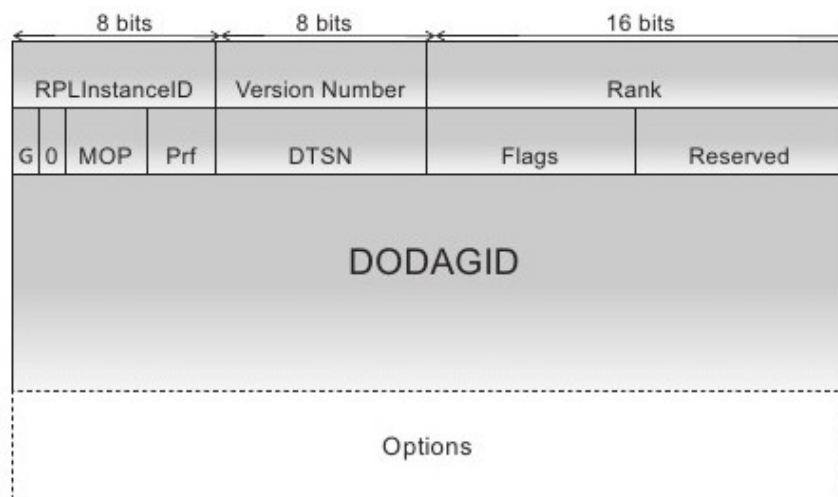


Figura 4: Formato dei messaggi DIO

I messaggi DIO (Figura 4) sono inviati dalla radice del DODAG per costruire un nuovo DAG e vengono mandati in multicast a tutta la struttura del DODAG.

Questi messaggi contengono informazioni importanti riguardanti la rete e permettono ai nodi di scoprire istanze di RPL con i parametri delle loro configurazioni e quindi di scegliere un set di parent nodes.

Il formato di questi messaggi contiene varie informazioni:

-*RPLinstance ID*: un numero di 8 bit che contiene l'informazione che dice a quale istanza di RPL appartiene il DODAG.

-*Version Number*: contiene la versione del DODAG che viene incrementata ogni volta che le informazioni vengono aggiornate e aiuta a mantenere tutti i nodi sincronizzati con gli stessi aggiornamenti.

-*Rank*: un campo di 16 bit che specifica il rango del nodo che sta mandando il DIO

-*Destination Advertisement Trigger Sequence Number (DTSN)*: un campo di 8 bit che viene utilizzato per mantenere i vari path nella giusta direzione lungo l'albero.

-*Grounded*: un campo che contiene un flag per sapere se il DODAG corrente soddisfa i requisiti dell'applicazione per il quale si sta utilizzando la rete.

-*Mode Of Operation (MOP)*: indica quale modalità di operazioni per RPL è stata scelta dal root. Sono possibili quattro differenti modalità, a seconda che si voglia supportare il routing in direzione discendente o ascendente lungo l'albero e se si voglia abilitare o meno il multicast. Ovviamente un nodo che si aggiunge al DODAG deve essere in grado di supportare le modalità scelte.

-*DODAGPreference (Prf)*: un campo di 3 bit che indica il grado di preferenza del root del DODAG correntemente analizzato rispetto agli altri presenti in rete.

-*DODAGID*: è un indirizzo di 128bit di Ipv6 fissato dalla radice del DODAG che identifica univocamente il DODAG stesso.

3.5.3 Destination Advertisement Object (DAO)

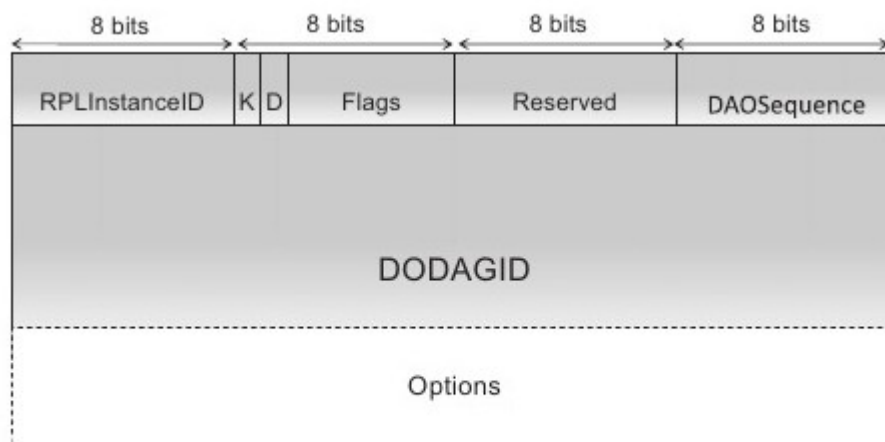


Figura 5: Formato dei messaggi DAO

I messaggi DAO (Figura 5) sono utilizzati per propagare informazioni sul routing inverso registrando il nodi visitati lungo il path percorso.

Questi messaggi sono mandati da tutti i nodi tranne il root e vengono usati per riempire le tabelle di routing con i prefissi dei nodi figli e per comunicare i propri indirizzi e prefissi ai propri nodi parent.

Dopo che un messaggio DAO partito da un nodo ha attraversato il DAG ed è arrivato alla radice, al suo interno è immagazzinato il path completo tra root e punto di partenza.

3.5.4 Destination Advertisement Object – Ack (DAO-ACK)

Il DAO-ACK è un pacchetto unicast mandato per rispondere alla ricezione di un DAO e contiene informazioni quali: *RPLInstanceID*, *DAOSequence* e *Status*.

3.6 Costruzione del DODAG

La costruzione del DODAG si basa sul processo di "scoperta dei vicini" (Neighbor Discovery o ND) che consiste in due operazioni principali: la trasmissione di messaggi DIO da parte della radice per costruire rotte discendenti verso i nodi, e la trasmissione di messaggi di controllo DAO in direzione ascendente dai nodi verso la radice.

Per creare un nuovo DODAG il nodo root trasmette un messaggio DIO nel quale dichiara un nuovo DODAGID, il suo rank per permettere ai nodi di comprendere il loro grado all'interno del nuovo DODAG e la funzione obiettivo utilizzata indentificata dall'Objective Code Point (OCP) all'interno del campo DIO configuration.

Questo messaggio sarà ricevuto da un nodo che, analizzando le informazioni, potrà a sua volta decidere se aggregarsi al DODAG o meno (*Figura 6*).

Quando un nodo disposto ad aggregarsi riceve un messaggio DIO esegue le seguenti operazioni:

- Aggiunge il mandante del DIO alla lista dei suoi parent.
- Calcola il proprio rank utilizzando l'appropriata funzione obiettivo specificata nel messaggio in modo che il grado sia maggiore di quello di ogniuno dei propri parent.
- Inoltra il messaggio DIO con le informazioni sul rank aggiornate.

Un nodo client sceglie il "most preferred parent", o parent preferito, tra i presenti nella lista e lo utilizza come nodo di default verso il quale dirigere il traffico.

Quando un nodo che è già associato al DODAG riceve un messaggio DIO questo può procedere in tre differenti modi:

- Scarta il messaggio secondo criteri precedentemente codificati in RPL.
- Processa il DIO mantenendo la propria posizione all'interno del DODAG.
- Migliora la propria posizione ottenendo un rank minore grazie alla funzione obiettivo.

Ogni volta che un nodo cambia la propria posizione all'interno del DODAG questo deve scartare dalla sua lista tutti i parent che abbiano un rank minore di quello appena calcolato per evitare eventuali loop.

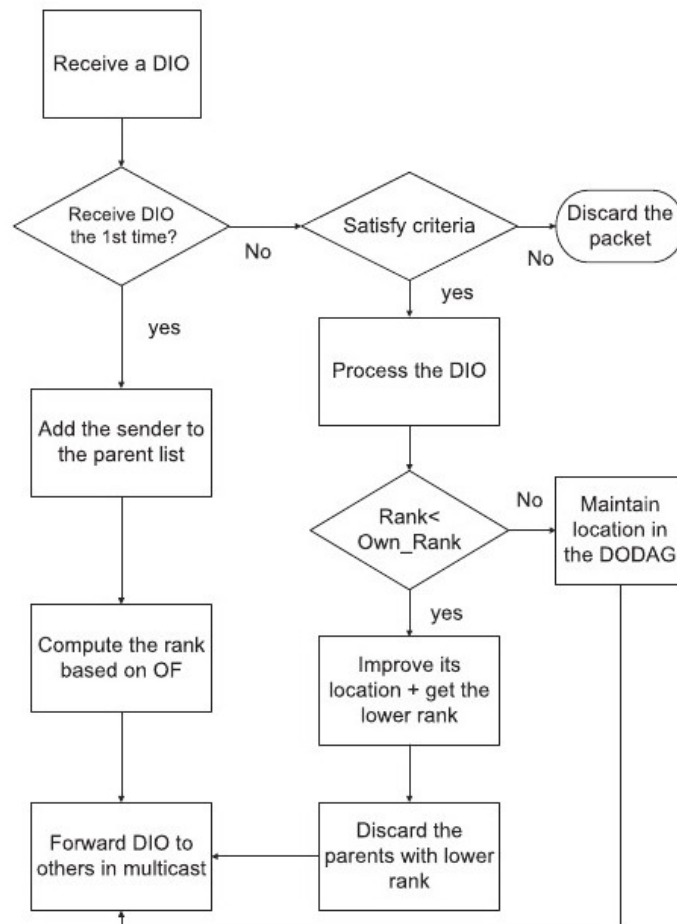


Figura 6: Operazioni di un router RPL in un DODAG

Dopo aver completato la costruzione di un DODAG ogni nodo ha un path per inoltrare il traffico verso la radice ed ovviamente quello principale è costituito dai parent preferiti.

Se nel campo Mode of Operation all'interno del DIO il Base Object è settato in maniera differente da zero allora le rotte discendenti dalla radice devono essere mantenute; in questo caso quindi ogni nodo deve mandare messaggi DAO di controllo per determinare rotte inverse a quelle appena stabilite dalla costruzione del DODAG.

Risalendo le rotte i nodi visitati vengono registrati nei pacchetti ed in questo modo vengono stabilite rotte tra la radice e i nodi client.

RPL stabilisce due Mode Of Operation:

-*Storing mode*: in questa modalità un DAO viene mandato da un nodo figlio al nodo parent selezionato che ha l'abilità di immagazzinare i messaggi ricevuti dai figli per aggregare le informazioni e mandarle ai propri parent.

-*Non-storing mode*: in questa modalità i messaggi DAO sono mandati alla radice ed i nodi non hanno capacità di immagazzinamento degli stessi ma solo di inserire le informazioni nei pacchetti che poi inoltrano verso la radice.

Per mantenere i DODAG vengono impostati dei "trickle timer" che fanno partire dei messaggi DIO a determinati intervalli che sono dipendenti dallo stato della rete, in modo da ottimizzare al meglio la manutenzione di RPL.

3.7 Modifiche Apportate e Sviluppo

Per risolvere il problema della comunicazione tra i nodi del dato sulla percentuale del database nei vari server si è scelto di aggiungere questa informazione all'interno dei messaggi DIO (vedi *Figura 4*).

Come in tutti i protocolli in via di sviluppo, nei messaggi sono presenti diversi campi riservati ad utilizzi futuri; inserendo la percentuale del database presente nel root del DODAG nei messaggi di servizio RPL otteniamo due vantaggi principali: il primo è ovviamente soddisfare la necessità della circolazione di questa tipologia di informazioni per i nostri scopi ed il secondo è che, di fatto, non aggiungiamo realmente niente di nuovo al protocollo, se non la modifica di un integer che comunque avrebbe occupato spazio in memoria senza essere utilizzato: evitando, in questo modo, un aumento del consumo energetico da parte della cpu.

3.8 Modifica della Funzione Obiettivo

Le modalità appena descritte di attenzione al consumo energetico sono state utilizzate anche nella modifica della funzione obiettivo (FO).

La FO di partenza che è stata utilizzata è quella denominata "mrhof" che, in parole povere, utilizza la metrica dell'ETX per assegnare rank e decidere il routing all'interno della rete.

La modifica operata in questo lavoro consiste nell'implementare, al posto del singolo studio dell'ETX, una combinazione lineare della metrica precedente e della distribuzione del database delle feature.

Quindi se:

ETX = Vecchia Metrica (dipendente solo dal dato su ETX)

Percentuale = Percentuale Del Database (nel root del DODAG)

Coefficiente = Coefficiente per decidere il bilanciamento della combinazione lineare (che varia da 0 a 100)

Allora la nuova metrica utilizzata nella funzione obiettivo verrà calcolata nel seguente modo:

$$NuovaMetrica = \frac{((ETX * (100 - Coefficiente)) - (Percentuale * Coefficiente))}{100}$$

Dove il meno è dovuto al fatto che nelle funzioni obiettivo vengono privilegiati i valori più bassi.

Dopo aver normalizzato entrambi in campi "ETX" e "Percentuale" sullo stesso range di valori avremo quindi una metrica controllata dal coefficiente della combinazione lineare che ci permetterà di decidere quale delle due metriche avvantaggiare nelle nostre simulazioni.

Anche in questo caso la modifica al codice non rappresenta un aumento apprezzabile dell'utilizzo della cpu e quindi non impatta significativamente le risorse energetiche dei nodi.

4 - Valutazione Prestazioni

In questo capitolo vengono spiegati gli strumenti utilizzati e le scelte fatte per ultimare l'analisi dello scenario in esame: dalla scelta del software più adatto alle modifiche eseguite sul protocollo RPL; per poi passare a come sono state eseguite le simulazioni e la raccolta dei dati; questi vengono poi analizzati, e ,dopo averli correlati tra loro e con le varie situazioni di rete, vengono tratte le conclusioni di questo lavoro e proposti alcuni sviluppi futuri per continuare a lavorare sul tipo di soluzione qui studiata.

4.1 Scelta del software

L'obiettivo di questo lavoro è verificare la possibilità di distribuire le informazioni necessarie al riconoscimento di immagini all'interno di una VSN e valutare le prestazioni di questo tipo di soluzioni.

Per svolgere questo compito è stato scelto un simulatore di reti di sensori che, ovviamente, doveva essere in grado di implementare RPL.

Online si possono trovare varie opzioni Open Source per questo tipo di necessità, quelle principali e più utilizzate sono visibili nella *Tabella 2* [13]:

Simulator	NS-3	Castalia OMNet++	Cooja/MPSim
WSN platforms	n/a	n/a	Tmote Sky, ESB, MicaZ
GUI	MSF, NetAnim, Ns3Generator	topology definition, result analysis and visualization	Friendly GUI
Wireless channel	802.11, YANS, Free space/two-ray/shadowing /small-scale path loss model	Lognormal shadowing, experimentally measured, path loss map, packet reception rates map, temporal variation, unit disk	Multipath ray tracing with support for attenuating for obstacles, unit disk, directed graph
PHY layer	LTE, LAN, Mobility module	CC1100, CC2420	CC2420, TR1001
MAC layer	802.11, CSMA/CA(CD), QoS, mesh, bridge	TMAC, SMAC, Tunable MAC	CSMA/CA, TDMA, X-MAC, LPP, NulMAC, ContikiMAC, SicslowMAC
Network layer	AODV, dsdv, olsr, internet	Simple tree, multi path rings	RPL, AODV
Transport layer	UDP, TCP	None	UDP(not complete), TCP
Energy model	Yes	Yes	Yes

Tabella 2: Elementi chiave dei simulatori RPL disponibili

-**NS-3**: è un simulatore implementato completamente in C++ a differenza delle versioni precedenti, e quindi trae beneficio da molti nuovi utili meccanismi come l'utilizzo di templates e puntatori.

-**OMNet++**: un altro simulatore scritto in C++ basato su eventi discreti utilizzati come modello per le comunicazioni nella rete, multiprocessori e altri sistemi distribuiti o paralleli. Rappresenta un approccio di tipo "framework" e dal suo lancio i suoi modelli di simulazione sono stati sviluppati principalmente per l'ambito delle reti (come per esempio reti wireless, reti ad-hoc, reti di sensori etc.).

-**Cooja**: fornisce un ambiente di simulazione che connette MSPsim (emulatore hardware) ed il sistema operativo Contiki OS. Cooja è in grado di simulare un programma che utilizza Contiki senza alcuna modifica e possiede il modulo ContikiRPL che lo equipaggia di tutti gli strumenti necessari per la maggior parte degli studi riguardanti l'argomento; per questi motivi è uno degli strumenti più utilizzati ed è stato scelto per svolgere le simulazioni che ci interessano.

4.2 Contiki OS e la situazione iniziale

Contiki [14] è un sistema operativo open source progettato per operare nel campo dei sensori e nelle reti LLN.

Dal sito dell'organizzazione è possibile scaricare le varie release a seconda del bisogno, prima di procedere con le simulazioni riguardanti questo lavoro molto tempo è stato speso per cercare la giusta versione da utilizzare poichè, come in molti casi con questo tipo di software, ogniuna ha diversi bug che vanno affrontati e risolti prima di poter operare con soddisfazione.

Ogni versione è installata su una macchina virtuale che ha già al suo interno tutto il necessario per eseguire simulazioni di rete ed analisi basilare dei risultati.

Per i nostri scopi, dopo aver testato le ultime cinque versioni rilasciate, abbiamo scelto l'ultima (versione 2.7) vista la presenza di una nuova e più chiara interfaccia grafica per Cooja e di un minor numero di bug da sistemare prima di poter iniziare a lavorare.

La prima cosa da scegliere in una simulazione è la tipologia di nodi che si vogliono utilizzare, Cooja è in grado di emulare varie tipologie di sensori come: Sky Motes, Z1motes, MicaZ motes etc.(Tabella 3).

MicaZ mote...
Eth1120...
Trxeb1120...
Trxeb2520...
Exp2420 mote (MSP430F5438)...
Exp1101 mote (MSP430F5438)...
Exp1120 mote (MSP430F5438)...
CC430 mote...
Sky mote...
EXP430F5438 mote...
Wismote mote...
Z1 mote...
Sky mote...
ESB mote...

Tabella 3: Tipologie dei Motes Disponibili in Cooja

Noi abbiamo scelto gli Sky Mote (ovvero Tmotes, in *Figura 7*) poichè sono i più utilizzati nella letteratura sulle valutazioni riguardanti RPL.

Contiki fornisce vari esempi di codice da installare sui mote; tra questi abbiamo scelto "udp-rpl" per simulare il tipo di comunicazione presente nel matching distribuito: il codice fa in modo che i nodi della rete instaurino comunicazioni udp con i server-root e, dopo aver mandato un messaggio al server, ricevano una risposta; esattamente come se dovessero mandare un insieme di feature al nodo contenente il database e attendere una risposta con il matching relativo all'immagine che si sta analizzando.

Dopo aver operato le modifiche necessarie per far eseguire le operazioni volute ai nodi abbiamo cominciato con le simulazioni.



Figura 7: Sky mote e relative informazioni all'interno di Cooja

4.3 Simulazioni con le reti di controllo

Dopo la scelta dei nodi e del codice da installare sugli stessi un passaggio importante è quello di capire che tipo di rete simulare, che dimensioni utilizzare e quanti server inserire. In questo lavoro si è scelto di emulare una griglia di nodi che siano a portata di comunicazione radio solo con i vicini nelle 4 direzioni cardinali (nord, sud, est, ovest): un reticolo (vedi Figura 8) nel quale sostituire, in posizioni predeterminate, i server-root verso i quali sarà convogliato il traffico.

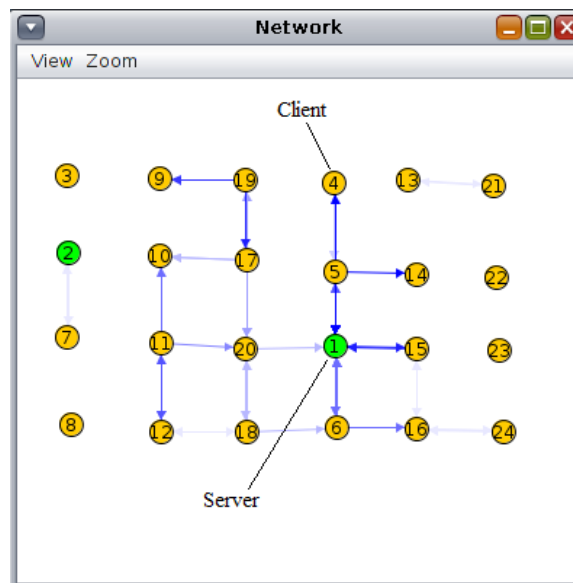


Figura 8: Esempio di topologia di rete utilizzata

Le dimensioni scelte per le reti sono di 24, 28, 38 e 42 nodi ai quali accoppiare 2,3 e 4 server a seconda della dimensione della rete.

4.4 Simulazioni e Raccolta dei Dati

Come già indicato in precedenza le simulazioni sono state svolte utilizzando Cooja all'interno di una virtual machine che operava su di un computer portatile

[Virtual Machine con 1 GB di ram e processore singolo che opera su un computer equipaggiato con Processore Intel Core i3-3120M @ 2.50 Ghz con 6GB di ram].

Le simulazioni svolte per le reti di controllo sono state eseguite con reti da 24 nodi con uno e due server alternati in posizioni centrali e laterali per coprire le varie possibilità; nelle reti da 28 e 38 nodi sono stati inseriti tre server e in quelle da 42 nodi invece quattro.

Tutte queste reti sono state poi simulate con la versione modificata di RPL per utilizzare la nuova funzione obiettivo con coefficiente settato a 50 e 75, ovviamente non avrebbe senso simulare o studiare una situazione in cui la combinazione lineare è completamente sbilanciata verso la percentuale del database perchè ne risulterebbe un protocollo di routing che non tiene minimamente conto delle condizioni della rete risultando in traffico diretto solamente verso il server con la maggiore percentuale di database disponibile.

Le "distribuzioni" di database scelte sono state quelle di 70%-30%, 80%-20% e 90%-10% distribuite nelle varie posizioni predefinite.

I dati vengono raccolti automaticamente da Cooja grazie ad un'applicazione che salva il log degli output dei vari nodi che sono stati modificati per comunicare le informazioni che ci interessano.

I log-files sono poi stati importati in vari fogli elettronici ed organizzati per evidenziare le metriche di interesse per il caso in esame.

4.5 Risultati

Le analisi delle simulazioni preliminari per stabilire dei riferimenti hanno mostrato dati abbastanza intuitivi: come il fatto che il posizionamento dei server all'interno della rete influenza grandemente il ritardo nella VSN.

Per valutare le prestazioni di questa modifica al protocollo di rete abbiamo deciso di concentrarci su due metriche principali quali il ritardo totale medio e il numero di connessioni/comunicazioni andate a buon fine; in seguito abbiamo poi analizzato i dati anche dal punto di vista dell'accuratezza raggiunta nell'analisi delle feature nei vari casi simulati.

4.5.1 Risultati di RPL senza modifica

Come già detto le prime simulazioni ci hanno permesso di stabilire dei dati di riferimento: Sul ritardo:

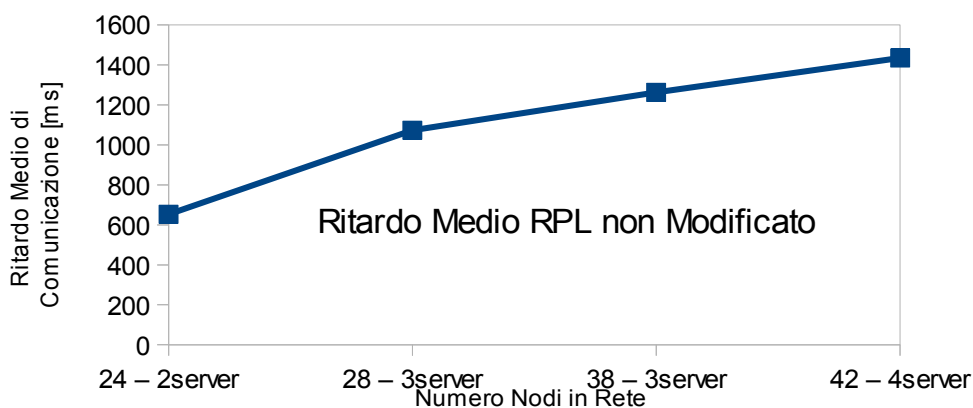


Grafico 1: Ritardo Medio in reti che utilizzano RPL senza modifica

E sulle connessioni:

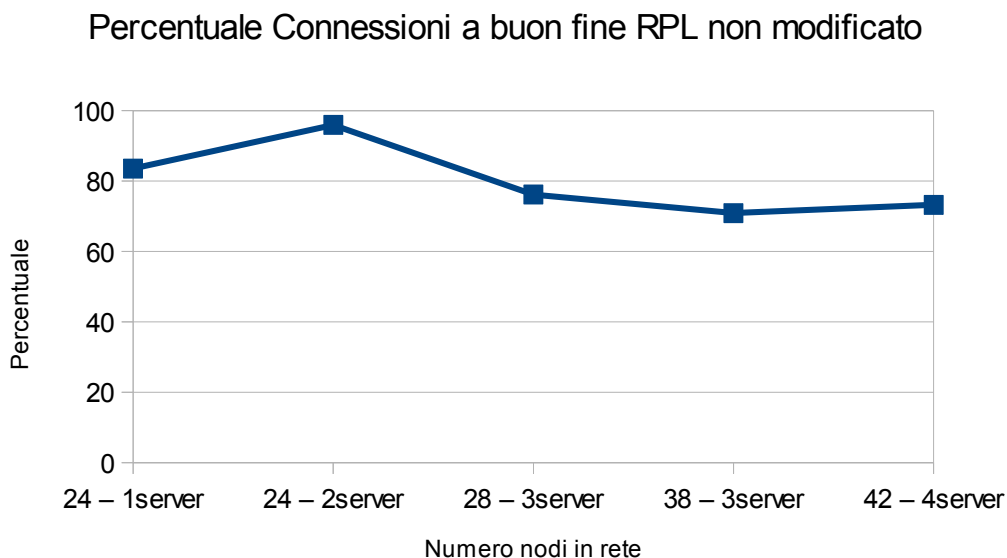


Grafico 2: Percentuale Connessioni a buon fine in reti che utilizzano RPL senza modifica

Si vede chiaramente dai *Grafici 1 e 2* che utilizzando una versione non modificata di RPL abbiamo, senza troppe sorprese, un ritardo che aumenta con il numero di nodi presenti nella rete ed un maggior numero di comunicazioni perse: tutti dati intuitivi e ovviamente derivanti dalla presenza di un maggiore traffico nella rete diretto verso i nodi server-root.

4.5.2 Risultati di RPL modificato

Passiamo adesso ad analizzare i risultati delle simulazioni che utilizzano la versione di RPL che utilizza la nuova funzione obiettivo "ibrida" con coefficiente impostato a 50 e 75.

4.5.2.1 Connessioni totali

La prima metrica presa in esame è quella delle comunicazioni totali andate a buon fine, e, di conseguenza, quelle perse. Questa metrica è importante perchè ci da un'idea del consumo energetico necessario per comunicare con il server da parte dei nodi poichè, visto

che la modifica inserita non influenza il costo energetico dal punto di vista della memoria o della cpu, l'incremento o decremento del numero delle trasmissioni è l'unico vero nuovo impatto sul consumo in questa situazione.

I dati sono divisi per numero di nodi (e quindi server abbinati), percentuali utilizzate nei database nei root ed infine per coefficiente utilizzato.

Qui di seguito due grafici di esempio e poi un grafico con tutti i dati visualizzati insieme:

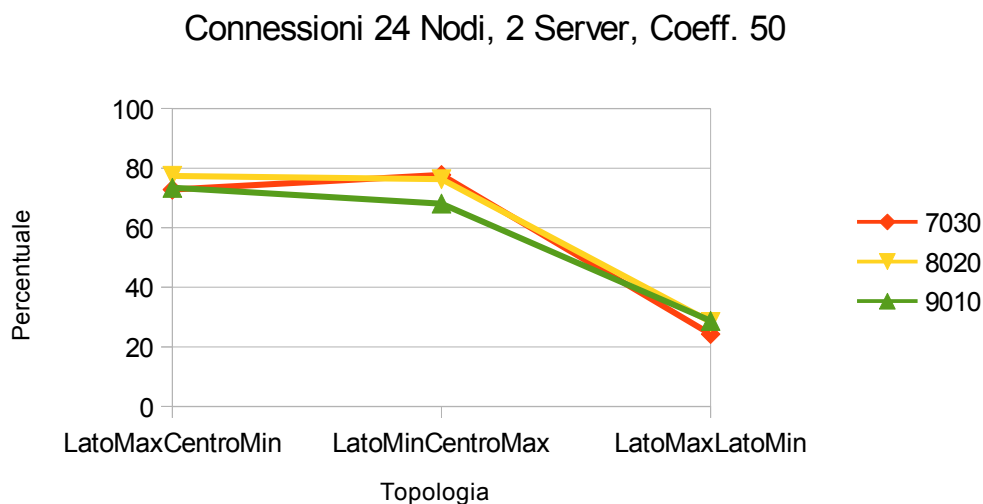


Grafico 3: Percentali Delle Connessioni a buon fine in reti con 24 Nodi, 2 Server e Coefficiente settato a 50

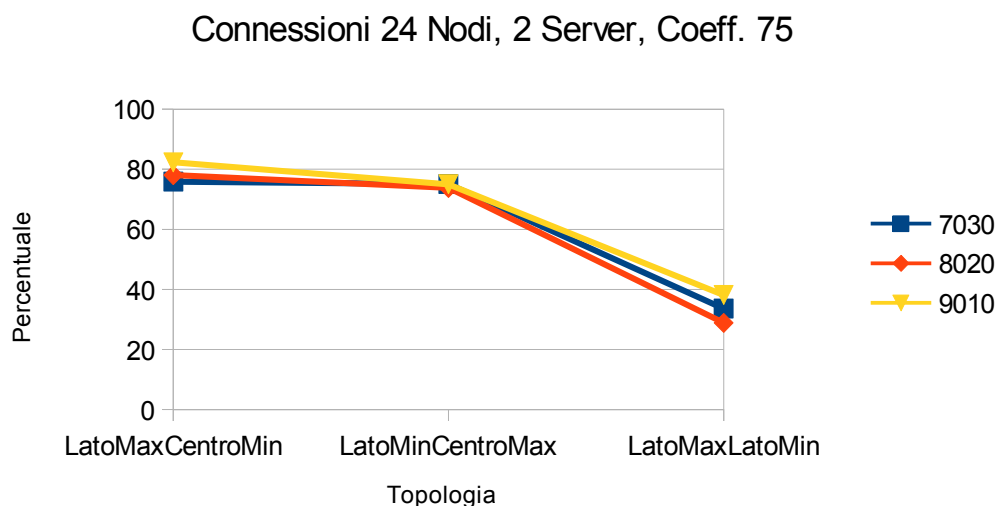


Grafico 4: Percentali Delle Connessioni a buon fine in reti con 24 Nodi, 2 Server e Coefficiente settato a 75

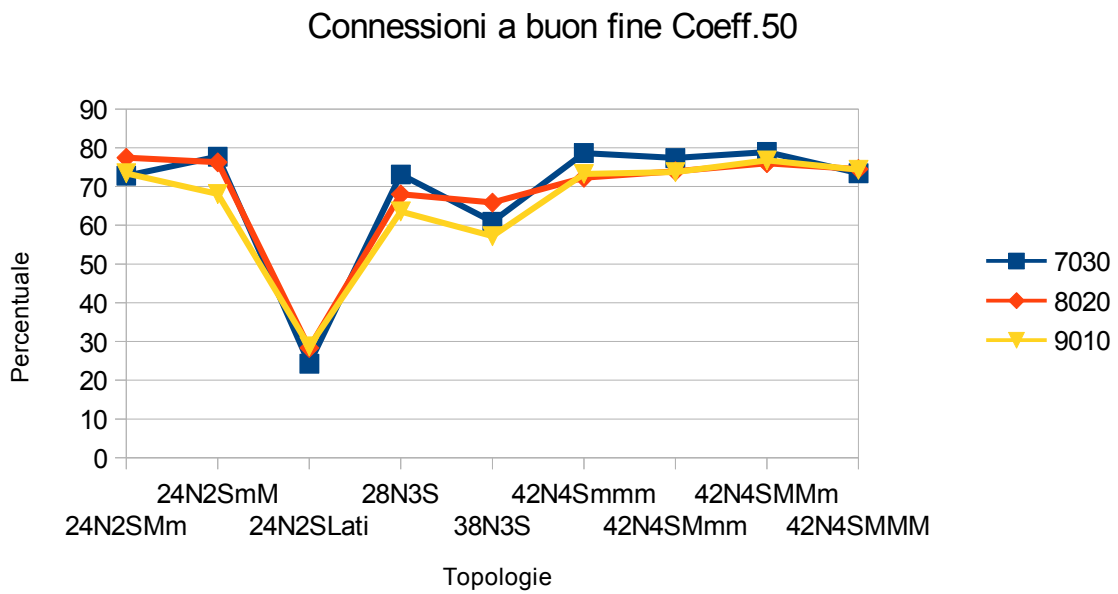


Grafico 5: Percentuali delle Connessioni a buon fine nelle diverse topologie con Coeff. 50

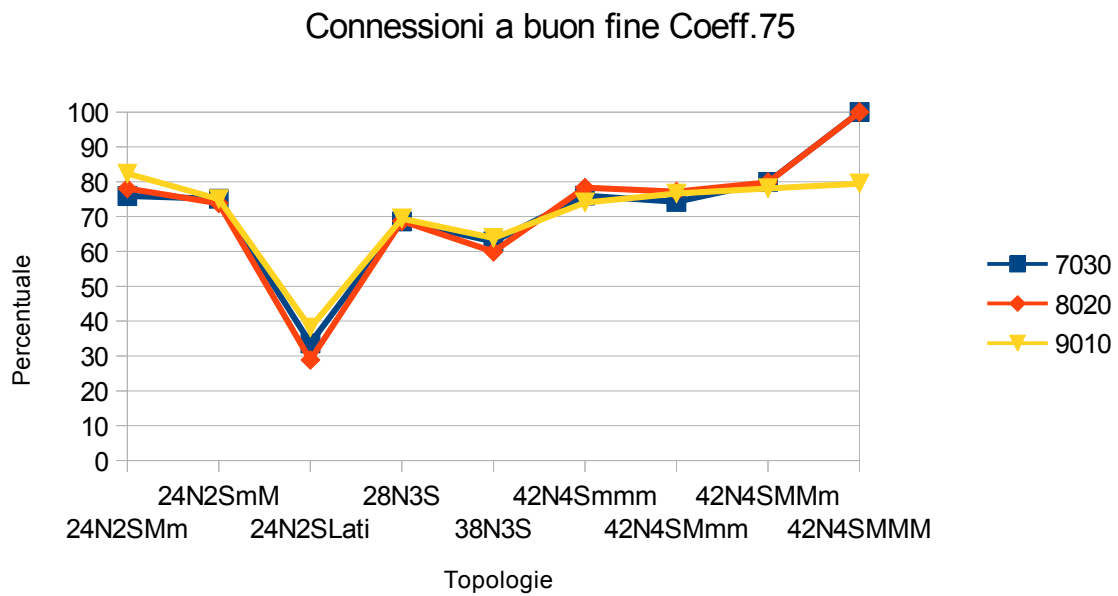


Grafico 6: Percentuali delle Connessioni a buon fine nelle diverse topologie con Coeff. 75

TOPOLOGIE:		
	# Nodi	# Server
24N2SMm	24	2
24N2SmM	24	2
24N2SLati	24	2
28N3S	28	3
38N3S	38	3
42N4Smmm	42	4
42N4SMmm	42	4
42N4SMMm	42	4
42N4SMMM	42	4

Tabella 4: Legenda dei grafici per riconoscere le topologie

Dove "M" significa che nella posizione più a sinistra è installata la Maggiore parte del database mentre "m" indica ovviamente la minima parte; in questo modo possiamo distinguere le varie topologie nei vari grafici.

Dai dati rappresentati nei *Grafici 5 e 6* possiamo estrapolare varie informazioni: notiamo subito come, nei vari casi simulati con due server, posizionare le radici dei dodag ai lati/limiti della rete causi un crollo vero e proprio delle comunicazioni andate a buon fine; mentre possiamo notare come l'aumento dei nodi e dei server in rete garantisca un miglioramento dal punto di vista delle connessioni fino ad arrivare ad un 26% nel caso migliore possibile.

Altro elemento importante degno di nota è come la situazione inizi ad essere vantaggiosa quando si passa da reti di 38 nodi a quelle con 42 e più, e come, tra le varie percentuali utilizzate in rete la divisione 80% e 20% del database nelle varie distribuzioni possibili dia quasi sempre dei risultati migliori rispetto alle altre.

Per visualizzare ancora meglio l'impatto della nuova funzione obiettivo possiamo mettere in relazione i dati ottenuti con quelli di riferimento e visualizzare solo la differenza di questi due nel *Grafico 7*:

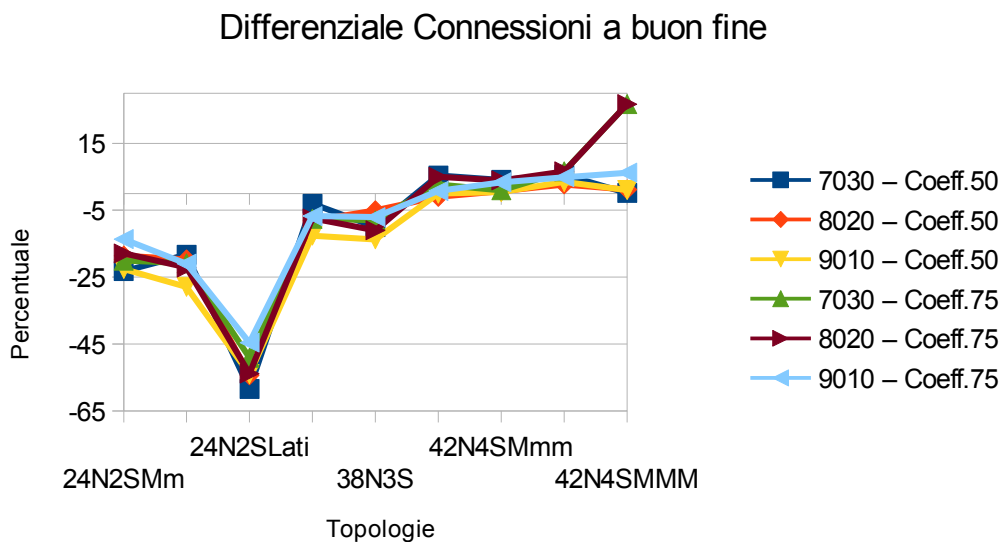


Grafico 7: Differenziale delle percentuali delle Connessioni a buon fine nelle reti con modifica rispetto a quelle di riferimento

Delta – Connessioni						
	24N2SMm	24N2SmM	24N2SLati	28N3S	38N3S	
7030 – Coeff.50	-23,2	-18,28	-58,45	-3,08	-10,01	
8020 – Coeff.50	-18,56	-19,7	-54,35	-8,18	-5,03	
9010 – Coeff.50	-22,56	-27,89	-53,99	-12,61	-13,72	
7030 – Coeff.75	-20,05	-20,85	-49,07	-7,55	-8,04	
8020 – Coeff.75	-17,9	-22,14	-53,85	-7,47	-10,98	
9010 – Coeff.75	-13,63	-20,96	-44,57	-6,74	-7,04	
	42N4Smmm	42N4SMmm	42N4SMMMm	42N4SMMM		
7030 – Coeff.50	5,37	4,08	5,64	0,17		
8020 – Coeff.50	-1,01	0,68	2,72	1,22		Dati
9010 – Coeff.50	-0,03	0,45	3,47	1,04		Percentuali
7030 – Coeff.75	2,74	0,94	6,66	26,74		
8020 – Coeff.75	5	3,91	6,61	26,74		
9010 – Coeff.75	0,76	3,36	4,86	6,2		

Tabella 5: Differenziale delle percentuali delle Connessioni a buon fine nelle reti con modifica rispetto a quelle di riferimento

4.5.2.2 Ritardo Medio

I dati sul ritardo medio in rete rispecchiano le osservazioni fatte sulle connessioni come possiamo vedere di seguito nel *Grafico 8*:

Di nuovo, visualizziamo la differenza percentuale del ritardo in rete con quello di riferimento per avere una visuale di insieme più immediata:

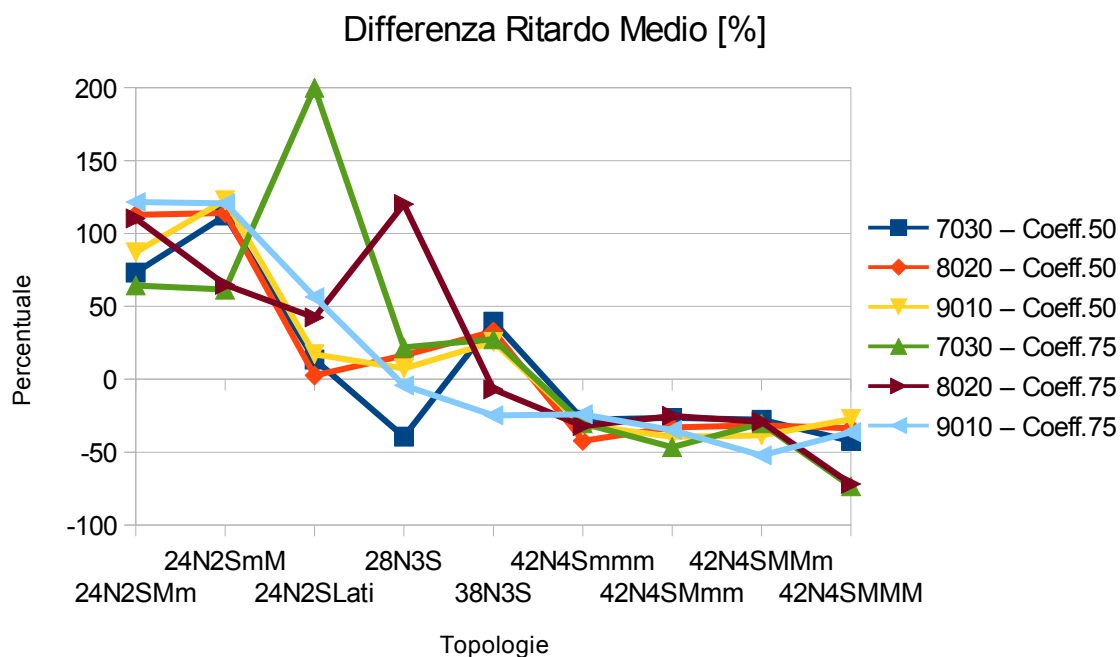


Grafico 8: Differenziale del Ritardo Medio in rete Rispetto ai dati di Riferimento

Delta - Ritardo					
	24N2SMm	24N2SmM	24N2SLati	28N3S	38N3S
7030 - Coeff. 50	73,15	112,25	13,56	-39,45	39,49
8020 - Coeff. 50	112,72	113,97	2,56	16,42	32,74
9010 - Coeff. 50	87,05	122,75	17,12	7,44	25,57
7030 - Coeff. 75	64,35	61,5	200	21,75	27,41
8020 - Coeff. 75	110,3	64,78	42,41	120,13	-6,96
9010 - Coeff. 75	121,56	120,65	56,42	-4,22	-24,78
	42N4Smmm	42N4SMmm	42N4SMMMm	42N4SMMM	
7030 - Coeff. 50	-27,76	-26,46	-27,76	-42,41	Dati Percentuali
8020 - Coeff. 50	-42,26	-33,15	-31,63	-33,56	
9010 - Coeff. 50	-31,52	-39,57	-38,58	-27,35	
7030 - Coeff. 75	-30,25	-46,53	-30,28	-73,53	
8020 - Coeff. 75	-32,11	-25,11	-29,19	-71,84	
9010 - Coeff. 75	-24,05	-34,81	-52,19	-36,32	

Tabella 6: Differenziale del Ritardo Medio in rete Rispetto ai dati di Riferimento

Notiamo come il ritardo sia inizialmente più alto rispetto ai dati di riferimento per quanto riguarda le simulazioni con due server ma quando i numeri dei nodi e dei server cominciano a salire il ritardo medio comincia a scendere e ad essere avvantaggiato dalla modifica ancora prima delle "connessioni".

In questo caso, al contrario della metrica precedente, possiamo notare una netta differenza nell'impatto del coefficiente nella funzione obiettivo sui risultati finali e troviamo i casi migliori nelle reti di 42 nodi con coefficiente fissato a 75 utilizzato con le distribuzioni da 80/20 e 70/30.

4.5.2.3 Accuratezza

Con il termine "Accuratezza" si intende tenere conto della percentuale di informazione, e quindi di database, raggiunta da ogni nodo in una comunicazione: se queste simulazioni devono rispecchiare una VSN che realizza riconoscimento di immagini, raggiungere un nodo che possiede un 30% del database significherà ottenere un risultato che potrà essere accurato al 30% rispetto ad un riconoscimento operato sulla totalità del database.

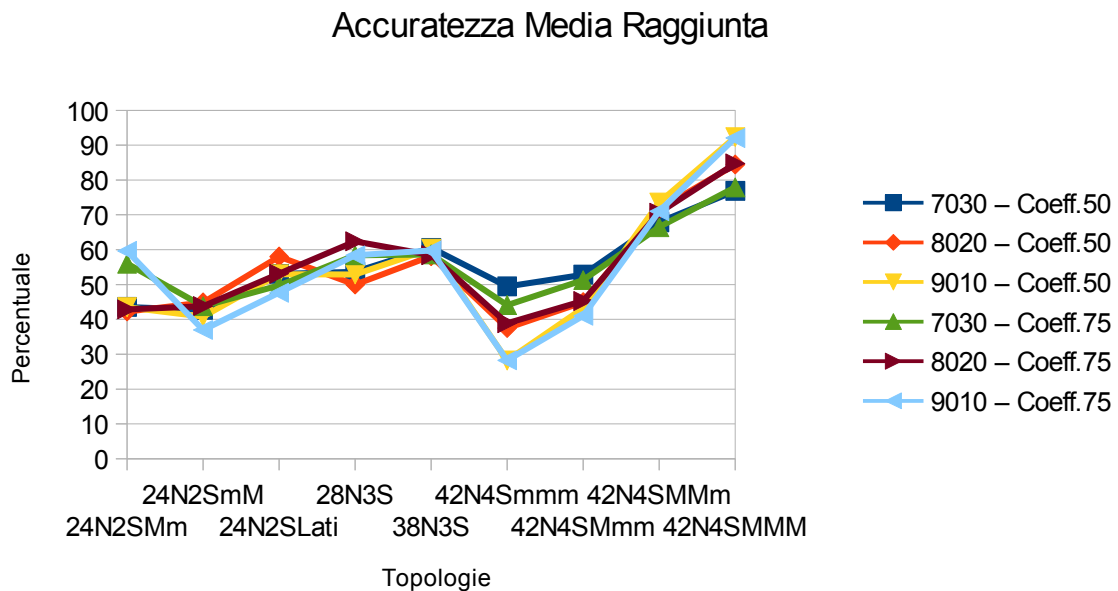


Grafico 9: Accuratezza media raggiunta dalle connessioni nelle varie topologie di rete

Accuratezza Media					
	24N2SMm	24N2SmM	24N2SLati	28N3S	38N3S
7030 – Coeff.50	43,58	42,71	53,04	53,68	60,53
8020 – Coeff.50	42,27	44,86	58,03	50	58,13
9010 – Coeff.50	43,49	40,77	52,96	52,89	60,25
7030 – Coeff.75	56	43,78	49,78	58,4	58,63
8020 – Coeff.75	42,87	43,79	53,08	62,41	58,44
9010 – Coeff.75	59,64	36,95	47,65	58,37	59,74
	42N4Smmm	42N4SMmm	42N4SMMM	42N4SMMM	
7030 – Coeff.50	49,47	52,84	67,88	76,89	Dati Percentuali
8020 – Coeff.50	37,5	44,95	71,66	84,43	
9010 – Coeff.50	28,21	43,24	73,55	92,3	
7030 – Coeff.75	44,09	51,28	66,49	77,89	
8020 – Coeff.75	38,91	45,27	70,63	84,73	
9010 – Coeff.75	28,23	41,05	71,05	92,05	

Tabella 7: Accuratezza media raggiunta dalle connessioni nelle varie topologie di rete

Oltre all'accuratezza media raggiunta in rete rappresentata nel *Grafico 9* è interessante osservare i dati sulla distribuzione specifica all'interno delle varie topologie di rete (*Tabelle da 8 a 13*):

Distribuzione Accuratezza			
7030 – Coeff.50	100,00%	70,00%	30,00%
24N2SMm		33,94	66,06
24N2SmM		31,78	68,22
24N2SLati		57,61	42,39
28N3S	12,42	37,47	50,11
38N3S	24,44	33,55	42,01
42N4Smmm	23,53	0	76,47
42N4SMmm	24,83	13,66	61,52
42N4SMMM	22,72	54,94	22,35
42N4SMMM	22,97	77,03	0

Tabella 8: Distribuzione Accuratezza 7030 Coeff.50

Distribuzione Accuratezza			
8020 – Coeff.50	100,00%	80,00%	20,00%
24N2SMm		37,11	62,89
24N2SmM		41,44	58,56
24N2SLati		63,38	36,62
28N3S	9,38	37,5	53,13
38N3S	22,9	33,02	44,08
42N4Smmm	21,87	0	78,13
42N4SMmm	21,28	13,21	65,51
42N4SMMM	22,94	55,52	21,54
42N4SMMM	22,17	77,83	0

Tabella 9: Distribuzione Accuratezza 8020 Coeff.50

Distribuzione Accuratezza			
9010 – Coeff.50	100,00%	90,00%	10,00%
24N2SMm		41,86	58,14
24N2SmM		38,46	61,54
24N2SLati		53,7	46,3
28N3S	14,93	36,82	48,26
38N3S	24,42	35,33	40,25
42N4Smmm	20,23	0	79,77
42N4SMmm	22,54	16,2	61,27
42N4SMMM	19,94	57	23,06
42N4SMMM	22,97	77,03	0

Tabella 10: Distribuzione Accuratezza 9010 Coeff.50

Distribuzione Accuratezza			
7030 – Coeff.75	100,00%	70,00%	30,00%
24N2SMm		65	35
24N2SmM		34,45	65,55
24N2SLati		49,44	50,56
28N3S	10,05	53,42	36,53
38N3S	21,73	33,54	44,72
42N4Smmm	20,13	0	79,87
42N4SMmm	22,61	13,62	63,77
42N4SMMm	23,51	50,07	26,42
42N4SMMM	26,28	73,72	0

Tabella 11: Distribuzione Accuratezza 7030 Coeff.75

Distribuzione Accuratezza			
8020 – Coeff.75	100,00%	80,00%	20,00%
24N2SMm		38,12	61,88
24N2SmM		39,64	60,36
24N2SLati		55,13	44,87
28N3S	16,7	48,42	34,88
38N3S	24,31	31,65	44,04
42N4Smmm	23,64	0	76,36
42N4SMmm	23,21	11,17	65,62
42N4SMMm	21,59	55,6	22,81
42N4SMMM	23,65	76,35	0

Tabella 12: Distribuzione Accuratezza 8020 Coeff.75

Distribuzione Accuratezza			
9010 – Coeff.75	100,00%	90,00%	10,00%
24N2SMm		62,05	37,95
24N2SmM		33,68	66,32
24N2SLati		47,06	52,94
28N3S	15,83	42,66	41,51
38N3S	25,95	32,98	41,07
42N4Smmm	20,25	0	79,75
42N4SMmm	22,69	13,29	64,02
42N4SMMm	26,15	46,9	26,95
42N4SMMM	20,49	79,51	0

Tabella 13: Distribuzione Accuratezza 9010 Coeff.75

Analizzando questi dati vediamo come in reti più piccole con soli due server possiamo raggiungere un'accuratezza maggiore impostando il coefficiente a 75 e installando la maggior parte del database in posizione centrale; quando invece cominciamo a spostare l'attenzione su reti di dimensioni più importanti con più server e con un server esterno che possiede tutto il database notiamo come, in realtà, l'accuratezza del 100% non dipenda dal coefficiente utilizzato ma sia più o meno stabile e ovviamente dovuta al posizionamento dei nodi attorno al server "centrale", ma se ci spostiamo in altre parti della rete e controlliamo altri valori di accuratezza possiamo notare come questi dipendano in realtà dal ritardo in rete.

4.5.3 Analisi Ritardo-Accuratezza (l'impatto della funzione obbiettivo)

Vale quindi la pena prendere in esame la relazione Ritardo Medio – Accuratezza per capire come questa cambia a seconda del tipo di rete, del numero dei server e dei coefficienti in gioco.

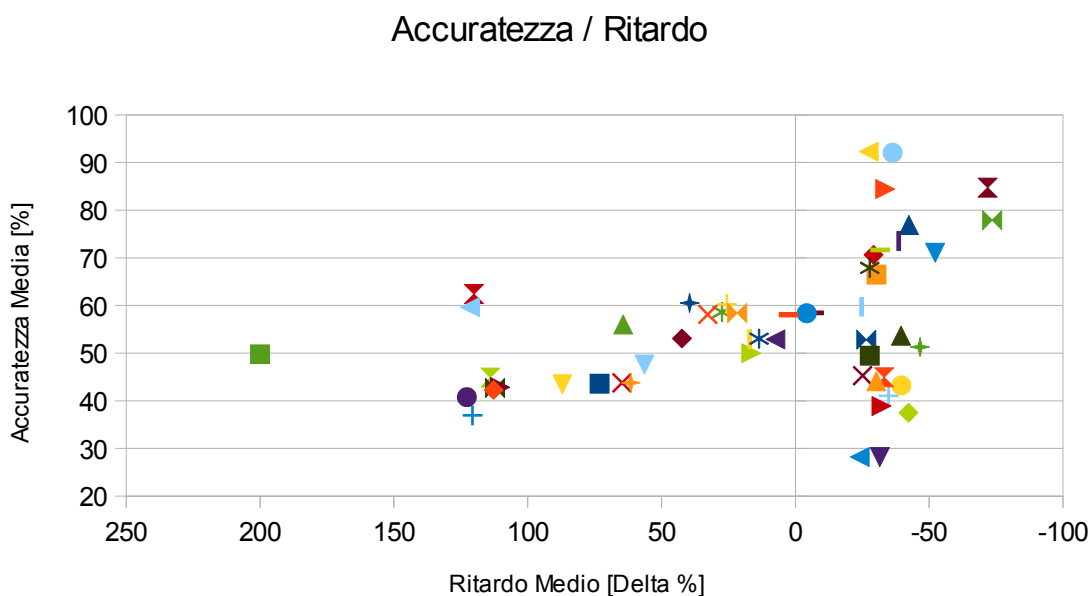


Grafico 10: Accuratezza Raggiunta in relazione Al Ritardo Medio in rete nelle varie topologie analizzate

Ogni punto nel *Grafico 10* rappresenta una determinata topologia di rete, per semplicità viene omessa la legenda poichè l'unico dato interessante sarebbe notare la distribuzione rispetto al coefficiente ma da questo punto di vista non c'è reale differenziazione tra i punti.

Dal grafico si nota subito come l'accuratezza raggiunta dipenda più dal ritardo totale sviluppato nella rete che dai coefficienti utilizzati; infatti con ogni distribuzione possibile del database per avere dati alti di accuratezza è necessario avere un guadagno notevole sul ritardo in rete.

5 - Conclusioni

Questa analisi ci permette di trarre varie conclusioni su questo tipo di scenario.

Come in ogni altro tipo di rete anche in questo caso all'aumentare dei nodi aumenta ovviamente anche il traffico e quindi il ritardo sperimentato dalle comunicazioni andate a buon fine mentre quelle fallite aumentano di numero perchè i server o la rete stessa non riescono a gestire le richieste del traffico.

La cosa interessante da questo punto di vista è che nel momento in cui implementiamo una funzione obiettivo "arricchita" da una nuova metrica come nel caso in esame, al crescere della rete aumentano anche i vantaggi per tutte le metriche analizzate; sembrerebbe che un dato aggiuntivo preso in esame nella funzione obiettivo aiuti ad organizzare meglio il traffico in rete mitigando gli effetti della pura quantità di richieste.

Dal punto di vista energetico il cambiamento implementato si dimostra vantaggioso non tanto dal punto di vista computazionale, per il quale è di fatto indifferente, ma per il guadagno nelle comunicazioni andate a buon fine che di conseguenza abbassano notevolmente la necessità di ritrasmissioni che ovviamente utilizzerebbero energia aggiuntiva.

In generale vediamo quindi che la nuova funzione obiettivo ha un impatto positivo su tutto lo spettro ed influenza maggiormente il ritardo ed il numero di connessioni nelle quali all'aumentare del coefficiente migliora anche la situazione; mentre per quanto riguarda l'accuratezza possiamo dire che la funzione obiettivo abbia un impatto secondario poichè il vero guadagno viene dalla drastica riduzione del ritardo medio in rete.

Concludendo possiamo sicuramente affermare che questo tipo di soluzione si è rivelata vantaggiosa per il caso in esame e potrebbe essere implementata facilmente e con successo.

L'ideale passo successivo per questo tipo di soluzione sarà sicuramente eseguire maggiori simulazioni studiando a fondo tutto il range dei coefficienti, possibilmente applicati anche

ad altre tipologie di reti e con una più vasta quantità di nodi e server per testare i limiti della scalabilità di questo tipo di modifica alla funzione obiettivo.

Sarebbe ulteriormente interessante eseguire dei test che utilizzino funzioni obiettivo con coefficienti che si evolvono con i bisogni della rete stessa e tengono conto di soluzioni basate sulla qualità del servizio (QOS): potrebbe per esempio esserci un bisogno di riconoscimenti con priorità alta dal punto di vista del ritardo o dell'accuratezza e quindi di conseguenza aggiustare i coefficienti all'interno della rete per favorire un certo tipo di bilanciamento.

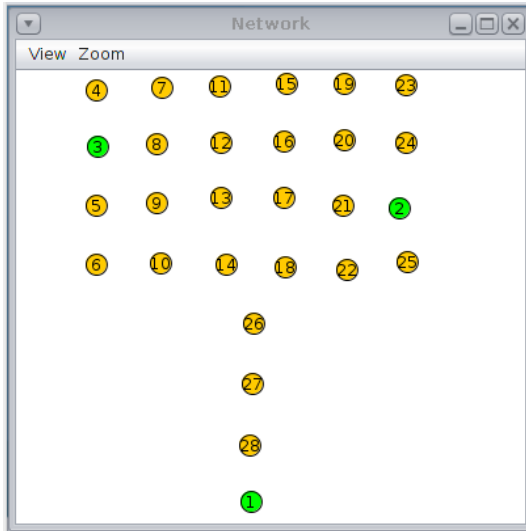
Bibliografia

- [1] Stanislava Soro, Wendi Heinzelman, "A Survey of Visual Sensor Networks", *Advances in Multimedia Volume 2009*, Article ID 640386.
- [2] Youssef Charfi, Bell Canada, Naoki Wakamiya, Masayuki Murata, "Challenging Issues In Visual Sensor Networks", *IEEE Wireless Communications* - April 2009.
- [3] Ian F. Akyildiz, Tommaso Melodia, Kaushik R. Chowdhury, "A survey on wireless multimedia sensor networks", I.F. Akyildiz et al. / *Computer Networks* 51 (2007) 921–960.
- [4] Theodore Zahariadist, Kanellina Petrakou², Stamatis Voliotis², "Enabling QoS in Visual Sensor Networks", 48th International Symposium ELMAR-2006, 07-09 June 2006, Zadar, Croatia.
- [5] K. Obraczka, R. Manduchi, J. J. Garcia-Luna-Aveces, "Managing the Information Flow in Visual Sensor Networks", 0-7803-7442-8102 @ 2002 IEEE.
- [6] Alessandro Redondi, "Analisi visuale energeticamente efficiente per reti di sensori wireless multimediali", Tesi di dottorato, Dipartimento di Elettronica, Informazione e Bioingegneria al Politecnico di Milano XXVI ciclo.
- [7] David G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999, pp. 1150–1157.
- [8] Herbert Bay, Tinne Tuytelaars, and Luc J. Van Gool, "Surf: Speeded up robust features," in *ECCV* (1), 2006, pp. 404–417.
- [9] Stefan Leutenegger, Margarita Chli, and Roland Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *ICCV*, 2011, pp. 2548–2555.

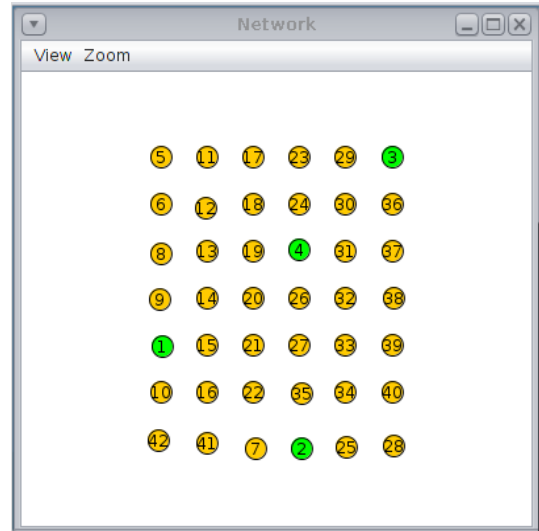
- [10] T. Winter, Ed., P. Thubert, Ed., A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", Internet Engineering Task Force (IETF) Request for Comments: 6550 March 2012.
- [11] P. Thubert, Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", Internet Engineering Task Force (IETF) Request for Comments: 6552 March 2012.
- [12] Olfa Gaddour, Anis Koubâa, "RPL in a nutshell: A survey", Survey Paper Computer Networks 56 (2012) 3163–3178.
- [13] Leila Ben Saad, Cedric Chauvenet, Bernard Tourancheau, "Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: two cases studies", Author manuscript, in "International Conference on Sensor Technologies and Applications SENSORCOMM 2011 (2011)".
- [14] Contiki OS disponibile all'indirizzo: <http://www.contiki-os.org/>

Appendice

Esempi Topologie di Rete:



Topologia di rete - 28 Nodi



Topologia di rete - 42 Nodi

Esempio Dati Raccolti:

```
Applications Places
ReticoloP35100802038Tot75.txt (~/Desktop/ReticoloP) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Cut Copy Paste Find
ReticoloP35100802038Tot75.txt
00:00.521 ID:6 Rime started with address 0.18.116.6.0.6.6.6
00:00.522 ID:35 fe80::212:7423:23:2323
00:00.523 ID:30 MAC 00:12:74:1e:00:1e:1e:1e Contiki 2.7 started. Node id is set to 30.
00:00.528 ID:35 Created a connection with the server :: local/remote port 8765/5678
00:00.529 ID:6 MAC 00:12:74:06:00:06:06:06 Contiki 2.7 started. Node id is set to 6.
00:00.529 ID:30 CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26
00:00.531 ID:2 Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7402:0002:0202
00:00.533 ID:2 Starting 'UDP server process'
00:00.535 ID:2 UDP server started
00:00.535 ID:6 CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26
00:00.538 ID:2 created a new RPL dag
00:00.541 ID:30 Tentative link-local IPv6 address fe80:0000:0000:0000:0212:741e:001e:1e1e
00:00.543 ID:30 Starting 'UDP client process'
00:00.543 ID:2 Server IPv6 addresses: aaaa::212:7402:2:202
00:00.545 ID:2 aaaa::ff:fe00:1
00:00.545 ID:30 UDP client process started
00:00.546 ID:6 Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7406:0006:6066
00:00.548 ID:2 fe80::212:7402:2:202
00:00.548 ID:6 Starting 'UDP client process'
00:00.550 ID:30 Client IPv6 addresses: aaaa::212:741e:1e:1e1e
00:00.550 ID:6 UDP client process started
00:00.553 ID:30 fe80::212:741e:1e:1e1e
00:00.554 ID:2 Created a server connection with remote address :: local/remote port 5678/8765
00:00.555 ID:6 client IPv6 addresses: aaaa::212:7406:6:606
00:00.558 ID:6 fe80::212:7406:6:606
00:00.558 ID:30 Created a connection with the server :: local/remote port 8765/5678
00:00.563 ID:6 Created a connection with the server :: local/remote port 8765/5678
Plain Text Tab Width: 8 Ln 7, Col 98 INS
```

Processo di Inizializzazione Nodi

```

Applications Places
ReticoloP35100802038Tot75.txt (~/Desktop/ReticoloP) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
ReticoloP35100802038Tot75.txt
03:21.503 ID:7 DATA send to 1 'Hello 3'
03:21.609 ID:28 DATA send to 1 'Hello 3'
03:21.780 ID:2 DATA rcv 'Hello 3 from the client' from 7
03:21.782 ID:2 DATA sending reply
03:22.064 ID:7 DATA rcv 'Reply'
03:22.615 ID:8 DATA rcv 'Reply'
03:24.300 ID:24 DATA send to 1 'Hello 3'
03:24.460 ID:34 DATA send to 1 'Hello 3'
03:24.616 ID:1 DATA rcv 'Hello 3 from the client' from 34
03:24.617 ID:1 DATA sending reply
03:24.816 ID:34 DATA rcv 'Reply'
03:27.580 ID:3 DATA rcv 'Hello 3 from the client' from 28
03:27.582 ID:3 DATA sending reply
03:28.117 ID:2 DATA rcv 'Hello 3 from the client' from 24
03:28.118 ID:2 DATA sending reply
03:29.265 ID:31 DATA send to 1 'Hello 3'
03:29.952 ID:1 DATA rcv 'Hello 3 from the client' from 31
03:29.954 ID:1 DATA sending reply
03:30.688 ID:13 DATA send to 1 'Hello 3'
03:31.386 ID:37 DATA send to 1 'Hello 3'
03:31.490 ID:2 DATA rcv 'Hello 3 from the client' from 13
03:31.491 ID:2 DATA sending reply
03:31.506 ID:1 DATA rcv 'Hello 3 from the client' from 37
03:31.507 ID:1 DATA sending reply
03:31.675 ID:13 DATA rcv 'Reply'
03:31.779 ID:38 DATA send to 1 'Hello 3'
03:32.133 ID:1 DATA rcv 'Hello 3 from the client' from 38
Plain Text Tab Width: 8 Ln 7, Col 98 INS

```

Comunicazione tra i Nodi -1

```

Applications Places
ReticoloP35100802038Tot75.txt (~/Desktop/ReticoloP) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
ReticoloP35100802038Tot75.txt
15:13.464 ID:34 DATA send to 1 'Hello 15'
15:15.480 ID:5 DATA send to 1 'Hello 15'
15:15.627 ID:3 DATA rcv 'Hello 15 from the client' from 5
15:15.629 ID:3 DATA sending reply
15:15.755 ID:5 DATA rcv 'Reply'
15:16.909 ID:3 DATA rcv 'Hello 15 from the client' from 35
15:16.910 ID:3 DATA sending reply
15:16.919 ID:3 DATA rcv 'Hello 15 from the client' from 34
15:16.921 ID:3 DATA sending reply
15:19.237 ID:24 DATA send to 1 'Hello 15'
15:19.435 ID:38 DATA send to 1 'Hello 15'
15:19.557 ID:1 DATA rcv 'Hello 15 from the client' from 38
15:19.559 ID:1 DATA sending reply
15:19.859 ID:21 DATA send to 1 'Hello 15'
15:19.954 ID:38 DATA rcv 'Reply'
15:20.027 ID:37 DATA send to 1 'Hello 15'
15:20.056 ID:1 DATA rcv 'Hello 15 from the client' from 37
15:20.057 ID:1 DATA sending reply
15:20.140 ID:37 DATA rcv 'Reply'
15:20.198 ID:33 DATA send to 1 'Hello 15'
15:21.227 ID:3 DATA rcv 'Hello 15 from the client' from 21
15:21.228 ID:3 DATA sending reply
15:21.674 ID:1 DATA rcv 'Hello 15 from the client' from 33
15:21.675 ID:1 DATA sending reply
15:21.699 ID:21 DATA rcv 'Reply'
15:22.838 ID:7 DATA send to 1 'Hello 15'
15:25.254 ID:3 DATA rcv 'Hello 15 from the client' from 7
Plain Text Tab Width: 8 Ln 7, Col 98 INS

```

Comunicazione tra i Nodi -2

Dati Completi:

Conessioni:

Riferimento

	Iniziate	ArrivateServer	Concluse	Pserver	Pconcluse
Reticolo1sCentrale24Tot	158	157	132	99,37	83,54
Reticolo1sLato24tot	145	140	110	96,55	75,86
2s					
Reticolo2sCentroLato24tot	322	322	309	100	95,96
Reticolo2sLatoLato24tot	451	375	373	83,15	82,71
3s					
Reticolo2s1sLatoLatoAlto28Tot	642	536	489	83,49	76,17
3s					
Reticolo2s1sLatoLatoAlto38Tot	299	286	212	95,65	70,9
4s					
Reticolo6x7s4tot42	359	344	263	95,82	73,26

2s

	Iniziate	ArrivateServer	Concluse	Pserver	Pconcluse	DeltaPConclus
50						
7030						
Lato30Centro7024Tot507030	246	218	179	88,62	72,76	-23,2
Lato70Centro3024Tot507030	233	214	181	91,85	77,68	-18,28
LatoLato24Tot507030	272	92	66	33,82	24,26	-58,45
8020						
Lato20Centro8024Tot508020	208	194	161	93,27	77,4	-18,56
Lato80Centro2024Tot508020	198	181	151	91,41	76,26	-19,7
LatoLato24Tot508020	201	71	57	35,32	28,36	-54,35
9010						
Lato10Centro9024Tot509010	203	172	149	84,73	73,4	-22,56
Lato90Centro1024Tot509010	238	208	162	87,39	68,07	-27,89
LatoLato24Tot509010	282	108	81	38,3	28,72	-53,99

75

	Iniziate	ArrivateServer	Concluse	Pserver	Pconcluse	DeltaPConclus
7030						
Lato30Centro7024Tot757030	220	200	167	90,91	75,91	-20,05
Lato70Centro3024Tot757030	229	209	172	91,27	75,11	-20,85
LatoLato24Tot757030	220	89	74	40,45	33,64	-49,07
8020						
Lato20Centro8024Tot758020	196	181	153	92,35	78,06	-17,9
Lato80Centro2024Tot758020	191	169	141	88,48	73,82	-22,14
LatoLato24Tot758020	201	77	58	38,31	28,86	-53,85
9010						
Lato10Centro9024Tot759010	215	195	177	90,7	82,33	-13,63
Lato90Centro1024Tot759010	200	190	150	95	75	-20,96
LatoLato24Tot759010	215	102	82	47,44	38,14	-44,57

3s

		50					
		Iniziate	ArrivateServer	Conclude	Pserver	Pconclude	DeltaPConclus
		7030					
ReticoloP3S100703028Tot50		498	443	364	88,96	73,09	-3,08
ReticoloP3S100703038Tot50		716	626	436	87,43	60,89	-10,01
		8020					
ReticoloP3S100802028Tot50		478	416	325	87,03	67,99	-8,18
ReticoloP3S100802038Tot50		715	642	471	89,79	65,87	-5,03
		9010					
ReticoloP3S100901028Tot50		494	402	314	81,38	63,56	-12,61
ReticoloP3S100901038Tot50		766	651	438	84,99	57,18	-13,72

		75					
		Iniziate	ArrivateServer	Conclude	Pserver	Pconclude	DeltaPConclus
		7030					
ReticoloP3S100703028Tot75		494	438	339	88,66	68,62	-7,55
ReticoloP3S100703038Tot75		727	635	457	87,35	62,86	-8,04
		8020					
ReticoloP3S100802028Tot75		575	539	395	93,74	68,7	-7,47
ReticoloP3S100802038Tot75		746	654	447	87,67	59,92	-10,98
		9010					
ReticoloP3S100901028Tot75		494	436	343	88,26	69,43	-6,74
ReticoloP3S100901038Tot75		736	655	470	88,99	63,86	-7,04

4s

		50					
		Iniziate	ArrivateServer	Conclude	Pserver	Pconclude	DeltaPConclus
		7030					
ReticoloP4S10030303042Tot50		744	697	585	93,68	78,63	5,37
ReticoloP4S10070303042Tot50		759	725	587	95,52	77,34	4,08
ReticoloP4S10070703042Tot50		858	810	677	94,41	78,9	5,64
ReticoloP4S10070707042Tot50		798	753	586	94,36	73,43	0,17
		8020					
ReticoloP4S10020202042Tot50		818	759	591	92,79	72,25	-1,01
ReticoloP4S10080202042Tot50		756	719	559	95,11	73,94	0,68
ReticoloP4S10080802042Tot50		762	715	579	93,83	75,98	2,72
ReticoloP4S10080808042Tot50		862	812	642	94,2	74,48	1,22
		9010					
ReticoloP4S10010101042Tot50		747	692	547	92,64	73,23	-0,03
ReticoloP4S10090101042Tot50		757	710	558	93,79	73,71	0,45
ReticoloP4S10090901042Tot50		752	707	577	94,02	76,73	3,47
ReticoloP4S10090909042Tot50		755	714	561	94,57	74,3	1,04
		75					
		Iniziate	ArrivateServer	Conclude	Pserver	Pconclude	DeltaPConclus
		7030					
ReticoloP4S10030303042Tot75		829	785	630	94,69	76	2,74
ReticoloP4S10070303042Tot75		1089	1035	808	95,04	74,2	0,94
ReticoloP4S10070703042Tot75		747	723	597	96,79	79,92	6,66
ReticoloP4S10070707042Tot75		818	818	818	100	100	26,74
		8020					
ReticoloP4S10020202042Tot75		768	736	601	95,83	78,26	5
ReticoloP4S10080202042Tot75		749	698	578	93,19	77,17	3,91
ReticoloP4S10080802042Tot75		760	741	607	97,5	79,87	6,61
ReticoloP4S10080808042Tot75		892	892	892	100	100	26,74
		9010					
ReticoloP4S10010101042Tot75		762	706	564	92,65	74,02	0,76
ReticoloP4S10090101042Tot75		727	692	557	95,19	76,62	3,36
ReticoloP4S10090901042Tot75		762	742	595	97,38	78,08	4,82
ReticoloP4S10090909042Tot75		1003	971	797	96,81	79,46	6,2

Ritardo:

Riferimento	Ritardo[ms]
Reticolo1sCentrale24Tot	934,18
Reticolo1sLato24tot	2816,73
2s	
Reticolo2sCentroLato24tot	1537,59
Reticolo2sLatoLato24tot	653,09
3s	
Reticolo2s1sLatoLatoAlto28Tot	1072,78
3s	
Reticolo2s1sLatoLatoAlto38Tot	1261,78
4s	
Reticolo6x7s4tot42	1433,84

2s

		Ritardo[ms]	Differenza[ms]	Differenza[%]
	50			
	7030			
Lato30Centro7024Tot507030		1130,83	477,74	73,15
Lato70Centro3024Tot507030		1386,17	733,08	112,25
LatoLato24Tot507030		741,62	88,53	13,56
	8020			
Lato20Centro8024Tot508020		1389,24	736,15	112,72
Lato80Centro2024Tot508020		1397,43	744,34	113,97
LatoLato24Tot508020		669,84	16,75	2,56
	9010			
Lato10Centro9024Tot509010		1221,62	568,53	87,05
Lato90Centro1024Tot509010		1454,79	801,7	122,75
LatoLato24Tot509010		764,9	111,81	17,12
	75			
	7030			
Lato30Centro7024Tot757030		1073,37	420,28	64,35
Lato70Centro3024Tot757030		1054,71	401,62	61,5
LatoLato24Tot757030		2757,5	2104,41	322,22
	8020			
Lato20Centro8024Tot758020		1373,44	720,35	110,3
P2SLato80Centro2024Tot758020		1076,16	423,07	64,78
LatoLato24Tot758020		930,06	276,97	42,41
	9010			
Lato10Centro9024Tot759010		1446,96	793,87	121,56
Lato90Centro1024Tot759010		1441,07	787,98	120,65
LatoLato24Tot759010		1021,55	368,46	56,42

3s

	50	Ritardo[ms]	Differenza[ms]	Differenza[%]
	7030			
ReticoloP3S100703028Tot50		649,57	-423,21	-39,45
ReticoloP3S100703038Tot50		1760,11	498,33	39,49
	8020			
ReticoloP3S100802028Tot50		1248,93	176,15	16,42
ReticoloP3S100802038Tot50		1674,87	413,09	32,74
	9010			
ReticoloP3S100901028Tot50		1152,59	79,81	7,44
ReticoloP3S100901038Tot50		1584,41	322,63	25,57
	75			
	7030			
ReticoloP3S100703028Tot75		1306,09	233,31	21,75
ReticoloP3S100703038Tot75		1607,66	345,88	27,41
	8020			
ReticoloP3S100802028Tot75		2361,54	1288,76	120,13
ReticoloP3S100802038Tot75		1173,98	-87,8	-6,96
	9010			
ReticoloP3S100901028Tot75		1027,48	-45,3	-4,22
ReticoloP3S100901038Tot75		949,11	-312,67	-24,78

4s

	50	Ritardo[ms]	Differenza[ms]	Differenza[%]
	7030			
ReticoloP4S10030303042Tot50		1035,87	-397,97	-27,76
ReticoloP4S10070303042Tot50		1054,39	-379,45	-26,46
ReticoloP4S10070703042Tot50		1036,02	-397,82	-27,75
ReticoloP4S10070707042Tot50		825,74	-608,1	-42,41
	8020			
ReticoloP4S10020202042Tot50		827,88	-605,96	-42,26
ReticoloP4S10080202042Tot50		958,48	-475,36	-33,15
ReticoloP4S10080802042Tot50		980,29	-453,55	-31,63
ReticoloP4S10080808042Tot50		952,59	-481,25	-33,56
	9010			
ReticoloP4S10010101042Tot50		981,91	-451,93	-31,52
ReticoloP4S10090101042Tot50		866,41	-567,43	-39,57
ReticoloP4S10090901042Tot50		880,68	-553,16	-38,58
ReticoloP4S10090909042Tot50		1041,66	-392,18	-27,35
	75			
	7030			
ReticoloP4S10030303042Tot75		1000,07	-433,77	-30,25
ReticoloP4S10070303042Tot75		766,66	-667,18	-46,53
ReticoloP4S10070703042Tot75		999,72	-434,12	-30,28
ReticoloP4S10070707042Tot75		379,56	-1054,28	-73,53
	8020			
ReticoloP4S10020202042Tot75		973,45	-460,39	-32,11
ReticoloP4S10080202042Tot75		1073,84	-360	-25,11
ReticoloP4S10080802042Tot75		1015,35	-418,49	-29,19
ReticoloP4S10080808042Tot75		403,82	-1030,02	-71,84
	9010			
ReticoloP4S10010101042Tot75		1089,02	-344,82	-24,05
ReticoloP4S10090101042Tot75		934,65	-499,19	-34,81
ReticoloP4S10090901042Tot75		685,46	-748,38	-52,19
ReticoloP4S10090909042Tot75		913,01	-520,83	-36,32

Accuratezza:

2s

50					
	Alta	Bassa	Alta[%]	Bassa[%]	Acc.Media
7030					
Lato30Centro7024Tot507030	74	144	33,94	66,06	43,58
Lato70Centro3024Tot507030	68	146	31,78	68,22	42,71
LatoLato24Tot507030	53	39	57,61	42,39	53,04
8020					
Lato20Centro8024Tot508020	72	122	37,11	62,89	42,27
Lato80Centro2024Tot508020	75	106	41,44	58,56	44,86
LatoLato24Tot508020	45	26	63,38	36,62	58,03
9010					
Lato10Centro9024Tot509010	72	100	41,86	58,14	43,49
Lato90Centro1024Tot509010	80	128	38,46	61,54	40,77
LatoLato24Tot509010	58	50	53,7	46,3	52,96
75					
7030					
Lato30Centro7024Tot757030	130	70	65	35	56
Lato70Centro3024Tot757030	72	137	34,45	65,55	43,78
LatoLato24Tot757030	44	45	49,44	50,56	49,78
8020					
Lato20Centro8024Tot758020	69	112	38,12	61,88	42,87
Lato80Centro2024Tot758020	67	102	39,64	60,36	43,79
LatoLato24Tot758020	43	35	55,13	44,87	53,08
9010					
Lato10Centro9024Tot759010	121	74	62,05	37,95	59,64
Lato90Centro1024Tot759010	64	126	33,68	66,32	36,95
LatoLato24Tot759010	48	54	47,06	52,94	47,65

3s

50							
	100 Alta	Bassa	100[%]	Alta[%]	Bassa[%]	Acc media	
7030							
ReticoloP3S100703028Tot50	55	166	222	12,42	37,47	50,11	53,68
ReticoloP3S100703038Tot50	153	210	263	24,44	33,55	42,01	60,53
8020							
ReticoloP3S100802028Tot50	39	156	221	9,38	37,5	53,13	50
ReticoloP3S100802038Tot50	147	212	283	22,9	33,02	44,08	58,13
9010							
ReticoloP3S100901028Tot50	60	148	194	14,93	36,82	48,26	52,89
ReticoloP3S100901038Tot50	159	230	262	24,42	35,33	40,25	60,25
75							
7030							
ReticoloP3S100703028Tot75	44	234	160	10,05	53,42	36,53	58,4
ReticoloP3S100703038Tot75	138	213	284	21,73	33,54	44,72	58,63
8020							
ReticoloP3S100802028Tot75	90	261	188	16,7	48,42	34,88	62,41
ReticoloP3S100802038Tot75	159	207	288	24,31	31,65	44,04	58,44
9010							
ReticoloP3S100901028Tot75	69	186	181	15,83	42,66	41,51	58,37
ReticoloP3S100901038Tot75	170	216	269	25,95	32,98	41,07	59,74

50											
	100I	II	III	100[%]	I[%]	III[%]	III[%]	Alta[%]	Bassa[%]	Acc media	
7030											
ReticoloP4S10030303042Tot50	164	100	265	168	23,53	14,35	38,02	24,1	0	76,47	46,47
ReticoloP4S10070303042Tot50	180	99	287	159	24,83	13,66	39,59	21,93	13,66	61,52	52,84
ReticoloP4S10070703042Tot50	184	114	331	181	22,72	14,07	40,86	22,35	54,94	22,35	67,88
ReticoloP4S10070707042Tot50	173	109	302	169	22,97	14,48	40,11	22,44	77,03	0	76,89
8020											
ReticoloP4S10020202042Tot50	166	109	310	174	21,87	14,36	40,84	22,92	0	78,13	37,5
ReticoloP4S10080202042Tot50	153	95	302	169	21,28	13,21	42	23,5	13,21	65,51	44,95
ReticoloP4S10080802042Tot50	164	103	294	154	22,94	14,41	41,12	21,54	55,52	21,54	71,66
ReticoloP4S10080808042Tot50	180	105	343	184	22,17	12,93	42,24	22,66	77,83	0	84,43
9010											
ReticoloP4S10010101042Tot50	140	105	290	157	20,23	15,17	41,91	22,69	0	79,77	28,21
ReticoloP4S10090101042Tot50	160	115	267	168	22,54	16,2	37,61	23,66	16,2	61,27	43,24
ReticoloP4S10090901042Tot50	141	97	306	163	19,94	13,72	43,28	23,06	57	23,06	73,55
ReticoloP4S10090909042Tot50	164	98	291	161	22,97	13,73	40,76	22,55	77,03	0	92,3
75											
	100I	II	III	100[%]	I[%]	III[%]	III[%]	Alta[%]	Bassa[%]	Acc media	
7030											
ReticoloP4S10030303042Tot75	158	119	336	172	20,13	15,16	42,8	21,91	0	79,87	44,09
ReticoloP4S10070303042Tot75	234	141	412	248	22,61	13,62	39,81	23,96	13,62	63,77	51,28
ReticoloP4S10070703042Tot75	170	84	278	191	23,51	11,62	38,45	26,42	50,07	26,42	66,49
ReticoloP4S10070707042Tot75	215	171	215	217	26,28	20,9	26,28	26,53	73,72	0	77,89
8020											
ReticoloP4S10020202042Tot75	174	99	304	159	23,64	13,45	41,3	21,6	0	76,36	38,91
ReticoloP4S10080202042Tot75	162	78	283	175	23,21	11,17	40,54	25,07	11,17	65,62	45,27
ReticoloP4S10080802042Tot75	160	110	302	169	21,59	14,84	40,76	22,81	55,6	22,81	70,63
ReticoloP4S10080808042Tot75	211	211	234	236	23,65	23,65	26,23	26,46	76,35	0	84,73
9010											
ReticoloP4S10010101042Tot75	143	88	298	177	20,25	12,46	42,21	25,07	0	79,75	28,23
ReticoloP4S10090101042Tot75	157	92	282	161	22,69	13,29	40,75	23,27	13,29	64,02	41,05
ReticoloP4S10090901042Tot75	194	86	262	200	26,15	11,59	35,31	26,95	46,9	26,95	71,05
ReticoloP4S10090909042Tot75	199	147	393	232	20,49	15,14	40,47	23,89	79,51	0	92,05