

SALES: Scalable context-Aware middleware for mobiLe Environments

Web Site: <http://www-lia.deis.unibo.it/Research/SALES>

Aggiornato al 14/03/2011

Introduzione

Nell'ultimo decennio, la straordinaria diffusione di dispositivi mobili con connettività wireless ha promosso la nascita di nuovi interessanti scenari nei quali gli utenti richiedono accesso continuo e indipendente dalla localizzazione attuale alle proprie risorse. Partendo da questi scenari, tradizionalmente catalogati come ubiquitous computing, si sono venuti man mano a delineare nuovi scenari in cui è necessario fornire consapevolezza alle singole applicazioni del proprio contesto di esecuzione. Un'applicazione context-aware è in grado di monitorare il proprio contesto di esecuzione e di adattarsi a seconda dei cambiamenti di quest'ultimo.

Tradizionalmente, il contesto raggruppa vari attributi, a seconda della necessità del progettista, che possono spaziare in modo molto differenziato in ambito tecnico o meno: per esempio, se l'applicazione deve fornire flussi multimediali, il contesto potrà contenere la banda disponibile e il dispositivo utilizzato dall'utente per la visione (attributo tecnico) per adattare il video di conseguenza. D'altro canto, se l'applicazione deve ottenere informazioni sulle risorse disponibili nel luogo in cui attualmente l'utente si trova, il contesto può contenere dati di alto livello utilizzati per la descrizione dell'ambiente.

Come si vede immediatamente considerando qualche semplice esempio, la conoscenza del contesto di esecuzione è fondamentale per la realizzazione di tali nuovi scenari. Ipotizziamo che uno studente voglia stampare le diapositive associate ad una lezione mediante il suo PDA: se lo studente è appena arrivato, potrebbe non conoscere i servizi di stampa disponibili e potrebbe perdere tempo in un'inutile ricerca che potrebbe invece essere portata a termine in modo del tutto automatico dal supporto. Infatti, un servizio di stampa context-aware potrebbe recuperare i dati di contesto associati al luogo fisico in cui lo studente si trova e potrebbe suggerire il servizio di stampa più vicino. Inoltre, potrebbe ridirigere automaticamente il comando di stampa emesso dal PDA per raggiungere direttamente quella specifica stampante. Successivamente, durante la lezione, il professore decide di condividere il video di una conferenza con gli studenti presenti ma senza pubblicare il video sul sito web del corso: lo scambio manuale del video con tutti gli studenti comporterebbe un inutile spreco di tempo. In questo caso, un servizio di condivisione dei file basato su contesto potrebbe essere utilizzato per condividere contenuti solo con particolari persone, in particolare soltanto gli studenti presenti nell'aula durante la lezione, evitando altre soluzioni più macchinose e complesse.

Negli ultimi anni, le applicazioni basate su contesto hanno quindi ricevuto molta attenzione sia nell'ambito industriale che accademico. Tale attenzione si è però gradualmente spostata dalle singole applicazioni alle infrastrutture di supporto (middleware) in grado di semplificare il progetto, lo sviluppo e il deployment di tali applicazioni. In particolare, un middleware può semplificare lo sviluppo delle applicazioni fornendo supporto a numerosi problemi:

1. **Rappresentazione, Accesso e Distribuzione del Contesto** – Tutti i dati di contesto devono essere opportunamente rappresentati e distribuiti all'interno del sistema. D'altronde, ogni singolo dispositivo deve aver accesso continuo a tali dati per garantire la correttezza dell'adattamento e del middleware e delle applicazioni. Ad esempio, considerando il servizio di stampa, il dispositivo deve poter aver accesso al descrittore associato al luogo in cui si trova e tale accesso deve essere garantito durante gli spostamenti. In particolare, la disseminazione e il ritrovamento dei dati di contesto è un

problema fondamentale all'interno di tali sistemi dato che introduce una mole di traffico non banale e strettamente proporzionale alla popolazione corrente.

2. **Eterogeneità dei dispositivi e delle comunicazioni wireless adottate** – Per supportare scenari pervasivi, il middleware deve supportare l'eterogeneità sia in termini di dispositivi sia di comunicazioni wireless. Ad esempio, all'interno di un'area universitaria possiamo trovare dispositivi molto differenti in termini di risorse, come tradizionali computer portatili e telefoni cellulari, che comunicano tra loro sfruttando standard wireless differenti, come IEEE 802.11 a/b/g (Wi-Fi) e Bluetooth (BT). Tutte queste differenti offerte devono essere considerate dal middleware per consentire adattamenti automatici e dinamici dipendenti dalla comunicazione wireless e dal dispositivo su cui è posto in esecuzione. Ad esempio, se il middleware deve fornire un servizio per un telefono cellulare con una semplice connessione BT, dovrà adattare automaticamente sia l'erogazione di servizi che i protocolli di gestione per evitare di saturare la banda.

3. **Gestione della mobilità e della localizzazione** – Tutti i dispositivi coinvolti nello scenario principale sono mobili. Di conseguenza, il middleware deve gestire la mobilità per offrire continuità nell'erogazione dei servizi e per supportare la disseminazione dei dati verso i singoli dispositivi. D'altra parte, se la mobilità deve essere gestita per mantenere il collegamento logico al sistema, il middleware deve anche essere in grado di utilizzare un sistema di localizzazione dato che informazioni accurate su tale attributo possono essere sia utilizzate all'interno del contesto sia utilizzate per introdurre protocolli di gestione adattativi e basati sulla staticità del nodo.

4. **Scalabilità** – Come accennato sopra, un middleware context-aware deve consentire efficacemente la distribuzione dei dati di contesto. Per garantire la scalabilità finale, il middleware dovrebbe sfruttare standard wireless differenti (in modo da utilizzare la banda presente su ognuno di essi) e dovrebbe anche sfruttare differenti modalità di comunicazione, sia basate su infrastrutture sia su comunicazioni ad-hoc. D'altronde, se le infrastrutture fisse permettono di garantire la distribuzione dei dati di contesto, le comunicazioni ad-hoc consentono lo scambio diretto di messaggi tra nodi senza introdurre traffico aggiuntivo sull'infrastruttura. Inoltre, i dati dovrebbero essere distribuiti solo ai nodi che ne fanno richiesta evitando disseminazioni in aree non richieste. Infine, il middleware deve rafforzare principi di località fisica e logica per ridurre sprechi di banda inutili. Ad esempio, gli avvisi emessi dal servizio di stampa dell'esempio riportato precedentemente dovrebbero essere mantenuti vicini al luogo in cui si trova la stampante dato che la loro utilità decresce in maniera forse più che proporzionale con la distanza (difficilmente l'utente è disposto a percorrere grandi distanze per recuperare dei documenti stampati).

5. **Sicurezza e privacy** – Le informazioni di contesto possono contenere dati sensibili da proteggere adeguatamente. Se necessario, il middleware deve garantire integrità, riservatezza e disponibilità dei dati, consentendo l'accesso alle sole persone autorizzate. Per esempio, se il contesto contiene un attributo per la localizzazione attuale dell'utente, bisogna assolutamente garantire che tali dati siano disponibili solamente a livello di infrastruttura e magari soltanto ad alcuni utenti autorizzati, ad esempio agli amici, per evitare violazione della privacy. Per di più, la gestione della sicurezza non è per nulla semplice dato che, ad esempio, la protezione della riservatezza dei dati mediante crittografia potrebbe compromettere l'utilizzabilità dei meccanismi di distribuzione dei dati di contesto.

Partendo da queste problematiche, il nostro lavoro intende offrire pieno supporto agli scenari context-aware, in particolar modo mascherando problematiche di scalabilità e di adattabilità. Il progetto **Scalable context-Aware middleware for mobiLe EnvironmentS (SALES)** nasce con tali ambizioni, utilizzando tecnologie wireless differenti, comunicazioni ad-hoc e basate su infrastrutture e proponendo un approccio collaborativo e mirato alla disseminazione dei dati di contesto.

Tutte le tesi correlate sono mirate alla realizzazione di casi di studio applicativi o di moduli di gestione di SALES, e prevedono quindi un'apposita fase di realizzazione di un prototipo utile alla valutazione delle scelte effettuate.

Architettura distribuita di SALES

L'architettura distribuita di SALES può essere suddivisa in due macro-livelli principali. Il livello superiore è basato su nodi fissi e potenti utili per ottenere una distribuzione veloce e affidabile dei dati di contesto, mentre il livello inferiore sfrutta dispositivi meno potenti, che comunicano mediante collegamenti wireless ad-hoc, per ottenere maggiore affidabilità ed effettuare distribuzione meno costosa. Infine, 'versatili' dispositivi utente multi-homed collegano i due livelli per realizzare un sistema unico in cui disseminare e recuperare dati di contesto.

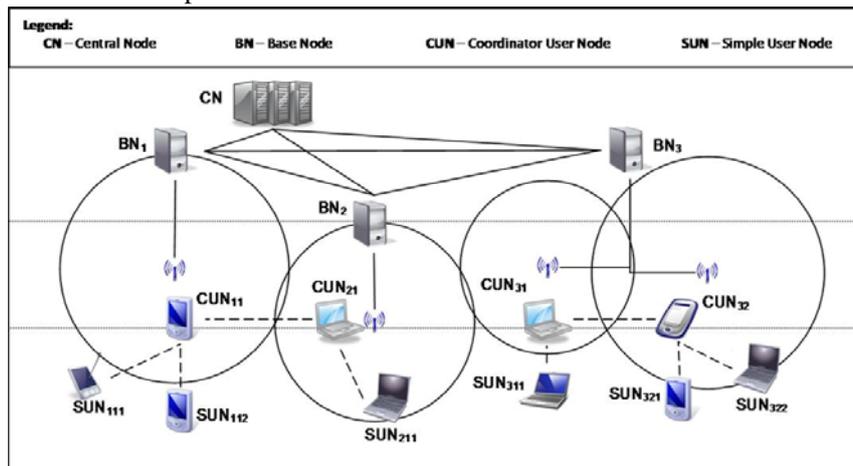


Figura 1. Architettura distribuita di SALES

Più nel dettaglio, il middleware distingue quattro tipi di nodi differenti:

Central Node (CN) – Il CN è la radice dell'architettura distribuita. È l'unica entità statica delegata alla memorizzazione persistente dei dati di contesto e fornisce i dati solo su richiesta, e, in quanto tale, tipicamente realizzato in architettura cluster per garantire scalabilità. Il CN è nascosto dai nodi appartenenti al livello inferiore (BN) ed è quindi in grado di scambiare messaggi solo con questi.

Base node (BN) - Ogni BN è un nodo fisso associato a uno o più WiFi APs disponibili nell'ambiente. Un BN garantisce la connessione con i nodi utenti presenti nei luoghi serviti ed effettua cache dei dati di contesto per ridurre le richieste verso il CN. Ogni BN condivide inoltre le cache locali con i propri pari per ridurre ulteriormente le richieste inviate al livello superiore. Secondo gli specifici dati di contesto, i BN possono adottare politiche sia reattive che proattive per la gestione dei dati dal CN.

Coordinator User Node (CUN) - I dispositivi utenti sono raggruppati in cluster per adattare la disseminazione dei dati di contesto. Ogni cluster prevede un coordinatore locale eletto dinamicamente, ossia un CUN, che controlla la disseminazione dei dati e la mobilità all'interno del cluster. Dato che anche il coordinatore è tipicamente un dispositivo mobile, ogni CUN esegue protocolli di gestione della mobilità col suo BN, se disponibile, e con i nodi serviti per permettere il discovery e la formazione dei cluster. SALES sfrutta inoltre i CUN per costruire un'infrastruttura di instradamento multi-hop e basata su contesto utile per ridurre sia il carico introdotto sui livelli superiori sia i tempi di risposta. Questa infrastruttura di instradamento è inoltre l'unica scelta disponibile per sostenere la disseminazione e la ricerca dei dati in scenari privi di infrastruttura. Come detto prima, i CUN sono tradizionalmente dispositivi multi-homed dato che gestiscono contemporaneamente sia comunicazioni basate su infrastruttura che su modalità ad-hoc.

Simple User Node (SUN) - Ogni dispositivo utente che non è un coordinatore è un SUN. Ogni SUN, come un tradizionale CUN, interagisce, producendo e recuperando dati di contesto. Inoltre, un SUN può scambiare messaggi solo con il proprio CUN e con i propri pari mediante comunicazioni ad-hoc, e deve mantenere un'associazione con un CUN disponibile per avere accesso al sistema.

Architettura software del middleware SALES

Il middleware deve risolvere differenti problemi legati allo scenario principale supportato. In prima istanza, SALES deve gestire la mobilità, l'eterogeneità delle comunicazioni wireless, e la scalabilità del sistema finale.

Partendo dalla gestione della mobilità, sia i SUN e i CUN devono effettuare discovery dell'infrastruttura presente e degli altri nodi disponibili. Per di più, ogni singolo nodo utente deve monitorare continuamente il collegamento logico col nodo padre, introducendo un overhead ulteriore in termini di banda. Pertanto, SALES cerca di adattare i protocolli di gestione della mobilità in base alla copertura garantita dallo standard wireless utilizzato e alla mobilità dei singoli dispositivi per aumentare il periodo di verifica del link logico con il padre e ridurre il consumo di banda. Inoltre, SALES deve evitare la saturazione dei canali wireless, adattando la trasmissione dei dati in base alla banda attuale, e deve sfruttare le differenti modalità, ad-hoc e basata su infrastruttura, per ridurre il carico introdotto e aumentare la scalabilità. Infine, dato che la disseminazione dei dati di contesto introduce un overhead direttamente proporzionale alla popolazione corrente ed ai dati distribuiti, SALES garantisce la scalabilità disseminando solo i dati necessari ed evitando politiche di disseminazione basate su inondazione.

In particolare, il middleware realizza percorsi di instradamento dei dati mediante il concetto di "context query". Le context query rappresentano le necessità del livello applicativo e devono essere propagate per la disseminazione dei dati all'interno dell'architettura distribuita di SALES. Attualmente, le query possono adottare anche tecniche probabilistiche, quali i filtri di Bloom, per ridurre la dimensione finale dei messaggi utilizzati per la disseminazione.

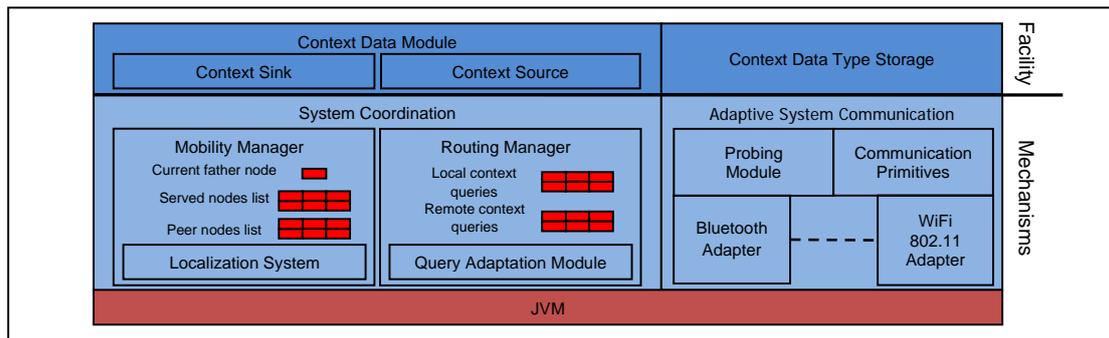


Figura 2. Architettura logica del middleware SALES

Da un punto di vista di alto livello, SALES adotta l'architettura software a due livelli mostrata in Figura 2:

Facility layer – Questo layer implementa tutte quelle funzionalità di alto livello necessarie alla creazione, al ritrovamento e alla distribuzione dei dati di contesto. Ogni dato di contesto è associato univocamente ad un apposito descrittore che fornisce sia la struttura del dato che le politiche di disseminazione per quel particolare tipo di dato. Il *Context Data Type Storage* consente la memorizzazione e l'accesso a tali descrittori. Ogni dato di contesto è associato ad un apposito *Context Data Module* che contiene una source e una sink: la *Context Source* consente la creazione e la distribuzione di nuovi dati di contesto, mentre la *Context Sink* permette il ritrovamento dei dati di contesto mediante le context query.

Mechanisms layer – Questo layer realizza tutte le funzionalità di basso livello utili alla comunicazione inter-nodo e alla formazione della gerarchia. L'*Adaptive System Communication* offre primitive generiche per l'invio/la ricezione dei messaggi, nasconde i dettagli delle singole tecnologie wireless e impone il rispetto della banda disponibile. Invece, ogni tecnologia wireless ha un apposito adattatore, quali il *Bluetooth Adapter* e il *WiFi Adapter*, che implementa le reali funzionalità di invio/ricezione sulla specifica tecnologia. Sopra tali adapter, il *Communication Primitives* realizza

primitive di invio sia one-way che request/response, mentre il *Probing Module* monitora periodicamente la banda disponibile su ogni collegamento. Il *System Coordination* utilizza le API di comunicazioni fornite dall'*Adaptive System Communication* per supportare mobilità e decisioni di routing. Il *Mobility Manager* sfrutta il *Localization System* per ritrovare la localizzazione attuale e per realizzare protocolli di mobilità adattativi. Infine, il *Routing Manager* impone le strategie di distribuzione dei dati e delle query, e usa il *Query Adaptation Module* per adattare dinamicamente la rappresentazione delle query.

Dettagli implementativi

L'architettura distribuita e software di SALES è stata realizzata sia su tecnologia Java che in C++ all'interno del simulatore ns2. Mentre la prima implementazione è utile a testare alcune problematiche che possono emergere solo durante l'utilizzo di testbed reali, la seconda consente lo studio dei protocolli di distribuzione in ambienti su larga scala, densamente popolati, e con nodi mobili. Conseguentemente, le due implementazioni sono complementari tra loro e consentono uno studio a largo spettro delle soluzioni proposte. Nel seguito, sono riportati alcuni dettagli inerenti le singole implementazioni.

Implementazione su tecnologia Java

- L'implementazione è basata su piattaforma J2SE 1.6.
- Sistema di localizzazione naive.
- La configurazione e il bootstrap dei nodi utente è gestita completamente in modo statico.
- L'*Adapter Bluetooth* è realizzato solo parzialmente (supporta solo piconet).

Implementazione su ns2

- L'implementazione è basata su simulatore ns2-34 in linguaggio C++/TCL.
- Sistema di localizzazione naive.
- La configurazione e il bootstrap dei nodi utente è gestita dal simulatore.
- Nessun supporto Bluetooth.

Tesi disponibili per laurea specialistica:

1. Algoritmi intelligenti per la distribuzione in architetture ibride infrastruttura/ad-hoc – Nell'architettura a due livelli di SALES, è assolutamente conveniente utilizzare le comunicazioni ad-hoc tra i nodi mobili per poter effettuare la distribuzione dei dati di contesto. In generale, bisogna notare che il traffico introdotto sull'infrastruttura fissa deve essere minimizzato sia per motivi economici che di scalabilità del sistema. Per raggiungere tale obiettivo, è auspicabile utilizzare l'infrastruttura fissa solo per ricevere ed elaborare piccole informazioni di gestione, quale qualità di contesto, distribuzione delle sorgenti, profili utente, etc., per riconfigurare i processi di distribuzione sulla MANET. In questo modo, possiamo massimizzare i dati distribuiti sui collegamenti ad-hoc e minimizzare il carico sull'infrastruttura. Scopo fondamentale della tesi è pertanto lo studio e la realizzazione di protocolli di distribuzione dei dati di contesto adattativi e basati sulla collaborazione tra infrastruttura fissa e mobile. In una prima fase, bisognerà analizzare alcune delle proposte presenti in letteratura ed evidenziare quali aspetti possono essere utilizzati per riconfigurare la distribuzione dei dati di contesto. In una seconda, bisognerà implementare e testare la soluzione all'interno dell'implementazione ns2 per evidenziare possibili vantaggi e svantaggi rispetto la soluzione attualmente adottata da SALES.

2. Introduzione della tecnologia UMTS nell'implementazione ns2 di SALES – La disponibilità di connessioni verso infrastrutture fisse è fondamentale per garantire l'accesso ai dati di contesto in ambienti non serviti da tradizionali infrastrutture wireless a corto raggio, quale BT e Wi-Fi. Sia in scenari cittadini che veicolari, bisogna utilizzare tecnologie cellulari, quali GSM/UMTS/LTE, per raggiungere l'infrastruttura fissa di SALES e poter ritrovare le informazioni di contesto necessarie. Allo stato attuale, ns2 non supporta in modo nativo infrastrutture e comunicazioni UMTS ma esistono numerose estensioni e progetti di ricerca che implementano tale supporto. Scopo fondamentale di questa tesi è quello di introdurre il supporto UMTS nell'implementazione attuale di SALES. Al tempo stesso, bisognerà compiere uno studio critico sull'utilizzo di tecnologie UMTS e sugli eventuali vantaggi/svantaggi rispetto all'utilizzo di tecnologia a medio raggio WiFi.

3. Studio di algoritmi di broadcast e comparazione basata su ns2 – La distribuzione dei dati di contesto all'interno di SALES è guidata da query per ridurre il traffico finale introdotto. Purtroppo, tale protocollo di scambio dati non risulta essere sempre la soluzione vincente. Ad esempio, se dobbiamo distribuire un dato di contesto a tutti i nodi presenti in una particolare area fisica, non conviene assolutamente utilizzare un protocollo pull basato su query/risposta verso ognuno di essi; conviene invece sfruttare la natura broadcast delle comunicazioni wireless per minimizzare il numero finale di messaggi scambiati. Scopo fondamentale di tale tesi è quello di estendere i protocolli di distribuzione di SALES con soluzioni broadcast limitate. In una prima fase, bisognerà realizzare alcune soluzioni presenti in letteratura nell'implementazione ns2 di SALES. In una seconda, bisognerà effettuare uno studio comparativo delle soluzioni implementate per identificare vantaggi e svantaggi dei singoli protocolli.

4. Aggregazione delle query di contesto – Ogni nodo SALES partecipa attivamente alla propagazione delle query di contesto e dei dati. In particolare, ogni nodo mobile, sia CUN che SUN, può dover distribuire le query ricevute sia ai propri pari che al livello superiore. Similmente, un BN riceve le query dai CUN serviti e le propaga verso pari e CN. Attualmente, SALES distribuisce le query di contesto ricevute una a una, introducendo così un overhead proporzionale al numero totale di query vive nel sistema. Tuttavia, le query emesse nella stessa area geografica presentano di solito similarità e coperture parziali/totali che possono essere sfruttate per ridurre il numero di query effettivamente

distribuito. Ad esempio, si pensi a una query che richiede tutte le stampanti presenti su un piano e a un'altra che richiede le stesse informazioni solo per una particolare aula: ovviamente, la distribuzione della seconda query non è necessaria dato che la prima produrrà già l'instradamento dei dati necessari. Pertanto, scopo fondamentale di tale tesi è di investigare possibili strategie di rappresentazione e di aggregazione delle query con lo scopo fondamentale di ridurre il numero totale di query propagate.

5. Introduzione di un sistema di localizzazione – Attualmente, SALES utilizza un sistema di localizzazione fittizio. Obiettivo principale di questa tesi è di integrare un sistema di localizzazione reale basato sui beacon emessi dagli APs IEEE 802.11 e di rendere i protocolli di gestione adattativi secondo la tecnologia wireless e la mobilità dei dispositivi. Inoltre, dato che i SUN non hanno accesso all'infrastruttura fissa, si intende investigare sistemi di localizzazione collaborativi in cui i CUN condividono i propri dati di localizzazione, funzionando come ancore virtuali. Come vincolo obbligatorio, il sottosistema di localizzazione deve essere basato sulle API JSR-179 Location.

6. Rappresentazione delle query di contesto – In SALES, l'ottimizzazione della rappresentazione utilizzata per la propagazione delle query di contesto è fondamentale: è necessario adottare rappresentazioni probabilistiche che consentano di ridurre la dimensione dei messaggi scambiati, ma allo stesso tempo garantire un numero di falsi positivi contenuti per evitare la diffusione di dati non utili. Attualmente, SALES utilizza una rappresentazione basata su filtri di Bloom: lo scopo principale di questa tesi è quello di esaminare altre soluzioni presenti in letteratura per capire cosa offrano in più rispetto ai filtri di Bloom e come possano essere integrate.

7. Sicurezza – Gli ambienti cooperativi introducono numerose problematiche di sicurezza dovute all'autenticazione dei dispositivi, all'integrità, alla riservatezza e alla disponibilità delle informazioni. SALES ipotizza un ambiente completamente fidato in cui sia le query di contesto che i dati sono distribuiti come testo in chiaro. Per di più, ogni singolo CUN ha completo accesso al traffico dei SUNs serviti. Anche se possiamo concedere fiducia ai nodi contenuti nell'infrastruttura fissa, l'area utente rimane completamente scoperta e vulnerabile a possibili attacchi. Per questo, obiettivo principale di questa tesi è di identificare le possibili minacce alla sicurezza di SALES e di proporre possibili soluzioni ad alcune di esse.

Tesi disponibili per laurea triennale:

1. **Porting di SALES su piattaforma Android** – Il middleware SALES è interamente scritto per piattaforma Java SE 1.6. Scopo fondamentale di tale tesi è di effettuare il porting del middleware su piattaforma Google Android. Uno studio preliminare della portabilità è stato già effettuato ed ha evidenziato la necessità di effettuare alcune modifiche sul codice di SALES, principalmente dovute all'utilizzo di librerie e classi non disponibili sulla piattaforma Android. Al tempo stesso, bisognerà effettuare una fase di testing estiva per evidenziare moduli e/o componenti di SALES onerosi e non facilmente eseguibili su dispositivi mobili con risorse limitate. Per la realizzazione e il testing finale, saranno messi a disposizione del laureando cellulari basati su piattaforma Android.

2. **Estensione del supporto Bluetooth per la realizzazione di Scatternet** – La tecnologia Bluetooth impone molti limiti alla comunicazione tra nodi: per esempio, due dispositivi master non possono comunicare direttamente e una piconet può contenere solo 8 dispositivi attivi, master incluso. Il problema principale è che ogni CUN deve essere un master BT, e questo vieta le comunicazioni tra CUN su BT. Il problema è ancora più rilevante se consideriamo che i CUN hanno tradizionalmente una interfaccia BT ed una WiFi, e questa è già preclusa alla comunicazione con l'infrastruttura di SALES. L'obiettivo principale di questa tesi è di realizzare un semplice algoritmo, basato su soluzioni presentate in letteratura, per consentire la formazione di scatternet e la comunicazione inter-CUN.

3. **Configurazione dinamica dei dispositivi utenti** – Attualmente, la configurazione dei CUN e dei SUN è completamente statica. Vogliamo pertanto introdurre un protocollo di configurazione dinamica sui nodi utente per l'assegnazione degli indirizzi IP sulle interfacce di rete (quando necessario). L'obiettivo principale di questa tesi è quindi quello di introdurre protocolli di configurazione dinamici, tenendo conto di alcuni problemi derivanti dagli ambienti mobili. In particolare, si vogliono investigare tre differenti soluzioni. Nella prima soluzione, ogni CUN offre un servizio di DHCP verso i SUN serviti e ogni BN lavora allo stesso modo verso i CUN serviti; in tal modo, riusciamo a supportare sia ambienti dotati che privi di infrastruttura, anche se tale soluzione comporta l'introduzione di protocolli di coordinamento tra CUN per evitare conflitti sull'indirizzamento. Nella seconda soluzione, è possibile utilizzare i nodi fissi per distribuire gli indirizzi sia a CUN sia ai SUN: in particolare, utilizzando i CUN come DHCP relay, riusciremmo a garantire subito l'assenza di conflitti sugli indirizzi ma, come contro, non riusciremmo a sostenere ambienti privi di infrastruttura. Infine, nella terza soluzione, possiamo ipotizzare l'utilizzo di ambo le precedenti, sfruttando i CUN come DHCP server in ambienti privi d'infrastruttura e come DHCP relay in ambienti con infrastruttura.