



UNIVERSITÀ DEGLI STUDI DI FIRENZE
Scuola di Ingegneria - Dipartimento di Ingegneria
dell'Informazione

Corso di Laurea Triennale in
INGEGNERIA INFORMATICA

Progettazione di un protocollo di diffusione dell'informazione per reti veicolari basato sul principio del Network Coding

Tesi di Laurea di

Daniele Mugnai

Relatori:

Prof. Francesco Chiti

Dott. Alessio Bonadio

Anno Accademico 2016/2017

Indice

1	Ringraziamenti	1
2	Introduzione	3
3	Panoramica	5
3.1	Le Reti MANET	5
3.2	Le Reti VANET	6
3.2.1	Introduzione	6
3.2.2	Applicazioni	7
3.2.3	Comunicazioni	10
3.2.4	IEEE 802.11p	10
3.2.5	IEEE 1609	11
3.2.6	Trasmissione dei Messaggi	13
3.3	Network Coding	14
3.3.1	Introduzione	14
3.3.2	Codifica	14
3.3.3	Decodifica	15
3.4	State of Art	15
3.4.1	BitTorrent	15
3.4.2	Code Torrent	17

3.4.3	CarTorrent	17
3.4.4	Avalanche	18
4	Design e Implementazione	20
4.1	Protocollo di Rete	20
4.2	Network Coding	22
4.2.1	Codifica	23
4.2.2	Decodifica	23
4.2.3	Coefficienti	25
4.3	Architettura e Funzionamento	25
5	Analisi e Testing	27
5.1	Tecnologie Utilizzate	27
5.1.1	OMNET++	27
5.1.2	Veins	28
5.1.3	SUMO	29
5.1.4	Eigen	29
5.2	Scenari di Test	29
5.2.1	Rete a Due Nodi	30
5.2.2	Rete a Tre Nodi	32
5.2.3	Rete a Quattro Nodi	34
5.2.4	Analisi Conclusiva	37
6	Conclusioni	39
	Bibliografia	42

Capitolo 1

Ringraziamenti

“La fortuna favorisce la mente preparata.”.

Louis Pasteur

Prima di entrare nella trattazione, sento il dovere di ringraziare tutti coloro che hanno partecipato e contribuito in questi anni di studio. Dopo tre anni e mezzo finalmente è arrivato il giorno tanto sperato.

Vorrei ringraziare il Prof. Francesco Chiti, relatore della mia tesi, per la disponibilità proposta e cortesia dimostratemi, e per tutto l'aiuto fornito durante lo sviluppo. Inoltre vorrei ringraziare il Dott. Alessio Bonadio, correlatore di Tesi, per la pazienza e l'aiuto durante questi tre mesi. Con lui vorrei ringraziare tutti le persone del laboratorio Daconets.

Il ringraziamento più grande va alla mia famiglia, che, con il loro incrollabile sostegno morale ed economico, mi hanno permesso di raggiungere questo traguardo. Grazie per avermi insegnato il valore del lavoro, dell'impegno e del sacrificio; per avermi fatto capire che senza questi non è possibile fare nulla.

Un ringraziamento ai compagni di studi, per essermi stati vicini sia nei momenti difficili, sia nei momenti felici: sono stati per me più veri amici che semplici compagni. In particolar modo vorrei ringraziare Alessio e Wolmer,

con i quali ho stretto un grande rapporto di amicizia. Insieme a loro vorrei Ringraziare Sofia, per essermi stata vicino in questo ultimo anno di studio.

Ultimi ma non meno importanti sono i ringraziamenti verso i miei amici. Ci siamo sempre sostenuti a vicenda, nella buona e nella cattiva sorte, durante le fatiche e lo sconforto che hanno caratterizzato il nostro percorso nei momenti di gioia e soddisfazione al raggiungimento del traguardo.

Un sentito grazie a tutti!

Capitolo 2

Introduzione

Negli ultimi anni con la forte affermazione delle reti Wireless sempre più preformanti e stabili, il decisivo aumento del traffico stradale e l'avanzamento dell'Intelligenza Artificiale, si sta cercando di studiare sistemi che migliorino la sicurezza e qualità stradale.

In Italia 610 persone su 1000 hanno un'auto; considerando che tra quei 390 sono anche incluse le persone al di sotto dei 18 anni, così come chi non ha la patente, praticamente ogni persona in grado di guidare possiede un'automobile. Si è dunque deciso di trovare soluzioni riguardanti i problemi connessi alla crescente densità automobilistica, ad esempio il maggior rischio di incidenti, congestioni stradali, inquinamento ecc. Mentre nel passato le migliorie erano avvenute sui sistemi utilizzati (AIRBAG, ABS, CRUISE CONTROL, ecc), adesso l'idea chiave è quella dell'auto "smart". I veicoli sono dotati di intelligenza artificiale, scambiamo informazioni fra di loro. L'idea che sta alla base della comunicazione fra i Veicoli è quella delle Reti VANET.

Le Vanet rappresentano il cuore dell'ITS ("Intelligent Transportation Systems"). I veicoli che percorrono una data strada, inviano ai propri vicini (veicoli che si trovano all'interno dell'area di copertura del raggio d'azione

del trasmittente) delle informazioni come ad esempio lo stato del territorio, i dati riguardanti la propria velocità e direzione, il tipo di strada, il traffico presente, etc. I veicoli che ricevono questi messaggi dai nodi trasmittenti "acquisiscono" informazioni utili alla viabilità. Alla ricezione di ogni messaggio, un veicolo valuta le informazioni contenute al suo interno per poi prendere delle decisioni viarie (stesso comportamento delle formiche in prossimità della scia olfattiva). Successivamente queste informazioni vengono inoltrate ai propri vicini oppure scartate. La realizzazione di sistemi di comunicazione veicolo-veicolo e veicolo-strade è stata possibile grazie agli studi compiuti negli ultimi 10 anni. In questo periodo infatti, si è avuto un notevole sviluppo nel campo delle reti wireless e nel campo delle reti "auto-organizzanti", reti che non richiedono alcuna infrastruttura fissa.

Proprio perché lo scambio dei messaggi deve avvenire nel modo più veloce e sicuro, si sono studiate diverse tecniche di trasmissione dei messaggi. Lo scopo di questa tesi è quello di mostrare i risultati ottenuti utilizzando come tecnica di trasmissione dei messaggi il Network Coding. L'emergere del Network Coding ha cambiato il modo di pensare la comunicazione di rete: invece di indirizzare semplicemente i dati, come avveniva prima con il routing, i nodi intermedi possono ricombinare i diversi pacchetti in entrata in due o più pacchetti in uscita.

In questo caso, quindi, i flussi di dati non devono essere più tenuti separati ma vengono combinati per adattare meglio il flusso di informazioni e per soddisfare le esigenze di specifici modelli di traffico. Questa nuova interpretazione di diffusione dei dati, rivoluziona il nostro modo di gestire, operare e comprendere le reti di comunicazione. Ci si aspetta che sia una tecnologia fondamentale per le reti del futuro.

Capitolo 3

Panoramica

3.1 Le Reti MANET

Le Reti MANET(Mobile Ad-hoc Network) sono reti definite come un sistema di nodi dotati di mobilità e capacità di organizzarsi dinamicamente in strutture senza topologie fisse : esse permettono le comunicazioni wireless tra nodi senza il supporto di infrastrutture. Rispetto alle reti infra-strutturate le MANET presentano alcune differenze:

- Alta mobilità dei nodi:
- Assenza del controllo e della gestione della rete
- Problema del terminale Esposto
- Problema del terminale Nascosto
- Protocollo CSMA-CA

3.2 Le Reti VANET

3.2.1 Introduzione

Le reti VANET(Vehicular ad hoc network) sono un sottoinsieme delle reti MANET. La differenza principale rispetto alle reti MANET è la tipologia dei nodi: in una rete VANET i nodi sono costituiti da veicoli. Vengono definite anche come "Mobile Ad Hoc Network for InterVehicle Communications" (IVC). A differenza delle MANET, le Vanet possiedono funzionalità aggiuntive:

- Strutture altamente energetiche a causa della velocità dei veicoli;
- Disponibilità di molti percorsi dinamici creati real-time;
- I veicoli possono utilizzare GPS(Global Position System) per identificare la loro posizione;
- Frequenti disconnessioni: l'alta velocità dei veicoli e il loro spostamento porta a frequenti disconnessioni.

Possiamo definire una Rete VANET come un flusso di veicoli che comunicano con altri veicoli e con RSU nel proprio raggio di trasmissione. Queste reti sono costituite da veicoli e RSU(Road Side Unit).

I veicoli devono essere dotati di OBU(On Board Unit) ovvero dispositivi wireless che permettono la comunicazione fra il veicolo e i dintorni. Gli RSU,invece, sono infrastrutture statiche.

All'interno della rete possiamo,dunque, identificare due tipi di comunicazione:

- Vehicle-To-Vehicle communication (V2V). Con questa sigla si indicano tutti i protocolli e le applicazioni che prevedono che i veicoli possano comunicare tra di loro senza una infrastruttura di base.
- Vehicle-To-Infrastructure communication (V2I). Con questa sigla si indicano tutti i protocolli e le applicazioni che prevedono che i veicoli comunichino con un'infrastruttura di base, e che mandino e ricevano informazioni da questa entità.

Chiaramente la tipologia IVC è potenzialmente più semplice da realizzare dal punto di vista pratico, poiché non occorre piazzare nessuna struttura statica, e tutto è demandato ai singoli veicoli. Questo però rende le scelte progettuali più complesse, poiché si tratta di un'architettura completamente distribuita, con alta mobilità e di conseguenza caratteristiche che possono cambiare molto rapidamente nel tempo.

Al contrario, la tipologia I2V richiede il piazzamento di strutture esterne in posti strategici, come ad esempio incroci, semafori, oppure anche semi-fisse. Con quest'ultima categorizzazione ci si riferisce a dispositivi che possono essere posizionati per segnalare condizioni temporanee, come ad esempio il restringimento di corsie dovuto a lavori in corso, condizioni anormali dell'asfalto, strade chiuse ecc.

Chiaramente queste due tipologie non sono completamente separate, ma possono essere impiegate entrambe facendo comunicare i veicoli tra di loro e con una infrastruttura di rete.

3.2.2 Applicazioni

Si possono identificare quattro categorie di applicazioni per Vanet:

- 1 Safety Oriented

2 Commercial Oriented

3 Convenience Oriented

4 Productive Oriented

La prima categoria comprende il monitoraggio della strada circostante, l'avvicinamento di veicoli, la superficie stradale, le curve stradali ecc. Questa tipologia di applicazioni può essere classificata come:

- Real-time traffic: i dati riguardanti il traffico possono essere memorizzati nelle RSU e possono essere disponibili ai veicoli nel momento in cui vengono richiesti. Tutto questo risulta utile per risolvere problematiche relative a congestioni stradali, incolonnamenti stradali ecc.
- Real-time traffic: i dati riguardanti il traffico possono essere memorizzati nelle RSU e possono essere disponibili ai veicoli nel momento in cui vengono richieste. Tutto questo risulta utile per risolvere problematiche relative a congestioni stradali, incolonnamenti stradali ecc.
- Post-Crash Notification: un veicolo coinvolto in un incidente dovrebbe inviare in broadcast messaggi di warning riguardanti la sua posizione in modo tale che i soccorsi (polizia, 118 ecc.) possano raggiungere il luogo senza problemi.
- Road Hazard Control Notification: i veicoli possono notificare ad altri le condizioni della strada
- Cooperative Collision Warning: avviso due guidatori potenzialmente coinvolti in un incidente di correggere il loro andamento stradale
- Traffic Vigilance: possono essere installate fotocamere negli RSU così da controllare comportamenti di guida errati.

La seconda categoria fornisce al guidatore intrattenimento e servizi commerciali. Possono essere classificate in questo modo:

- Internet Access: i veicoli possono accedere ad internet attraverso le RSU se gli RSU si comportano come router;
- Map download: le mappe di una regione possono essere scaricate all'occorrenza prima di guidare in una nuova area.
- Real-time video: il veicolo può richiedere la trasmissione di video in tempo reale
- Value-added advertisement: servizio utile per negozi e attività per attirare i clienti. Annunci come pompe di benzine, ristoranti ecc.

La terza categoria di applicazioni si occupa della gestione del traffico con l'obiettivo di migliorarne l'efficienza. Possono essere classificate come:

- Route Diversion: pianificazione di percorsi alternativi in caso di congestioni stradali.
- Electronic Toll Collection: il pagamento di tasse stradali può essere effettuato elettronicamente attraverso un'apparecchiatura. Questa deve essere in grado di leggere OBU del veicolo.
- Parking Availability: Notifica la presenza di parcheggi in città metropolitane, aiutando a cercare il posto auto in una certa area
- Active Prediction: notifica in anticipo al guidatore la presenza di percorsi alternativi che ottimizzino il consumo di carburante.

Infine l'ultima categoria si aggiunge alle sopra citate. Questa si può classificare come:

- Environmental Benefits: attraverso la rilevazioni di dati ambientali real-time si può creare informazioni utilizzabili che supportino scelte di trasporto “ecologiche” da parte degli utenti.
- Time Utilization:: se un viaggiatore scarica la sua email, può trasformare una congestione stradale in un’attività produttiva.

3.2.3 Comunicazioni

Come accennato in precedenza esistono due tipi di comunicazioni: V2V e V2I. Per quanto riguarda le comunicazioni V2V, i veicoli si scambiano informazioni e dati senza una infrastruttura superiore; mentre per quanto riguarda le comunicazioni V2I, gli RSU forniscono i veicoli di specifiche informazioni richieste. La necessità di RSU è dovuta al fatto che la richiesta di informazioni dai veicoli è sempre in crescita; in più una comunicazione attraverso RSU risulta più stabile e meno frequente a disconnessioni.

L’utilizzo di questo tipo di comunicazioni ha motivato una modifica dello standard IEEE 802.11. L’IEEE 802.11p, noto come standard Dedicated Short-Range Communications (DSRC) ,supporta lo scambio di dati tra veicoli ad alta velocità e tra automobili e l’infrastruttura fissa nella banda con licenza di 5,9-2,15 Ghz.

3.2.4 IEEE 802.11p

Lo standard 802.11p viene approvato dall’IEEE nel giugno del 2010 con lo scopo di rendere il livello fisico e MAC di 802.11 idonei alla comunicazione tra veicoli. L’architettura dello standard IEEE 802.11p WAVE è una modifica a quella dello standard IEEE 802.11. Le modifiche principali sono state fatte a

livello MAC in quanto era necessario intervenire per rendere la comunicazione tra i veicoli più rapida e allo stesso tempo efficiente.

Il Basic Service Set (BSS) è un gruppo di nodi connessi a un Access Point (AP) grazie a un collegamento wireless. Quando un nodo vuole accedere al BSS deve acquisire un service Set Identification (SSID).

Le operazioni di autenticazione e associazione sono troppo onerose per essere adottate in IEEE 802.11p in quanto le comunicazioni tra veicoli richiedono, soprattutto in caso di comunicazioni di sicurezza, la capacità di scambiare dati rapidamente, il che rende inefficiente la procedura di adesione utilizzata in 802.11. Inoltre Le comunicazioni V2V o V2I possono avvenire anche ad alte velocità pertanto il tempo di comunicazione deve poter essere molto breve.

In 802.11p viene introdotta una modalità che consente a due stazioni di poter comunicare direttamente tra loro, anche se non fanno parte dello stesso BSS, eliminando le procedure di associazione e autenticazione.

Dato che l'integrità dei dati trasmessi è un requisito fondamentale per le VANET, l'autenticazione viene gestita dai livelli superiori attraverso lo standard 1609.

3.2.5 IEEE 1609

Lo standard 1609 definisce il sistema WAVE(Wireless Access in Vehicular Environments) con l'obiettivo di garantire comunicazioni di tipo V2V e V2I. La famiglia di standard IEEE 1609 fornisce protocolli per lo scambio wireless di dati (1609.1), per la sicurezza dei messaggi (1609.2), per i servizi di rete (1609.3), per la gestione del livello MAC (1609.4).

L'architettura WAVE, definita dallo standard 1609.0, indica che le applicazioni in esecuzione su OBU e RSU possono comunicare tra loro secondo lo

stack protocollare in Figura.

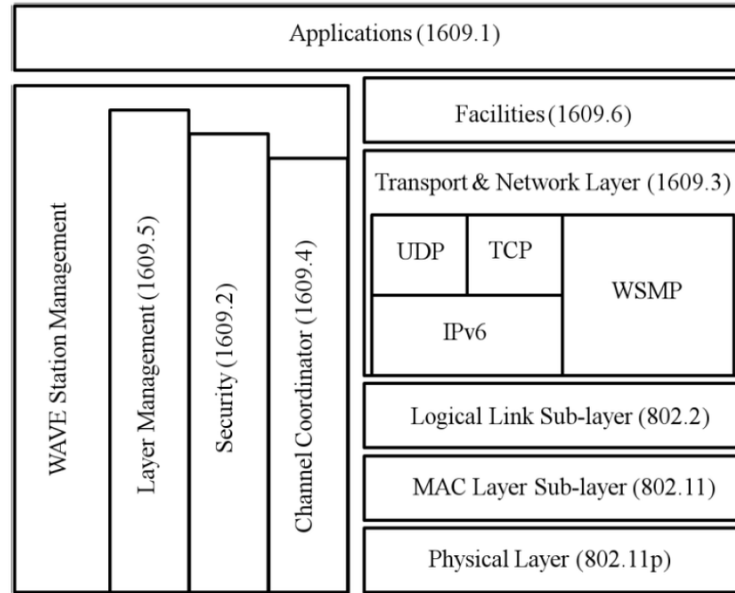


Figura 3.1: Protocol Stack Vanet

Nel sistema WAVE, al protocollo TCP, viene aggiunto il WSMP (WAVE Short Message Protocol) per rendere lo stack di protocolli maggiormente orientato ai servizi. WSMP consente di assegnare un PSID (Provider Service Identifier) ad un dispositivo che fornisce un servizio invece della coppia IP e trasporto. Un'applicazione che fornisce un servizio, invia in broadcast una frame di management contenente il proprio PSID.

Se esiste un'applicazione sul dispositivo interessata al servizio, il PSID viene memorizzato in modo che tutti i messaggi inviati dal provider vengono consegnati all'applicazione stessa. Inoltre l'architettura WAVE utilizza diversi canali di frequenza per lo scambio di messaggi. Il control channel è utilizzato per l'invio di messaggi di controllo mentre i service channel sono utilizzati per i messaggi relativi ai servizi offerti dalle applicazioni.

3.2.6 Trasmissione dei Messaggi

Il fulcro centrale delle Vanet è lo scambio di messaggi. Data la sua natura dinamica una rete Vanet necessita di avere comunicazioni estremamente veloci per poter far fronte ad eventi imprevisti(incidenti, ingorghi stradali, inversione di marcia repentina, etc.).

Rispetto a una rete infrastrutturale l'invio di un pacchetto fra nodi in una Vanet è decisamente più difficile. Stabilire le comunicazioni non è un compito banale, in quanto il percorso tra due nodi che vogliono comunicare non è conosciuto a priori, a causa dell'alta mobilità di cui i nodi sono dotati in queste reti.

Il paradigma delle reti MANET e di conseguenza il paradigma delle VANET si differenzia da quello della telefonia mobile dal fatto che l'infrastruttura che svolge il compito di routing è anch'essa in movimento e non geograficamente fissa come nei cellulari. Un messaggio viene trasmesso da un veicolo all'altro sino a raggiungere il nodo destinatario. Il sistema abituale adottato per la creazione delle rotte è il broadcast . Il più semplice algoritmo di broadcast che si utilizza è il Flooding (inondamento).

Nel Flooding, ogni nodo al ricevimento di un messaggio lo ritrasmette ai propri vicini una volta soltanto. Come si può notare, quest'algoritmo è molto semplice da implementare. Tuttavia, il suo utilizzo nella rete può portare a un numero esponenziale di messaggi talvolta ridondanti e questo può ridurre drasticamente le prestazioni della rete stessa. Il volume dei messaggi in broadcast è proporzionale alla densità dei veicoli.

3.3 Network Coding

Il Network Coding è un nuovo modo di intendere l'invio di dati attraverso la rete: i pacchetti non sono più trattati come singoli flussi d'informazione ma vengono elaborati dai nodi interni della rete allo scopo di costruire un unico pacchetto in uscita a partire da un certo numero di pacchetti in ingresso. I messaggi che fanno uso di network coding contengono quindi più dati, il che significa, che con meno trasmissioni si possono consegnare più informazioni. D'altra parte però, la dimensione dei messaggi può aumentare. Ovviamente, il costo computazionale aumenta per via dei processi di codifica e decodifica, i quali, avvengono anche nei nodi intermedi.

3.3.1 Introduzione

Il network coding è una tecnica matematica che permette di migliorare il rendimento e l'efficienza di una rete. A differenza del Flooding, un nodo prende diversi pacchetti e li combina insieme per poi trasmetterli. Questo implica che ogni nodo debba eseguire alcune operazioni sui pacchetti da trasmettere. L'idea principale è quella che il pacchetto codificato da trasmettere deve avere la stessa dimensione di un pacchetto in ingresso. Gli algoritmi per codificare e decodificare vengono spiegati nel corso del capitolo.

3.3.2 Codifica

Con il Network Coding i pacchetti in uscita sono una combinazione lineare dei pacchetti originali, in cui le operazioni sono effettuate nel campo F_2^q , $q < 8$. Si assuma di avere N pacchetti originali da inviare $M_1..M_n$. Con il Network Coding ogni pacchetto è associato a una serie di coefficienti $g_1..g_n$ appartenenti a F . Ogni pacchetto X^k codificato nella rete risulterà $X^k = \sum_{i=1}^n g_i M_i$.

3.3.3 Decodifica

Un nodo riceve il pacchetto X^k e lo memorizza nell'ultima riga della matrice di decodifica. Si consideri un nodo che ha ricevuto e memorizzato una set di m pacchetti codificati $[Y_1, \dots, Y_m]$ e sia X la matrice di decodifica. Ogni elemento $x_{i,j}$ della matrice della codifica sarà un numero appartenente a F_2^q , ove l'indice i indica la i -esima riga (i -esimo pacchetto codificato ricevuto) mentre l'indice j indica il pacchetto originale.

Se l'elemento $x_{i,j} = 0$ ciò significa che non è stato ricevuto il pacchetto originale; se $x_{i,j} \neq 0$ indica che il pacchetto è ricevuto. La matrice viene, poi, trasformata in una matrice triangolare usando un'eliminazione Gaussiana. Un pacchetto ricevuto è detto innovativo, se fa aumentare il rango della matrice, mentre se non è innovativo, viene ridotto ad una fila di zeri da un'eliminazione gaussiana e viene ignorato.

3.4 State of Art

3.4.1 BitTorrent

BitTorrent è un esempio di sistema P2P che utilizza tecniche per diffondere simultaneamente diversi frammenti di un file tra i diversi peer.

L'acquisizione di un file avviene attraverso la raccolta di suoi frammenti. Inoltre, come il numero di peer aumenta, diventa più difficile fare la pianificazione ottimale della distribuzione di frammenti ai ricevitori. Una possibile soluzione è quella di utilizzare un procedimento euristico che dà la priorità a scambi di frammenti più rari a livello locale. Questi spesso non corrispondono a quelli che sono globalmente più rari. Le conseguenze sono, tra le altre, download più lenti e trasferimenti in fase di stallo.

Per ogni file che deve essere distribuito, viene creata una rete composta da peer. Oltre ai peers generali, possiamo avere dei nodi speciali: il registrar, che attiva la scoperta di peer e tiene traccia della topologia, il logger, che aggrega i messaggi di analisi di peers e registrar ed i seeds, che sono peer che hanno acquisito il contenuto completo.

L'origine del file sorgente è il primo nodo della rete. Per raggiungere una rete, i peers contattano il registrar, e vengono connessi ad un piccolo numero (di solito 4-8) di nodi vicini. I vicini, per ogni nodo d'arrivo, sono scelti a caso tra i nodi partecipanti, che accettano la connessione a meno che non abbiano già raggiunto il loro numero massimo di vicini. Ogni nodo mantiene le informazioni topologiche locali, vale a dire, l'identità dei vicini ed il registrar registra la lista dei peers attivi.

Un peer può entrare e uscire dalla rete in qualsiasi momento. Per evitare che si formino isole isolate di peers (cluster), i nodi, periodicamente, abbandonano un vicino e si riconnettono ad uno nuovo, chiedendo al registrar di selezionare a caso un nuovo vicino dalla lista dei peers attivi. Per quanto riguarda la propagazione dei contenuti, la sorgente divide il file da distribuire in N blocchi. Ogni peer in possesso di un blocco agisce come un server per esso. Per decidere quali dei suoi blocchi un peer deve far partire, ad esempio, può scegliere una tra le diverse strategie di propagazione dei contenuti, inviando:

- o un blocco casuale all'inizio della distribuzione
- o un blocco tra i più rari a livello locale
- o un blocco più raro a livello globale delle piccole reti P2P

3.4.2 Code Torrent

Si tratta di un protocollo di file sharing P2P. Un nodo(veicolo) intende condividere un file(seed node). Il file F da condividere viene diviso in n parti $p_1..p_n$. Da notare che non avviene la trasmissione di una parte ma vengono inviati coded frames.

Un coded frames è definito come combinazione lineare di alcune parti p_k . Un coded frames c viene definito come $c = \sum_{k=1}^n e_k p_k$, dove e_k è un numero appartenente da F . Nel header del coded frames è memorizzato encoding vector $e = [e_1..e_n]$, utile per la futura codifica.

Sia E la matrice dei coefficienti, C la matrice dei coded frames ricevuti e $P = [p_1..p_n]$. Algebricamente si ha che calcolando $P = E^{-1}C$ si ottengono i pacchetti originali. Quando un nodo richiede un file, ogni nodo in possesso di una qualsiasi parte $p_{i,j}$ o di un coded frames, genera un nuovo coded frames e lo invia. Il nodo continua a richiedere coded frames dai suoi vicini fino a quando non vengono raccolti n coded frames linearmente indipendenti. Per ottenere le n parti di file $p_1..p_n$, un nodo deve ottenere n parti linearmente indipendenti.

3.4.3 CarTorrent

I ricercatori dell'UCLA Engineering con a capo Mario Gerla e Giovanni Pau hanno implementato un protocollo basato su BitTorrent per condividere informazioni via wireless fra automobili. Il nome del progetto è CarTorrent. I client CarTorrent disseminano la loro informazione sulla disponibilità dei pezzi in broadcast. Ogni messaggio è inoltrato fino a che non raggiunge un nodo lontano dall'origine. In questo modo i nodi possono conoscere le caratteristiche topologiche della rete; queste vengono poi utilizzate per scegliere

il pezzo più vicino. In pratica, se A e B sono due nodi che possiedono il pacchetto k che C desidera, C sceglie il nodo più vicino a se.

Code-torrent e Cartorrent risultano simili. Attraverso Car-torrent un peer necessita di ricevere solamente k pezzi da decodificare. Code-torrent implica un over-head di coded piece. Questo over-head implica una memorizzazione maggiori di blocchi e un maggior tempo di download.

3.4.4 Avalanche

Si tratta di un protocollo P2P creato da Pablo Rodriguez e Christos Gkantsidis di Microsoft che tenta di alleggerire il carico utilizzando Network Coding. Invece di distribuire i frammenti originali del file, i peers producono combinazioni lineari dei frammenti che sono già validi. Quando un peer ha abbastanza combinazioni linearmente indipendenti dei frammenti originali, riesce a decodificare e ricostruire il file originale. Microsoft Secure Content Downloader (MSCD) utilizza questo protocollo per la distribuzione P2P di Windows Update.

Analogamente a BitTorrent, la sorgente divide il file in n blocchi B_1, B_2, \dots, B_k e carica le diverse codifiche, di questi blocchi, a diversi utenti a caso. Gli utenti aiutano a distribuire il file, facendo l'upload dei blocchi in loro possesso, agli altri utenti nella rete. Mediante tali operazioni, l'informazione riesce a disperdersi in maniera rapida e il file è spedito ad ogni utente nella rete. In una rete di distribuzione di contenuti, i nuovi utenti possono collegarsi, come nodo nuovo, in qualsiasi momento, fino a quando il processo di distribuzione è attivo. Il nuovo utente si metterà in contatto con un tracker (un server centralizzato) per poter ottenere una lista dei peer, adiacenti al nodo, con cui comunicare. Come già ribadito, ciò che distingue Avalanche da BitTorrent o da qualsiasi altra rete P2P è l'utilizzo del Network Coding,

che a differenza del routing, permette la codifica dei nodi all'interno della rete e quindi uno throughput massimo quando l'informazione viene spedita in ambiente multicast.

Dopo aver ricevuto abbastanza blocchi codificati linearmente indipendenti, un nodo è in grado di decodificare l'intero file . In BitTorrent, il mittente trasmette in maniera casuale, il suo pacchetto più raro a livello locale o il più raro a livello mondiale. In Avalanche, il mittente non conosce la percentuale di pacchetti uguale ai suoi tra i vicini o nella rete, ma genera una combinazione lineare casuale di tutti i suoi pacchetti. Si evince che l'applicazione del Network Coding può consentire la riduzione del tempo di download di file perché un blocco codificato caricato da un nodo contiene informazioni su ogni blocco posseduto da quel nodo. Inoltre, se un nodo lascia la rete prima della fine del processo di distribuzione, è più probabile che i nodi rimanenti abbiano tutte le informazioni necessarie per recuperare l'intero file .

Capitolo 4

Design e Implementazione

In questo paragrafo introduciamo il protocollo di rete e l'algoritmo di Network Coding. Non verrà analizzata l'implementazione con frammenti di codice ma verranno presentati frammenti di pseudo codice (più leggibile) e specificando per ogni frammento, la logica di dominio che si è utilizzata.

4.1 Protocollo di Rete

Come analizzato in precedenza le reti VANET sono caratterizzate da un alta mobilità; questo non permette connessioni stabili e affidabili. Per far fronte a questo problema è stato eliminato il livello di Rete dallo Stack Protocol-lare TCP/IP. Dato la grande capacità di sviluppo del livello applicazione, è stato inserito come substrato ad esso il livello di Network Coding. Lo stack protocollare risultante è quello in figura.

APPLICATION
NETWORK CODING
UDP
LLC

Tabella 4.1: Stack Protocolli VANET

Il motivo principale per cui è stato omesso il livello di Rete è dovuto alle difficoltà legate al routing su una rete in movimento. L'alta mobilità e velocità dei nodi fa sì che il percorso fra due nodi cambi real-time. Risulta quindi molto complesso riuscire a memorizzare una tabella di routing: la metrica del percorso che un pacchetto deve effettuare risulta variabile e non affidabile.

Per quanto riguarda il livello trasporto è stato utilizzato il protocollo UDP. A differenza del TCP, l'UDP è un protocollo connectionless che non gestisce la trasmissione dei pacchetti persi. Risulta quindi poco affidabile; in compenso è molto rapido ed efficiente per applicazioni time-sensitive. Proprio per questo sulle Rete VANET è stato utilizzato il protocollo UDP.

La novità risulta l'inserimento del "cuscinetto" di Network Coding alla base del livello applicazione. Il funzionamento di questo verrà mostrato nelle sezioni seguenti.

4.2 Network Coding

Sia A la matrice di codifica

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots \\ a_{21} & a_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

dove

$$a_{i,j} = \begin{cases} 0 & \text{se non è stato ricevuto il pacchetto} \\ F_2^q & \text{se il pacchetto è stato ricevuto e decodificato} \end{cases}$$

Sia I la matrice di decodifica. Essa coincide con la matrice identità.

$$Y = \begin{bmatrix} 1 & 0 & \dots \\ 0 & 1 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Si tratta perciò di risolvere l'equazione

$$AX = Y \tag{4.1}$$

ovvero

$$X = A^{-1}Y \tag{4.2}$$

Il problema dunque è quello di ricavare l'inversa di A . La matrice A è la matrice dei pacchetti originali. Dal punto di vista algebrico vi sono diversi algoritmi per la risoluzione. Nel nostro caso viene utilizzato il Metodo di eliminazione di Gauss.

4.2.1 Codifica

La fase iniziale del Network Coding è quella di codifica. Sia A un nodo che possiede N pacchetti da inviare. L'algoritmo di combinazione viene illustrato successivamente.

Algorithm 1 Coding

```
1: procedure COMBINE
2:   Sia  $M[1...n+1]$  un array di messaggi
3:   Sia  $C$  il numero di pacchetti da combinare
4:   Crea nuovo messaggio Combined
5:   for  $i = 1$  to toCombine do
6:     Scegli un messaggio  $m(i)$  da  $M$ 
7:     Inserisci  $m(i)$  nell'header
8:   end for
9: end procedure
```

Quando il Nodo A ha a disposizione un pacchetto combinato lo invia in broadcast. A questo punto se un altro nodo è nel raggio di trasmissione il pacchetto viene ricevuto da questo altrimenti viene perso.

4.2.2 Decodifica

La fase di decodifica si ha quando un nodo riceve un messaggio. L'algoritmo viene mostrato qui sotto.

Algorithm 2 Decoding

```
1: procedure DECODING
2:   Sia X la matrice di codifica
3:   Sia p un messaggio combinato ricevuto
4:   Sia n il numero di pacchetti combinati su p
5:   for  $i = 1$  to n do
6:     Sia g il coefficiente associato al pacchetto
7:     Sia i la riga
8:     Sia j l'ID di un pacchetto combinato
9:     Inserisci g nella posizione M(i,j)
10:  end for
11:  Incrementa i
12: end procedure
```

Una volta decodificato il messaggio ricevuto e dopo aver popolato la matrice di codifica, il nodo può, se è possibile, decidere di risolvere il sistema lineare precedentemente illustrato.

Algorithm 3 Solve

```
procedure SOLVE
  Sia X la matrice di codifica
  if X è invertibile then
    Risolvi l'inversa di X
  else
    Attendi nuovo pacchetto
  end if
end procedure
```

4.2.3 Coefficienti

Come abbiamo visto in precedenza la matrice di codifica viene popolata con numeri appartenenti a campi finiti (Galois). I campi di Galois sono strutture costituite da un numero finito di elementi nei quali le operazioni aritmetiche sono chiuse. Questo significa che qualunque operazione aritmetica su due elementi nel campo avrà come risultato ancora un elemento appartenente al campo.

Il problema legato ai coefficienti è quello di evitare combinazioni lineari indipendenti. Il problema consiste nel fatto che, col Network Coding, si ha una certa probabilità di avere combinazioni lineari dipendenti, mentre a noi servirebbe che fossero indipendenti. Uno studio, sul trattato [9], dimostra che questa probabilità è legata alla dimensione del campo F_2^q e che, per campi di piccole dimensioni, questa probabilità diventa trascurabile.

4.3 Architettura e Funzionamento

Nella nostra implementazione abbiamo considerato una rete P2P. Veicoli e RSU sono trattati allo stesso modo. L'unica differenza fra i due è la loro mobilità: gli RSU hanno mobilità nulla a differenza dei veicoli. Riepilogando quello che abbiamo trattato precedentemente:

- Un nodo chiamato sorgente ha un insieme di pacchetti da inviare;
- Combina i pacchetti seguendo l'algoritmo COMBINE
- Invia i pacchetti combinati
- Un nodo destinatario riceve i pacchetti combinati

- Il nodo destinatario decodifica i pacchetti secondo l'algoritmo DECODE
- Il nodo destinatario ricava i pacchetti originali attraverso l'algoritmo SOLVE

Collaboration1::Interaction1::SequenceDiagram1

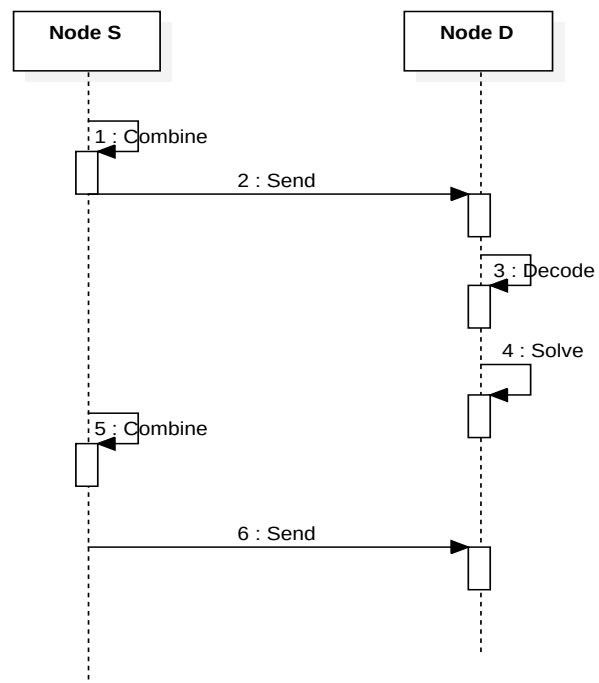


Figura 4.1: Sequence Diagram

Capitolo 5

Analisi e Testing

5.1 Tecnologie Utilizzate

5.1.1 OMNET++

OMNET++ è una piattaforma di simulazione di reti:

- Modulare
- Ad eventi discreta
- Object Oriented(C++)
- Open-Source

Il suo dominio di utilizzo è appunto la rete; fornisce un supporto per network sensor, wireless ad hoc-network, Internet protocol, ecc. Omnet++ fornisce allo sviluppatore un IDE Eclipse-Based. Attraverso OMNet++ è possibile costruire scenari di network communication attraverso la definizione di gerarchie di moduli scritti in C++. La struttura dei moduli, le relazioni e i link di comunicazione tra di essi è specificato attraverso i Network Descrip-

tion files (o NED). La logica di dominio degli stessi è invece incorporata in gerarchie di classi C++ che rispecchiano fedelmente le gerarchie dei NED.

5.1.2 Veins

Veins è un framework open-source per simulazioni di Vanet (Vehicle in Network Simulation). Si basa su due simulatori: OMNET++ e SUMO. Veins fornisce all'utente una suite di modelli per simulare Vanet più realisticamente possibile senza sacrificare la velocità.

Veins fornisce una parte di codice, che può essere utilizzato senza venir modificato, e una parte di ambiente di sviluppo ideata per essere estesa da parte dell'utente. L'utente deve solamente dedicarsi all'implementazione della propria applicazione: il framework si occupa di tutto il resto: modellazione degli strati di protocollo inferiori e della mobilità dei nodi, avendo cura di impostare la simulazione, assicurandone la corretta esecuzione e raccogliendo i risultati durante e dopo la simulazione.

Con Veins ogni simulazione viene eseguita eseguendo due simulatori in parallelo: OMNeT++ (per la simulazione di rete) e SUMO (per la simulazione del traffico stradale). Entrambi i simulatori sono collegati tramite un socket TCP. Il protocollo per questa comunicazione è stato standardizzato come Traffic Control Interface (TraCI). Ciò consente la simulazione accoppiata bidirezionale del traffico stradale e del traffico di rete. Il movimento dei veicoli nel simulatore del traffico stradale SUMO si riflette come movimento dei nodi in una simulazione OMNeT++. I nodi possono quindi interagire con la simulazione del traffico stradale in corso, ad es. Per simulare l'influenza dell'IVC sul traffico stradale.

5.1.3 SUMO

SUMO è una suite di simulazione del traffico gratuita e aperta, disponibile dal 2001. SUMO consente la modellazione di sistemi di traffico intermodale compresi veicoli stradali, mezzi pubblici e pedoni. Incluso con SUMO è una vasta gamma di strumenti di supporto che gestiscono attività come la ricerca di percorsi, la visualizzazione, l'importazione di rete e il calcolo delle emissioni. SUMO può essere migliorato con modelli personalizzati e fornisce varie API per controllare a distanza la simulazione.

5.1.4 Eigen

A livello applicazione per realizzare questo lavoro è stato necessario utilizzare una libreria esterna per l'algebra. Uno dei difetti di Veins è la poca apertura verso la compatibilità con altre Librerie. Perciò abbiamo deciso di utilizzare la libreria Eigen. Eigen è una template library scritta in C++ per l'algebra lineare:matrice,vettori,risoluzioni e algoritimi inerenti. Si tratta di una libreria Open-Source licenziata sotto MPL2. La libreria è implementata utilizzando tecniche di programmazione a template. La libreria si presenta molto semplice all'utilizzo, molto simile a MATLAB. La libreria dunque è stata utilizzata nella implementazione,gestione e risoluzione di matrici. La risoluzione non risulta ottimale date i vincoli della libreria. Una maggior precisione dei dati può essere realizzata attraverso librerie più complesse o cambiando ambiente di sviluppo(MATLAB).

5.2 Scenari di Test

Nella sezione seguente vengono mostrati tre scenari di test e analisi. La scelta di questi scenari è dovuta al fatto che qualunque topologia di rete è ricon-

ducibile alla combinazione di queste. I test sono stati effettuati mettendo a confronto l'utilizzo di Network Coding e il Flooding. Il numero di messaggi utilizzato per il testing è crescente: da 4 fino a 128. Per numero di messaggi maggiori è necessario utilizzare librerie matematiche più efficienti.

Le reti analizzate sono le seguenti:

- Rete a due nodi
- Rete a tre nodi
- Rete con quattro nodi

5.2.1 Rete a Due Nodi

Si tratta della connessione più semplice da realizzare.

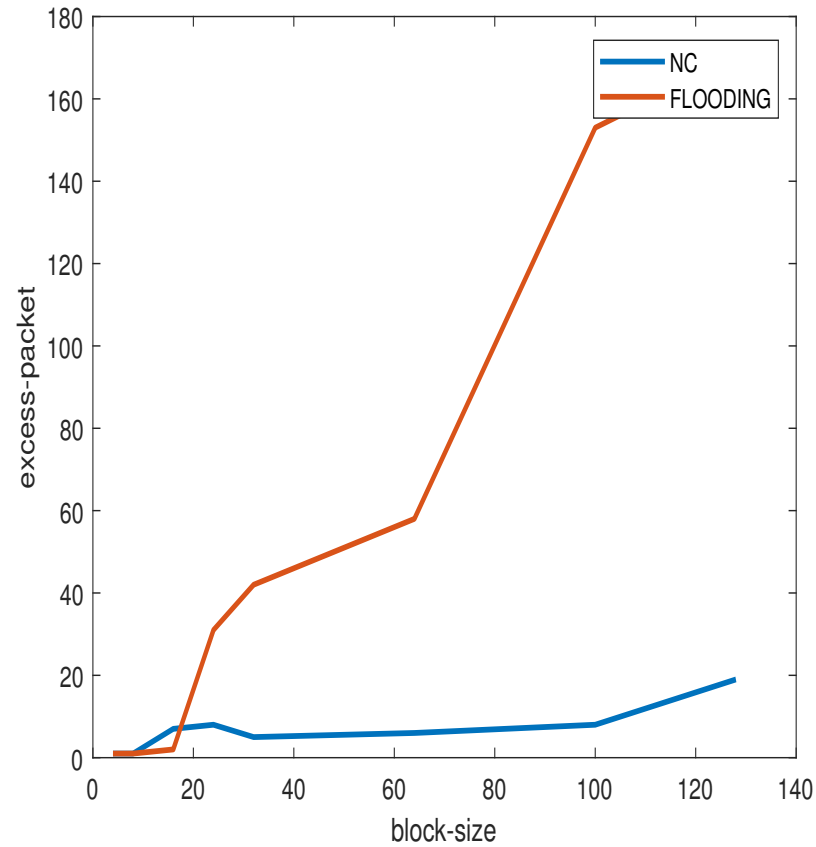


Figura 5.1: Overhead su rete a due nodi

Analizzando il grafico è facile vedere che per un numero piccolo di messaggi il flooding risulta più efficiente. Questo è dovuto al fatto che il Network Coding su un numero piccolo di combinazione ha maggiori probabilità di creare una combinazione lineare dipendente. Risulta perciò necessario un numero di pacchi maggiori per ottenere i pacchetti originali. Per un numero di pacchetti elevato il Network Coding risulta più efficace.

5.2.2 Rete a Tre Nodi

In questo scenario è stato introdotto un nuovo nodo che chiamiamo Relay. Poiché il nodo sorgente e destinatario sono posti fuori dal loro raggio di trasmissione, abbiamo inserito un nodo Relay. Questo ha un comportamento passivo, ovvero non genera alcun messaggio. Il suo ruolo è quello di storage di messaggi combinati; combina i messaggi fra quelli che già ha ricevuto per poi ritrasmettere il messaggio.

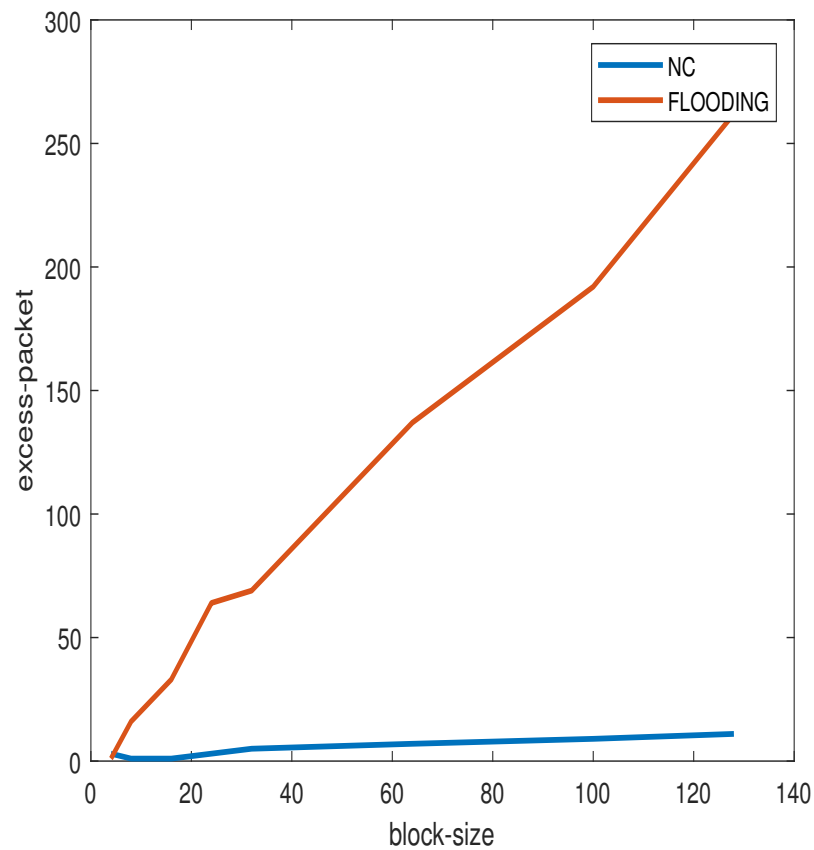


Figura 5.2: Overhead su rete a tre nodi

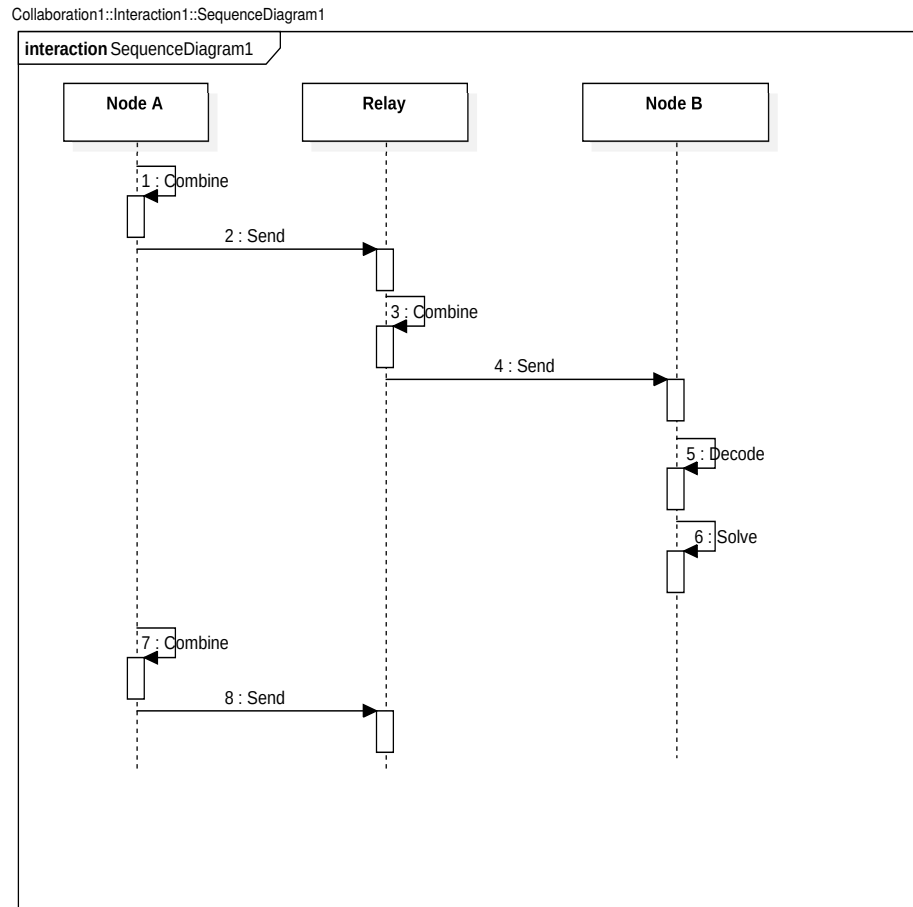


Figura 5.3: Sequence Diagram con Relay

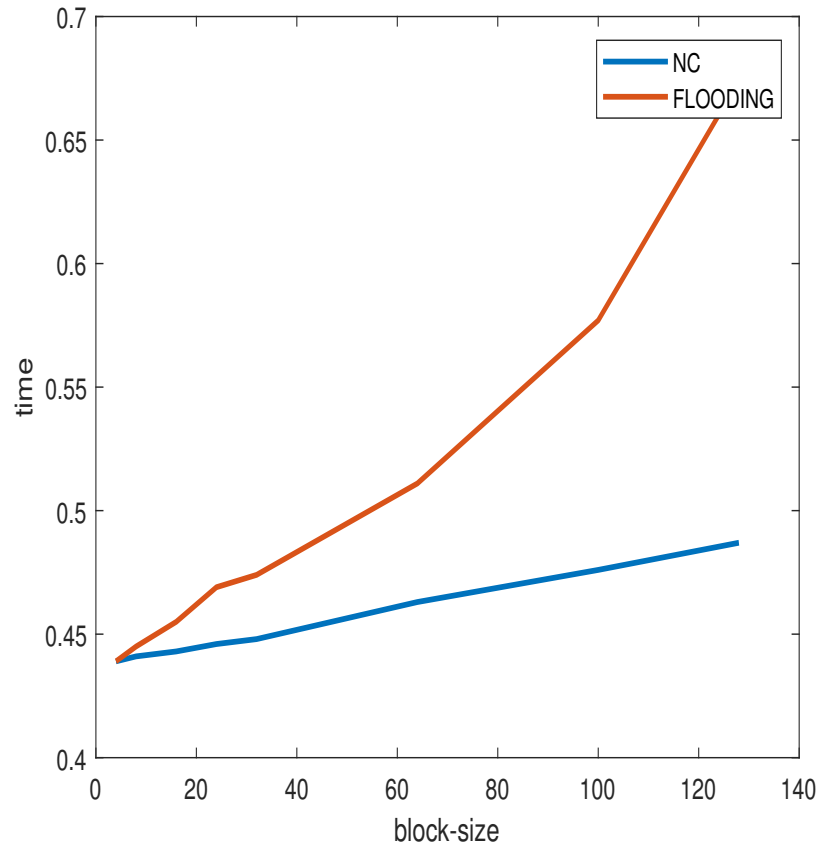


Figura 5.4: Variazione tempo di trasmissione su rete a tre nodi

In questo caso il Network Coding risulta molto più performante sia per un numero piccolo di messaggi sia per un numero elevato. Il nodo Relay gioca un ruolo chiave: combinando a sua volta altri pacchetti migliora la combinazione di pacchetti così che il numero di combinazioni lineare indipendenti siano poco più di quelle necessarie necessarie.

5.2.3 Rete a Quattro Nodi

Aggiungendo un ulteriore nodo Relay, si viene a costituire una rete a diamante come in figura.

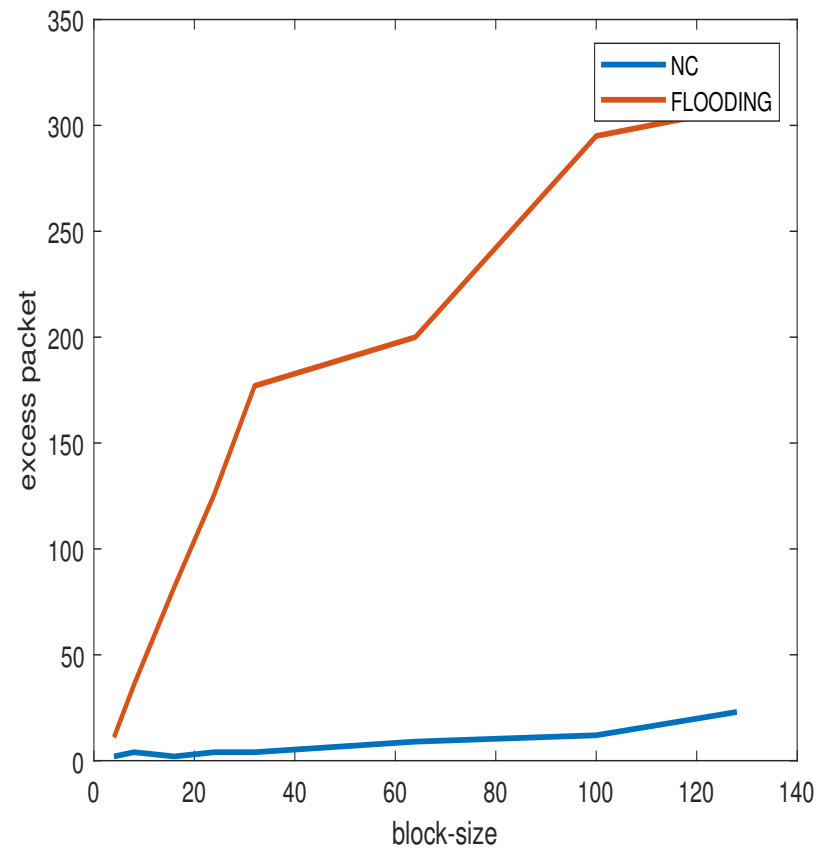


Figura 5.5: Overhead su rete a quattro nodi

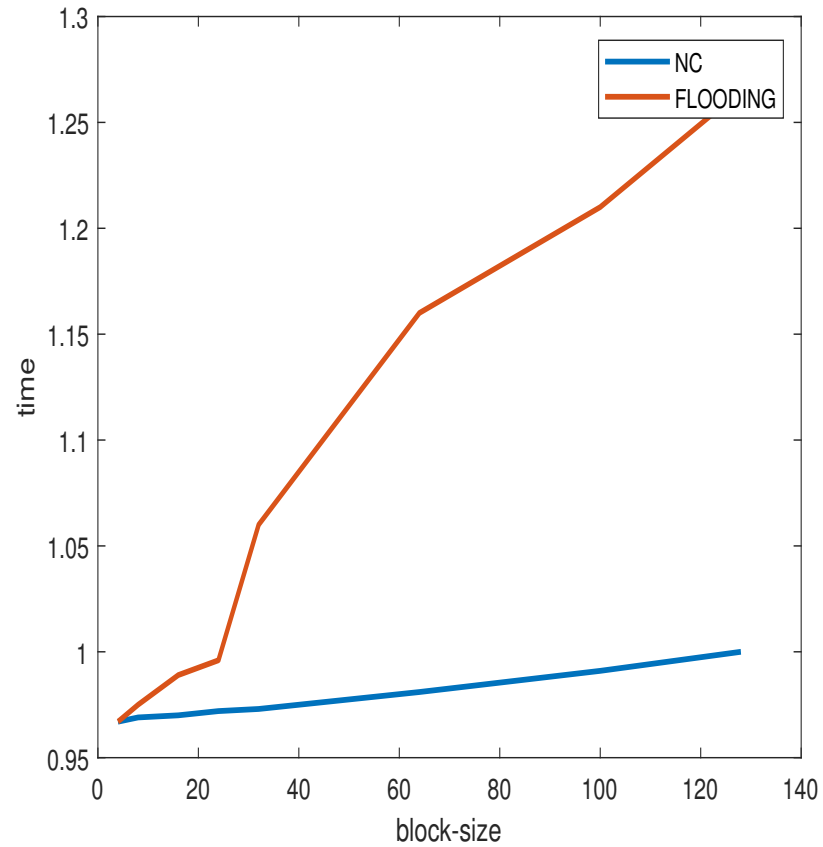


Figura 5.6: Variazione tempo di trasmissione su rete a quattro nodi

In questo scenario il Network Coding risulta molto performante: il gap fra i due andamenti è molto grande e cresce all'aumentare dei messaggi. Due Relay fanno sì che le combinazioni sia molto diverse: anche nel caso di combinazioni uguali, la scelta dei coefficienti diversi fa sì che due righe della matrice siano difficilmente linearmente dipendenti, alleggerendo il costo per risolvere il sistema.

5.2.4 Analisi Conclusiva

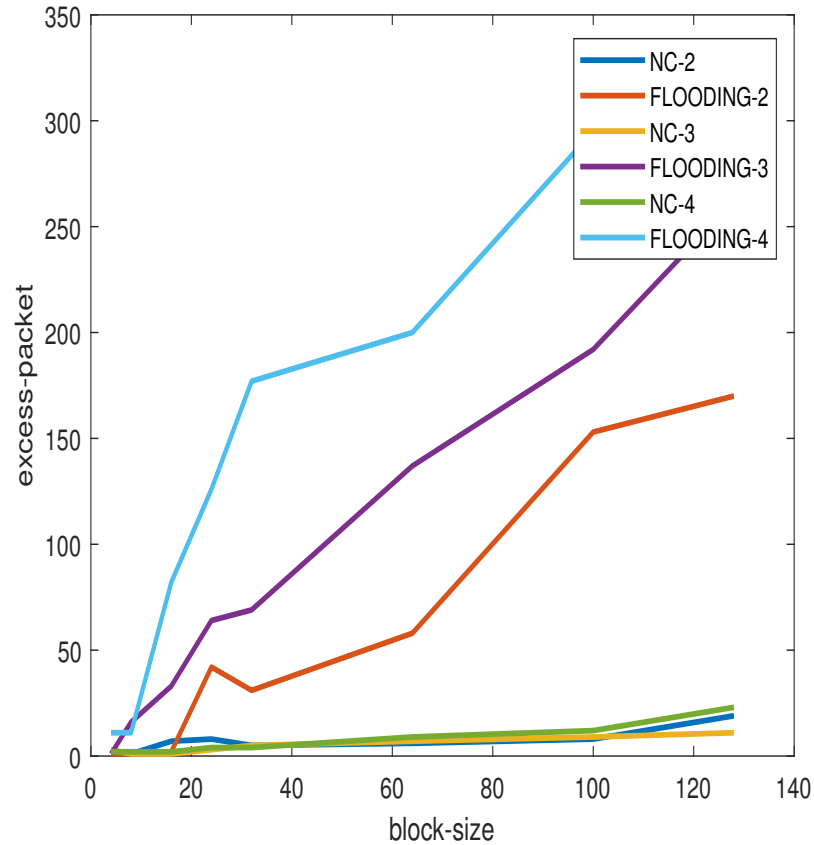


Figura 5.7: Riepilogo Test

La trasmissione totale dei messaggi attraverso Flooding necessita un numero di messaggi elevato, risultando meno performante: se un pacchetto viene perso durante la trasmissione, è necessario la ritrasmissione di tutti. Attraverso il Network Coding, invece, se un pacchetto viene perso, vi sarà sicuramente un pacchetto combinato che tiene una parte di questo, facendo sì che l'informazione possa essere ricostruita.

In tutto questo va considerato che il Network Coding, a differenza del Flooding, necessita di un costo computazionale maggiore. La risoluzione di

un sistema lineare(matrice inversa), è molto complesso per matrici di grosse dimensioni.

Analizzando il grafico conclusivo si possono concludere le seguenti considerazioni:

- Al variare della complessità della rete, il Network Coding non varia la propria complessità;
- All'aumentare della complessità della rete, attraverso il Flooding, i messaggi necessari per la ricostruzione dell'informazione risulta crescente;

Capitolo 6

Conclusioni

Le informazioni raccolte durante la stesura della tesi forniscono un quadro generale di come vengono instradate le informazioni sulla rete, in presenza del Network Coding: i pacchetti non sono più trattati come singoli flussi d'informazione ma vengono elaborati dai nodi interni della rete allo scopo di costruire un unico pacchetto in uscita a partire da un certo numero di pacchetti in ingresso.

Il network coding risulta perciò molto performante per situazioni di Disseminazioni di informazioni: per quanto riguarda la trasmissione istantanea di un messaggio

Sono state citate tutte le scelte di progetto rilevanti che sono state prese durante la fase di design del software. Naturalmente non sono state riportate. Per un'approfondimento ulteriore è richiesta una analisi approfondita del codice.

Il lavoro di tesi è stato proposto dal Prof. Francesco Chiti ed è stato prodotto da Mugnai Daniele.

Elenco delle figure

3.1	Protocol Stack Vanet	12
4.1	Sequence Diagram	26
5.1	Overhead su rete a due nodi	31
5.2	Overhead su rete a tre nodi	32
5.3	Sequence Diagram con Relay	33
5.4	Variazione tempo di trasmissione su rete a tre nodi	34
5.5	Overhead su rete a quattro nodi	35
5.6	Variazione tempo di trasmissione su rete a quattro nodi	36
5.7	Riepilogo Test	37

Elenco delle tabelle

4.1	Stack Protocollore VANET	21
-----	------------------------------------	----

Bibliografia

- [1] A. Ashtaiwi and A. Saoud, “Performance comparison of position based routing protocols for vanets,” in *2017 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)*, Oct 2017, pp. 78–84.
- [2] Q. Wang, D. Xie, and X. Ji, “Network codes-based content-centric transmission control in vanet,” in *2015 International Conference on Connected Vehicles and Expo (ICCVE)*, Oct 2015, pp. 157–162.
- [3] Z. Yang, M. Li, and W. Lou, “Codeplay: Live multimedia streaming in vanets using symbol-level network coding,” in *The 18th IEEE International Conference on Network Protocols*, Oct 2010, pp. 223–232.
- [4] J. Fukuyama, R. Vuyyuru, R. Guha, W. Chen, and J. Lee, “A fast and distributed algorithm for vehicular network coding,” in *2010 IEEE Vehicular Networking Conference*, Dec 2010, pp. 197–201.
- [5] F. Liu, Z. Chen, and B. Xia, “Data dissemination with network coding in two-way vehicle-to-vehicle networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2445–2456, April 2016.

- [6] M. H. Firooz and S. Roy, “Collaborative downloading in vanet using network coding,” in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 4584–4588.
- [7] M. Gerlay, R. G. Cascella, Z. Caoy, B. Crispo, and R. Battiti, “An efficient weak secrecy scheme for network coding data dissemination in vanet,” in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2008, pp. 1–5.
- [8] A. Magnetto, S. Spoto, R. Gaeta, M. Grangetto, and M. Sereno, “Fountains vs torrents: The p2p toroverde protocol,” in *2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Aug 2010, pp. 417–420.
- [9] Y. Wu, P. A. Chou, and K. Jain, “A comparison of network coding and tree packing,” in *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.*, June 2004, pp. 143–.
- [10] U. Lee, J.-S. Park, J. Yeh, G. Pau, and M. Gerla, “Code torrent: Content distribution using network coding in vanet,” in *Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking*, ser. MobiShare ’06. New York, NY, USA: ACM, 2006, pp. 1–5. [Online]. Available: <http://doi.acm.org/10.1145/1161252.1161254>
- [11] E.-K. Lee, M. Gerla, G. Pau, U. Lee, and J.-H. Lim, “Internet of vehicles: From intelligent grid to autonomous cars and vehicular fogs,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 9, p. 1550147716665500, 2016. [Online]. Available: <https://doi.org/10.1177/1550147716665500>