

**UNIVERSITÀ DEGLI STUDI DI PADOVA**

DIPARTIMENTO D'INGEGNERIA DELL'INFORMAZIONE

LAUREA TRIENNALE IN INGEGNERIA INFORMATICA

# Implementazione di applicazioni per il traffico veicolare in ns-3

**Relatore:**

Dr. Giordani Marco

**Candidato:**

Poletti Lorenzo

**Co-Supervisore:**

Drago Matteo

Anno Accademico 2020/2021

# Abstract

Con l'aumento del numero dei sensori installati sulle auto e della mole di dati prodotti, si è reso necessario ricercare nuove tecnologie che permettessero migliori prestazioni della rete nella comunicazione tra veicoli. Una possibile soluzione è stata trovata nelle onde millimetriche, aprendo quindi un campo di ricerca per l'implementazione e valutazione di applicazioni in casi d'uso frequenti, tra qui il platooning tra veicoli. Esso permette a veicoli connessi di viaggiare in colonna a breve distanze ed elevate velocità, coordinando le proprie manovre e favorendo operazioni di guida più efficienti. Data l'impossibilità di sperimentare sul campo, si è fatto necessario l'uso di simulatori come ns-3.

A questo scopo è stato realizzato MilliCar, un modulo di ns-3 che gestisce proprio la comunicazione tra veicoli con onde millimetriche.

In questo contesto, dopo aver presentato gli strumenti implementativi, in questo lavoro di tesi sarà mostrata la realizzazione di un'applicazione tramite ns-3 e MilliCar, basata sulle specifiche tecniche dello standard rilasciato dall'ente 3GPP per il caso studio del platooning.

I risultati simulativi ottenuti saranno infine analizzati per valutare le prestazioni di rete al variare delle caratteristiche dello scenario (ad esempio il numero di veicoli nel platoon) e dell'applicazione implementata (ad esempio la dimensione dei pacchetti trasmessi), in relazione ai requisiti previsti dallo standard.

# Indice

|  |           |
|--|-----------|
| <b>Abstract</b>                          | <b>I</b>  |
| <b>1 Introduzione</b>                    | <b>1</b>  |
| 1.1 Platooning . . . . .                 | 1         |
| 1.2 Simulazione . . . . .                | 2         |
| 1.2.1 Millimeter Wave (mmWave) . . . . . | 3         |
| 1.2.2 MilliCar . . . . .                 | 3         |
| 1.2.3 Applicazione . . . . .             | 4         |
| 1.3 Contributi . . . . .                 | 4         |
| <b>2 Implementazione</b>                 | <b>6</b>  |
| 2.1 MilliCar . . . . .                   | 7         |
| 2.2 Applicazione . . . . .               | 8         |
| <b>3 Simulazione</b>                     | <b>12</b> |
| 3.1 Caso baseline . . . . .              | 12        |
| 3.2 Casi dello standard . . . . .        | 13        |
| 3.3 Caso limite . . . . .                | 16        |
| <b>4 Conclusioni</b>                     | <b>18</b> |
| <b>Bibliografia</b>                      | <b>19</b> |

# Capitolo 1

## Introduzione

Negli ultimi anni il numero di sensori nei veicoli e la quantità di dati raccolti sono aumentati considerevolmente in vista della necessità di creare un ambiente sempre più automatizzato e sicuro nel campo della guida autonoma [1].

Tale obiettivo è raggiungibile grazie alla creazione di un network tra veicoli (V2V) e infrastrutture (V2I), che nel complesso (V2X) si scambiano informazioni necessarie all'adempimento di numerosi processi automatici [2], come per esempio:

- **Cooperative lane merge:** Un veicolo host permette l'immissione di un veicolo remoto nella sua corsia.
- **Infrastructure assisted environment perception:** Durante l'immissione di un veicolo in una nuova strada, questo può ottenere dall'infrastruttura informazioni sull'ambiente circostante, come posizione di oggetti statici o dinamici. Ciò è utile al fine di aumentare il livello di confidenza dei propri sensori o estenderne il raggio visivo [2].
- **Obstructed view assist:** Un veicolo con la vista ostruita, ad esempio da un palazzo oppure ad un incrocio, richiede ai veicoli nelle vicinanze di condividere l'immagine delle loro videocamere in modo da procedere alla manovra in maggiore sicurezza.
- **Platooning:** Un gruppo di veicoli a distanza ridotta si muove in modo coordinato al fine di diminuire i consumi di carburante, aumentare l'efficienza di guida, ridurre gli ingorghi stradali e le emissioni di gas serra.
- **In-vehicle entertainment:** I passeggeri all'interno del veicolo in movimento o stazionario possono usufruire di servizi per l'intrattenimento, ma non solo, come video streaming, gaming, realtà virtuale, lavoro da remoto.

Questi sono solo alcuni dei possibili casi d'uso e, per questa tesi, ci baseremo sul platooning.

### 1.1 Platooning

Con il termine platooning si indica un insieme di veicoli virtualmente collegati che condividono informazioni di stato come velocità ed intenzioni come l'intenzione ad accelerare, decelerare, od uscire dal platooning. Utilizzando tale sistema è possibile diminuire il consumo medio di carburante dei veicoli (riducendo quindi le emissioni di gas serra grazie al coordinamento delle manovre di guida), aumentare l'efficienza e ridurre gli ingorghi stradali.

Il gruppo è formato da un leader che svolge il ruolo di coordinatore (platoon leader) e da un

numero variabile di elementi in coda (platoon members) [3]. Le operazioni possibili per un veicolo facente parte del platooning sono molteplici:

- **Entrare e uscire:** In prima istanza dei veicoli possono comunicare per formare un platooning e successivamente altri possono richiederne l'accesso oppure, se giunti a destinazione, uscirne.
- **Annunci e avvisi:** Un veicolo non appartenente al platooning deve essere a conoscenza della sua esistenza per evitare di disturbarne il funzionamento muovendosi in mezzo ad esso. Inoltre, questi avvisi sono necessari per conoscere le procedure di entrata al platooning per un veicolo esterno.
- **Comunicazione di gruppo:** I veicoli all'interno del platooning devono essere in grado di comunicare tra loro per coordinarsi su che strada prendere, quando accelerare, decelerare o in generale su che operazioni di guida svolgere. Queste decisioni confidenziali sono comunicate tramite messaggi criptati.

Tali operazioni devono rispettare specifici requisiti di rete, che possono essere suddivisi in due gruppi a seconda del grado di autonomia del platooning che vuole essere garantita [3]:

- **Gruppo 1 (densità normale):** La distanza tra i veicoli è di almeno 2 metri. Se il platooning si muove a 100 km/h, la latenza massima consentita è di 25 ms con pacchetti da 300-400 bytes.
- **Gruppo 2 (densità elevata):** La distanza tra i veicoli è di 1 metro. Se il platooning si muove a 100 km/h, la latenza massima consentita è di 10 ms con pacchetti da 50-1200 bytes.

Come si può intuire, date le elevate velocità e le distanze tra veicoli così ridotte, il rischio di errori durante la comunicazione deve essere il più basso possibile, supportando un'affidabilità di almeno il 90%. Allo stesso modo, per garantire tempi di risposta celeri tra veicoli, una latenza bassa è necessaria, non più di 25 ms per il gruppo 1 mentre al massimo 10 ms per il gruppo 2 [3].

Oltre ai gruppi verrà tenuto in conto, come vedremo in seguito, del livello di automazione, che identifica quanto un sistema V2X possa essere gestito senza il supporto di un agente esterno. Maggiore il livello, più i requisiti di funzionamento saranno stringenti, data la maggior quantità di dati da gestire e le possibili implicazioni negative in caso di un malfunzionamento del sistema di guida autonoma.

## 1.2 Simulazione

Per studiare il comportamento della rete nel caso d'uso del platooning i parametri che entrano in gioco sono molteplici: numero e distanza tra veicoli, architettura della rete, presenza o meno di ostruzioni durante la comunicazione e parametri dell'applicazione come dimensione del pacchetto e intervallo d'invio.

Uno studio sul campo con tutte queste variabili è impraticabile sia per motivi di costi che di scarsa flessibilità. Per questo si è deciso di ricorrere alle simulazioni e, nello specifico, all'uso del software ns-3 [4].

Prima di passare al contributo offerto da questa tesi è bene spiegare quali strumenti e tecnologie sono stati impiegati in combinazione con ns-3 per realizzare le simulazioni.

### 1.2.1 Millimeter Wave (mmWave)

Lo stato dell'arte della comunicazione tra veicoli (V2V) avviene tramite il protocollo Dedicated Short-Range Communication (DSRC) che permette un massimo raggio di trasmissione di 1km con data rate dell'ordine di 2-6 Mbps, fino ad un massimo nominale di circa 30 Mbps.. Mentre, la comunicazione V2I fa uso della tecnologia 4G-LTE che raggiunge i 100 Mbps in condizioni ottimali di traffico e di carico della rete. [5].

La crescente presenza di sensori nei veicoli a guida autonoma, come radar per stimare posizione e velocità di elementi nelle vicinanze, videocamere per rendere l'esperienza di guida più sicura eliminando punti ciechi, segnalando limiti di velocità e dando migliore visibilità notturna tramite infrarossi, oppure LIDAR che tramite tecnologia laser genera mappe di profondità ad alta risoluzione dei dintorni, hanno incrementato notevolmente il traffico dati a cui la rete V2V può essere sottoposta [1].

Una possibile soluzione è stata trovata nell'impiego delle onde millimetriche (mmWave), ovvero uno spettro di frequenze che va dai 10 ai 300 GHz. Questo largo range spettrale, unito a modulazioni di ordine superiore (modulazioni con più di 4 simboli) e le ridotte dimensioni delle antenne per la lunghezza d'onda nell'ordine dei millimetri, consentono la creazione di array di antenne tramite tecniche multiple-input multiple-output (MIMO), raggiungendo velocità di connessioni nominali prossime ai gigabit-per-secondo.

Nonostante i numerosi vantaggi nell'uso di questa tecnologia, è giusto considerare anche le tre principali problematiche [5]:

- **Scarsa penetrabilità del segnale:** A differenza dei segnali a basse frequenze, le onde millimetriche hanno una scarsa penetrabilità nelle strutture solide. Ciò può portare all'improvvisa perdita di collegamento anche solo per il movimento di un ostacolo.
- **Attenuazione del percorso:** Poiché nello spazio vuoto l'attenuazione del segnale è proporzionale al quadrato della frequenza, con onde millimetriche questo valore diventa molto grande e perciò il raggio di comunicazione decresce. Inoltre, la pioggia può causare scattering del segnale essendo circa della stessa unità di grandezza delle lunghezze d'onda.
- **Allineamento delle antenne:** Per risolvere il problema dell'attenuazione e garantire una connettività elevata non si può far uso di una trasmissione omnidirezionale, quindi la scelta migliore ricade su una direzionale. Questo però richiede un allineamento preciso delle antenne tra trasmettitore e ricevitore, con un alto rischio di errore se non viene gestito in modo efficace ed efficiente.

Per poter valutare nella pratica il funzionamento in ambito V2V delle onde millimetriche e quindi anche l'effetto che queste problematiche hanno sui casi d'uso, è stato realizzato un modulo per il simulatore ns-3 chiamato MilliCar.

### 1.2.2 MilliCar

MilliCar è un modulo open-source per ns-3 che permette la creazione di simulazioni in ambito V2V per veicoli operanti a frequenze mmWave [6] [7]. Si occupa di implementare il canale con propagazione, dissolvenza e beamforming secondo gli standard del 3GPP per reti V2X per frequenze superiori a 6 GHz, il layer fisico (PHY), il medium access control (MAC) e include helper ed esempi per facilitare l'interazione con il modulo [8].

Nello specifico:

- **Canale:** Basato sullo standard per le onde millimetriche rilasciato dal 3GPP che modella la propagazione e la dissolvenza del segnale.

In particolare sono stati definiti gli scenari "V2V-Urban" e "V2V-Highway" che rappresentano come avviene la propagazione se ci troviamo in una città oppure su un'autostrada. In più, lo stato assunto dal canale fra due veicoli può essere Line-Of-Sight (LOS, i veicoli sono sulla stessa strada senza ostacoli in mezzo), Non-Line-Of-Sight (NLOS, il canale è ostruito da un oggetto statico) e Vehicle Non-Line-Of-Sight (NLOSv, i veicoli sono sulla stessa strada ma il canale è ostruito da un altro veicolo) [6].

- **Layer fisico:** Si occupa della trasmissione e ricezione del segnale sul canale di comunicazione. Durante la fase di ricezione viene applicato il modello d'errore.
- **Layer MAC:** Implementa la gestione dell'accesso al canale da parte di più utenti, lo scheduling delle risorse, l'invio e la ricezione su più canali logici e il link adaptation.
- **Integrazione con layer di alto livello:** In ultimo, offre l'integrazione con i livelli più alti del protocol stack tramite un'estensione del NetDevice, permettendo la creazione di connessioni end-to-end.

Il funzionamento a basso livello non è il punto principale di questa tesi, ma ci servirà per comprendere alcuni comportamenti dell'applicazione in fase di discussione dei risultati.

### 1.2.3 Applicazione

Per poter costruire un simulatore è necessario prima di tutto definire il funzionamento dell'applicazione. L'applicazione è un'astrazione che rappresenta l'invio e la ricezione dei dati. Viene definita attraverso la tipologia di comunicazione impiegata a livello di trasporto (TCP, UDP) e la dimensione del pacchetto, l'intervallo d'invio, il numero di pacchetti da inviare e altri parametri. Parametri che, nel nostro caso, sono stati forniti dal 3GPP, ovvero un accordo tra un insieme di enti per la regolamentazione degli standard dei sistemi di telecomunicazione mobile, che assieme all'IEEE sta procedendo a standardizzare le comunicazioni in reti V2X operanti a onde millimetriche con rispettivamente l'introduzione dei modelli IEEE 802.11bd e 3GPP NR V2X [9].

Attualmente tra i diversi standard dei casi d'uso, è stato fornito anche quello del platooning che differenzia le applicazioni in tre categorie basate sul livello di automazione: lowest, low e highest[2].

Ad esempio il lowest, come espresso dal nome, ha requisiti meno stringenti come una dimensione del payload di 300-400 bytes, prevedendo l'invio di 30 pacchetti al secondo con una massima latenza di 25 ms. Al contrario il caso d'uso highest ha un payload di 50-1200 bytes, prevede la trasmissione di 30 pacchetti al secondo con una massima latenza di 10 ms[10].

Dopo la definizione delle caratteristiche dell'applicazione, per poter iniziare a simulare al fine di analizzare le prestazioni della rete come latenza, packet loss rate o data-rate, è necessario installare l'applicazione nei dispositivi secondo la topologia del sistema in questione. Questa operazione verrà mostrata più nel dettaglio nella sezione 2.2.

## 1.3 Contributi

L'obiettivo di questa tesi è quello di implementare, usando ns-3 e MilliCar, un'applicazione che modelli il platooning sulla base delle specifiche previste dallo standard 3GPP e, successivamente, realizzare simulazioni di rete sulla base di tale applicazione al variare dei livelli di automazione, numero e distanza dei veicoli e stato del canale.

Conclusa questa parte seguirà una fase di valutazione delle performance secondo le metriche di latenza media, latenza media dell'ultimo veicolo nel platooning, packet loss rate e throughput medio in base ai dati ottenuti dalle simulazioni.

Come verrà mostrato nel capitolo 3, i risultati sono quasi sempre conformi a quanto previsto dallo standard, a parte alcune situazioni limite dove l'attenuazione del segnale gioca un ruolo importante nell'aumentare la packet loss.

Il documento è organizzato nel modo seguente: nel capitolo 2 vengono trattati in modo più specifico gli strumenti utilizzati per implementare l'applicazione e come questa opera, nel capitolo 3 vengono mostrati e discussi gli esiti delle simulazioni e infine, nel capitolo 4, vengono raccolte le conclusioni.



## Capitolo 2

# Implementazione

Come già accennato nel capitolo introduttivo, data l'impossibilità di svolgere uno studio sul campo è stato necessario fare uso di un software simulativo: ns-3.

ns-3 è un simulatore di rete ad eventi discreti utilizzato nel campo della ricerca per sviluppare nuovi protocolli e analizzare sistemi complessi. [4].

Data la natura open-source è possibile realizzare moduli che ne estendono le funzionalità, come MilliCar che introduce la comunicazione V2V. Sia i moduli che le simulazioni sono scritti in C++.

ns-3 è organizzato in diverse cartelle, tra cui: *src* in cui sono presenti tutti i moduli che vengono usati nelle simulazioni, *build*, in cui sono salvati i file per il simulatore e *scratch*, che contiene gli script con esempi e scenari che possono essere compilati subito [11].

Gli elementi tipici [4] di una simulazione sono:

- **Nodo:** Rappresenta l'astrazione di un dispositivo all'interno della rete, come può essere un computer, cellulare o automobile.
- **Applicazione:** Come già introdotto nel primo capitolo, l'applicazione è un'astrazione di un programma che può inviare e ricevere dati. Da questo modello generico è possibile realizzare comportamenti più specifici, come per esempio applicazioni client/server che scambiano un traffico UDP.
- **Canale:** Come nella realtà, i nodi possono essere collegati tramite WiFi, Ethernet o altre tecnologie per creare un network.
- **Net device:** Rappresenta concettualmente l'hardware e il software installato su un nodo per connettersi ad un canale. Per esempio nel caso di un computer sarebbe la scheda di rete con relativo driver.
- **Helper:** In ns-3 per semplificare l'atto implementativo da parte dell'utente sono stati realizzati gli helper che accorpano operazioni frequenti come collegare net device a nodi, net device a canali o assegnare IP [11].

Nel nostro caso il nodo è un veicolo in movimento, l'applicazione è di tipo UDP e fa riferimento alle specifiche del 3GPP, il canale e NetDevice sono gestiti dagli helper di MilliCar.

## 2.1 MilliCar

Procediamo in questa sezione ad approfondire il funzionamento di MilliCar mostrando anche in che modo è stato utilizzato nello sviluppo di questa tesi.

Il canale e, di conseguenza, il comportamento dell'attenuazione del segnale dovuta alla propagazione è implementato attraverso la classe *MmWaveVehicularPropagationLossModel*, che adotta le strutture descritte alla sezione 1.2.2 di "V2V-Highway" e "V2V-Urban", ognuna delle quali può far riferimento alle 3 condizioni di canale (LOS, NLOS e NLOSv). Nel nostro caso la *ChannelCondition* è stata impostata a NLOS per lo scenario urbano e NLOSv per quello autostradale. La dissolvenza del canale e il beamforming sono gestite da un'altra classe chiamata *MmWaveVehicularSpectrumLossModel*, mentre il comportamento non isotropico dell'antenna per cui il segnale non viene irradiato in modo uniforme in tutte le direzioni è descritto in *MmWaveVehicularAntennaArrayModel*.

Passando al livello fisico troviamo *MmWaveSidelinkSpectrumPhy* che gestisce la trasmissione tramite il metodo *StartTxDataFrames* che genera il segnale rappresentato dalla struttura *MmWaveSidelinkSpectrumSignalParameters* e lo invia allo *SpectrumChannel* richiamando *StartTx*.

Quando il segnale viene ricevuto dallo *SpectrumChannel* il metodo *StartRx* controlla se può venire decodificato dopo l'applicazione del modello d'errore, valutando il superamento di una certa soglia di sensibilità associata al Modulation e Coding Scheme (MCS) da parte del Signal Interference Noise Ratio (SINR). Solo in caso positivo il segnale verrà inoltrato al livello successivo.

L'altra classe è *MmWaveSidelinkPhy* che si occupa di mantenere il sistema sincronizzato e gestire la trasmissione e ricezione dei transport block a livello fisico. MilliCar struttura i frame in base alle specifiche del 3GPP NR V2X [6], suddividendo il frame in sub frame, i quali contengono un numero variabile di slot, a loro volta sono formati da 14 simboli di tipo Orthogonal Frequency Division Multiplexing (OFDM). Il numero di slot dipende dalla *numerology* scelta, che può essere, in base allo standard, 2 o 3 a cui corrisponde un numero di 4 e 8 slot per subframe. I transport block vengono accumulati in un buffer in attesa di un slot libero, il cui invio è a carico del metodo *StartSlot*. La ricezione invece prevede le operazioni di invio al livello successivo e il mantenimento del beamforming.

Muovendoci al livello MAC troviamo *MmWaveSidelinkMac* che gestisce l'accesso al canale da parte di più utenti, lo scheduling delle risorse, l'invio e la ricezione su più canali logici e il link adaptation. Inoltre, supporta la trasmissione su slot differenti per ogni veicolo come previsto per le onde millimetriche in comunicazione direzionale. I pattern per l'assegnazione degli slot ai veicoli sono pre-configurati a livello MAC con uno schema fisso che prevede un'allocazione degli slot in base al sub frame. Per esempio nel caso di default ogni veicolo ha un singolo slot per sub frame.

All'inizio di ogni slot, il livello MAC controlla gli scheduling pattern ed eseguendo *DoSlotIndication* decide se eseguire un invio o una ricezione. Nel caso lo slot sia riservato per un invio, il metodo *ScheduleResources* suddivide le risorse disponibili sui canali logici e informa i livelli superiori. Poi, dopo aver creato il transport block utilizzando il Service Data Unit (SDUs) ottenuto dai livelli più alti, avviene l'inoltro al livello fisico tramite il metodo *AddTransportBlock*.

Se lo slot invece è in attesa di una possibile ricezione viene informato il livello fisico, il quale esegue le operazioni di de-multiplexing per far corrispondere ai pacchetti ricevuti il canale logico corretto.

Inoltre, il link adaptation è gestito dalla classe *MmWaveAmc* che basandosi sulle condizioni di stato del canale inoltrate dal livello fisico viene determinata la miglior modulazione e schema di codifica per la trasmissione.

Spostandoci più in alto arriviamo alla connessione end-to-end e quindi alla classe *MmWaveVehicularNetDevice* responsabile della comunicazioni tra nodi e questo avviene tramite il metodo *ActivateBearer* che accetta come parametri il *bearerID*, il RadioNetwork Temporary Identifier (RNTI) (che è un numero che identifica i dispositivi connessi nella rete) e gli indirizzi IP di entrambi i nodi coinvolti nella comunicazione. Il *bearerID* inoltre deve essere univoco per ogni connessione; questa consistenza viene garantita tramite l'uso dell'helper.

L'invio invece è a carico del metodo *Send* che dopo aver ricevuto i pacchetti dal livello IP lo inoltra verso il PDCP. Al contrario, in ricezione, il PDCP invia al NetDevice che poi a sua volta procede all'inoltro al livello successivo.

MilliCar dispone di un helper chiamato *MmWaveVehicularHelper* che si occupa di formare e configurare il canale e i suoi parametri, calcolare la struttura del frame impostando la numerologia, installare i net device sui nodi istanziando il livello fisico, MAC, configurando le antenne e connettendo i nodi al canale. Infine con *PairDevices* connette due a due i dispositivi assicurandosi dell'univocità del *bearerID* [6].

Concludiamo questa sezione invece presentando in tabella 1 alcuni parametri fissi adottati nelle simulazioni che saranno utili per delineare alcuni comportamenti nei risultati:

| Parametro                | Valore  | Descrizione                                     |
|--------------------------|---------|---|
| AntennaElements          | 16      | Numero di antenne                               |
| CenterFreq               | 28 GHz  | Frequenza a cui è modulato il segnale trasmesso |
| Bandwidth                | 100 MHz | Larghezza di banda                              |
| Numerology               | 3       | Specifica quante slot per sub frame             |
| UseAmc                   | True    | Adaptive modulation and coding                  |
| Shadowing                | True    | Effetto shadowing                               |
| IsotropicAntennaElements | True    | Antenne isotropiche                             |

Tabella 1: Parametri standard per applicazioni in ambito platooning.

## 2.2 Applicazione

Per parlare del funzionamento dell'applicazione è necessario spiegare la topologia della rete, che nel caso del platooning è rappresentata da un numero variabile di veicoli (nodi), di cui un leader che funge da server e tutti gli altri client, disposti in colonna su una sola corsia (Figura 1).

Il leader esegue un numero di applicazioni "server" uguale al numero di veicoli client. Ognuna di queste applicazioni fa riferimento ad una porta differente e si occupa di gestire il traffico in entrata.

D'altra parte ogni client ha installato un'applicazione "client" che genera un traffico UDP che viene inviato al server alla porta assegnata. Poiché lo scopo di questa tesi è valutare le performance del platooning, l'applicazione server si limita a tener conto di quanti pacchetti sono arrivati e in che tempistiche. Al contrario i client, simulano l'invio dati secondo i parametri dei 3 livelli di automazione definiti dal 3GPP [10] e riportati in tabella 2 per il caso d'uso del platooning.

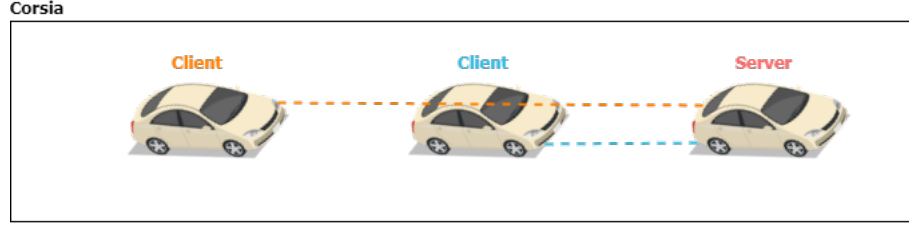


Figura 1: Topologia rete platooning con 3 veicoli.

| Grado di automazione | Payload (bytes) | $T_x$<br>Transmission rate (messaggi/sec) | Latenza massima end-to-end (ms) | Affidabilità (%) | Data rate (Mbps) |
|----------------------|-----------------|---|---------------------------------|------------------|------------------|
| Lowest               | 300-400         | 30  | 25                              | 90               | 0.072 - 0.096    |
| Low                  | 6500            | 50  | 20                              |                  | 2.6              |
| Highest              | 50-1200         | 30  | 10                              | 99.99            | 0.012 - 0.288    |

Tabella 2: Parametri standard per applicazioni in ambito platooning.

Per entrare più nello specifico, a livello pratico sfruttando il *MobilityHelper* sono stati creati  $n$  (con  $n$  da 3-6) nodi disposti in colonna, con il leader come primo. Dopo aver inizializzato i nodi ad essere veicoli che comunicano a onde millimetriche con *MmWaveVehicularHelper*, è stata creata l'applicazione configurandola con i parametri della tabella, in particolare impostando *ns3::UdpClient::Interval*, ovvero l'intervallo di tempo tra un invio e l'altro, con  $\frac{1\text{messaggio}}{T_x}$  millisecondi e *ns3::UdpClient::PacketSize*, ovvero la dimensione del pacchetto, con il valore di payload.

Quindi si è proceduto ad installare tramite *UdpEchoServerHelper* le  $n-1$  applicazioni server sul nodo leader, ognuna in una porta differente.

Inoltre, sempre nella creazione del server, si è provveduto ad inserire il tracing fornito da ns-3 dal metodo *TraceConnectWithoutContext* con cui sono stati registrati il numero e la latenza dei pacchetti in arrivo. Poi, per ogni client è stata installata un'applicazione con *UdpClientHelper* puntante ad una porta del server.

La velocità dei veicoli è stata scelta in base allo scenario, quindi nel caso urbano dove il canale è di tipo NLOS la velocità è di 50 km/h, mentre in caso autostradale con canale NLOSv la velocità è di 100 km/h. Tutta questa struttura appena descritta è la nostra simulazione, che è stata valutata variando i parametri di channel condition (NLOS e NLOSv), numero di veicoli e distanza tra questi. In modo schematico:

**Per ogni applicazione:** lowest, low e highest.

- **Condizione del canale:** NLOS e NLOSv.

- **Distanza tra veicoli nel platooning:** 1 metro (high density) e 5 metri (low density).

- \* **Numero di veicoli per platooning:** 3, 4, 5 e 6.

La scelta di questi parametri è giustificata dalla necessità di valutare le prestazioni della rete in condizioni del sistema più o meno stressanti.

Il canale in modalità NLOS presuppone un caso particolare in cui tutti i veicoli siano in Non-Line-Of-Sight con il leader. Questo può capitare per esempio in uno scenario urbano durante la svolta ad un incrocio con palazzi o altri oggetti che possono ostruire la vista al resto del platooning dal leader per qualche secondo (Figura 2).

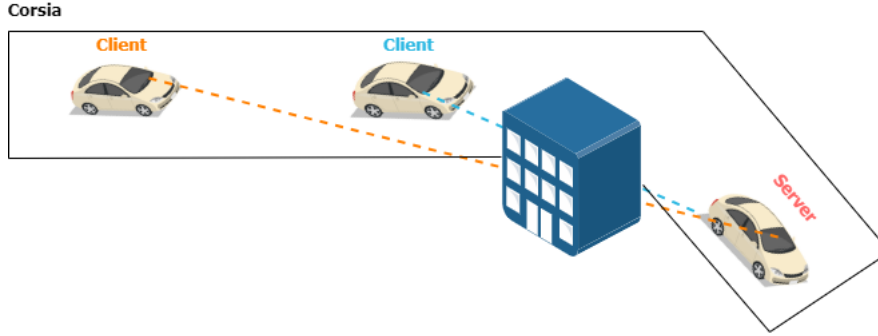


Figura 2: Canale NLOS in caso urbano.

Un caso più comune invece è quello con canale NLOSv, dove i veicoli sono rispetto al leader ostruiti da altri veicoli. Questa è la condizione normale di funzionamento per tutti i veicoli ad eccezione del secondo che risulterà in Line-Of-Sight, rispetto al leader come primo (come precedentemente illustrato in figura 1). Però, nella versione corrente di MilliCar il canale può essere impostato solo globalmente, per cui nonostante non sia vero in pratica, anche il secondo veicolo avrà un canale di tipo NLOSv. In ogni caso questa differenza non incide sui risultati, come si potrà osservare dalle prestazioni del caso baseline nel capitolo 3.

La scelta della distanza è motivata dalla ridotta propagazione delle onde millimetriche, quindi è utile vedere in che misura ci sono delle differenze. Nonostante la distanza di 1 metro tra veicoli in uno scenario urbano con canale NLOS non sia realistica, poiché implica la presenza di un palazzo o altre strutture tra il server e i client, è stata valutata comunque come caso di confronto.

Infine, poiché stiamo parlando di platooning è naturale voler osservare come il numero di veicoli influenzi le prestazioni della rete.

Oltre ai casi sopracitati ne sono stati valutati altri due, uno per avere un riferimento in condizioni ottimali, mentre l'altro per esacerbare ancora di più il carico dell'applicazione sulla rete.

- **Baseline:** Caso di riferimento con condizioni ottimali per ogni tipo di applicazione con 2 veicoli, distanza 1 metro e channel condition LOS.
- **Limite:** Caso estremo dell'applicazione più stringente (highest) variando il numero di messaggi al secondo tra 50, 100, 500, 1000 e 5000 con 6 veicoli, distanza 5 metri e channel condition NLOS.

L'unico parametro costante, oltre a quelli discussi nel paragrafo precedente, è stato il numero di pacchetti inviati per ogni veicolo fissato a 100 in modo d'avere un output dati indipendentemente dall'inter-packet interval dell'applicazione.

Le varie casistiche sono state valutate secondo le seguenti metriche:

- **Latenza media fra tutti i veicoli:** Calcolato sommando la latenza dei singoli pacchetti, mediato sul totale dei pacchetti ricevuti.
- **Latenza media dell'ultimo veicolo:** Calcolato sommando la latenza dei pacchetti provenienti dal veicolo più distante dal leader e mediato sul totale dei pacchetti ricevuti.

- **Throughput medio:** Calcolato dividendo il numero di pacchetti ricevuti moltiplicato per la loro dimensioni in bit e diviso per il tempo impiegato per la trasmissione dal primo all'ultimo pacchetto, mediato sul numero dei veicoli client.
- **Packet loss rate:** In percentuale è il numero di pacchetti ricevuti su quelli inviati.

## Capitolo 3

# Simulazione

Proseguiamo ora con la discussione dei risultati delle simulazioni presentate nel capitolo precedente partendo con l'analizzare il caso baseline (Sezione 3.1), per poi continuare con i tre casi applicativi secondo lo standard del 3GPP al variare dei parametri discussi nella sezione 2.2 (Sezione 3.2) e infine presentando il caso limite (Sezione 3.3).

### 3.1 Caso baseline

Lo studio del caso baseline serve come punto di riferimento in condizioni ottimali. I prossimi grafici avranno sull'asse orizzontale il grado di automazione mentre su quella verticale la metrica. Dalla figura 3a è possibile vedere un throughput più accentuato nel caso low, motivato dalle dimensioni maggiori del payload (6500 bytes) e dal basso inter-packet interval (20 ms) dell'applicazione, in linea con i data rate espressi nella tabella 2.

In condizioni di canale ottime e con solo due veicoli, riscontriamo delle latenze in media basse, nell'ordine di meno di un 1 ms, come visibile in figura 3b. Inoltre, si nota che il caso low è quello con tempi leggermente maggiori, sempre attribuite alla dimensione del pacchetto e l'inter-packet interval che assieme rendono il processo d'invio, ricezione e gestione fino ai livelli alti più lungo. Infine, la packet loss rate in tutti i casi è risultata dello 0% (e per tale motivo è stato omesso il grafico raffigurante tale metrica), il che è ragionevole operando ad onde millimetriche in condizione di canale LOS e con data-rate di al massimo 2.6 Mbps.

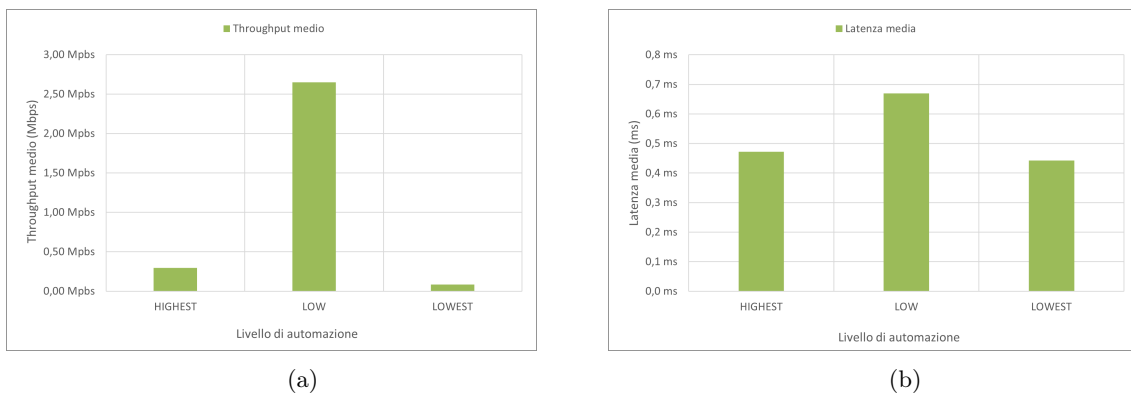


Figura 3: Grafici caso baseline

### 3.2 Casi dello standard

Iniziamo la valutazione delle prestazioni dal caso applicativo meno stringente. I grafici in questa sezione mostreranno sempre una metrica sull'asse verticale e il numero di veicoli in quella orizzontale, con quattro diverse serie di dati che rappresentano le possibili combinazioni tra canale (N,V rispettivamente NLOS e NLOSv) e distanza tra veicoli (1,5 metri).

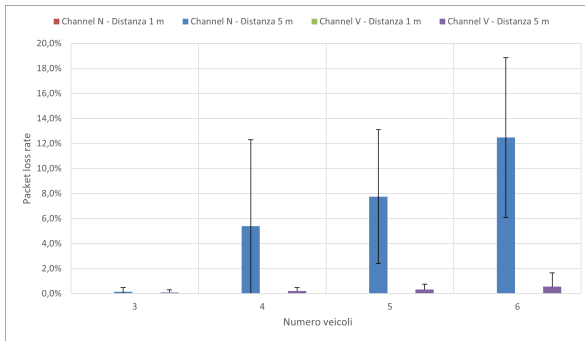
#### Lowest

Il primo grafico, quello in figura 4a, mostra come la packet loss rimanga pressoché allo 0% in qualsiasi condizione di canale se la distanza tra veicoli è di 1 metro. Al contrario, aumenta se questa passa a 5 metri e in particolare, in modo ancor più evidente, se il canale è in condizione N e il numero di veicoli cresce.

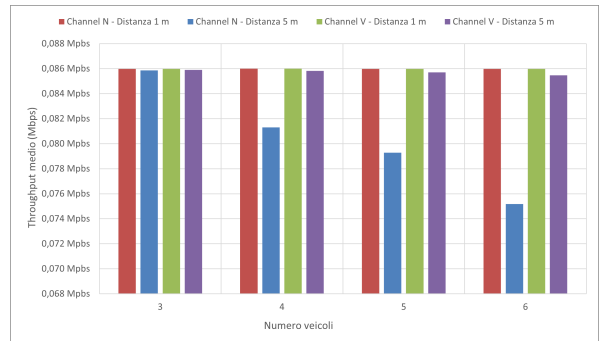
Una perdita di pacchetti così alta è attribuibile all'elevata attenuazione del segnale sul canale. Infatti, oltre a dover coprire distanze più ampie vi è un elemento di ostruzione che riduce il SINR, aumentando la probabilità di un errore nella decodifica a livello fisico. Il comportamento è simile, seppur meno marcato e all'interno dei requisiti di affidabilità previsti dal 3GPP e riportati in tabella 2, per il caso V.

È bene ricordare, come già discusso nella sezione 2.2, che la condizione di canale N per tutti i veicoli è rara e si limita a pochi casi particolari.

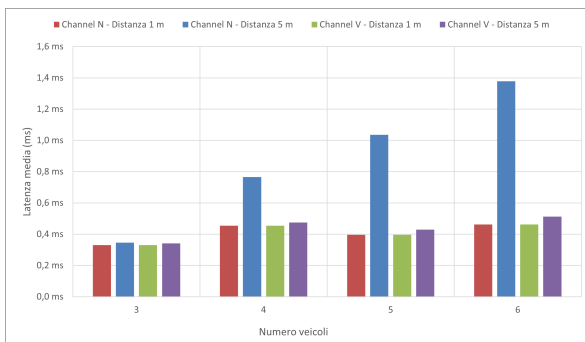
Passiamo ora al grafico del throughput medio, mostrato in figura 4b, che racchiude dei risultati riconducibili al precedente ragionamento. Infatti, con una perdita di pacchetti significativa il data-rate medio diminuirà di conseguenza, come visibile nel caso di un canale N e distanza di 5 metri.



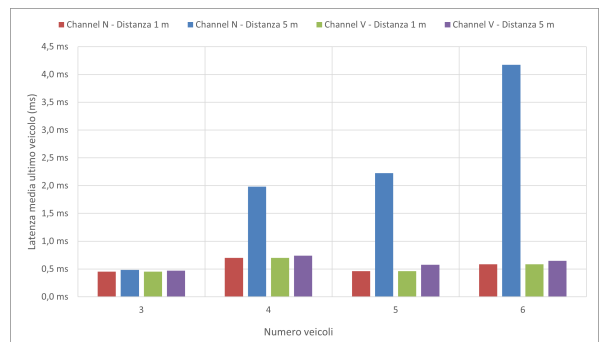
(a) Packet loss rate



(b) Throughput medio



(c) Latenza media globale



(d) Latenza media dell'ultimo veicolo

Figura 4: Statistiche nel caso lowest



Oltre ciò, si misurano valori in linea con quanto previsto nella tabella 2 con un range da 0.074 a 0.086 Mbps.

Dal terzultimo grafico, figura 4c, è visibile uno stacco netto tra gli altri casi e quello con canale N e distanza 5. Facendo un'analisi più approfondita sui tempi di latenza medi per serie di pacchetti, ovvero step singoli d'invio per ogni veicolo client, sono emersi valori maggiori per il caso con canale N rispetto allo stesso caso con canale V. Seppur il grafico in figura 5 sia composto da una serie di soli 10 step, è abbastanza per notare questa discrepanza.

Unito all'elevata packet loss del caso N, che rende il denominatore "pacchetti ricevuti" minore e di conseguenza, in media, la latenza maggiore, troveremo una tendenza coerente con quanto presente nel grafico 4c.

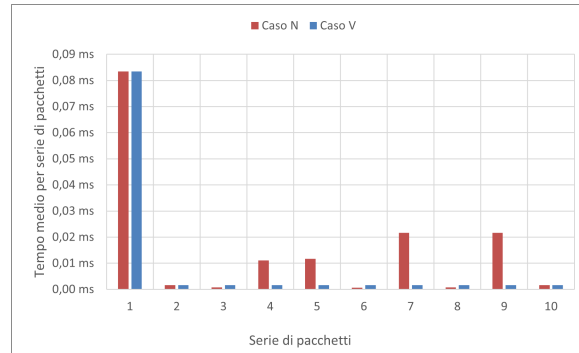


Figura 5: Latenza media per serie di pacchetti

Questo trend tende invece ad essere indipendente dal numero dei veicoli negli altri casi in figura 4c, in quanto la dimensione del platooning e quindi le distanze relative risultano essere molto basse (inferiori ai 50 metri) e con questi valori la discriminante "distanza" non affligge in alcun modo la trasmissività del segnale.

Infine troviamo il grafico 4d della latenza media per l'ultimo veicolo appartenente al platooning, che in modo analogo al precedente segue lo stesso ragionamento. Oltre all'andamento anche i valori sono pressoché uguali al caso medio, attribuite alle corte distanze, con l'unica differenza fatta per il caso con distanza 5 m e canale N, il quale presenta un valore di circa il quadruplo rispetto la media nel caso con 6 veicoli. Poiché i valori di latenza a cui corrisponde una packet loss dello 0% sono uguali alla media, è verosimile pensare che nei casi in cui i picchi di questa sono più alti (canale N, distanza 5 m), l'ultimo veicolo abbia un ruolo più determinante nell'incrementarli. Quindi, di conseguenza, si conclude come sia ragionevole avere tempi di latenza maggiori se ci si trova in coda al platooning.

## Low

Sia questo che il prossimo caso seguono in modo simile gli andamenti osservati a quello precedente, verranno dunque sottolineati con più enfasi gli aspetti per cui differiscono. Nella figura 6a si possono notare valori leggermente superiori di packet loss rispetto sia al caso lowest che highest quando il canale è N e la distanza è 5 metri. Questo è probabilmente dovuto ad una maggiore variabilità dei risultati ottenuti sulle 10 run indipendenti: a riprova di questo fatto sono stati inseriti gli intervalli di confidenza al 95%.

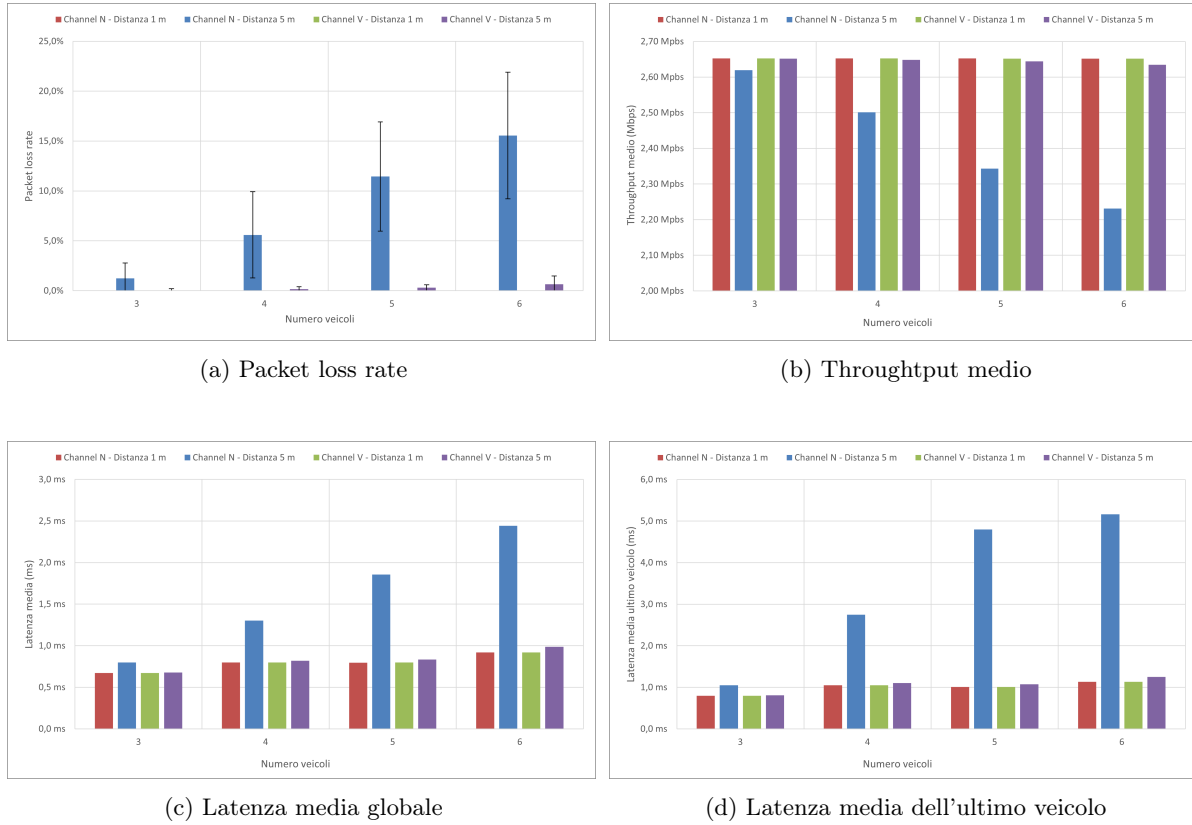


Figura 6: Statistiche nel caso low

Il throughput medio si attesta fino a picchi di 2.65 Mbps, in linea con i risultati previsti. Come visto per il caso baseline, riscontriamo anche qui tempi di latenza medi più lunghi, sempre legati alla dimensione maggiore del pacchetto e il ridotto inter packet interval, che gioca un ruolo nel congestionare maggiormente il canale e rallentare l'elaborazione dei dati verso i livelli più alti.

### Highest

Il throughput mantiene buoni valori compresi tra 0.255 e 0.295 Mbps (figura 7b). Sono invece riscontrabili dei tempi di latenza lievemente maggiori rispetto al caso lowest, attribuibili ad una maggiore dimensione del pacchetto (1200 bytes).

In generale, ad eccezione fatta per canale N e distanza 5 metri, i risultati di tutti i grafici si mostrano in linea con i requisiti di affidabilità, latenze minime e data rate richiesti dal 3GPP per il platooning.

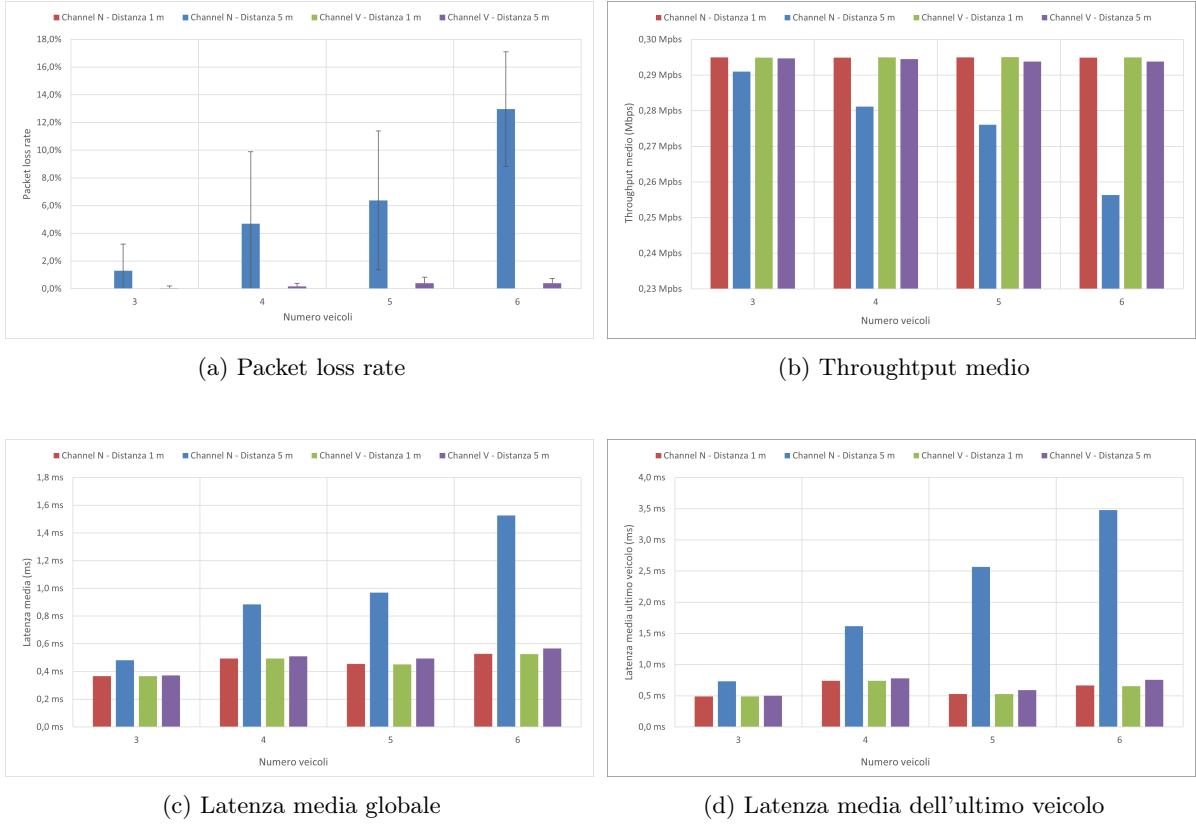


Figura 7: Statistiche nel caso highest

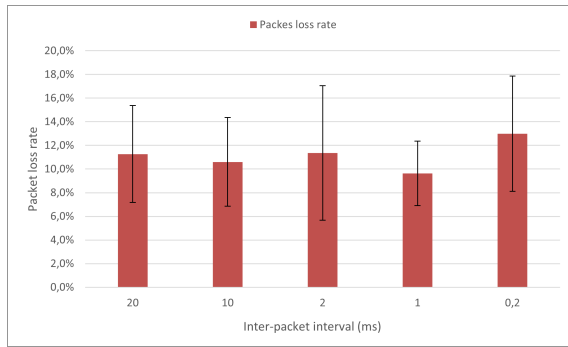
### 3.3 Caso limite

Concludiamo con il valutare uno scenario al limite al variare del valore di inter-packet interval, considerando quindi diversi livelli di congestione del canale, che oltre ad avere i requisiti applicativi più stringenti si trova nella situazione peggiore emersa negli scorsi paragrafi: 6 veicoli, 5 metri di distanza fra questi e canale di tipo N.

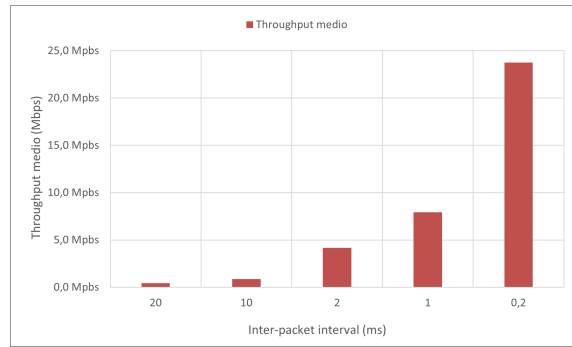
Dalla figura 8a è riscontrabile una packet loss pressoché stabile ma senza dubbio inaccettabilmente alta rispetto ai valori previsti dal caso d'uso del platooning per un alto livello di automazione, segno del fatto che l'inter packet interval sembra non giocare un ruolo fondamentale quanto lo stato del canale e la distanza, Il throughput medio aumenta considerevolmente al diminuire degli intervalli d'invio, come prevedibile, raggiungendo picchi di 25 Mbps, che sono comunque gestibili con onde millimetriche.

La latenza aumenta al diminuire degli intervalli, supportando le considerazioni fatte nella sezione precedente sulla necessità del server di dover elaborare una maggiore quantità di pacchetti in tempi più brevi.

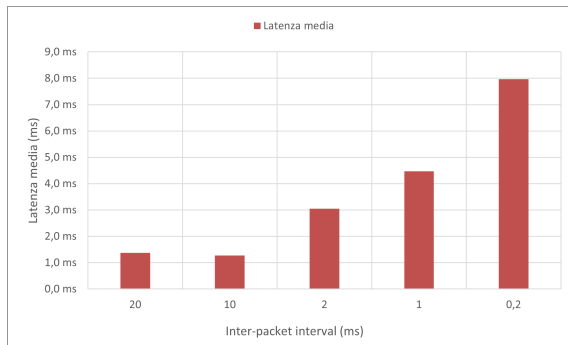
Nonostante, secondo gli standard visti, le applicazioni non funzionino con questi parametri in uno scenario reale, il sistema mantiene buone prestazioni sia in termini di latenza che di throughput. Questo non è altrettanto vero per l'affidabilità, che nel caso highest è fissata al 99,99 % (Tabella 2).



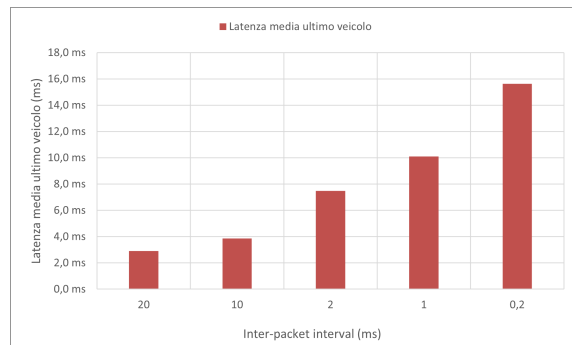
(a) Packet loss rate



(b) Throughput medio



(c) Latenza media globale



(d) Latenza media dell'ultimo veicolo

Figura 8: Statistiche nel caso limite

## Capitolo 4

# Conclusioni

La realizzazione di questa tesi è stata motivata dalla necessità di implementare e valutare un'applicazione per il traffico veicolare, in particolare nel caso studio del platooning.

A tale scopo, è stato utilizzato il software di simulazione ns-3 e il modulo MilliCar, ed è stato possibile realizzare un'applicazione che rispettasse le specifiche (in termini di caratterizzazione dell'applicazione) previste dallo standard 3GPP per comunicazioni V2X.

Il passaggio successivo è stato di quello di simulare in scenari differenti l'esecuzione dell'applicazione, che ha riscontrato comportamenti eccellenti per quasi tutte le casistiche e metriche. Infatti, fatta eccezione per il caso con canale di tipo N e distanza tra veicoli di 5 metri, i requisiti di latenza minima, affidabilità e throughput sono stati soddisfatti ampiamente.

Questa eccezione sembra attribuibile alla tipologia del canale, che presenta condizioni peggiori in termini di distanza da percorrere e ostruzione del segnale. Ciò porta così ad un aumento della packet loss media e di conseguenza ad una riduzione dell'affidabilità del sistema.

In futuro si potrebbero implementare ed analizzare le prestazioni di altri casi d'uso previsti dal 3GPP, come il collective perception of environment, l'automated cooperative driving for short distance grouping oppure il sensor and state map sharing.

# Bibliografia

- [1] J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, “Millimeter-Wave Vehicular Communication to Support Massive Automotive Sensing,” en, *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, Dec. 2016, ISSN: 0163-6804. DOI: 10.1109/MCOM.2016.1600071CM. [Online]. Available: <http://ieeexplore.ieee.org/document/7786130/>.
- [2] *C-v2x use cases volume ii: Examples and service level requirements*, English, version 1.0, 5GAA, 113 pp., 20/10/2020.
- [3] *3rd generation partnership project; technical specification group services and system aspects; study on enhancement of 3gpp support for 5g v2x services (release 16)*, English, version 16.2.0, 3GPP, Dec. 2018, 76 pp., 2018-12.
- [4] *Ns-3 / a discrete-event network simulator for internet systems*, <https://www.nsnam.org/>.
- [5] M. Giordani, A. Zanella, and M. Zorzi, “Millimeter wave communication in vehicular networks: Challenges and opportunities,” in *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, May 2017, pp. 1–6. DOI: 10.1109/MOCAST.2017.7937682.
- [6] M. Drago, T. Zugno, M. Polese, M. Giordani, and M. Zorzi, “MilliCar – An ns-3 Module for mmWave NR V2X Networks,” en, *arXiv:2002.10347 [cs]*, Feb. 2020, arXiv: 2002.10347. [Online]. Available: <http://arxiv.org/abs/2002.10347>.
- [7] *Millicar - an ns-3 module for mmwave nr v2x networks*, <https://github.com/signetlabdei/millicar>.
- [8] T. Zugno, M. Drago, M. Giordani, M. Polese, and M. Zorzi, “NR V2X Communications at Millimeter Waves: An End-to-End Performance Evaluation,” en, *arXiv:2005.10148 [cs]*, May 2020, arXiv: 2005.10148. [Online]. Available: <http://arxiv.org/abs/2005.10148>.
- [9] —, “Towards Standardization of Millimeter Wave Vehicle-to-Vehicle Networks: Open Challenges and Performance Evaluation,” en, *arXiv:1910.00300 [cs]*, May 2020, arXiv: 1910.00300. [Online]. Available: <http://arxiv.org/abs/1910.00300>.
- [10] *3rd generation partnership project; technical specification group services and system aspects; enhancement of 3gpp support for v2x scenarios; stage 1(release 16)*, English, version 16.2.0, 3GPP, Jun. 2019, 18 pp., 2019-06.
- [11] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, “End-to-End Simulation of 5G mmWave Networks,” en, *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2237–2263, 2018, ISSN: 1553-877X, 2373-745X. DOI: 10.1109/COMST.2018.2828880. [Online]. Available: <https://ieeexplore.ieee.org/document/8344116/>.