



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

**PROPAGAZIONE DI MESSAGGI TRA VEICOLI
CON MODELLO URBANO REALISTICO**

RELATORE

PROF. CLAUDIO ENRICO PALAZZI

Co-RELATORE

DOTT. ARMIR BUJARI

CANDIDATO

MARCO ROMANELLI

ANNO ACCADEMICO 2016-2017

AI SOLITI VECCHI AMICI

Abstract

Le reti veicolari (*Vehicular Ad-hoc Networks*, VANET) stanno riscuotendo un crescente interesse grazie ai recenti progressi tecnologici e ai molti campi d'applicazione possibili, dalla sicurezza dei veicoli e delle strade allo sviluppo di servizi multimediali intra-veicolari. All'intero di questo contesto è stato ideato Fast Broadcast, un protocollo che sfrutta una stima del raggio trasmissivo per la rapida propagazione di informazioni. Per rappresentare al meglio il mezzo fisico si rende necessario lo sviluppo di modelli di propagazione del segnale che prendano in considerazione i vari effetti di degradazione, come la distanza, cammini multipli da riflessione, ombreggiatura da ostacoli, eccetera. Quest'ultima ricopre un ruolo essenziale se lo scenario considerato è urbano o suburbano, dove gli edifici costituiscono un'ostruzione che non può essere ignorata. Uno dei modelli per l'ombreggiatura da ostacoli presenti in letteratura sfrutta dati reali sulla topologia stradale e sulla geometria degli ostacoli per ottenere simulazioni più realistiche. Si andrà, quindi, a valutare il protocollo Fast Broadcast all'interno di due diversi scenari urbani reali. Successivamente, sarà presa in considerazione la possibilità che alcuni veicoli non siano in grado di partecipare allo scambio dei messaggi (per esempio non hanno capacità di comunicazione), includendo all'interno dello scenario una rete di sensori per aiutare la propagazione. Per permettere la possibilità che questi dispositivi siano posizionati a un'altezza diversa dal suolo si andrà a proporre un'estensione al modello a ostacoli che consideri lo spazio tridimensionale.

Indice

ABSTRACT	v
ELENCO DELLE FIGURE	ix
ELENCO DELLE TABELLE	xi
ELENCO DEGLI ALGORITMI	xiii
ELENCO DEGLI ACRONIMI	xv
1 INTRODUZIONE	I
2 PROTOCOLLO FAST BROADCAST	5
2.1 Fase di stima	6
2.2 Fase di inoltro	7
2.3 Estensione a due dimensioni	9
3 MODELLO A OSTACOLI	13
3.1 Modelli di radiopropagazione	13
3.2 Modellazione di ostacoli	14
3.3 Il modello matematico	16
3.4 L'implementazione	17
3.5 Estensione a tre dimensioni	19
4 APPLICATIVI	25
4.1 Network Simulator 3	25
4.2 Simulation of Urban MObility	27
5 SIMULAZIONI	31
5.1 Scenario a griglia	31
5.1.1 Risultati	33
5.2 Scenario urbano reale	38
5.2.1 Scenario Los Angeles	39
5.2.2 Scenario Padova	45
5.3 Scenario urbano con sensori	49
5.3.1 Risultati	50

6 CONCLUSIONI	55
APPENDICE A TABELLE DEI RISULTATI	57
A.1 Scenario a griglia	57
A.2 Scenario Los Angeles	59
A.3 Scenario Padova	61
A.4 Scenario con sensori	63
BIBLIOGRAFIA	64

Elenco delle figure

2.1	Esempio di propagazione di un messaggio in una e due dimensioni (fonte: TheDrive e Car2Car).	II
3.1	Esempio di scenario urbano.	17
3.2	File dati estratto da OSM in cui compaiono informazioni sull'altezza di un edificio.	20
3.3	Rappresentazione di un edificio in cui è presente un valore di l'altezza. . . .	21
3.4	Rappresentazione in 2D e 3D degli edifici utilizzati per l'esempio.	24
3.5	Degradazione del valore RSS mentre il secondo veicolo è in movimento. Ogni linea indica una trasmissione, mentre il colore rappresenta l'RSS. . . .	24
4.1	Un estratto del file dati generato da OSM.	28
4.2	Un estratto del file dati sugli edifici dopo la conversione con Polyconvert. . .	28
4.3	Scenario urbano raffigurante il centro di Padova (IT) simulato con SUMO; a destra un dettaglio con i veicoli presenti.	29
4.4	Procedimento per estrarre le informazioni da OSM e generare i file per la simulazione con ns-3.	29
5.1	Scenario a griglia: copertura dei veicoli in totale e sulla circonferenza. . . .	35
5.2	Scenario a griglia: numero di messaggi di inoltro inviati e ricevuti.	36
5.3	Scenario a griglia: numero di salti.	37
5.4	Una vista delle aree selezionate per le simulazioni; a sinistra la città di Padova e a destra Los Angeles (fonte: OSM). Il cerchio rosso evidenzia la posizione del nodo di partenza mentre i tre rimanenti rappresentano il perimetro della circonferenza e i due intervalli di confidenza.	40
5.5	Scenario L.A.: numero di salti.	42
5.6	Scenario L.A.: copertura dei veicoli in totale e sulla circonferenza.	43
5.7	Scenario L.A.: numero di messaggi di inoltro inviati e ricevuti.	44
5.8	Scenario Padova: copertura dei veicoli in totale e sulla circonferenza. . . .	46
5.9	Scenario Padova: numero di messaggi di inoltro inviati e ricevuti.	47
5.10	Scenario Padova: numero di salti.	48
5.11	Scenario sensori: copertura totale dei veicoli.	52
5.12	Scenario Padova: numero di messaggi di inoltro ricevuti.	53

Elenco delle tabelle

5.1	Configurazione dei parametri per le simulazioni.	32
5.2	Parametri della topologia per gli scenari urbani.	39
5.3	Parametri della topologia per lo scenario con sensori.	50
5.4	Statistiche sull'altezza degli ostacoli.	51
5.5	Scenario sensori.: numero di salti.	52
A.1	Scenario a griglia: copertura dei veicoli in totale.	57
A.2	Scenario a griglia: copertura dei veicoli sulla circonferenza.	58
A.3	Scenario a griglia: numero di salti.	58
A.4	Scenario a griglia: numero di messaggi di inoltro inviati.	58
A.5	Scenario a griglia: numero di messaggi di inoltro ricevuti.	58
A.6	Scenario L.A.: copertura dei veicoli in totale.	59
A.7	Scenario L.A.: copertura dei veicoli sulla circonferenza.	59
A.8	Scenario L.A.: numero di salti.	59
A.9	Scenario L.A.: numero di messaggi di inoltro inviati.	60
A.10	Scenario L.A.: numero di messaggi di inoltro ricevuti.	60
A.11	Scenario Padova: copertura dei veicoli in totale.	61
A.12	Scenario Padova: copertura dei veicoli circonferenza.	61
A.13	Scenario Padova: numero di salti.	61
A.14	Scenario Padova: numero di messaggi di inoltro inviati.	62
A.15	Scenario Padova: numero di messaggi di inoltro ricevuti.	62
A.16	Scenario con sensori: copertura dei veicoli in totale.	63
A.17	Scenario con sensori: numero di messaggi di inoltro ricevuti.	63

Elenco degli algoritmi

2.1	Invio di un messaggio Hello.	7
2.2	Ricezione di un messaggio Hello.	7
2.3	Invio di un messaggio di inoltro.	9
2.4	Gestione di un messaggio di inoltro.	9
3.1	Algoritmo per determinare il numero di intersezioni con i bordi dell'ostacolo e la distanza interna percorsa fra due punti.	19
3.2	Modifica all'algoritmo per includere la terza dimensione.	22
5.1	Semplice algoritmo di inoltro.	49

Elenco degli acronimi

BSP	Partizione binaria dello spazio (<i>Binary Search Partition</i>)
CGAL	Computational Geometry Algorithms Library
CMFR	<i>Current-turn Maximum Front Range</i>
CMBR	<i>Current-turn Maximum Back Range</i>
CSMA	<i>Carrier Sense Multiple Access</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
EGRESS	<i>Enviroment for Generating REalistic Scenarios for Simulations</i>
LMBR	<i>Last-turn Maximum Back Range</i>
LMFR	<i>Last-turn Maximum Front Range</i>
LOS	Visuale (<i>Line of sight</i>)
MANET	Reti Ad-hoc mobili (<i>Mobile Ad-hoc Networks</i>)
MRP	Modello di radiopropagazione
NLOS	Non in visuale (<i>Non Line Of Sight</i>)
ns-3	Network Simulator 3
OLOS	Visuale ostacolata (<i>Obstructed Line of Sight</i>)
OSM	OpenStreetMap
RSS	Potenza del segnale in ricezione (<i>Received signal strength</i>)
SUMO	Simulator for Urban Mobility
VANET	Reti Ad-hoc veicolari (<i>Vehicular Ad-hoc Networks</i>)

1

Introduzione

I progressi tecnologici dell'ultimo decennio nell'hardware, nel software e nelle telecomunicazioni hanno permesso la larga diffusione di unità computazionali all'interno degli autoveicoli. Questo ha portato a un incremento dell'interesse della ricerca scientifica in questo campo e, in particolare, sullo studio delle reti veicolari o VANET (*Vehicular Ad-hoc Networks*).

Le VANET rappresentano una famiglia delle reti Ad-hoc mobili (*Mobile Ad-hoc Network*, MANET) nelle quali i movimenti sono strutturati, i veicoli possono essere consapevoli della propria posizione nello spazio (ad esempio tramite geolocalizzazione GPS) e spesso sono equipaggiati con attrezzatura che permette le comunicazioni inter-veicolari (es. antenne radio). Queste reti sono alla base di una vasta gamma di necessità, che spaziano dalla sicurezza stradale alle applicazioni multimediali, dalla distribuzione dei dati sul traffico allo sviluppo

di una rete infrastrutturale urbana.

Una caratteristica comune è la necessità di un sistema che propaghi le informazioni con il minor ritardo possibile; per esempio un pericolo inatteso sulla careggiata potrebbe causare l'impossibilità di proseguire lungo il percorso ed è necessario informare i veicoli che seguono. Oppure all'interno di una trasmissione di contenuti multimediali (giochi multiplayer, video) fra due o più veicoli alte latenze potrebbero inficiare la fruizione dei contenuti [1] [2]. Una delle soluzioni proposte si chiama Fast Broadcast: un protocollo per ridurre il tempo di propagazione di un messaggio da una sorgente a una destinazione tramite una stima del raggio trasmisivo effettivo e il conseguente utilizzo di questa per ridurre il numero di salti necessari al raggiungimento dell'obiettivo [3].

La valutazione di protocolli di questo tipo su scenari urbani con migliaia di agenti risulta problematica principalmente a causa dei costi proibitivi (oltre alle implicazioni sulla privacy) che comporterebbe una sperimentazione nel mondo reale. Spesso i ricercatori ricadono, quindi, sull'utilizzo di modelli simulati eventualmente integrati con risultati ottenuti da esperimenti sul campo in ambienti ridotti e controllati. In particolare, l'accuratezza dei modelli di propagazione e di mobilità rappresenta la chiave per una buona valutazione delle prestazioni dei protocolli di rete veicolari [4]. All'interno di scenari urbani e suburbani gli edifici ostruiscono la naturale propagazione di un segnale radio nello spazio e, di conseguenza, al fine di eseguire simulazioni più affidabili questa ostruzione non può essere ignorata. Una fra le diverse soluzioni proposte negli anni permette di calcolare l'attenuazione del segnale per un singolo edificio in funzione del numero di intersezioni con le pareti esterne e la distanza interna percorsa [5]. Qualche anno più tardi, il modello è stato ripreso, implementato per uno software di simulazioni e integrato con informazioni reali sulla geometria degli edifici e sulla topologia stradale [6]. Un aspetto positivo di questo modello risiede nella sua implementazione, rilasciata come modulo aggiuntivo per Network Simulator 3 (ns-3) [7], ultima

versione della nota famiglia di simulatori di rete, successore di ns-2 e l'unica a essere attualmente mantenuta. ns-3 è assieme a OMNeT++ [8] uno dei simulatori più diffusi in ambito accademico e di ricerca, grazie alla loro natura *open-source*, all'ambiente di configurazione delle simulazioni in C++ e a una comunità di sviluppo attiva. Considerato questo e la presenza di una implementazione sia del modello a ostacoli che del protocollo Fast Broadcast per ns-3 la scelta del simulatore da utilizzare è ricaduta su quest'ultimo.

I principali contributi di questo lavoro si possono così riassumere.

- La valutazione del protocollo Fast Broadcast in un contesto urbano reale, utilizzando mappe geografiche e dati sugli ostacoli presenti di due città con caratteristiche topologiche diverse (Sezione 5.2).
- La proposta e l'analisi di uno scenario nel quale alcuni veicoli non sono in grado di partecipare allo scambio dei messaggi (ad esempio per l'impossibilità di comunicare) e nel quale è prevista la presenza di sensori che coopera nel processo di disseminazione. (Sezione 5.3).
- Lo studio di un modello per l'ombreggiatura da ostacoli presente in letteratura e una sua possibile estensione nello spazio tridimensionale (Capitolo 3).

Il presente lavoro è strutturato come segue. Nel capitolo successivo (Capitolo 2) verrà presentato il protocollo Fast Broadcast e una proposta di estensione in due dimensioni. Il Capitolo 3 analizzerà il modello a ostacoli partendo da una breve panoramica sulle soluzioni presenti in letteratura, entrerà nel dettaglio del modello adottato per concludere presentando una possibile soluzione per rendere il modello tridimensionale. Il Capitolo 4 darà una veloce panoramica sui software utilizzati e il metodo utilizzato per la creazione degli scenari. Il Capitolo 5 illustrerà i diversi gruppi di simulazioni effettuati e i relativi risultati ottenuti. Infine, il Capitolo 6 si concluderà riassumendo i risultati ed evidenziando alcuni possibili sviluppi.

2

Protocollo Fast Broadcast

La tecnica del Fast Broadcast [3] è nata con lo scopo di propagare un messaggio fra veicoli utilizzando meno salti possibile; questo è reso possibile tramite una stima dinamica del raggio trasmissivo reale. Il protocollo si compone di due fasi: la prima, chiamata “Fase di stima”, permette a ogni veicolo di avere una stima aggiornata del proprio raggio trasmissivo da utilizzare nella seconda fase. La seconda, invece, è detta “Fase di inoltro” e viene attivata nel momento in cui si rende necessario inviare un messaggio al resto dei veicoli presenti nell’area di interesse.

2.1 FASE DI STIMA

Durante questa fase ogni veicolo effettua una stima del suo raggio trasmissivo, sia di fronte che alle sue spalle, attraverso l'invio di messaggi *Hello*. Per ottenere una stima sempre aggiornata il tempo è suddiviso in turni e le informazioni raccolte durante un certo turno rimangono attive per tutto il turno successivo, per poi essere scartate. Una breve durata dei turni permette di cogliere meglio variazioni del raggio trasmissivo mantendendo tuttavia un elevato scambio di messaggi; gli autori suggeriscono un tempo pari a un secondo.

Le informazioni sulla stima sono rappresentate dai campi *Current-turn Maximum Front Range* (CMFR) e *Current-turn Maximum Back Range* (CMBR). Il primo esprime la stima della massima distanza in avanti dalla quale un altro veicolo nell'area di interesse può ricevere messaggi che provengono dal veicolo considerato. Il secondo, invece, stima la massima distanza all'indietro. Questi valori sono costantemente aggiornati con i valori ricevuti nei messaggi Hello fino alla fine del turno, dopodiché vengono salvati nei campi *Latest-turn Maximum Front Range* (LMFR) e *Latest-turn Maximum Back Range* (LMBR) rispettivamente. Questo permette di combinare una stima calcolata nel tempo (con diversi messaggi Hello) con le ultime informazioni sul mezzo trasmissivo.

Nel dettaglio, la procedura di invio di un messaggio Hello prevede inizialmente che il veicolo aspetti un tempo d'attesa casuale e, dopo aver verificato l'assenza di altre trasmissioni in corso e/o collisioni, proceda all'invio del messaggio contenente la stima del raggio trasmissivo in avanti. Al messaggio vengono aggiunte altre informazioni utili al protocollo, come ad esempio la posizione aggiornata del veicolo.

Si fa notare che per ogni turno non più di un messaggio Hello viene inviato fra veicoli all'interno dell'area delimitata dal raggio trasmissivo. In questo modo il numero totale di messaggi Hello generati rimane limitato nell'ordine di $\mathcal{O}(1)$.

In fase di ricezione, un veicolo determina la distanza fra lui e chi ha spedito il messaggio: se proviene da davanti sarà aggiornato il campo CMFR, altrimenti se proviene dal retro verrà aggiornato il campo CMBR. In ogni caso, la nuova stima sarà calcolata come il massimo fra il vecchio valore, la distanza fra i due veicoli e la stima inviata dall'altro veicolo (e ricevuta nel messaggio di Hello).

Gli algoritmi 2.1 e 2.2 riassumono l'invio e la ricezione di un messaggio Hello.

Algoritmo 2.1 Invio di un messaggio Hello.

```

1: per ogni turno fai
2:   tempoInvio ← generaCasualmente(dimTurno)
3:   aspetta(tempoInvio)
4:   se  $\neg$  (RilevatomsgInoltro()  $\vee$  RilevataCollisione()) allora
5:     msgInoltro.portataMassima ← max(LMR, CMR)
6:     msgInoltro.posizione ← rilevaPosizione()
7:     inviaInBroadcast(msgInoltro)

```

Algoritmo 2.2 Ricezione di un messaggio Hello.

```

1: portata ← msgInoltro.portataMassima
2: posizioneMittente ← helloMsg.posizione
3: posizione ← rilevaPosizione()
4: distanza ← calcolaDistanza(posizione, posizioneMittente)
5: CMR ← max(CMR, distanza, portata)

```

2.2 FASE DI INOLTRO

Lo scopo di questa fase consiste nel determinare il veicolo migliore per inoltrare un messaggio e viene “attivata” nel momento in cui ci sia bisogno di propagare velocemente una certa informazione agli altri veicoli. Questo viene fatto sfruttando la conoscenza sulla stima del raggio trasmissivo ottenuta nella fase precedente: ogni veicolo si assegna una priorità nell'inoltro basandosi sulla distanza dal veicolo che ha trasmesso il messaggio; la priorità cresce all'aumentare

della distanza e sarà quindi il veicolo più lontano ad avere maggiore probabilità di effettuare l'inoltro. La fase ha inizio con la generazione di un messaggio di *Broadcast* (inoltro), dove si trovano due valori di interesse: la posizione del veicolo e la stima del raggio trasmissivo; quest'ultima, caratteristica proprio di questo protocollo, rappresenta quanto all'indietro ci si aspetta che arrivi il segnale prima che diventi troppo debole da non essere rilevabile.

Questo valore sarà poi sfruttato per determinare quale veicolo dovrà inoltrare il messaggio in modo da minimizzare il numero di salti (*hop*) e di conseguenza ridurre il tempo di propagazione. Nel dettaglio, quando un veicolo deve inviare o inoltrare un messaggio Broadcast calcola il valore del campo MaxRange come il massimo fra l'LMBR e il CMBR.

In fase di ricezione, un veicolo attende una quantità di tempo determinata in maniera casuale partendo dal valore di una finestra di contesa (*contention window*). Il valore di questa finestra varia fra un valore minimo (*CWMin*) e un valore massimo (*CWMax*), in funzione della distanza dal veicolo che ha inviato il messaggio e la stima del raggio trasmissivo secondo la Formula 2.1. È facile vedere come maggiore è la distanza dal mittente e minore è l'intervallo d'attesa.

$$\left\lfloor \left(\frac{\text{PortataMassima} - \text{Distanza}}{\text{PortataMassima}} \times (\text{CWMax} - \text{CWMin}) \right) + \text{CWmin} \right\rfloor \quad (2.1)$$

Se durante l'attesa viene ricevuto lo stesso messaggio proveniente dai veicoli che seguono, ciò significa che il messaggio è già stato propagato in avanti e quindi non è necessario che il veicolo corrente lo inoltri. Se, invece, a inviarlo è un veicolo che precede significa che il messaggio è già stato inoltrato da un altro mezzo; la procedura quindi dev'essere fatta ripartire includendo i parametri appena ricevuti. Nel caso in cui l'attesa finisce senza aver ricevuto nessun altro messaggio (o collisioni), il veicolo può procedere con l'inoltro includendo nell'invio la stima del raggio trasmissivo.

Gli algoritmi 2.3 e 2.4 riassumono quanto detto.

Algoritmo 2.3 Invio di un messaggio di inoltro.

- 1: `msgInoltro.portataMassima` $\leftarrow \max(\text{LMR}, \text{CMR})$
 - 2: `msgInoltro.posizione` $\leftarrow \text{rilevaPosizione}()$
 - 3: `inviaInBroadcast(msgInoltro)`
-

Algoritmo 2.4 Gestione di un messaggio di inoltro.

- 1: `cwnd` $\leftarrow \text{calcolaCwnd}()$
 - 2: `rcw` $\leftarrow \text{generaCasualmente}(\text{cwnd})$
 - 3: `aspetta(rcw)`
 - 4: **se** `stessoMessaggioDaDietro()` **allora**
 - 5: `esci()`
 - 6: **altrimenti se** `stessoMessaggioDaDavanti()` **allora**
 - 7: `riavviaProceduraInoltro()`
 - 8: **altrimenti**
 - 9: `msgInoltro.portataMassima` $\leftarrow \max(\text{LMR}, \text{CMR})$
 - 10: `msgInoltro.posizione` $\leftarrow \text{rilevaPosizione}()$
 - 11: `inviaInBroadcast(msgInoltro)`
-

2.3 ESTENSIONE A DUE DIMENSIONI

Nello studio originale del protocollo i veicoli erano distribuiti lungo una strada rettilinea con un singolo senso di marcia. Ciò comportava che presi due veicoli fosse facile definirne la provenienza e la direzione, ossia chi dei due fosse davanti e chi dietro e di conseguenza a chi inoltrare il messaggio. Se si considera uno scenario nel quale i veicoli circolano in una rete stradale a due o più sensi di marcia, quali sono i veicoli che si trovano davanti e quali invece dietro? Si prenda l'esempio in Figura 2.1: se il primo veicolo a sinistra ha bisogno di propagare un messaggio il più lontano possibile, allora è facile intuire che a inoltrarlo dovrà essere il veicolo più a destra. Ora invece si consideri lo scenario nell'immagine sottostante: se è il veicolo A a dover inviare il messaggio, quale mezzo dovrà inoltrarlo, B o C? Non è

più sufficiente la differenziazione davanti-dietro. Una soluzione a questo problema è stata proposta in [9]: qui la distinzione viene effettuata basandosi sulla posizione del veicolo che ha generato il messaggio. La fase di stima non richiede modifiche sostanziali: viene eliminato il concetto di direzione e conseguentemente si utilizza solo uno dei due campi CMBR e CMFR (creando così un CMR e un LMR). Nella fase di inoltro, la soluzione prevede di controllare la distanza del mittente e la distanza del destinatario rispetto all'origine del messaggio. La prima modifica necessaria è quindi la presenza nel messaggio di inoltro della posizione del veicolo che l'ha generato (oltre alla posizione del mezzo che lo inoltra). In fase di ricezione, si controlla che la distanza origine-mezzo sia maggiore o uguale a quella origine-mittente: in tal caso il veicolo è un candidato per inoltrare il messaggio; se così non avviene il messaggio viene scartato.



Figura 2.1: Esempio di propagazione di un messaggio in una e due dimensioni (fonte: TheDrive e Car2Car).

3

Modello a ostacoli

3.1 MODELLI DI RADIOPROPAGAZIONE

Un modello di radiopropagazione (MRP) simula gli effetti dell'attenuazione del segnale radio (segnali elettromagnetici nello spazio libero o etere, in contrasto con la propagazione guidata) dovuta alla distanza, cammini multipli per effetto della riflessione, ombreggiatura causata dalla presenza di ostacoli. L'utilizzo di un'idonea rappresentazione per questo tipo di ostacoli è, quindi, necessaria nel contesto di simulazioni di reti VANET in ambienti urbani e suburbani. Nel corso degli anni, diversi MRP sono stati proposti. Il più semplice di questi si chiama modello a disco unitario (*unit-disk model*), nel quale i veicoli possono comunicare fra loro se si trovano entro una certa soglia di distanza, mentre non possono altrimenti [10]. Un modello molto utilizzato nelle simuazioni di VANET è il modello a doppio raggio (*Two-ray*

ground-reflection model), nel quale il segnale in fase di ricezione è composto da una componente in linea diretta e una seconda derivante dalla riflessione causata dal terreno [11]. In [12] e in [13], modelli più sviluppati prendono in considerazione anche le proprietà riflettenti delle superfici e degli ostacoli.

Tuttavia, un approccio diretto di questo tipo difficilmente scala al numero di nodi necessario in un classico scenario VANET e per questo motivo spesso tali modelli si affidano a una fase di pre-elaborazione che può richiedere tempi elevati [14]. Per ovviare a questo problema, alcune proposte astraggono dalle informazioni sui singoli edifici, modellando l'ambiente urbano in modo omogeneo così da creare un modello analitico per l'ombreggiatura [15]. Questo tipo di modelli riducono il costo computazionale e generalmente si raffrontano bene con i risultati reali; ciò nonostante, non riescono a catturare effetti a livello mesoscopico come eventuali spazi fra edifici che permetterebbero trasmissioni a corto raggio [16].

Sono affetti dallo stesso problema anche i modelli (puramente) probabilistici, in quanto non tengono in considerazione la geometria urbana sottostante. Questi modellano l'effetto dell'ombraggiatura tramite distribuzione stocastiche, fra cui Rice, Rayleigh, Nakagami-m, lognormale e Weibull [10] [17].

3.2 MODELLAZIONE DI OSTACOLI

Nelle simulazioni di alcuni scenari, come possono essere le VANET, un'accurata rappresentazione della topologia dell'ambiente è necessaria in quanto limita non solo la mobilità dei veicoli ma interferisce anche con le trasmissioni radio [18] [19]. L'attenuazione radio viene spesso modellata in modo deterministico basandosi principalmente sulla distanza della visuale (*Line Of Sight*, LOS) fra i veicoli, combinando eventualmente un modello stocastico per considerare l'ombreggiatura.

Negli ultimi anni, diversi studi hanno cercato di rappresentare meglio l'effetto ombreggiatura causato dagli edifici. In [16] gli autori utilizzavano tecniche di geocodifica inversa (*reverse geocoding*) per estrapolare informazioni sulla topologia della rete stradale in stile Manhattan. Da qui veniva derivata la geometria degli edifici utilizzata per differenziare i casi in cui i due veicoli comunicanti avessero la visuale libera (LOS) o meno (*Non Line Of Sight*, NLOS) e calcolare di conseguenza la potenza del segnale in ricezione. La configurazione a griglia non è tuttavia applicabile in ogni scenario urbano, speciamente in città italiane storiche, dove la rete stradale è molto irregolare. In [4], invece, le informazioni sugli edifici erano codificate direttamente all'interno dell'ambiente sviluppato, chiamato EGRESS (*Enviroment for Generating REalistic Scenarios for Simulations*), e permettevano di definire tre tipologie di edifici e altrettanti percorsi per simulare la mobilità dei nodi; il modello permetteva anche di configurare, con alcuni limiti, diversi tipi di materiale dei pavimenti e dei muri degli edifici. Questa categorizzazione non permetteva però una buona scalabilità, se si considera che in pochi chilometri quadrati di una città media ci sono migliaia di edifici. Gli autori di [6] integravano nel noto simulatore di reti Network Simulator 3 (ns-3) [7] un modello a ostacoli empirico precedentemente sviluppato [5] ed estraevano le informazioni per la geometria degli edifici da una piattaforma gratuita chiamata OpenStreetMap (OSM) [20]; queste venivano successivamente elaborate da un software per la simuazione della mobilità di veicoli e utilizzate nel simulatore, differenziando i dati sugli edifici e quelli sulla mobilità dei veicoli; il modello di propagazione teneva conto della distanza percorsa dal segnale internamente all'ostacolo e del numero di muri esterni che questo attraversava. La semplicità computazionale, l'assenza di pre-elaborazioni, la possibilità di integrare informazioni reali sull'ambiente e l'integrazione con ns-3 erano alcuni dei punti a favore di questo modello.

3.3 IL MODELLO MATEMATICO

Il modello a ostacoli utilizzato viene descritto per la prima volta in [5], dove viene presentato come un'estensione dei cosiddetti modelli di attenuazione del segnale. In generale, questi possono essere scritti nella forma dell'Equazione 3.1, dove P è la potenza del trasmettitore (e ricevitore), G i guadagni delle antenne e L il termine che indica le attenuazioni dovute alla trasmissione.

$$P_r[dBm] = P_t[dBm] + G_t[dB] + G_r[dB] - \sum L_x[dB] \quad (3.1)$$

$$L_{TwoRayGround} = 10 \lg \left(\frac{d^4 L}{h_t^2 h_r^2} \right) \quad [dB] \quad (3.2)$$

$$L_{LogNorm} = 10 \lg (X_\sigma) \quad [dB] \quad (3.3)$$

Questi modelli possono essere espressi come componenti di L e concatenati in modo da ottenere l'attenuazione totale risultante. Ad esempio, le equazioni 3.2 e 3.3 illustrano rispettivamente i modelli a doppio raggio e log-normale. Il modello in esame rappresenta la perdita di segnale dovuta a un ostacolo estendendo l'Equazione 3.1 col termine $L_{s,o}$, che unisce l'attenuazione causata dai bordi dell'ostacolo (attenuazione per-parete) e quella derivante dalla superficie interna (attenuazione per-metro):

$$L_{s,o} = \alpha n + \beta d_o \quad (3.4)$$

dove n è il numero di volte che il bordo dell'ostacolo viene intersecato dalla visuale e d_o è la distanza, in metri, attraversata all'interno dell'ostacolo. Il primo dei due parametri di calibrazione, α , espresso in decibel per metro (dB/m), descrive l'attenuazione che la trasmissione subisce a causa delle pareti esterne dell'ostacolo. Il secondo, β , espresso in decibel (dB per

parete), serve come misura di approssimazione della struttura interna dell'ostacolo. Tramite la regolazione di questo valore è possibile rappresentare diverse tipologie di ostacoli. Prendendo il caso di edifici in un ambiente cittadino, i valori predefiniti per i due parametri sono $\alpha = 9$ dB per parete e $\beta = 0,4$ dB/m. Nell'esempio raffigurato in Figura 3.1 la visuale fra i due veicoli interseca $n = 6$ muri e una distanza interna di $d_o = 32$ m e se si sostituiscono i valori in 3.4 si ottiene: $L_{s,o} = 9 * 6 + 0,4 * 32 = 66,8$ dB. Solo considerando questa attenuazione la potenza del segnale si riduce di molto, considerando che, a titolo esemplificativo, nello standard 802.11b a 11 Mb/s il limite per la ricezione è di -89 dBm; si evince come, in questo caso, sia difficile che la trasmissione da un veicolo possa essere ricevuta con successo da un secondo veicolo.

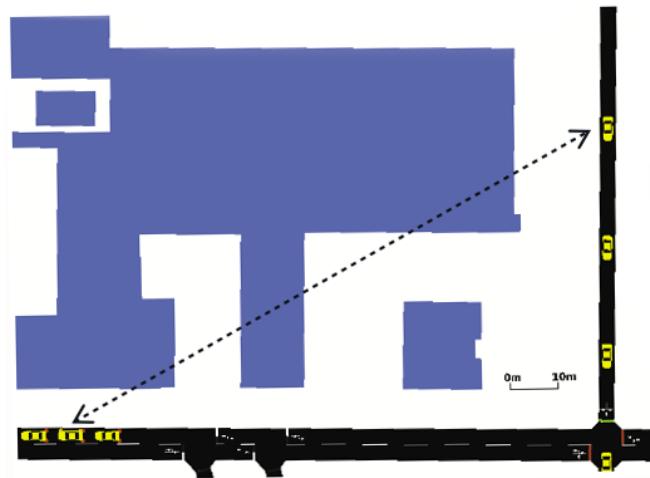


Figura 3.1: Esempio di scenario urbano.

3.4 L'IMPLEMENTAZIONE

Ripredendo il modello ideato in [5] e descritto nella sezione precedente, in [6] gli autori ne hanno sviluppato un'efficiente implementazione per il software di simulazioni ns-3, chiamata

semplicemente *Obstacle Shadowing* (ombreggiatura da ostacoli). Un ostacolo è rappresentato come un poligono bidimensionale, internamente riprodotto da una lista di vertici (x, y) , dove questo delimita il confine (i bordi) dell'ostacolo. Nella sua implementazione, distribuita come modulo aggiuntivo per il simulatore ns-3, sono state utilizzate le Computational Geometry Algorithms Library (CGAL), libreria scritta in C++ contenente algoritmi di geometria computazionale. Gli ostacoli sono raggruppati in una struttura algebrica dedicata, chiamata partizione binaria dello spazio (*Binary Search Partition*, BSP), utilizzata per motivi di ottimizzazione.

Nel dettaglio, quando bisogna calcolare l'attenuazione del segnale fra due elementi, nodi in ns-3, si prende in considerazione un riquadro di delimitazione che include i due e lo si estende di un certo raggio, ad esempio 200 metri; i potenziali ostacoli si trovano selezionando quelli il cui punto centrale risiede all'interno del riquadro, utilizzando il BSP per la ricerca. Ogni ostacolo selezionato viene controllato per verificare se interseca la visuale (in questo caso *obstructed-line-of-sight*, OLOS), fra i due nodi. Se questa verifica ha esito positivo viene calcolato il numero di intersezioni, la distanza interna percorsa e, successivamente, restituita la quantità di attenuazione secondo la Formula 3.4; questo processo è riassunto nell'Algoritmo 3.1. Infine, come ulteriore ottimizzazione, il valore calcolato viene salvato e riutilizzato nel caso i nodi non si siano spostati per più di 0,1 metri. Per quanto riguarda la struttura del codice in ns-3, il modello è implementato in tre classi: *Obstacle* contiene la rappresentazione geometrica dell'ostacolo come anche i parametri dell'attenuazione per-metro e per-muro. *Topology* legge il file contenente le informazioni sugli ostacoli (vedere capitolo successivo) e per ognuno di questi lo crea e lo posiziona all'interno della struttura dati BSP. La terza, *ObstacleShadowingPropagationLossModel*, estende ns-3 aggiungendo il modello di propagazione a ostacoli e richiama *Topology* nel momento in cui si rende necessario calcolare l'attenuazione del segnale fra due nodi, utilizzando l'Algoritmo 3.1. Il tutto è incluso in un

Algoritmo 3.1 Algoritmo per determinare il numero di intersezioni con i bordi dell'ostacolo e la distanza interna percorsa fra due punti.

```

1: procedura GETOBSTUCTEDDISTANCEBETWEEN( $p_1, p_2, B$ )
2: Input:  $p_1, p_2$ : posizione dei due veicoli;  $B$ : partizione binaria dello spazio (BSP) di ostacoli.
3: Output: Distanza interna percorsa  $m_o$  e il numero di intersezioni con i bordi  $n$ .
4:    $m_o \leftarrow 0$ ;  $n \leftarrow 0$ 
5:   Inizializza la portata massima  $r$ : distanza dal punto  $p_1$  o  $p_2$  al centro di un ostacolo, utilizzata per filtrare il sottoinsieme di ostacoli sufficientemente vicini.
6:   Crea un riquadro di delimitazione  $b$  per  $p_1$  e  $p_2$  ed estendila di  $r$  in tutte le direzioni.
7:   Calcola l'insieme di potenziali ostacoli  $O$  da quelli all'interno di  $b$  in  $B$ .
8:   per ogni ostacolo  $o \in O$  fai
9:     se (distanza( $p_1, o.\text{centro}$ )  $< r$ )  $\vee$  (distanza( $p_2, o.\text{centro}$ )  $< r$ ) allora
10:      per ogni spigolo  $e \in o$  fai
11:        se  $s$  interseca un raggio da  $p_1$  a  $p_2$  allora
12:           $n \leftarrow n + 1$ 
13:          Salva la distanza minima e massima da  $\{p_1, p_2\}$  al punto d'intersezione.
14:           $m_o \leftarrow m_o + \text{differenza fra i valori min e max calcolati al passo precedente.}$ 
restituisci  $m_o$  e  $n$ 
```

nuovo modulo chiamato obstacle (ostacolo).

3.5 ESTENSIONE A TRE DIMENSIONI

Come detto in precedenza, gli oggetti sono rappresentati da poligoni bidimensionali e, conseguentemente, il modello descritto lavora in una proiezione bidimensionale dell'ambiente (tridimensionale) di ns-3, prendendo quindi in considerazione solo le prime due componenti x e y della posizione di ogni nodo.

Uno dei principali motivi che portò gli ideatori del modello a questa scelta fu la mancanza di sufficienti informazioni tridimensionali sulla piattaforma dalla quale aquisivano i dati (OpenStreetMap, OSM). Grazie a una crescente diffusione di Internet sia a livello tecnologico che nella vita quotidiana delle persone, informazioni di questo tipo sono e saranno sempre

più disponibili. Dato, quindi, il supporto nativo alla terza dimensione di ns-3 (a differenza dei suoi predecessori) è naturale pensare di estendere il modello affinché tenga conto di questa componente e pertanto dell'esatta posizione del nodo nell'ambiente tridimensionale.

ALTEZZA DEGLI OSTACOLI Sebbene le forme degli ostacoli si possa descrivere esclusivamente in due dimensioni (punti bidimensionali che ne delineano il perimetro al suolo), OSM prevede la possibilità di definire l'altezza totale in metri e/o specificare il numero di piani (sopra e sotto il livello del suolo) nel caso di edifici. Queste informazioni possono essere sfruttate per creare una forma tridimensionale semplificata dell'ostacolo, sia che sia presente il vero valore dell'altezza sia che ci siano solo indicati il numero di piani, anche se in questo caso si tratterebbe solo di una stima e non della dimensione esatta.

```
<way id="62332429" version="13" timestamp="2016-11-30T11:53:34Z" changeset="44057933"
  uid="731498" user="Agno-phi">
  <nd ref="778692594"/>
  <nd ref="778692588"/>
  <nd ref="778692589"/>
  <nd ref="1809913218"/>
  <nd ref="778692593"/>
  <nd ref="778692594"/>
  <tag k="addr:city" v="Padova"/>
  <tag k="addr:housenumber" v="63"/>
  <tag k="addr:postcode" v="35121"/>
  <tag k="addr:street" v="Via Trieste"/>
  <tag k="alt_name" v="Torre Archimede; Tullio Levi Civita"/>
  <tag k="amenity" v="university"/>
  <tag k="building" v="university"/>
  <tag k="building:levels" v="8"/>
  <tag k="building:levels:underground" v="2"/>
  <tag k="name" v="Dipartimento di Matematica &quot;Tullio Levi-Civita&quot;"/>
  <tag k="operator" v="Università degli Studi di Padova"/>
  <tag k="website" v="http://www.math.unipd.it"/>
</way>
```

Figura 3.2: File dati estratto da OSM in cui compaiono informazioni sull'altezza di un edificio.

Nell'esempio indicato nel Figura 3.2, manca la misura dell'altezza ma è presente il numero di piani: è sufficiente moltiplicare questo numero, qui 8, per l'altezza media un piano, per esempio 2,7 metri, per ottenere una stima dell'altezza pari a 21,6 metri.

A questo punto è necessario decidere la rappresentazione interna dell'ostacolo tridimensionale e ci sono (almeno) due strade possibili: combinare la forma bidimensionale dell'ostacolo

```
<poly id="62332429" type="amenity" color="237,199,199" fill="1" layer="-1.00"
height="21.6" shape="1846.79,2221.98 1886.77,2209.99 1875.63,2173.09 1855.95,2178.99
1835.64,2185.08 1846.79,2221.98"/>
```

Figura 3.3: Rappresentazione di un edificio in cui è presente un valore di l'altezza.

e l'altezza per creare un poligono tridimensionale (con punti x, y, z) oppure scegliere di tenere le due informazioni separate e gestirle di conseguenza. Va però fatto notare che si conosce solo forma che questo ha al suolo, in quanto i punti sono bidimensionali, di conseguenza la forma tridimensionale sarebbe una proiezione in altezza della stessa. Mantere separate la forma e l'altezza permette, quindi, di ridurre i costi legati alla costruzione e alla memorizzazione dell'eventuale poliedro, senza perdita di informazioni.

MODIFICA ALL'ALGORITMO Il resto dell'algoritmo GetObstuctedDistanceBetween (Algoritmo 3.1) rimane sostanzialmente lo stesso e, nello specifico, la ricerca dei potenziali ostacoli rimane invariata. Infatti gli ostacoli che si interpongono fra due punti nello spazio tridimensionale saranno gli stessi che sul piano bidimensionale (creato dalla perdita della terza componente z) grazie alla forma semplificata dell'oggetto per i motivi esposti precedentemente. All'oggetto Obstacle è sufficiente aggiungere un campo con il valore dell'altezza.

Le modifiche significative si trovano nel calcolo del numero di intersezioni n , visto che per la distanza m_o basta aggiungere la componente z del punto di intersezione e il procedimento è lo stesso che nel caso bidimensionale. In origine, l'intersezione veniva calcolata fra il segmento (raggio) che unisce i due punti e lo spigolo dell'ostacolo considerato in quel momento, procedura poi iterata per ogni spigolo dell'ostacolo (riga 10). Ora bisogna trovare l'intersezione fra il raggio (segmento) e la parete dello spigolo corrente. Una soluzione soluzione consiste nel generare un piano in corrispondenza della parete, calcolare l'intersezione fra questo e il segmento e verificare, nel caso ci sia, che il punto trovato appartenga alla parete dell'oggetto e che non sia esterno. Per generare il piano a partire dal segmento dello spigolo

basta prendere i due vertici dello spigolo e un terzo vertice generato da uno dei due (a piacere) a cui si imposta la componente z diversa da 0. Per chiarire il concetto, si pensi a un semplice edificio: definita la forma alla base (al suolo) i muri sono perpendicolari al terreno e, a una certa altezza, si trova il tetto con la stessa forma della base; il punto d'intersezione che interessa si trova sulla faccia del muro corrente. L'ultimo passo, non necessario nel caso bidimensionale, consiste nel controllare se il raggio (fra i nodi) interseca la faccia superiore dell'oggetto (il tetto in un edificio), seguendo un procedimento simile ai casi delle pareti. Le modifiche sono riportate nell'Algoritmo 3.2.

Algoritmo 3.2 Modifica all'algoritmo per includere la terza dimensione.

```

1: procedura GETOBSTACLEDISTANCEBETWEEN( $p_1, p_2, B$ )
2:   ...
8:   per ogni ostacolo  $o \in O$  fai
9:     se ( $\text{distanza}(p_1, o.\text{centro}) < r$ )  $\vee$  ( $\text{distanza}(p_2, o.\text{centro}) < r$ ) allora
10:    per ogni spigolo  $e \in o$  fai
11:      Crea un piano  $pl$  passante per  $e$  e perpendicolare al suolo
12:      se  $pl$  interseca un raggio da  $p_1$  a  $p_2$  e il punto è interno alla parete allora
13:         $n \leftarrow n + 1$ 
14:        Salva la distanza minima e massima da  $\{p_1, p_2\}$  al punto
           d'intersezione.
15:         $m_o \leftarrow m_o +$  differenza fra i valori min e max calcolati al passo
           precedente.
16:      Crea un piano  $pls$  a partire dalla faccia superiore dell'oggetto
17:      se  $pls$  interseca un raggio da  $p_1$  a  $p_2$  e il punto è interno allora
18:         $n \leftarrow n + 1$ 
19:        Aggiorna la distanza minima e massima di conseguenza.
restituisci  $m_o$  e  $n$ 
  
```

ESEMPIO Si prenda lo scenario in Figura 3.4 che comprende un blocco di tre edifici in una zona suburbana nella periferia di Los Angeles. Si consideri un veicolo fermo sulla 11th Avenue (a sinistra) che trasmette a un veicolo che lentamente si sposta lungo la West 85th Street (in alto); lo scopo è quello di registrare la potenza del segnale in ricezione (*Received signal strength*,

RSS). In una prima prova è stato utilizzato il modulo originale (2D) mentre nella successiva l'estensione tridimensionale, sostituendo al primo veicolo con un drone ad un'altezza di 50 metri. Nella prima immagine in Figura 3.5 si vede la degradazione del valore RSS mentre il secondo veicolo si muove lungo la strada durante la prima prova. Non appena viene superato l'ostacolo, e la visuale passa da LOS a OLOS, la potenza del segnale cala bruscamente (passando da -72 dBm a -91 dBm); nella figura il colore rosso indica un valore RSS più basso e di conseguenza una degradazione del segnale maggiore. La seconda immagine rappresenta, invece, la seconda prova, dove si può vedere che il segnale viene bloccato solamente dal primo edificio (più alto). Questo comportamento non viene catturato dal modulo in due dimensioni in quanto non considera la terza componente z della posizione spaziale.

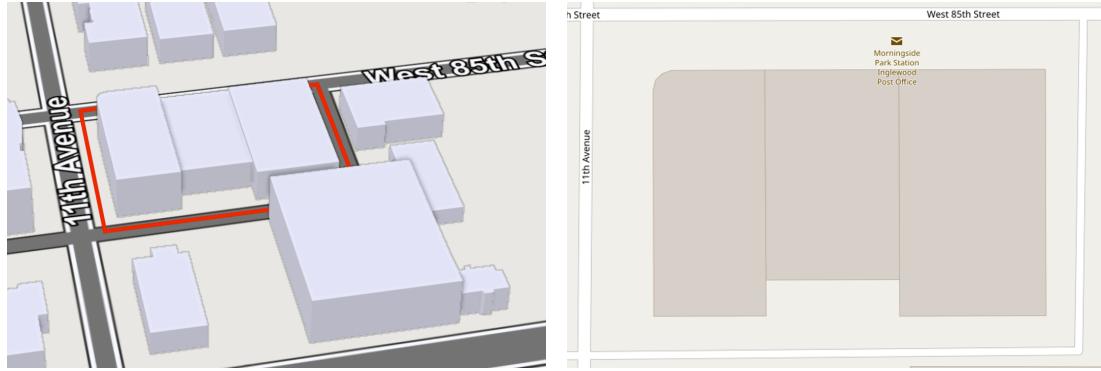


Figura 3.4: Rappresentazione in 2D e 3D degli edifici utilizzati per l'esempio.

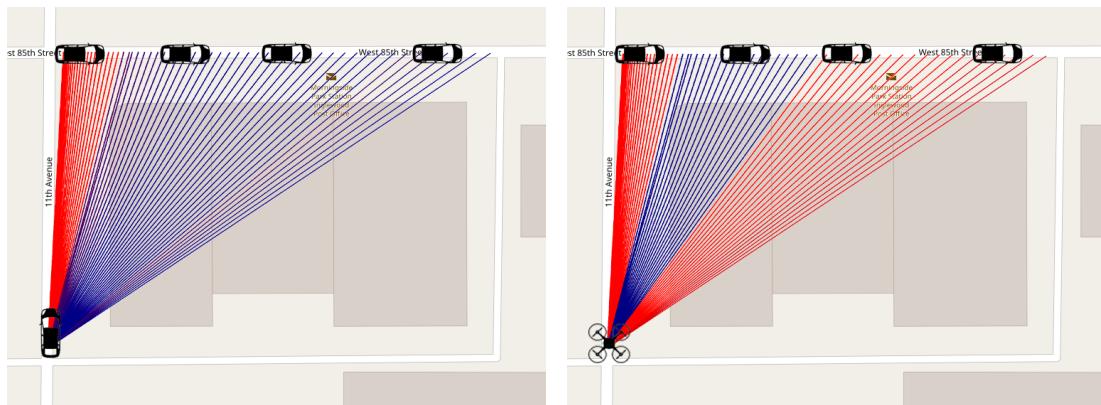


Figura 3.5: Degradazione del valore RSS mentre il secondo veicolo è in movimento. Ogni linea indica una trasmissione, mentre il colore rappresenta l'RSS.

4

Applicativi

4.1 NETWORK SIMULATOR 3

Network Simulator 3 (ns-3) [7] è un simulatore a eventi discreti utilizzato principalmente in ambito accademico e di ricerca, scritto in C++ e Python; è un software libero ed è distribuito sotto licenza GNU GPLv2. Creato nel 2006 da un gruppo coordinato da Tom Henderson, George Riley, Sally Floyd e Sumit Roy aveva come scopo quello di ovviare ad alcune grosse limitazioni della precedente versione (ns-2), come ad esempio la necessità di utilizzare un linguaggio di *scripting* dedicato per modellare le simulazioni o una maggiore scalabilità [21]. Attualmente è l'unica versione della serie a essere mantenuta. È d'obbligo menzionare che, sebbene tutte le versioni siano state scritte in C++ , le simulazioni scritte per ns-3 non sono retrocompatibili.

ELEMENTI CHIAVE In ns-3, il dispositivo di elaborazione di base (unità elementare) è chiamato *nodo* e fornisce i metodi per gestire i dispositivi all'interno delle simulazioni; può essere paragonato a un computer a cui è possibile aggiungere funzionalità come applicazioni, stack protocolari e schede periferiche. Sui nodi sono eseguite delle attività, le *applicazioni*, al fine di raggiungere un certo obiettivo. Per esempio, le applicazioni UdpEchoClientApplication e UdpEchoServerApplication compongono un'applicazione client/server per gestire e scambiare pacchetti nella rete. Le comunicazioni fra i nodi avvengono tramite degli speciali *canali* che possono essere definiti dallo sviluppatore e permettono di modellare concetti semplici, come un collegamento su cavo, fino a quelli più complessi, come uno *switch Ethernet*. Su un nodo è possibile installare uno o più *dispositivi di rete* che permettono la comunicazione con altri nodi attraverso i canali; un nodo può essere connesso a più canali attraverso diversi dispositivi di rete. La simulazione di una grande rete obbliga lo sviluppatore a creare migliaia di nodi, dispositivi di rete e canali. Per facilitare questo tipo di operazioni molto comuni sono state inseriti nel simulatore degli agenti, chiamati *Helper*.

STRUTTURA DI UNA SIMULAZIONE Per creare una semplice simulazione è necessario:

- impostare gli argomenti che saranno letti da riga di comando;
- impostare il valore degli attributi delle classi;
- creare i nodi;
- configurare il livello fisico e MAC;
- impostare il livello di rete, il protocollo di routing e gli indirizzi;
- configurare e installare le applicazioni;
- impostare la posizione iniziale dei nodi e l'eventuale mobilità;
- pianificare eventuali eventi definiti dall'utente;
- avviare la simulazione.

4.2 SIMULATION OF URBAN MOBILITY

Simulation of Urban MObility (SUMO) [22] è un simulatore per traffico veicolare su larga scala, *open source*, scritto in C++ e distribuito sotto licenza Eclipse Public V2. Al suo interno si trovano diversi applicativi utili, dall'importazione di dati reali alla creazione di percorsi per pedoni, diverse tipologie di veicoli, eccetera. Proprio queste *utility* sono state utilizzate per l'elaborazione delle informazioni sugli ostacoli (fornite poi al modulo *Obstacle Shadowing*) e il posizionamento dei veicoli. Per creare uno scenario derivato da un ambiente reale come quelli che saranno utilizzati nelle simulazioni, e poterlo utilizzare in ns-3, sono necessarie alcune elaborazioni intermedie.

Il primo passo consiste nell'utilizzare la piattaforma gratuita OpenStreetMap (OSM) [20] per ottenere le informazioni sul mondo reale (Figura 4.1) e da queste estrarre i dati sugli ostacoli (Figura 4.2). Il file ottenuto viene convertito dall'*utility* di SUMO Polyconvert che si occupa di tradurre alcune informazioni sugli ostacoli, come per esempio convertire la posizione da coordinate geografiche a cartesiane; il file risultante conterrà i poligoni che saranno letti dal modulo *Obstacle Shadowing* (Figura 4.2). Il secondo passo è la generazione delle posizioni dei veicoli. A questo scopo è stato creato uno *script* apposito che, sfruttando le librerie per Python messe a disposizione nel pacchetto di SUMO, partendo da un punto e percorrendo le strade ammissibili posiziona i veicoli a una certa distanza. L'articolo originale [6] utilizzava una funzione già presente in SUMO per il posizionamento casuale che però, non permettendo di specificare una distanza fra i veicoli, non andava bene per gli scenari desiderati in questo lavoro. Il file ottenuto dal processo viene convertito in un formato per ns-2 tramite lo *script* traceExporter (sempre compreso nell'pacchetto SUMO) e questo sarà poi eseguito dal modulo di ns-3 Ns2MobilityHelper in modo da posizionare i nodi durante la simulazione. Il processo completo è riassunto nella Figura 4.4.

Le immagini in Figura 4.2 mostrano l'utilizzo dell'interfaccia grafica inclusa nel pacchetto SUMO che permette la creazione, gestione e simulazione della rete.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="Overpass API">
<note>The data included in this document is from www.openstreetmap.org.
The data is made available under ODbL.</note>
<meta osm_base="2017-11-01T14:42:01Z"/>
<bounds minlat="45.3981000" minlon="11.8654000" maxlat="45.4171000" maxlon="11.8923000"/>
<node id="141542464" lat="45.4140872" lon="11.8775835" version="14" timestamp="2013-03-24T20:52:16Z" changeset="15484553" uid="161619" user="FvGordon"/>
<node id="141542468" lat="45.4115375" lon="11.8856004" version="21" timestamp="2011-11-28T16:48:29Z" changeset="9979546" uid="305001" user="voschix"/>
<node id="141542469" lat="45.4108736" lon="11.8880280" version="20" timestamp="2010-04-13T19:06:57Z" changeset="4416721" uid="209668" user="Tizianos">
<tag k="bicycle" v="yes"/>
<tag k="highway" v="crossing"/>
<tag k="supervised" v="no"/>
</node>
<node id="141542473" lat="45.4130128" lon="11.8903296" version="22" timestamp="2016-07-27T08:45:29Z" changeset="41057002" uid="519484" user="Vajotwo"/>
<node id="141542475" lat="45.4125225" lon="11.8917244" version="22" timestamp="2017-08-07T03:38:33Z" changeset="50904195" uid="15188" user="Polyglot"/>
```

Figura 4.1: Un estratto del file dati generato da OSM.

```
<?xml version="1.0" encoding="UTF-8"?>
<additional xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/additional_file.xsd">
<location netOffset="-724093.06,-5030502.58" convBoundary="-1393.39,-689.19,5170.08,3342.99"
origBoundary="11.845441,45.385455,11.929688,45.422127" projParameter="+proj=utm +zone=32
+ellps=WGS84 +datum=WGS84 +units=m +no_defs"/>

<poly id="101521983" type="building" color="255,230,230" fill="1" layer="-1.00" shape="1470.22,2228.07 1490.86,2294.02 1495.23,2306.17 1505.64,2301.85 1518.03,2295.92 1502.55,2274.79
1489.10,2255.81 1474.89,2227.76 1470.22,2228.07"/>
<poly id="103818689" type="tourism" color="207,245,201" fill="1" layer="-2.00" shape="1426.00,760.01 1434.96,804.28 1457.62,801.42 1456.46,796.34 1453.03,781.16 1447.86,750.40 1443.44,751.49
1426.00,760.01"/>
<poly id="103818853" type="building" color="255,230,230" fill="1" layer="-1.00" shape="365.89,1070.95 366.80,1071.47 368.08,1071.65 369.09,1071.38 369.59,1071.09 370.32,1070.35
370.80,1069.19 370.81,1068.14 370.26,1066.86 369.50,1066.15 368.45,1065.71 367.40,1065.67
366.40,1065.99 365.89,1066.35 365.22,1067.15 364.86,1068.13 364.83,1068.88 365.09,1069.89
365.89,1070.95"/>
<poly id="103818855" type="building" color="255,230,230" fill="1" layer="-1.00" shape="372.57,1079.18 383.31,1078.63 383.02,1075.05 372.29,1075.40 372.57,1079.18"/>
<poly id="103818856" type="building" color="255,230,230" fill="1" layer="-1.00" shape="414.10,1111.15 415.04,1105.53 415.39,1095.69 438.20,1104.15 432.25,1117.87 414.10,1111.15"/>
```

Figura 4.2: Un estratto del file dati sugli edifici dopo la conversione con Polyconvert.

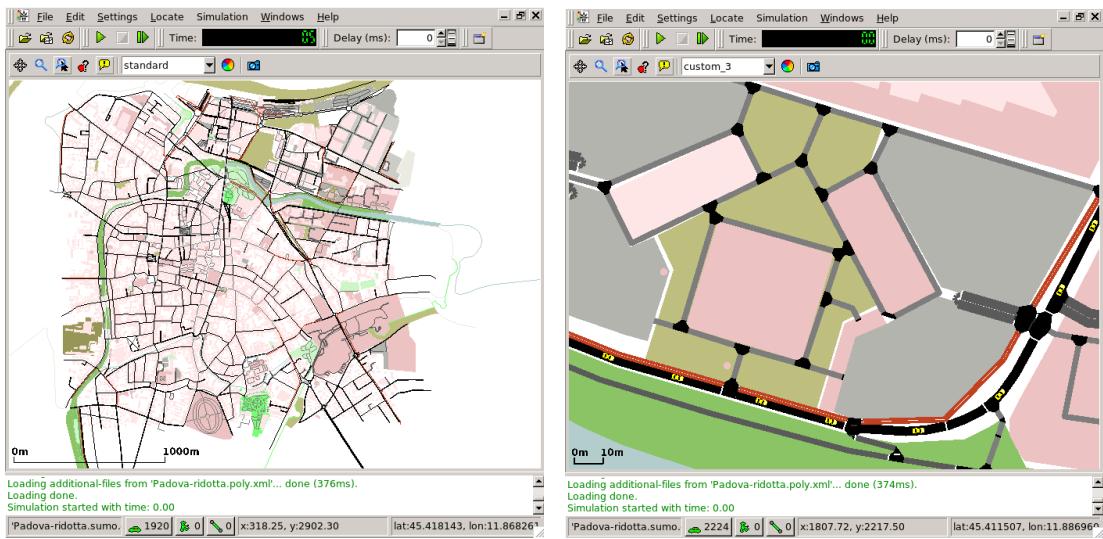


Figura 4.3: Scenario urbano raffigurante il centro di Padova (IT) simulato con SUMO; a destra un dettaglio con i veicoli presenti.

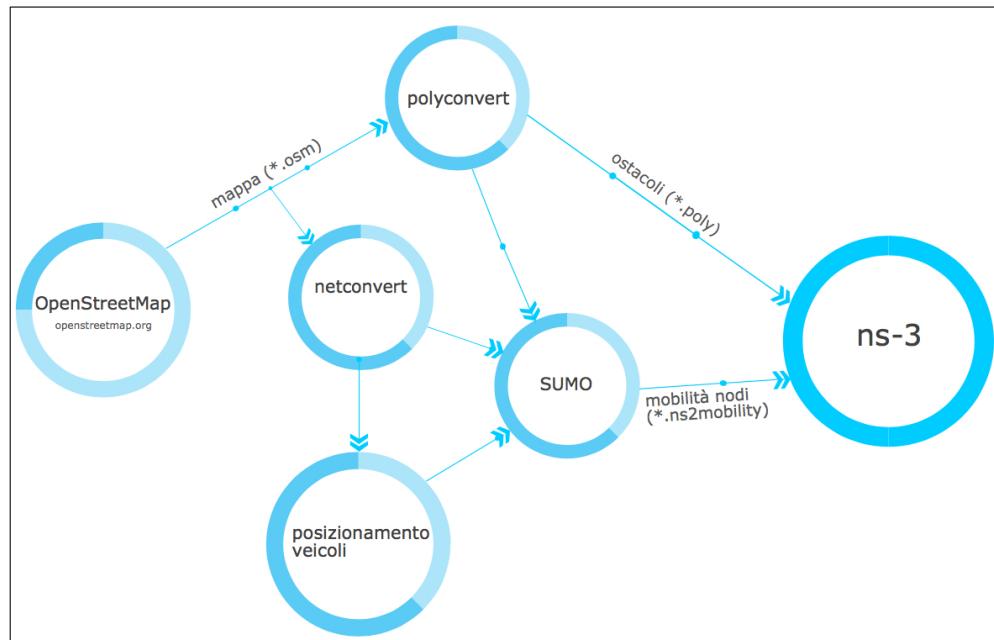


Figura 4.4: Procedimento per estrarre le informazioni da OSM e generare i file per la simulazione con ns-3.

5

Simulazioni

Dopo aver illustrato il protocollo in esame e il modello di propagazione utilizzato si passa ora alla fase di valutazione. Il capitolo procederà nel dettaglio dei diversi scenari affrontati, illustrandone le motivazioni e le caratteristiche e discutendo i risultati ottenuti. In Appendice A sono presentati i valori ottenuti dalle simulazioni in forma tabellare.

5.1 SCENARIO A GRIGLIA

Il primo gruppo di simulazioni ha lo scopo di analizzare l'impatto degli edifici sul comportamento del protocollo Fast Broadcast all'interno di uno scenario conosciuto, ossia quello presentato nella tesi originale [9] a cui sono stati aggiunti degli edifici. La configurazione dello scenario, dei nodi e della rete è la stessa ed è riassunta in Tabella 5.1. L'ambiente è una

città fittizia con strade a griglia in stile Manhattan di lunghezza 4000 metri e distanti l'una dall'altra 300 metri. I veicoli sono disposti a 12 metri di distanza per un totale di 8064. Il veicolo che da inizio alla fase di inoltro (generazione del primo messaggio di inoltro), che per facilità verrà chiamato veicolo *zero*, è posizionato al centro della griglia. Seguendo l'idea originale, è stata definita anche un circonferenza di raggio pari a 1000 ± 12 metri utilizzata per definire alcune metriche di valutazione; la circonferenza ha centro in corrispondenza del veicolo zero. Il raggio trasmissivo effettivo (fisico) assume due valori possibili: 300 o 500 metri.

Parametro	Valore
Lunghezza delle strade	4000 m
Distanza fra le strade	300 m
Distanza fra i veicoli	12 m
Circonferenza	1000 ± 12 m
Posizione del veicolo zero	centrale
Dimensioni pacchetto	164 byte
Standard trasmissione	802.11b
Frequenza	2.4 GHz
Banda del canale	22 MHz
Velocità di trasmissione	11 Mbps
Potenza trasmissione	7,5 dBm
Raggio trasmissivo	300-500 m
Codifica	DSSS*
Modello di propagazione	ns3::RangePropagation
Modello di ombreggiatura	A ostacoli
Simulazioni per configurazione	50

Tabella 5.1: Configurazione dei parametri per le simulazioni.

Per la valutazione del protocollo sono stati presi in esame diversi parametri:

- copertura totale: percentuale di veicoli, rispetto al numero totale, raggiunta dal messaggio di inoltro;

* Direct Sequence Spread Spectrum

- copertura sulla circonferenza: come sopra, ma rispetto al numero di veicoli presenti sulla circonferenza invece che sul totale;
- salti: numero medio di salti necessari a raggiungere i veicoli posti sul perimetro della griglia;
- messaggi di inoltro inviati: indica quante volte i veicoli hanno inoltrato il messaggio;
- messaggi di inoltro ricevuti: ogni volta che un veicolo riceve un messaggio di inoltro viene incrementato questo conteggio.

Per poter confrontare le prestazioni del protocollo Fast Broadcast, quest'ultimo è stato comparato con altri due metodi chiamati STATIC300 e STATIC500 , che utilizzano una stima fissa del raggio trmissivo rispettivamente di 300 e 500 metri. Il raggio trmissivo varia agendo sul modello di propagazione ns3::RangePropagation, implementazione di ns-3 del modello a disco unitario, nel quale una trasmissione viene ricevuta se è a una distanza minore o uguale della portata impostata. Gli edifici, non presenti nello scenario originale, sono stati creati manualmente in modo che ogni edificio fosse contenuto all'interno dello spazio creato dalle strade. Così facendo, risultano 169 edifici di 295x295 metri (i muri sono distanziati di 5 metri dalla strada).

Le simulazioni sono state eseguite sfruttando l'attrezzatura in dotazione al servizio di *High Performance Computing* dell'università di Padova, eseguendono un numero pari a 50 simulazioni per ogni configurazione.

5.1.1 RISULTATI

I grafici dalla Figura 5.1 alla 5.2[†] mostrano i risultati di questo gruppo di simulazioni. Come si può notare, la presenza degli ostacoli influisce molto poco sui quasi tutti i parametri presi

[†]Tutti i grafici mostrati utilizzano come misura di errore l'errore standard con intervallo di confidenza del 95%.

in esame. La differenza sulla copertura totale è, in generale, inferiore o vicina all'1%, tuttavia utilizzando i protocolli FB e STATIC300 si vede un leggero calo nel caso di raggio trmissivo pari a 300m e un leggero incremento con 500m, mentre è l'opposto con STATIC500 ; la differenza è comunque minima per ipotizzare la causa di questo comportamento. I veicoli sulla circonferenza (di raggio 1000m) ottengono una copertura leggermente migliore (in media dello 0,5%), tranne nel caso di STATIC500 con raggio trasmissivo 500m. Questo leggero aumento della copertura potrebbe essere dovuto a quello che gli autori del modello a ostacoli ([6]) chiamano “L'effetto cena con amici” (*The Dinner Party effect*): la presenza di ostacoli come edifici può aumentare, localmente, la probabilità di transmissioni con successo in quanto aumenterebbe il riuso delle spazio. Tale fenomeno può anche essere visto nel grafico in Figura 5.2 dove nel caso di raggio trasmissivo pari a 500m è presente un aumento, in media, del 5,6% nel numero di messaggi di inoltro ricevuti, mentre quelli inviati sono più simili al caso senza edifici. Anche il numero di salti necessari a raggiungere il perimetro della griglia rimane vicino e aumenta mediamente solo del 0,34%. Da tutto ciò si deduce che la presenza di edifici causa un aumento locale del traffico di messaggi ma il meccanismo di contesa e inoltro permette di mantenere limitata la propagazione di questi in avanti. Ciascun edificio ricopre un intero isolato (spazio interno racchiuso fra le strade) e si trova, per una differenza di pochi metri, a ridosso delle strade; la grandezza di questi edifici probabilmente blocca qualsiasi trasmissione fra due veicoli che si trovano agli estremi dell'isolato e, di conseguenza, limita la propagazione del messaggio all'interno del blocco che viene “forzato” a proseguire lungo la strada.

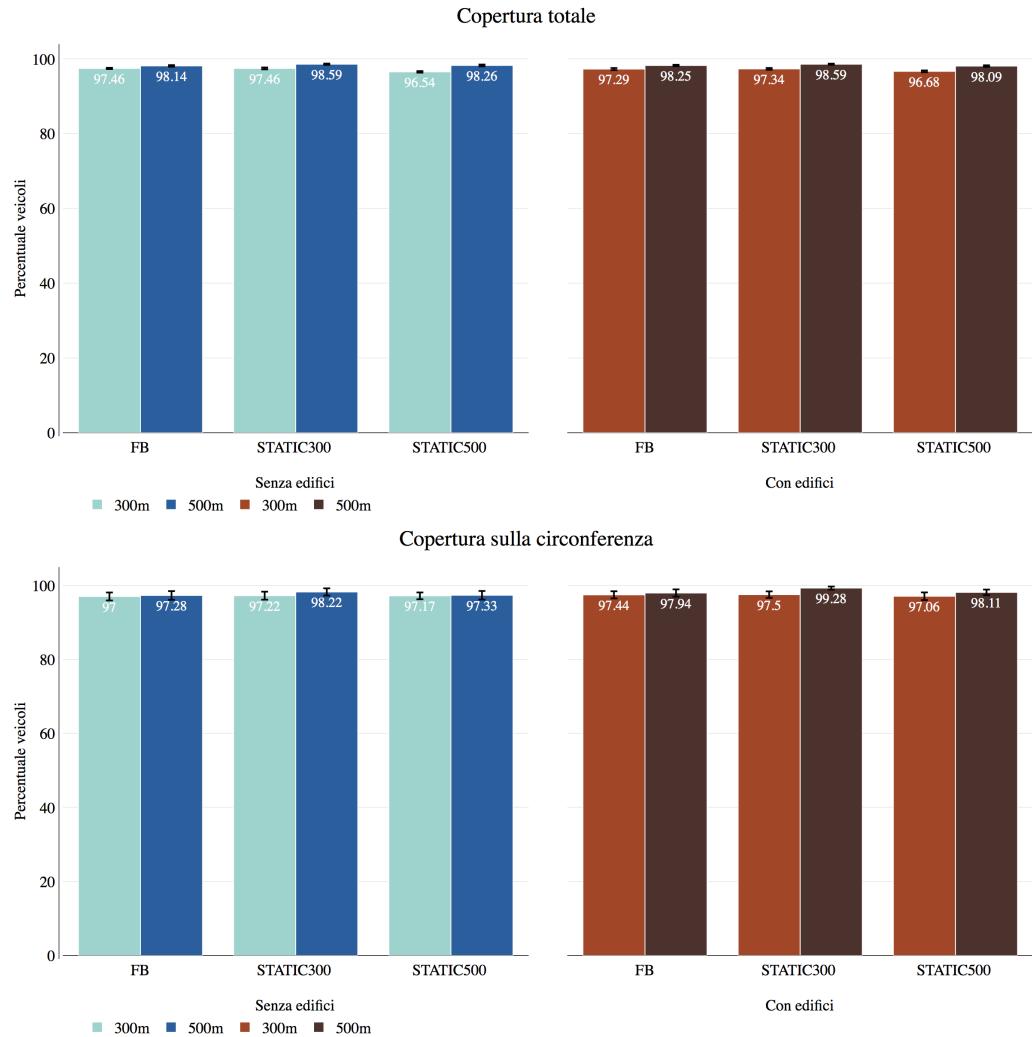


Figura 5.1: Scenario a griglia: copertura dei veicoli in totale e sulla circonferenza.

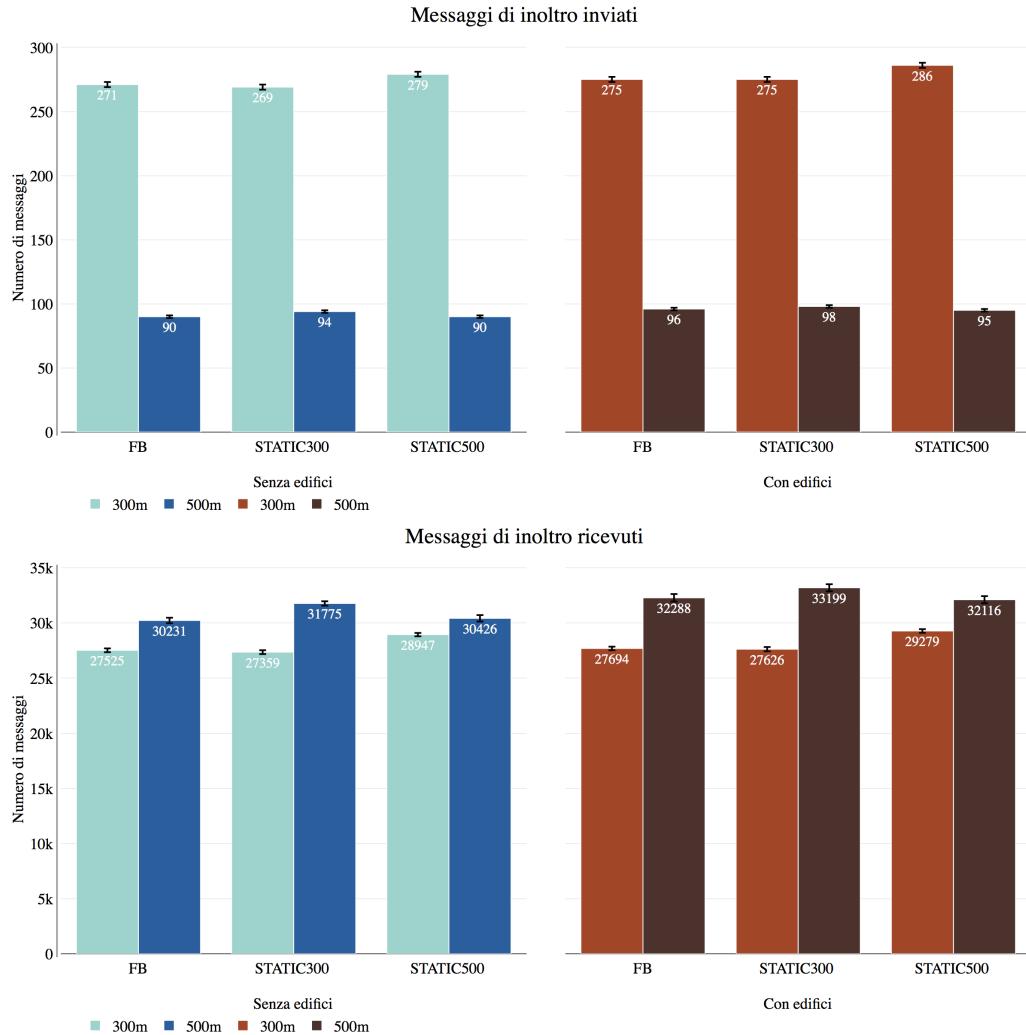


Figura 5.2: Scenario a griglia: numero di messaggi di inoltro inviati e ricevuti.

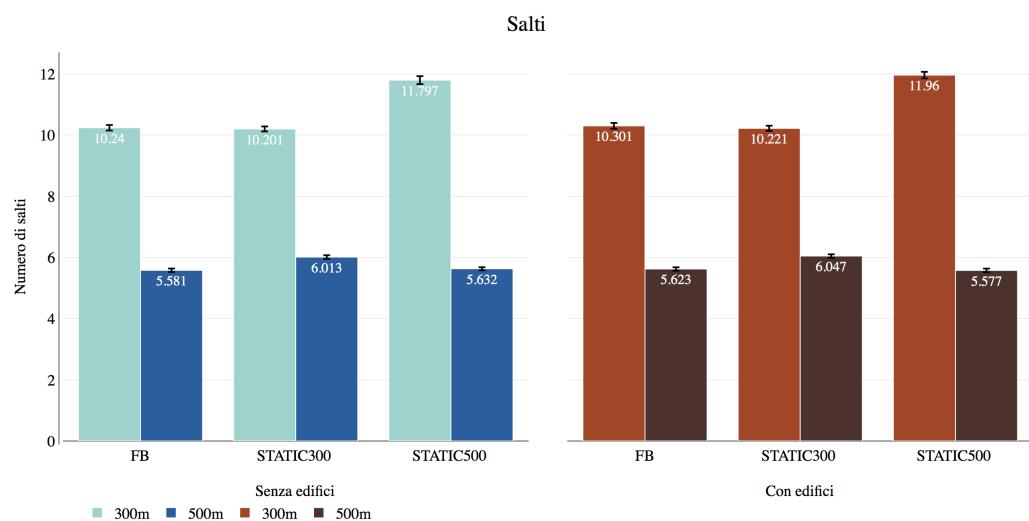


Figura 5.3: Scenario a griglia: numero di salti.

5.2 SCENARIO URBANO REALE

Le configurazioni precedenti avevano il difetto di essere poco veritieri, dal punto di vista della topologia stradale che sulla geometria degli edifici. Le strade formavano una griglia perfetta, con strade identiche e incroci esatti; gli edifici erano molto grandi e tutti della stessa forma e dimensione. In questo gruppo di simulazioni, invece, si è cercato di rappresentare un vero scenario realistico, con strade di lunghezza diversa e non perfettamente allineate, con edifici meno squadrati, a ridosso della strada o anche assenti in alcune zone (difficile in città, ma possibile).

A questo proposito sono state scelte due città, Padova (IT) e Los Angeles (California, USA) (Figura 5.4), definite due aree nella zona centrale di circa 5 chilometri quadrati e, seguendo il procedimento descritto nella Sezione 4.2, sono state estratte le informazioni necessarie alla configurazione delle simulazioni. Altre informazioni sugli scenari e sui parametri delle reti (se diverse dalle precedenti) sono elencati in Tabella 5.2. La scelta è ricaduta su Los Angeles poiché la sua rete stradale è paragonabile a una griglia in stile Manhattan, come nello scenario precedente, mentre Padova (sede anche dell'Università dove si è svolto questo lavoro) ha una topologia più irregolare, con strade più strette ed edifici a ridosso di queste, zone pedonali e ZTL. In questo modo si ha uno scenario che si avvicina al precente come concezione, una sua versione reale, mentre un secondo con caratteristiche proprie.

In questo gruppo di simulazioni sono state utilizzate le medesime metriche del caso precedente, fatta eccezione per il numero di salti che ora sono per raggiungere la circonferenza, invece che il perimetro della mappa. Questo perché nella configurazione a griglia lungo il bordo esterno erano sicuramente posizionati dei veicoli, mentre ora il limite “quadratico” dell’area è ideale.

Inoltre, per rappresentare al meglio il mezzo trasmissivo è stato utilizzato come modello di

propagazione base quello a doppio raggio (Two-Ray Ground), in quanto quello precedente (portata fissa) non teneva in considerazione diversi parametri essenziali, come la potenza della trasmissione, la frequenza del segnale o l'altezza delle antenne rispetto al suolo; inoltre l'assunzione che il segnale si propaghi uniformemente fino a una certa distanza prefissata è poco reale.

		Scenario	
		Padova	Los Angeles
Latitudine	N	45, 4171	33, 9654
	S	45, 3981	33, 9478
Longitudine	O	11, 8654	-118, 3260
	E	11, 8923	-118, 3055
Area approssimativa [km ²]		5	
Distanza fra veicoli [metri]		25	
Numero di veicoli		2224	1905
Numero di ostacoli		6322	8241
Potenza trasmissione [dBm]		4, 6-13, 4	
Modello di propagazione		ns3::TwoRayGround	
Simulazioni per configurazione		50	

Tabella 5.2: Parametri della topologia per gli scenari urbani.

5.2.1 SCENARIO LOS ANGELES

A differenza del precedente, quando si va a considerare l'effetto dell'ombreggiatura in questo scenario si notano maggiori cambiamenti. In primo luogo, come si può vedere dalla Figura 5.6, è presente una diminuzione della copertura dei veicoli che è più marcata con raggio trasmissivo 300m. Nelle simulazioni, infatti, le due diverse distanze massime raggiungibili dal segnale, 300 e 500 metri senza ostruzioni, sono state implementate tramite una diversa potenza in fase di trasmissione. Un segnale più potente ha, naturalmente, più probabilità di “superare” un ostacolo rispetto a un segnale con meno potenza. Inoltre, avendo una portata

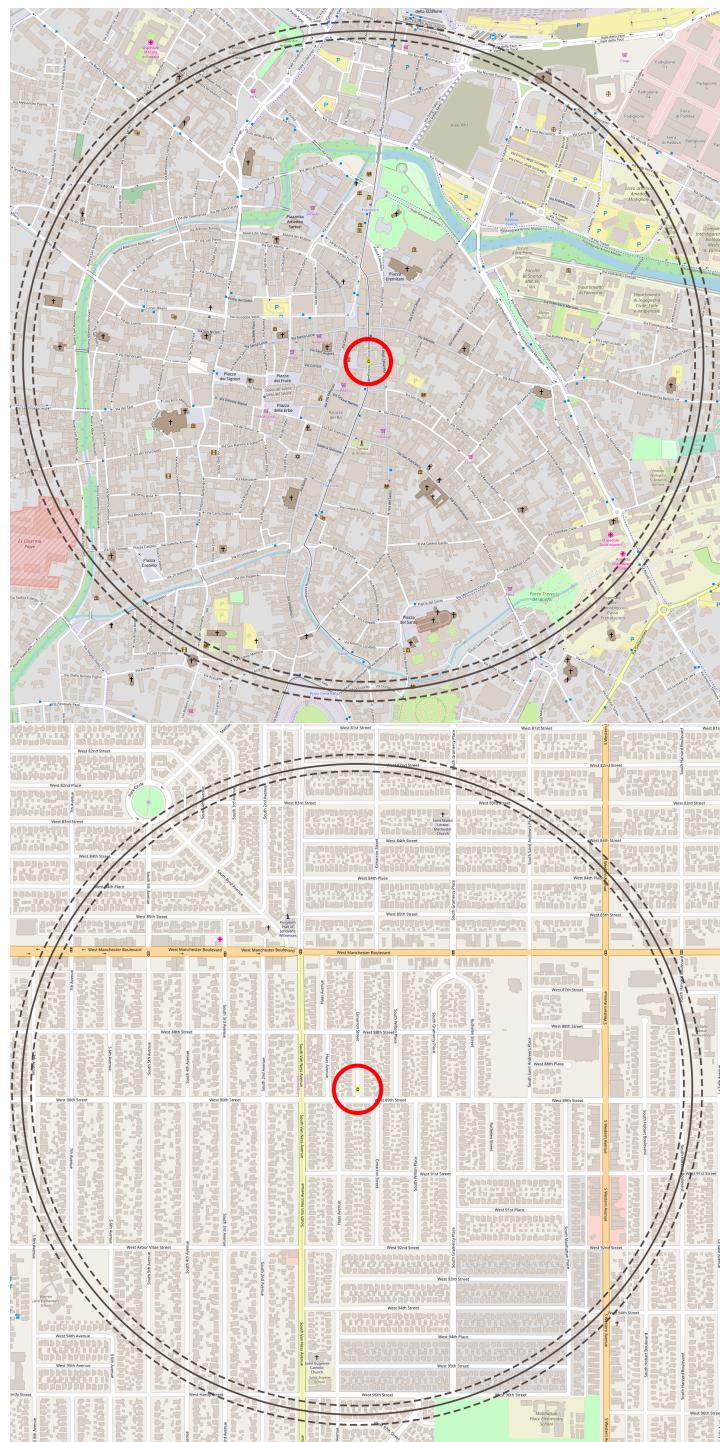


Figura 5.4: Una vista delle aree selezionate per le simulazioni; a sinistra la città di Padova e a destra Los Angeles (fonte: OSM). Il cerchio rosso evidenzia la posizione del nodo di partenza mentre i tre rimanenti rappresentano il perimetro della circonferenza e i due intervalli di confidenza.

maggiori potrebbe raggiungere altri elementi in ricezione, permettere degli inoltri e coprire ulteriori veicoli. Da notare è il cambio di tendenza nel caso del protocollo STATIC300 : in presenza di edifici si ha una copertura maggiore con raggio trasmissivo di 500m. Il motivo potrebbe essere che l'effetto dell'ombreggiatura influisce maggiormente rispetto allo non stimare correttamente il mezzo fisico. Questo fenomeno rimarrà costante anche nello scenario successivo.

Per quanto riguarda il numero medio di salti necessari a raggiungere i veicoli sulla circonferenza, in Figura 5.5, si ha un incremento del 40,6% nel caso di raggio trasmissivo 300m mentre solo del 9% nella controparte a 500m. Inoltre, si può osservare come il protocollo FB abbia prestazioni migliori delle alternative statiche con la giusta coppia valore-stima di circa l'1,5%, mentre nel caso STATIC300 -500m (stima errata) si ha un miglioramento del 77%. Considerando il numero di messaggi di inoltro che vengono scambiati nella rete (Figura 5.7), rimane anomalo il caso STATIC300 -500m, dove si ha un valore inferiore rispetto agli altri protocolli, anche se rimane coerente nel considerare l'effetto ombreggiatura (rispetto allo scenario senza). Mediamente, i messaggi inviati nel caso 300m subiscono un incremento del 194,6% e del 172% con 500m. Un andamento inverso caratterizza il numero totale di messaggi di inoltro ricevuti: se si considera un raggio trasmissivo di 500m si ha un aumento del 41% in presenza di edifici, mentre un calo del 31,3% con 300m. Ciò significa che nel primo caso gli ostacoli considerati causano la perdita di una grossa quantità di messaggi mentre così non avviene nel secondo. La causa di questo fenomeno potrebbe trovarsi nella topologia stradale presente in questo specifico scenario: gli isolati più piccoli hanno una dimensione di circa 50x175 metri mentre quelli più grandi (visibili nella mappa in basso a destra) sono lunghi circa 400 metri. Il raggio trasmissivo minore non riesce a superarli mentre è così per quello maggiore che raggiunge di conseguenza anche i veicoli posizionati sugli e nelle vicinanze degli incroci. Considerando gli isolati più piccoli con 300m si coprono (al massimo) due

incroci mentre se ne coprono fino a tre se si hanno 500m. Inoltre, potrebbe essere che il segnale generato nel caso di raggio trasmissivo minore non sia sufficiente a penetrare gli edifici presenti all'interno di un isolato (in larghezza), mentre è abbastanza forte quello con raggio trasmissivo maggiore.

Osservando i grafici si nota una maggior variabilità dei dati quando si considerano gli ostacoli. Questo può voler dire che per avere risultati più stabili è necessario eseguire più simulazioni con ostacoli rispetto alla controparte che non li considera. Questo fenomeno è più visibile con il protocollo FB che a differenza degli altri ha una componente probabilistica in più, ossia lo scambio iniziale di messaggi Hello. Ciò potrebbe voler dire anche che per effettuare una buona stima del raggio trasmissivo sono necessari più scambi di messaggi di quelli che si erano visti non considerando gli ostacoli.

Anche in questo scenario, il protocollo FB si comporta secondo le aspettative garantendo una buona copertura in entrambi i casi di mezzo trasmissivo, un numero di salti inferiore e quindi una consegna del messaggio più veloce e in generale un numero inferiore di messaggi inoltrati.

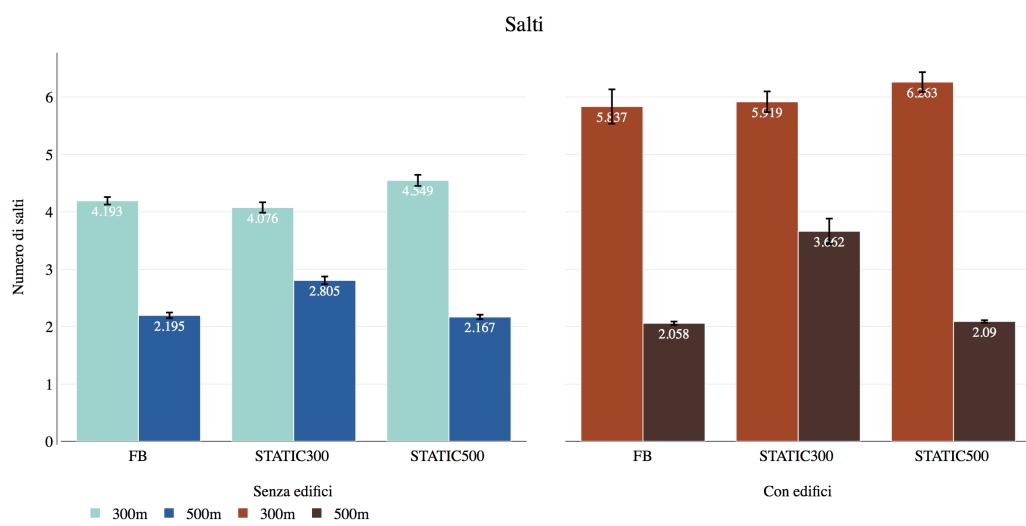


Figura 5.5: Scenario L.A.: numero di salti.

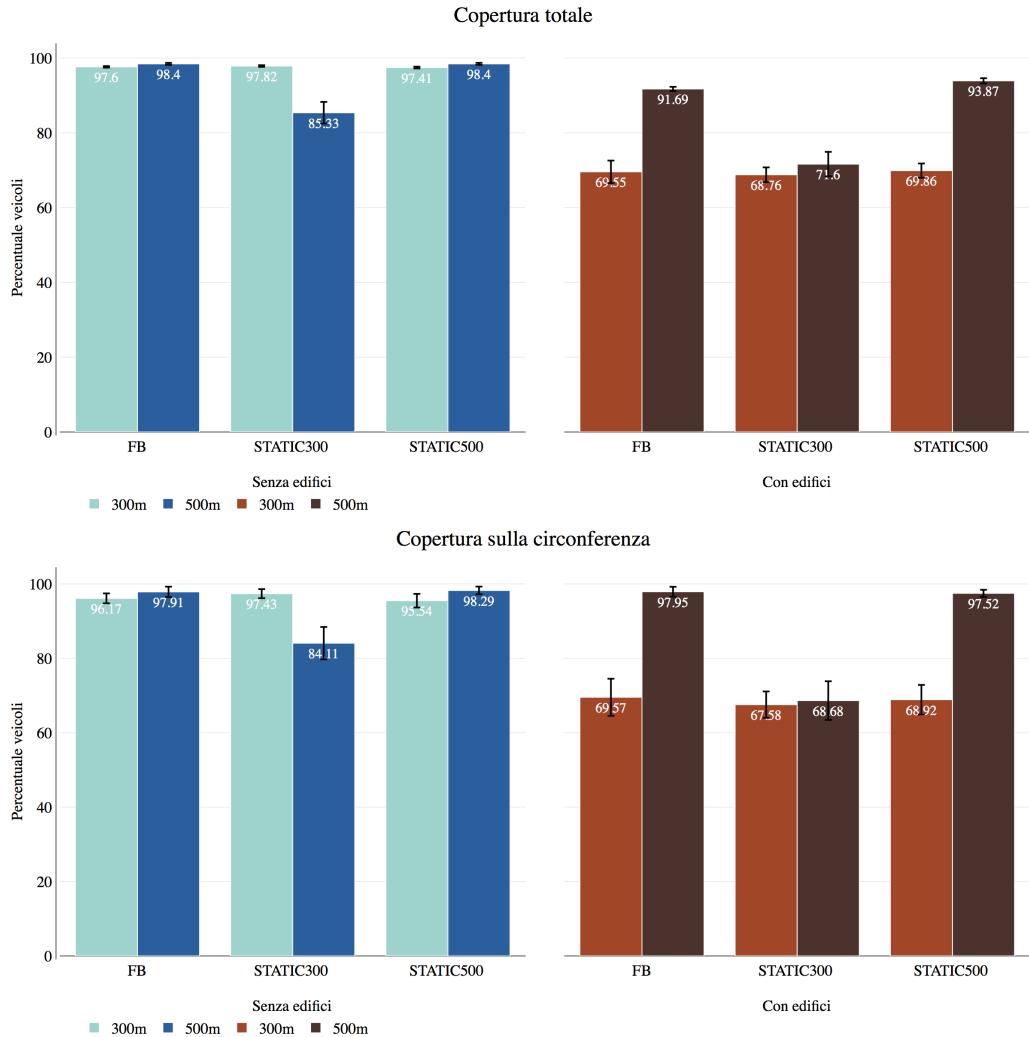


Figura 5.6: Scenario L.A.: copertura dei veicoli in totale e sulla circonferenza.

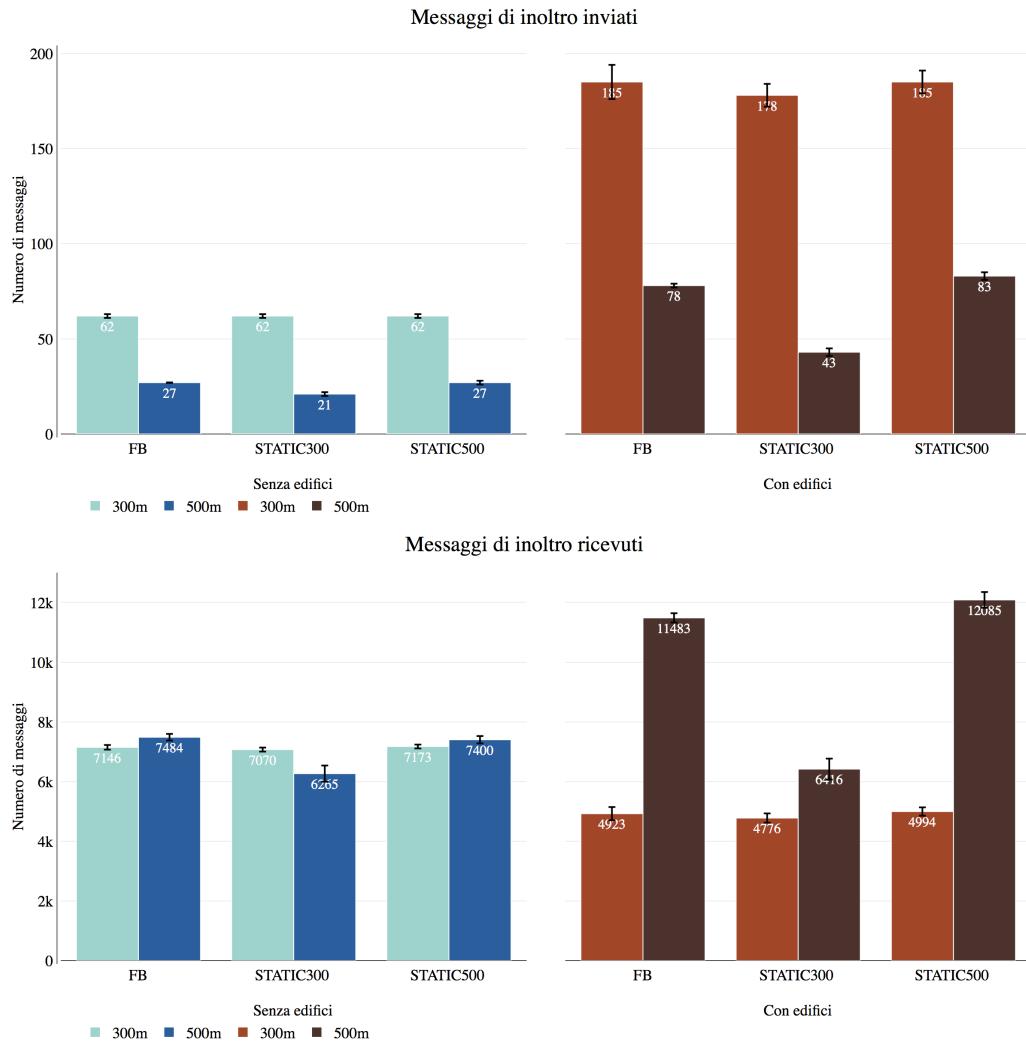


Figura 5.7: Scenario L.A.: numero di messaggi di inoltro inviati e ricevuti.

5.2.2 SCENARIO PADOVA

Similmente ai risultati ottenuti nelle simulazioni precedenti, anche nel centro cittadino padovano si subisce una brusca riduzione della copertura totale quando si va a considerare l'effetto ombreggiatura casuato dagli edifici (Figura 5.8). Più precisamente, in questo caso il calo è maggiore e pari al 50,5% nel caso 300m, a differenza del precente che si attestava al 30,5%; utilizzando il raggio trasmissivo pari a 500m la differenza di copertura totale si attesta intorno al 7,9% per entrambi gli scenari. Una sostanziale differenza di nota anche nella copertura dei veicoli che si trovano sulla circonferenza, dove si registra un calo che arriva fino al 77,3% con STATIC300 -300m. Se ne deduce che alcune caratteristiche intrinseche della città veneta impattano maggiormente un raggio trasmissivo minore che uno maggiore rispetto alle equivalenti della città californiana. Questo potrebbe essere dovuto alla geometria e topologia degli edifici della città veneta, a ridosso della careggiata, più stipati e con meno spazio libero, oppure dalla diverse struttura stradale, che vede strade più strette, rettilinei di lunghezza inferiore e cambi di direzione repentini.

Per quanto riguarda il numero di salti (per raggiungere la circonferenza) in Figura 5.10, i risultati sono coerenti con le aspettative e con lo scenario Los Angeles: il protocollo Fast Broadcast si comporta leggermente peggio in Padova che in L.A. ma rimane il migliore nell'adattarsi al cambiamento di raggio trasmissivo reale.

Rimane l'anomalia nel caso STATIC300 -500m del numero di messaggi di inoltro in Figura 5.12, dove si ha un valore inferiore rispetto al protocollo statico con stima corretta. Come nello scenario precedente, la causa potrebbe essere la maggior influenza dell'ombreggiatura rispetto all'effettuare una stima corretta del mezzo fisico. Considerando l'effetto degli edifici si ha un aumento dei messaggi inviati del 239% nel caso 300m e del 205% nell'altro, un incremento maggiore rispetto allo scenario precente probabilmente per le stesse cause descritte

in precedenza osservando gli effetti sulla copertura dei veicoli. Nel numero di messaggi di inoltro ricevuti si ha un calo nel caso 300m pari al 45,5%, mentre un aumento del 51,8% con il raggio trasmissivo maggiore, come nello scenario precedente.

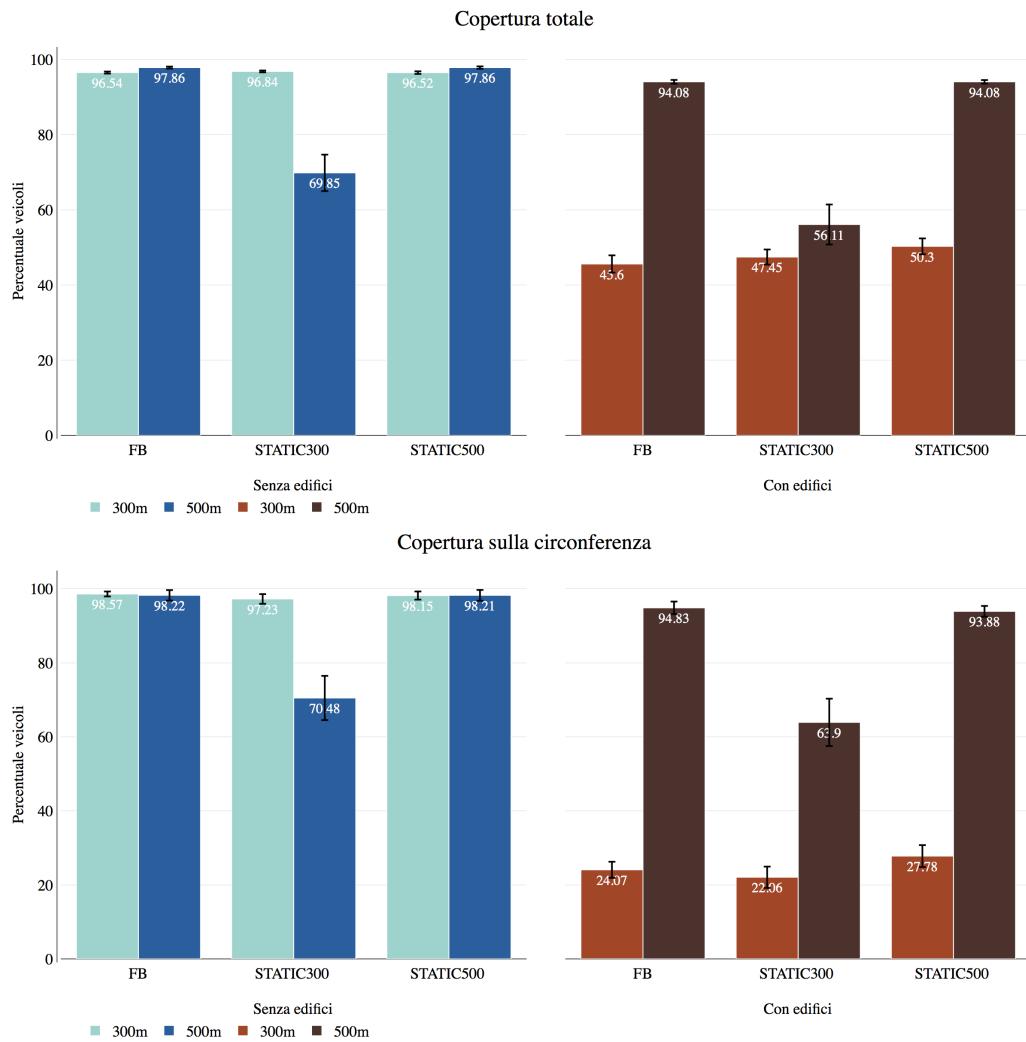


Figura 5.8: Scenario Padova: copertura dei veicoli in totale e sulla circonferenza.

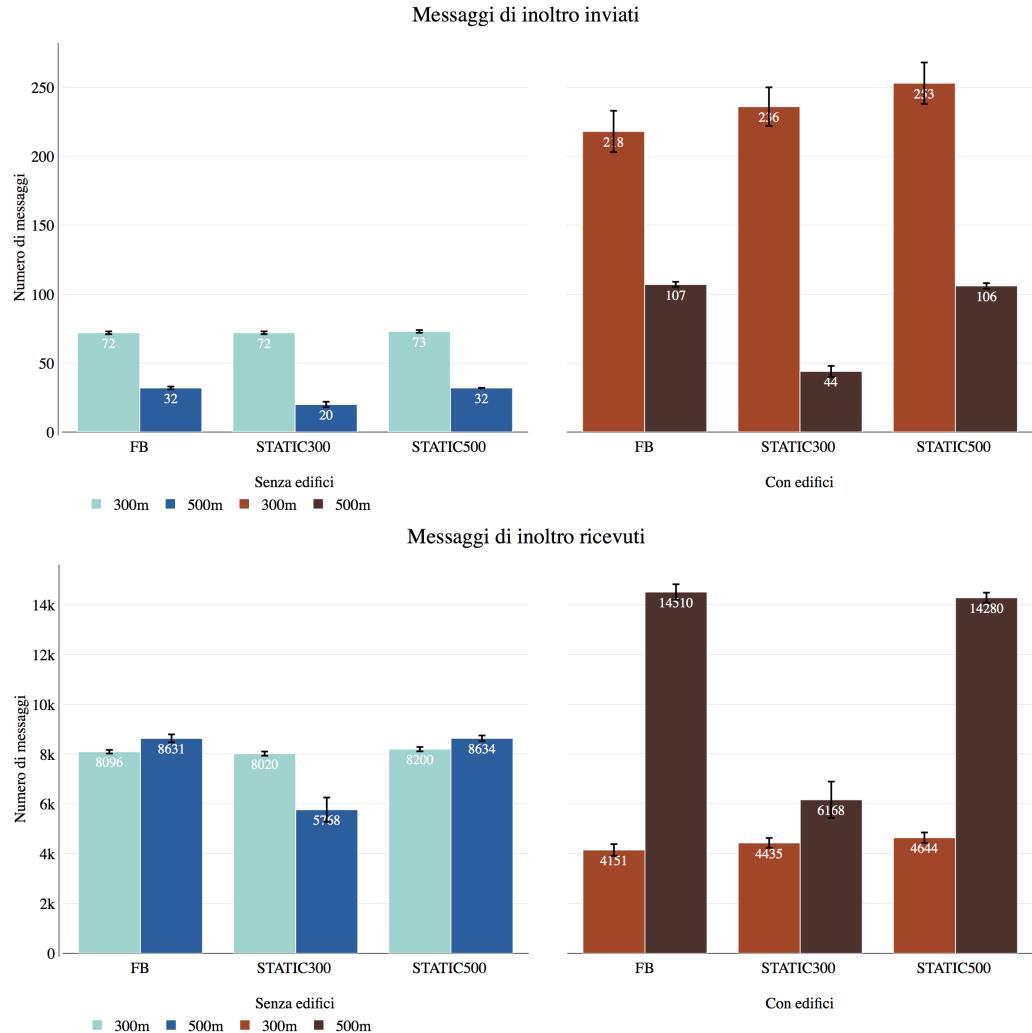


Figura 5.9: Scenario Padova: numero di messaggi di inoltro inviati e ricevuti.

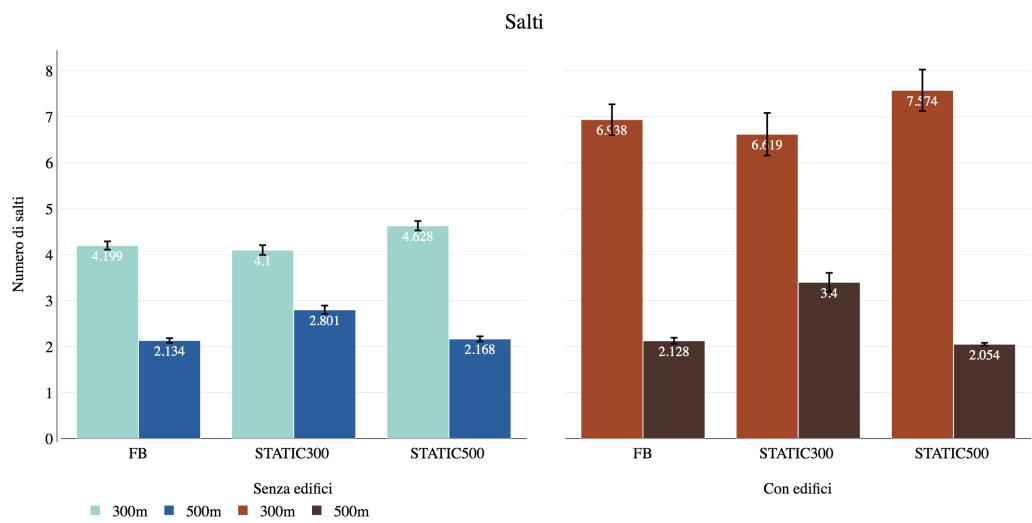


Figura 5.10: Scenario Padova: numero di salti.

5.3 SCENARIO URBANO CON SENSORI

Inizialmente si era detto che spesso le reti VANET sono caratterizzate da nodi con capacità di comunicazione intra-veicolare e che, grazie ai progressi tecnologici, sempre più veicoli possiedono queste caratteristiche. Tuttavia bisogna prendere atto che, almeno per il momento, non siamo nelle condizioni di poter affermare che ogni veicolo è grado di comunicare. Per questo motivo, si può pensare di rilassare tale assunzione e permettere che alcuni veicoli siano esclusi dal traffico; la loro presenza all'interno dello scenario rimane ma non partecipano allo scambio di messaggi.

Viene aggiunta una seconda assunzione: sparpagliati lungo la città sono posizionati dei sensori a basso costo che contribuiscono alla propagazione dei messaggi tramite il semplice Algoritmo 5.1 che inoltra (in broadcast) un messaggio solo la prima volta. Nel mondo rea-

Algoritmo 5.1 Semplice algoritmo di inoltro.

- 1: **se** (RilevatoMsg() \wedge visto() = NO) **allora**
 - 2: visto \leftarrow SI
 - 3: inviaInBroadcast(msg)
-

le questi sensori potrebbero essere dedicati come aiuto propagazione, cioè installati al solo scopo di aiutare la diffusione dei messaggi oppure essere presenti per altri motivi ma con la capacità di ricevere/inoltrare messaggi in caso di bisogno; ad esempio sensori per l'analisi delle polveri sottili, oppure per il monitoraggio del traffico stradale. Si assume, inoltre, che questi sensori siano posizionati a una certa altezza prefissata; potrebbe essere che, in fase di installazione si sia scelto di utilizzare delle infrastrutture pre-esistenti, come per esempio delle lanterne semaforiche, sulle quali montare i sensori in modo da ridurre i costi. Prendendo come spunto l'esempio indicato, si è visto che nel classico semaforo a tre tempi la parte contenente i segnali luminosi dev'essere posizionata fra i 15 e i 26, 5 piedi (rispettivamente 4, 5

e ho 8 metri) nel caso di strutture al di sopra della careggiata [23].

Nel dettaglio, lo scenario attuale prevede la disposizione di 295 veicoli distanti 50 metri (come nello scenario precedente) l'uno dall'altro e di 66 sensori posizionati agli incroci a un'altezza di 6 metri. Il raggio trmissivo è fissato a 300 metri, i veicoli implementano il protocollo Fast broadcast mentre i sensori il semplice inoltro visto in precedenza.

Nelle varie simulazioni si andrà a variare la percentuale di veicoli “disabilitati”, ovvero senza capacità di comunicazione, prendendo un campione casuale della popolazione. La configurazione sopra descritta è riassunta in Tabella 5.3, mentre alcune informazioni sulle altezze degli ostacoli[‡] (fra cui gli edifici) si possono vedere in Tabella 5.4, da cui si evince la diffusione di ostacoli con informazioni tridimensionali.

		Scenario
Latitudine	N	33, 9611
	S	33, 9509
Longitudine	O	-118, 3214
	E	-118, 3098
Area approssimativa [km ²]		1, 2
Distanza fra veicoli [metri]		50
Numero di veicoli		295
Posizione sensori		incroci
Altezza dei sensori [metri]		6
Numero di sensori		66
Simulazioni per configurazione		30

Tabella 5.3: Parametri della topologia per lo scenario con sensori.

5.3.1 RISULTATI

La Figura 5.11[§] mostra la percentuale dei veicoli raggiunti (sul totale di veicoli abilitati) al variare della percentuale di quelli disabilitati, ossia i veicoli che non partecipano allo scambio

[‡]Altezze predefinite per un piano e un sottotetto rispettivamente di 2, 7 e 2, 1 metri.

	Valore
Ostacoli totali	2791
Ostacoli con altezza	2714 (97, 24%)
Altezza massima [m]	21, 9
Altezza media [m]	4, 8
Altezza minima [m]	1, 4

Tabella 5.4: Statistiche sull'altezza degli ostacoli.

di messaggi. Per poter avere una base di riferimento, sono state effettuate anche delle simulazioni nelle quali i sensori non erano attivi, indicate in figura con l'etichetta “senza sensori”.

Mentre in questo caso la copertura totale non supera il 40%, attivando i sensori si ha un incremento sostanziale raggiungendo una copertura intorno al 97-98%. Inoltre, mentre senza i sensori la copertura soffre della mancanza di veicoli man mano che vengono disabilitati con i sensori questo non avviene. La presenza di sensori permette, quindi, non solo una copertura molto maggiore (in questo caso vicina al totale) dei veicoli, ma garantisce una robustezza maggiore in caso di riduzione dei veicoli con capacità di comunicazione.

La Tabella 5.5 mostra come i sensori permettano di raggiungere la circonferenza (questa volta di raggio 400 metri) senza che nessun veicolo inoltri il messaggio, solo grazie al semplice inoltro che i primi implementano. Tuttavia, questo miglioramento costa un incremento notevole del traffico generato, come si può vedere dalla Figura 5.12, che mostra la quantità di messaggi di inoltro ricevuti dai veicoli (attivi). Con il 10% di veicoli disabilitati si ha un incremento di due volte e mezza dei messaggi, che diventa di sei volte più grande nel caso 50%.

Concludendo, i sensori permettono una copertura quasi totale dei veicoli ma al costo di un incremento sostanziale del traffico di messaggi scambiati. È possibile implementare nei sensori algoritmi più intelligenti per l'inoltro, come ad esempio Fast Broadcast, anche se questo po-

[§]Errore standard con intervallo di confidenza dell'80%.

trebbe ridurre il numero di veicoli raggiunti. È necessario, quindi, trovare un compromesso che bilanci questi aspetti.

Percentuale veicoli disabilitati	Numero di salti	
	Senza sensori	Con sensori
10%	2,051	0
20%	2,215	
30%	2,126	
40%	2,103	
50%	2,023	

Tabella 5.5: Scenario sensori.: numero di salti.

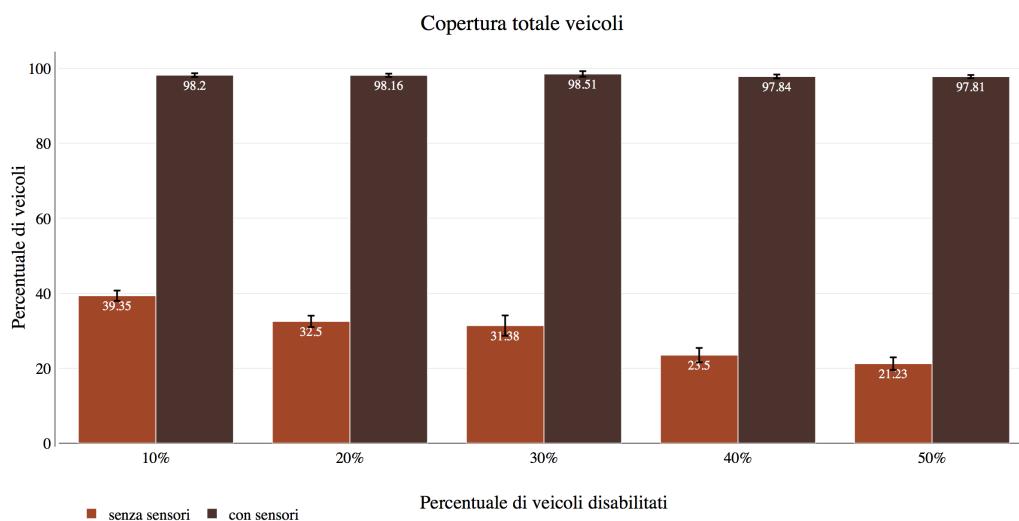


Figura 5.11: Scenario sensori: copertura totale dei veicoli.

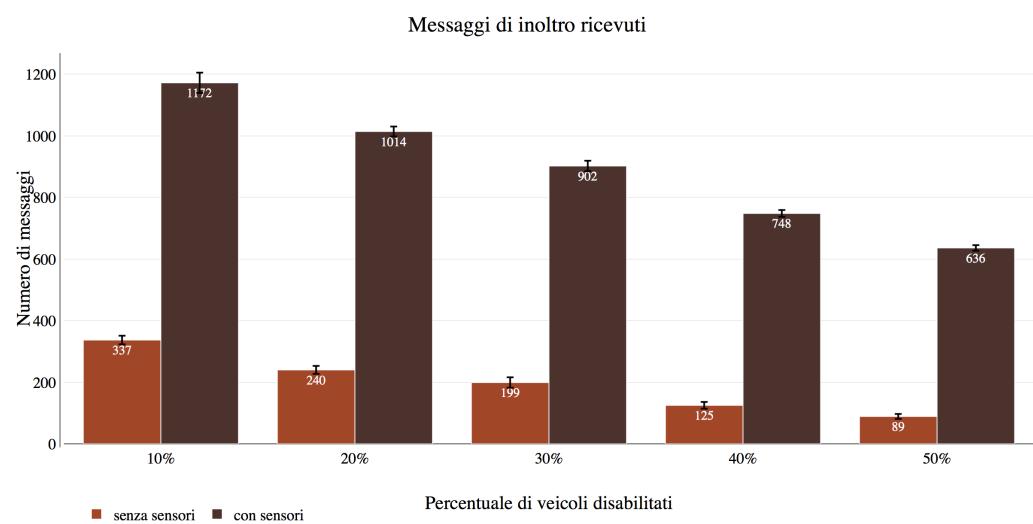


Figura 5.12: Scenario Padova: numero di messaggi di inoltro ricevuti.

6

Conclusioni

Lo scopo di questo lavoro è stato valutare il protocollo Fast Broadcast per la propagazione dei messaggi all'intero di un contesto urbano reale, con particolare attenzione all'impatto dell'effetto ombreggiatura derivato dalla presenza di ostacoli. Dalle prove effettuate è sorto evidente come questo sia un fattore chiave nelle simulazioni delle VANET, specialmente se si considera uno scenario cittadino dove abitazioni, uffici, ospedali, edifici in genere sono sempre presenti. L'utilizzo di dati reali sulla posizione e/o mobilità dei veicoli è altresì decisiva, permettendo di cogliere effetti causati dalle caratteristiche intrinseche dell'ambiente sottostante. Nel caso specifico, si è visto come a parità di portata del raggio trasmissivo la struttura del centro cittadino di Padova influisce maggiormente sullo scambio di messaggi rispetto allo scenario della città di Los Angeles, pertanto per ottenere dei risultati uniformi, o

se si vuole astrarre da uno scenario in particolare, è necessario utilizzare una trasmissione con una portata maggiore. Il protocollo Fast Broadcast si comporta come previsto, confermando che una stima dinamica sul mezzo trasmittivo permette di coprire più veicoli all'interno dell'area prefissata e in minor tempo. I risultati mostrano tuttavia che, in alcuni casi, l'effetto dell'ombreggiatura potrebbe impattare in modo maggiore rispetto allo non stimare correttamente il mezzo fisico. È opinione degli autori che un adeguato modello di rappresentazione dell'ambiente sia necessario durante lo studio di questa tipologia di scenari in quanto la presenza degli ostacoli non può essere esclusa. L'utilizzo di dati effettivi sulla geometria di questi permette di ridurre ulteriormente il divario fra simulazioni software ed esperimenti reali effettuati sul campo. In aggiunta, si è visto come la bassa copertura dei veicoli può essere migliorata in modo considerevole tramite l'aggiunta di dispositivi a basso costo, come per esempio vari tipi di sensori, incrementando tuttavia il traffico dati che viene generato.

Una prima modifica possibile alle configurazioni adottate consisterebbe nell'implementazione di un modello di mobilità per i veicoli, seguendo una delle diverse proposte presenti in letteratura oppure utilizzando tracce reali rese disponibili dagli utenti su alcune piattaforme di navigazione. Per quanto riguarda il protocollo, sarebbe interessante studiarne le prestazioni in presenza di diversi messaggi di inoltro generati da veicoli diversi, come anche l'utilizzo di un servizio a livello applicativo che produca un traffico dati costante fornito poi al protocollo.

In relazione al modello a ostacoli, la modifica tridimensionale proposta è una delle possibili soluzioni e ulteriori prove sono sicuramente necessarie per determinare in che misura l'utilizzo della terza dimensione impatti sulle simulazioni. Inoltre, sarebbe utile integrare nel simulatore ns-3 in maniera più profonda le informazioni sugli ostacoli in modo che non siano utilizzabili anche da altri moduli, ad esempio per permettere ai nodi di essere consapevoli di eventuali ostruzioni durante il movimento (si pensi a droni che si muovono intorno a degli edifici).

A

Tabelle dei risultati

A.1 SCENARIO A GRIGLIA

Protocollo e raggio trasmissivo		Copertura totale	
		Senza edifici	Con edifici
Fast Broadcast	300m	97, 46%	97, 29%
	500m	98, 14%	98, 25%
STATIC300	300m	97, 46%	97, 34%
	500m	98, 59%	98, 59%
STATIC500	300m	96, 54%	96, 68%
	500m	98, 26%	98, 09%

Tabella A.1: Scenario a griglia: copertura dei veicoli in totale.

Protocollo e raggio trasmissivo		Copertura sulla circonferenza	
		Senza edifici	Con edifici
Fast Broadcast	300m	97, 00%	97, 44%
	500m	97, 28%	97, 44%
STATIC300	300m	97, 22%	97, 50%
	500m	98, 22%	99, 28%
STATIC500	300m	97, 17%	97, 06%
	500m	97, 33%	98, 11%

Tabella A.2: Scenario a griglia: copertura dei veicoli sulla circonferenza.

Protocollo e raggio trasmissivo		Numero di salti	
		Senza edifici	Con edifici
Fast Broadcast	300m	10, 240	10, 301
	500m	5, 581	5, 623
STATIC300	300m	10, 201	10, 221
	500m	6, 013	6, 047
STATIC500	300m	11, 797	11, 960
	500m	5, 632	5, 577

Tabella A.3: Scenario a griglia: numero di salti.

Protocollo e raggio trasmissivo		Messaggi inviati	
		Senza edifici	Con edifici
Fast Broadcast	300m	271	275
	500m	90	96
STATIC300	300m	269	275
	500m	94	98
STATIC500	300m	279	286
	500m	90	95

Tabella A.4: Scenario a griglia: numero di messaggi di inoltro inviati.

Protocollo e raggio trasmissivo		Messaggi ricevuti	
		Senza edifici	Con edifici
Fast Broadcast	300m	27525	27694
	500m	30231	32288
STATIC300	300m	27359	27626
	500m	31775	33199
STATIC500	300m	28947	29279
	500m	30426	32116

Tabella A.5: Scenario a griglia: numero di messaggi di inoltro ricevuti.

A.2 SCENARIO LOS ANGELES

Protocollo e raggio trasmissivo	Copertura totale	
	Senza edifici	Con edifici
Fast Broadcast	300m	97, 60%
	500m	98, 40%
STATIC300	300m	97, 82%
	500m	85, 33%
STATIC500	300m	97, 41%
	500m	98, 40%

Tabella A.6: Scenario L.A.: copertura dei veicoli in totale.

Protocollo e raggio trasmissivo	Copertura sulla circonferenza	
	Senza edifici	Con edifici
Fast Broadcast	300m	96, 17%
	500m	97, 91%
STATIC300	300m	97, 43%
	500m	84, 11%
STATIC500	300m	95, 54%
	500m	98, 29%

Tabella A.7: Scenario L.A.: copertura dei veicoli sulla circonferenza.

Protocollo e raggio trasmissivo	Numero di salti	
	Senza edifici	Con edifici
Fast Broadcast	300m	4, 193
	500m	2, 195
STATIC300	300m	4, 076
	500m	2, 805
STATIC500	300m	4, 549
	500m	2, 167

Tabella A.8: Scenario L.A.: numero di salti.

Protocollo e raggio trasmissivo		Messaggi inviati	
		Senza edifici	Con edifici
Fast Broadcast	300m	62	185
	500m	27	78
STATIC300	300m	62	178
	500m	21	43
STATIC500	300m	62	185
	500m	27	83

Tabella A.9: Scenario L.A.: numero di messaggi di inoltro inviati.

Protocollo e raggio trasmissivo		Messaggi ricevuti	
		Senza edifici	Con edifici
Fast Broadcast	300m	7146	4923
	500m	7484	11483
STATIC300	300m	7070	4776
	500m	6265	6416
STATIC500	300m	7173	4994
	500m	7400	12085

Tabella A.10: Scenario L.A.: numero di messaggi di inoltro ricevuti.

A.3 SCENARIO PADOVA

Protocollo e raggio trasmissivo	Copertura totale		
		Senza edifici	Con edifici
Fast Broadcast	300m	96, 54%	45, 60%
	500m	97, 86%	94, 08%
STATIC300	300m	96, 84%	47, 45%
	500m	69, 85%	56, 11%
STATIC500	300m	96, 52%	50, 30%
	500m	97, 86%	94, 08%

Tabella A.11: Scenario Padova: copertura dei veicoli in totale.

Protocollo e raggio trasmissivo	Copertura sulla circonferenza		
		Senza edifici	Con edifici
Fast Broadcast	300m	98, 57%	24, 07%
	500m	98, 22%	94, 83%
STATIC300	300m	97, 23%	22, 06%
	500m	70, 48%	63, 90%
STATIC500	300m	98, 15%	27, 78%
	500m	98, 21%	93, 88%

Tabella A.12: Scenario Padova: copertura dei veicoli circonferenza.

Protocollo e raggio trasmissivo	Numero di salti		
		Senza edifici	Con edifici
Fast Broadcast	300m	4, 199	6, 938
	500m	2, 134	2, 128
STATIC300	300m	4, 100	6, 619
	500m	2, 801	3, 400
STATIC500	300m	4, 628	7, 574
	500m	2, 168	2, 054

Tabella A.13: Scenario Padova: numero di salti.

Protocollo e raggio trasmissivo		Messaggi inviati	
		Senza edifici	Con edifici
Fast Broadcast	300m	72	218
	500m	32	107
STATIC300	300m	72	236
	500m	20	44
STATIC500	300m	73	253
	500m	32	106

Tabella A.14: Scenario Padova: numero di messaggi di inoltro inviati.

Protocollo e raggio trasmissivo		Messaggi ricevuti	
		Senza edifici	Con edifici
Fast Broadcast	300m	8096	4151
	500m	8631	14510
STATIC300	300m	8020	4435
	500m	5768	6168
STATIC500	300m	8200	4644
	500m	8634	14280

Tabella A.15: Scenario Padova: numero di messaggi di inoltro ricevuti.

A.4 SCENARIO CON SENSORI

Percentuale veicoli disabilitati	Copertura totale	
	Senza sensori	Con sensori
10%	39, 35	98, 20
20%	32, 50	98, 16
30%	31, 38	98, 51
40%	23, 50	97, 84
50%	21, 23	97, 81

Tabella A.16: Scenario con sensori: copertura dei veicoli in totale.

Percentuale veicoli disabilitati	Messaggi ricevuti	
	Senza sensori	Con sensori
10%	337	1172
20%	240	1014
30%	199	902
40%	125	748
50%	89	636

Tabella A.17: Scenario con sensori: numero di messaggi di inoltro ricevuti.

Bibliografia

- [1] S. Biswas, R. Tatchikou e F. Dion, “Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety,” *IEEE Communications Magazine*, vol. 44, no. 1, pp. 74–82, gen. 2006.
- [2] L. Pantel e L. C. Wolf, “On the impact of delay on real-time multiplayer games,” in *Network and Operating System Support for Digital Audio and Video, 12th International Workshop, NOSSDAV 2002, Miami Beach, Florida, USA, May 12-14, 2002, Proceedings*, 2002, pp. 23–29. [Online] Disponibile a: <http://doi.acm.org/10.1145/507670.507674>
- [3] C. E. Palazzi, S. Ferretti, M. Roccati, G. Pau e M. Gerla, “How Do You Quickly Choreograph Inter-Vehicular Communications? A Fast Vehicle-to-Vehicle MultiHop Broadcast Algorithm, Explained,” in *in Proc. of the 3rd IEEE CCNC International Workshop on Networking Issues in Multimedia Entertainment (CCNC/NIME 2007), Las Vegas, NV, USA, IEEE Communications Society*, 2007.
- [4] K. N. Sridhar, S. Hao, M. C. Chan e A. L. Ananda, “Egress: Environment for generating realistic scenarios for simulations,” in *2006 Tenth IEEE International Symposium on Distributed Simulation and Real-Time Applications*, ott. 2006, pp. 15–24.
- [5] C. Sommer, D. Eckhoff, R. German e F. Dressler, “A computationally inexpensive empirical model of ieee 802.11p radio shadowing in urban environments,” in *2011 Eighth International Conference on Wireless On-Demand Network Systems and Services*, gen. 2011, pp. 84–90.
- [6] S. E. Carpenter e M. L. Sichitiu, “An obstacle model implementation for evaluating radio shadowing with ns-3,” in *Proceedings of the 2015 Workshop on Ns-3*, ser. WNS3 ’15. New York, NY, USA: ACM, 2015, pp. 17–24. [Online] Disponibile a: <http://doi.acm.org/10.1145/2756509.2756512>
- [7] “ns-3 Network Simulator.” [Online] Disponibile a: <https://www.nsnam.org> (Ultimo accesso 07/28/2017).
- [8] “OMNeT++.” [Online] Disponibile a: <https://www.omnetpp.org> (Ultimo accesso 19/10/2017).
- [9] M. Barichello, C. E. Palazzi e A. Bujari, “Propagazione rapida di messaggi in scenario veicolare urbano,” Tesi Magistrale, Università di Padova, 2017.

- [10] N. Akhtar, O. Ozkasap e S. C. Ergen, “Vanet topology characteristics under realistic mobility and channel models,” in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, apr. 2013, pp. 1774–1779.
- [11] T. S. Rappaport, *Wireless communications - principles and practice*. Prentice Hall, 1996.
- [12] A. Schmitz e M. Wenig, “The effect of the radio wave propagation model in mobile ad hoc networks,” in *Proceedings of the 9th ACM International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM ’06. New York, NY, USA: ACM, 2006, pp. 61–67. [Online] Disponibile a: <http://doi.acm.org/10.1145/1164717.1164730>
- [13] A.-K. H. Souley e S. Cherkaoui, “Realistic urban scenarios simulation for ad hoc networks,” 2005.
- [14] I. Stepanov e K. Rothermel, “On the impact of a more realistic physical layer on manet simulations results,” *Ad Hoc Netw.*, vol. 6, no. 1, pp. 61–78, gen. 2008. [Online] Disponibile a: <http://dx.doi.org/10.1016/j.adhoc.2006.08.001>
- [15] Q. Sun, S. Y. Tan e K. C. Teh, “Analytical formulae for path loss prediction in urban street grid microcellular environments,” *IEEE Transactions on Vehicular Technology*, vol. 54, no. 4, pp. 1251–1258, lug. 2005.
- [16] E. Giordano, R. Frank, G. Pau e M. Gerla, “Corner: A step towards realistic simulations for vanet,” in *Proceedings of the Seventh ACM International Workshop on VehiculAr InterNETworking*, ser. VANET’10. New York, NY, USA: ACM, 2010, pp. 41–50. [Online] Disponibile a: <http://doi.acm.org/10.1145/1860058.1860065>
- [17] T. Rappaport, *Wireless Communications: Principles and Practice*, 2a ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.
- [18] H. H. Nguyen, S. Krug e J. Seitz, “Simulation of 3d signal propagation based on real world terrains for ad-hoc network evaluation,” in *2016 9th IFIP Wireless and Mobile Networking Conference (WMNC)*, lug. 2016, pp. 131–137.
- [19] K. Amjad, M. Ali, S. Jabbar, M. Hussain, S. Rho e M. Kim, “Impact of dynamic path loss models in an urban obstacle aware ad hoc network environment,” *Journal of Sensors*, vol. 2015, 2015.
- [20] “Open Street Map (OSM).” [Online] Disponibile a: <https://www.openstreetmap.org> (Ultimo accesso 07/28/2017).

- [21] T. R. Henderson, S. Roy, S. Floyd e G. F. Riley, “Ns-3 project goals,” in *Proceeding from the 2006 Workshop on Ns-2: The IP Network Simulator*, ser. WNS2 ’06. New York, NY, USA: ACM, 2006. [Online] Disponibile a: <http://doi.acm.org/10.1145/1190455.1190468>
- [22] D. Krajzewicz, J. Erdmann, M. Behrisch e L. Bieker, “Recent development and applications of SUMO - Simulation of Urban MObility,” *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, dic. 2012.
- [23] Federal Highway Administration, *Manual on Uniform Traffic Control Devices*, 2a ed., U.S. Department of Transportation, Washington, D.C., mag. 2012.