

UNIVERSITA' DEGLI STUDI DI PISA

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea

RECUPERO DI DATI IN UNA RETE DI SENSORI: PROGETTAZIONE E VALUTAZIONE DI ALGORITMI ENERGETICAMENTE EFFICIENTI

Candidato:

Bachini Samuele

Relatori:

Prof. Giuseppe Anastasi

Prof. Francesco Marcelloni

Dr. Giovanni Mainetto

Dr. Marco Conti

Anno Accademico 2003/2004

INDICE

I INTRODUZIONE	1
II RETI DI SENSORI	6
2.1 Fattori che influenzano la progettazione di una rete di Sensori	10
2.1.1 Tolleranza ai guasti	11
2.1.2 Scalabilità	11
2.1.3 Costi di Produzione	11
2.1.4 Ambiente Operativo	12
2.1.5 Topologia della Rete di Sensori	12
2.1.6 Vincoli Hardware	13
2.1.7 Mezzo di Trasmissione	15
2.1.7 Consumo Energetico	15
2.2 Consumo Energetico in una Rete di Sensori	16
2.2.1 Unità Computazionale	18
2.2.2 Unità Trasceiver	19
2.2.3 Unità di Sensing	19
2.2.4 Unità Energetica	20
2.2.5 Convertitore DC-DC	22
2.3 Applicazioni per Reti di Sensori	23
2.3.1 Applicazioni Militari	23
2.3.2 Applicazioni Ambientali	24
2.3.3 Applicazioni Mediche	26
2.3.4 Applicazioni Domestiche	26
2.3.5 Applicazioni Commerciali	27
III STATO DELL'ARTE	29

3.1	Protocolli per la <i>Reliability</i> e per il controllo della Congestione	30
3.1.1	TCP	31
3.1.2	ESRT	31
3.1.3	PSFQ	38
3.1.4	CODA	41
3.2	Protocolli di Routing	43
3.2.1	Energy Efficient Route	43
3.2.2	Data Centric Route	44
3.2.3	Data Aggregation	45
3.2.4	Flooding	45
3.2.5	Gossiping	46
3.2.6	Spin	47
3.2.7	SAR	47
3.2.8	LEACH	49
3.2.9	Directed Diffusion	50
3.3	Protocolli per l'accesso al mezzo trasmissivo	57
3.3.1	SMACS	58
3.3.2	EAR	60
3.3.3	CSMA	60
3.3.4	IEEE802.11	61
IV	TINYDB	69
4.1	Base di Dati nel TinyDB	70
4.2	Linguaggio di Interrogazione nel TinyDB	72
4.2.1	Event-Based Query	75
4.2.2	Lifetime-Based Query	76
4.2.3	Aggregate Query	78
4.2.4	Trigger Query	79
4.2.5	Query Over Flash	80
4.3	Disseminazione delle Interrogazioni e Raccolta dei Risultati in TinyDB	82
4.3.a	Catalog	82
4.3.b	Sincronizzazione tra i Nodi Sensore	84
4.3.c	Albero di Routing e Albero di Routing Semantico	84

4.3.d	Lista dei nodi Vicini Attivi	89
4.3.1	Immissione dell'Interrogazione nella rete	91
4.3.2	Parsing e Validazione dell'Interrogazione	91
4.3.3	Ottimizzazione dell'Interrogazione	92
4.3.4	Traduzione dell'Interrogazione SQL in rappresentazione binaria	97
4.3.5	Disseminazione dell'Interrogazione	98
4.3.6	Elaborazione dell'Interrogazione	99
4.3.7	Elaborazione nella Rete di Interrogazioni Aggregate (TAG)	104
4.3.8	Adattamento del Rate di Trasmissione e Ottimizzazione del Consumo Energetico	109
4.3.9	Comportamento da Router	111
4.3.10	Visualizzazione dei Risultati	111
V NUOVA ARCHITETTURA PER RETI DI SENSORI		114
5.1	Cross-Layer	115
5.2	Descrizione della Nuova Architettura per Reti di Sensori	116
5.3	Linguaggio di Interrogazione	118
5.4	Albero di Routing	119
5.5	Metadati	121
5.6	Disseminazione delle Interrogazioni	122
5.7	Esecuzione delle Interrogazioni	124
5.8	Protocollo MAC	125
5.9	Protocollo di Scheduling	128
5.10	Gestione della Topologia della Rete	130
5.11	Reliability	132
5.12	Decisione del Periodo di Campionamento	135
5.13	Sincronizzazione	137
VI PROTOCOLLO PER LO SCHEDULING DELLA COMUNICAZIONE TRA NODI SENSORE		139
6.1	Modello di Scheduling	141
6.2	Diminuzione Talk Interval	149
6.3	Aumento Talk Interval	152

6.4	Fase di Startup	158
6.5	Protocollo Generale di Scheduling dei nodi	163
6.6	Cambio Communication Period	165
VII	PROTOCOLLO DI ROUTING	169
7.1	Costruzione Albero di Routing	170
7.1.1	Creazione Albero	170
7.1.2	Dynasty Discover Phase	174
7.1.3	Protocollo per la costruzione dell'Albero di Routing	176
7.2	Mantenimento Albero di Routing	179
7.2.1	Aggiunta di un Nodo alla Rete	181
7.2.2	Caduta di un Nodo	185
7.2.3	Meccanismo Anti-Loop	187
7.2.4	Nodo a basso livello energetico	189
7.2.5	Refresh Periodico	193
VIII	SIMULAZIONE PROTOCOLLO DI ROUTING	196
8.1	Indici di Prestazione	198
8.2	Risultati delle Simulazioni	200
8.2.1	Simulazioni con Rete omogenea dal punto di vista energetico	200
8.2.2	Simulazioni con Rete non omogenea dal punto di vista energetico	206
IX	CONCLUSIONI	218
	RINGRAZIAMENTI	221
	APPENDICI	222
A	TinyOS	222
B	TinyDB & TinyOS	228
C	Sintassi TinyDB	229
	BIBLIOGRAFIA	231

CAPITOLO I

INTRODUZIONE

I recenti progressi tecnologici nei sistemi micro-elettro-meccanici, nelle comunicazioni *wireless* – cioè nelle tecnologie di comunicazione che sono prive di qualunque tipo di ‘filo’ come mezzo trasmissivo – e nell’elettronica digitale hanno permesso lo sviluppo di una nuova classe di dispositivi detti Nodi Sensore (in inglese *Sensor Node*).

Ciascun Nodo Sensore è tipicamente capace di rilevare misure fisiche, quali temperatura, umidità, pressione atmosferica, ecc..., dall’ambiente circostante, può compiere semplici elaborazioni e comunicare misure e elaborazioni compiute ad altri modi tramite comunicazioni *wireless*. Un insieme di questi dispositivi forma un tipo di rete che in letteratura viene chiamata Rete di Sensori (*Sensor Network*).

Il basso costo unitario e le ridotte dimensioni rendono una Rete di Sensori potenzialmente utilizzabile per il monitoraggio di fenomeni ambientali in molti ambiti applicativi. Alcune aree applicative potenzialmente interessate alle Reti di Sensori sono certamente quella militare, quella della sicurezza e quella medica. Ad esempio, potrebbe essere possibile con questa tecnologia rilevare la presenza di prodotti chimici disciolti nell’acqua o nell’aria, così come rilevare presenze umane in edifici anziché il passaggio di mezzi blindati in punti critici, ecc...

La Rete di Sensori che prendiamo in considerazione assume che ci sia almeno una entità speciale, chiamata nodo *Sink*, che agisce come nodo *gateway* fra il mondo dei sensori e il mondo dei computer. Inizialmente, l’utente collegato a un computer comunica al nodo Sink le informazioni presenti nella rete sensoriale di cui necessita formulando una *interroga-*

zione apposita. Il nodo *Sink* provvede a compilare l'interrogazione, a generare del codice corrispondente all'interrogazione originale e ad 'iniettare' tale codice nella rete di sensori. Successivamente il nodo speciale, che riceverà le informazioni richieste dai nodi sensori, elaborerà i dati raccolti prima di fornire il risultato finale all'utente.

Le Reti di Sensori sono modellate come delle microbasi di dati distribuite. Il recupero delle informazioni da parte dell'utente viene specificato tramite un linguaggio di interrogazione simile all'SQL.

A differenza delle reti tradizionali, una Rete di Sensori impone un insieme di nuove limitazioni per quanto riguarda la progettazione dei protocolli. I dispositivi che compongono una Rete di Sensori hanno una memoria di dimensioni ridotte, vincoli molto stringenti dal punto di vista dei consumi energetici, minori capacità di elaborazione e di comunicazione. Inoltre la topologia di una Rete di Sensori cambia molto frequentemente a causa dell'alta probabilità di fallimento dei nodi. Una ulteriore peculiarità di una Rete di Sensori è che i nodi devono essere disposti in modo denso e quindi una rete di questo tipo può essere formata anche da centinaia di milioni di dispositivi. Ciò significa che il numero di nodi qui coinvolti è di qualche ordine di grandezza maggiore che nelle reti *wireless ad-hoc* tradizionali. Inoltre, il problema del risparmio dei consumi energetici diventa l'obiettivo su cui maggiormente concentrare la progettazione. A causa della estrema limitatezza e non rinnovabilità della risorsa energetica – non è ovviamente possibile ricaricare o sostituire le batterie in un Nodo Sensore perché sono disposti in un ambiente ostile e difficilmente accessibile. Date le molte differenze esistenti tra le reti tradizionali e le Reti di Sensori, appare inappropriato ed inefficiente pensare di riadattare a questo caso i protocolli usati nelle reti tradizionali. Per convincersene, basti pensare al fatto che, ad esempio, l'alto numero di nodi rende il protocollo di *Routing* basato su un semplice *flooding* assolutamente inapplicabile nel caso di una Rete di Sensori.

Benché le potenziali applicazioni per una Rete di Sensori siano molte e con caratteristiche diverse le une dalle altre, una particolarità le accomuna

e cioè hanno tutte la necessità di un'infrastruttura di Rete appositamente ideata per risolvere i problemi appena elencati. Senza un servizio di *Routing* che risulti scalabile all'incrementare del numero di dispositivi, l'energia spesa dai nodi per determinare il percorso per recapitare dati al nodo *Sink* comporterebbe un grande consumo energetico. Se tutti i protocolli di rete non venissero progettati tenendo conto dei vincoli energetici, i nodi della rete si ritroverebbero completamente scarichi soltanto dopo poche rilevazioni.

Una Rete di Sensori è composta da dispositivi omogenei sia dal punto di vista delle caratteristiche hardware sia da quello del Sistema Operativo mentre, per quanto riguarda le Reti tradizionali, si è cercata una standardizzazione dei protocolli per permettere ad entità disomogenee di comunicare tra loro. Gli organismi di standardizzazione, quali *IETF* e *OSI*, hanno definito protocolli per reti che si basano su una strutturazione a strati (*layering*). Con questa strutturazione, una rete viene organizzata in una gerarchia di entità logicamente distinte (gli strati). I servizi forniti da una entità ad un certo livello sono solamente basati sui servizi forniti dall'entità del livello inferiore. Questa modularizzazione semplifica non solo la progettazione ma anche la manutenzione e l'aggiornamento dei componenti del sistema rendendo ad esempio la modifica di un protocollo realizzato in un certo livello completamente trasparente al resto del sistema. Questo approccio a strati isolati non risulta completamente adatto al caso di una Rete di Sensori. Ad esempio, il problema cruciale delle reti di sensori, e cioè il risparmio energetico, è un problema che si risolve mettendo in essere una unica politica di gestione della risorsa energia che attraversa tutti gli strati del sistema e tiene anche conto delle relazioni fra strati. Non esiste cioè un unico strato che implementa questa politica, e, per ottenere i massimi risparmi possibili, occorre anche progettare delle politiche che tengano conto delle interrelazioni fra strati (*interlayering*). In altri termini, il consumo di energia può essere ottimizzato solo se l'intera pila di protocolli è resa *power-aware*.

In definitiva, noi riteniamo che il successo di questa nuova tecnologia sia fortemente legato alla creazione di una Architettura di Rete apposita.

È per questo che in questa Tesi abbiamo progettato una Architettura dedicata per Reti di Sensori. Tale architettura mantiene l'approccio a *strati* ma contemporaneamente permette la cooperazione tra protocolli appartenenti a livelli diversi tramite la condivisione di informazioni riguardanti lo stato della rete (*Network Status*). Ogni protocollo può accedere al *Network-Status* per condividere i propri dati con gli altri protocolli. La separazione dei livelli è ottenuta standardizzando gli accessi al *Network-Status*.

Molti sono i benefici che è possibile trarre usando un approccio *cross-layered*: sono infatti possibili ottimizzazioni per tutte le funzioni di rete ed una riduzione degli *overhead* evitando la duplicazione di dati su livelli diversi. Inoltre è possibile rendere ogni livello *contex-aware*: qualsiasi protocollo può essere progettato per essere consapevole dello stato della rete. All'interno di questa nuova Architettura ci siamo concentrati sulla realizzazione di due protocolli di Rete. Il primo è un protocollo di *scheduling* delle trasmissioni tra nodi Sensore e sfrutta la natura periodica delle applicazioni di *monitoring*. L'intento è quello di sincronizzare i nodi sensore e di concentrare la comunicazione tra di essi in istanti di tempo prestabiliti. Infatti, i nodi Sensore, quando inattivi, possono essere messi in una modalità di funzionamento chiamata *power-down mode* con conseguente risparmio energetico.

L'altro protocollo permette la creazione ed il mantenimento di un albero di *Routing* per recapitare comandi dal *Sink* ai nodi della Rete e, analogamente, per far pervenire le rilevazioni al nodo *Sink*. In una Rete di Sensori, ciascun nodo, dato il limitato raggio di trasmissione, non è capace di comunicare direttamente con il *Sink* ma in generale comunica con i propri vicini formando così una rete *multi-hop*. In un ambiente di questo tipo il *Routing* assume un'importanza fondamentale a causa della mancanza di un controllo centralizzato e della dinamicità della rete. Come già sappiamo, il vincolo stringente di una Rete Sensoriale è che i Nodi Sensore hanno limi-

tate capacità energetiche. Quindi, un protocollo di *Routing* che cerchi di minimizzare la latenza, e cioè il tempo necessario per recapitare dei dati dal nodo sorgente al nodo *Sink*, riesce a minimizzare anche il consumo energetico in quanto una latenza minima comporta un numero minimo di trasmissioni da effettuare. Tuttavia, un protocollo di questo tipo porta a non rispettare la *'fairness'* nella rete. Ad esempio, un *Routing* basato unicamente sullo *'shortest path'* finirebbe con l'utilizzare sempre lo stesso set di *hop* per recapitare pacchetti dalla sorgente al nodo destinatario. Questi nodi finiranno con lo scaricarsi precocemente rispetto agli altri e conseguentemente correranno il rischio di spegnersi creando così dei buchi nella rete. È quindi necessario cercare di minimizzare il numero di *hop* che deve percorrere un messaggio per arrivare dalla sorgente a destinazione ma al tempo stesso cercare di far rimanere omogeneo il livello energetico tra i nodi della rete in quanto un obiettivo prioritario è quello di cercare di massimizzare il tempo di vita della rete.

La tesi è organizzata come segue. Il Capitolo 2 descrive le caratteristiche principali di una Rete di Sensori ed alcune possibili applicazioni. Il Capitolo 3 riporta lo stato dell'arte sui protocolli proposti per le Reti di Sensori. Il Capitolo 4 descrive *TinyDB*, un sistema per la gestione dei dati in una Rete di Sensori. Il Capitolo 5 introduce la Nuova Architettura per le Reti di Sensori. Nel Capitolo 6 è riportata la descrizione del protocollo di *Scheduling* tra i Nodi Sensore mentre nel Capitolo 7 è dettagliato il protocollo per la costruzione ed il mantenimento dell'Albero di *Routing*. Infine, nel capitolo 8, sono riportati i risultati ottenuti dalla simulazione di quest'ultimo protocollo.